

I. PENDAHULUAN

1.1. Latar Belakang

Kebutuhan akan *internet* sudah menjadi prioritas yang utama untuk masyarakat. Ini dipengaruhi dengan perkembangan teknologi yang sudah sangat berkembang yang juga mempengaruhi gaya hidup masyarakat pada umumnya, mulai dari berkomunikasi, berbelanja, berkerja, belajar, hiburan, dan masih banyak lagi, ini bisa dilakukan dimana saja dan kapan saja selama masih terhubung dengan *internet*, menurut data Asosiasi Penyelenggara Jasa *Internet* Indonesia (APJII, 2019) Pengguna *internet* di Indonesia pada tahun 2018 adalah sebanyak 64,8% atau sekitar 171,17 juta jiwa dari total populasi penduduk di Indonesia yaitu 264,16 juta jiwa dengan penggunaan 21,6% di wilayah Sumatera, 55,7% di pulau Jawa, 6,6% di pulau Kalimantan, 5,2% di wilayah Bali dan Nusa Tenggara, dan 10,9% di wilayah Sulawesi, Maluku dan Papua. Hal ini bertambah sekitar 10,12% total dari jumlah pengguna *internet* pada tahun 2017 (Fadly, 2020). Pandemi *Covid-19* yang melanda di dunia makin membuat masyarakat harus bisa beradaptasi dengan keadaan saat ini yang mulanya biasa bekerja di kantor sekarang dipaksa untuk bisa bekerja secara jarak jauh di rumah masing masing. Begitu juga dengan pendidikan, yang biasanya dilakukan secara tatap muka sekarang harus mulai beradaptasi melakukan pembelajaran jarak jauh. Hal ini tentu saja mempengaruhi penggunaan internet menjadi lebih besar lagi.

PT. Lintas Data Multimedia adalah perusahaan swasta yang bergerak dibidang pemasangan dan penyedia layanan *internet*. Perusahaan ini berfokus pada pemasangan dan penyedia layanan *internet* pada individu seperti rumahan maupun pada instansi instansi perkantoran baik swasta maupun negeri di wilayah Lampung. Untuk pemasangan lebih sering menggunakan metode *Fiber to the X* ini juga tergantung lokasi dan wilayah pemasangan apakah sudah dapat dijangkau sinyal nya atau belum, jika belum, biasanya mereka akan menggunakan

gelombang yang akan ditembakkan dari *base transceiver station tower* yang kemudian diterima melalui *receiver*. Akan tetapi biasanya biaya yang dikeluarkan akan lebih mahal dibandingkan menggunakan *Fiber to X*.

Pada proses pemasangan jaringan pada lokasi biasanya dilakukan terlebih dahulu survei untuk menentukan keperluan alat alat bahan, pembuatan topologi dan juga informasi informasi terkait yang berada di lokasi tersebut. Kemudian setelah informasi telah didapat, tim survei akan memberikan arahan ke tim lapangan yang akan bertugas memasang jaringan di tempat itu, mulai dari alamat lokasi, keadaan tempat, alat dan bahan, topologi dan juga informasi lain yang terkait di lokasi tempat pemasangan tersebut.

Setelah tim lapangan yang bertugas memasang telah selesai melaksanakan tugasnya, pengawas akan menghubungi penanggung jawab dan memberitahukan pemasangan telah selesai. Penanggung jawab disini kemudian akan menanyakan hal hal terkait pemasangan apakah alat alat sudah terpasang dengan benar dan akan meminta foto sebagai bukti bahwa alat alat yang dibutuhkan telah terpasang dan juga melakukan tes pada jaringan apakah masih ada kendala ataupun tidak. Setelah itu penanggung jawab akan menghubungi pihak klient bahwa pemasangan telah selesai dilaksanakan dan dapat digunakan.

Disini tim survei maupun tim pemasangan masih menggunakan kertas *form* sebagai alat acuan dalam pemasangan jaringan. Mulai dari alamat, topologi, alat alat yang dibutuhkan, sampai informasi yang terkait lokasi pemasangan jaringan tersebut. Sehingga hal ini dapat menimbulkan kendala yang dapat terjadi seperti terduplikasinya data, kemudian adanya potensi kehilangan berkas, dan kebutuhan informasi yang cepat. Hal ini membuat pengumpulan data menjadi tidaklah efisien. Oleh karena itu berdasarkan permasalahan yang telah penulis uraikan, maka dibuatkan sebuah solusi Aplikasi Pelaporan Pemasangan Jaringan berbasis *Mobile* Menggunakan *Flutter* yang diharapkan dapat mampu memberikan informasi secara tepat dan akurat pada proses pemasangan jaringan ini.

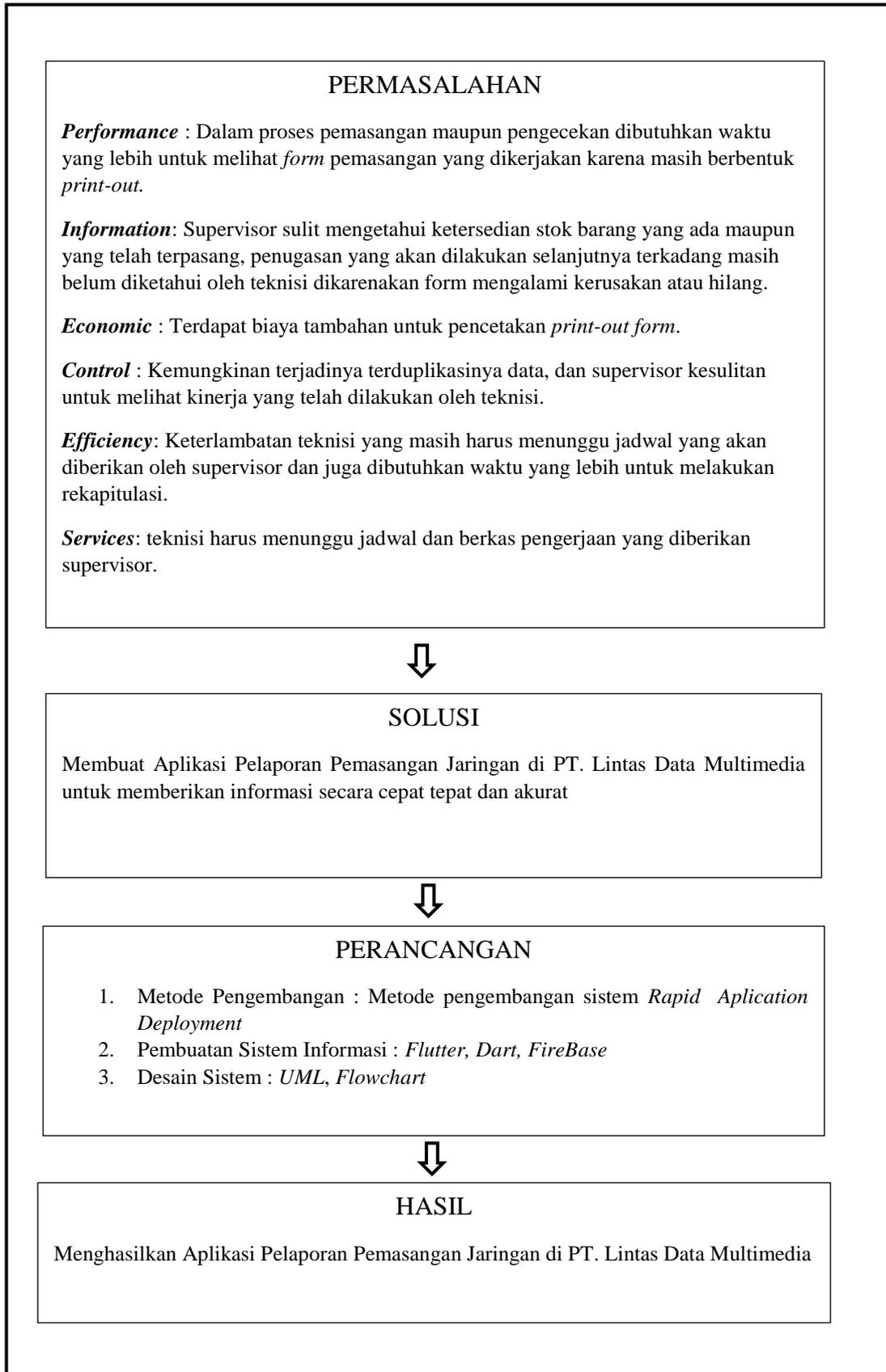
1.2. Tujuan

Tujuan dari pembuatan Aplikasi Pemasangan Jaringan pada PT Lintas Data Multimedia ini adalah untuk:

1. Menghasilkan sebuah aplikasi pelaporan pemasangan jaringan yang bisa digunakan oleh karyawan PT. Lintas Data Multimedia.
2. Memberikan informasi yang cepat tepat dan akurat terhadap lokasi pemasangan jaringan baik untuk tim pemasangan maupun penanggung jawab dari tim pemasangan tersebut.

1.3. Kerangka Pemikiran

Berdasarkan latar belakang dapat disusun suatu kerangka pemikiran yang disajikan sebagai berikut:



Gambar 1. Kerangka Pemikiran

1.4. Kontribusi

Aplikasi Pelaporan Pemasangan Jaringan di PT. Lintas Data Multimedia ini dapat memberikan kontribusi sebagai berikut:

1. Dengan adanya aplikasi ini memudahkan karyawan PT. Lintas Data Multimedia untuk mendapatkan informasi mengenai penugasan yang didapat.
2. Dengan adanya aplikasi ini dapat memudahkan penanggung jawab memberikan informasi secara cepat dan akurat kepada tim lapang untuk pemasangan jaringan
3. Dengan adanya aplikasi ini tim pemasang jaringan mendapatkan informasi secara cepat dan akurat dan juga dapat memberikan informasi kepada penanggung jawab secara cepat setelah melakukan pemasangan jaringan.

II. TINJAUAN PUSTAKA

2.1. Artikel Terkait

Artikel terkait yang dijadikan sumber acuan untuk Tugas Akhir ini diambil dari beberapa jurnal yang memiliki kesamaan dalam studi kasus atau metode pengembangan sistem yang digunakan. Dalam hal ini penulis menggunakan beberapa kajian sebagai bahan referensi dalam pembuatan Tugas Akhir ini, diantaranya adalah :

1. Jaya, dkk., (2018) dengan judul “Aplikasi Monitoring dan Evaluasi Kinerja Aparatur di Kejaksaan Negeri Mempawah” aplikasi ini dibuat dikarenakan dalam melakukan monitoring dan evaluasi kinerja aparatur masih menggunakan sistem yang konvensional dengan memantau kembali berkas berkas pendukung sehingga menjadikan hal ini tidaklah efektif. Hasil dari penelitian ini adalah aplikasi monitoring dan evaluasi kinerja aparatur dalam penanganan berkas perkara yang berbasis *android*, aplikasi ini mempermudah proses pengawasan dan evaluasi kerja aparatur di instansi.
2. Tjandra & Chandra., (2020) berjudul “Pemanfaatan *Flutter* dan *Electron Framework* pada Aplikasi Inventori dan Pengaturan Pengiriman Barang” berdasarkan jurnal tersebut aplikasi dibuat dikarenakan dalam pencatatan pemesanan masih dilakukan secara manual. Jika dalam jumlah pesanan terjadi dalam jumlah yang banyak mengakibatkan lambatnya pencatatan pemesanan oleh pemilik dan kemudian pencatatan inventori yang ada tidak tercatat dengan baik. Berdasarkan permasalahan ini menghasilkan sebuah aplikasi pemesanan pengiriman air secara *online* yang digunakan oleh pelanggan, aplikasi ini juga memberikan laporan penjualan secara teratur dan menampilkan stok barang yang akan habis.
3. Ramadhani, dkk., (2021) dengan judul “Pengembangan Aplikasi Monitoring Penyebaran Virus *Covid-19* Berbasis *Mobile Area* Pekanbaru dengan *Prototyping*” alasan aplikasi ini dibuat dikarenakan kebutuhan akan informasi

secara *real time* untuk masyarakat dirasa sangatlah diperlukan, sebenarnya sudah ada aplikasi *website* untuk ini tetapi masyarakat harus mengakses *website* tersebut untuk mendapatkan informasi mengenai *Covid-19*. Hasil dari penelitian ini adalah aplikasi *mobile* yang memberikan informasi secara *real time* dan juga terdapat pengembangan tambahan untuk cek status kesehatan mandiri dan juga status wilayah yang akan dikunjungi, aplikasi ini dikembangkan dengan *flutter* untuk menyesuaikan kebutuhan dari pemerintah kota Pekanbaru.

4. Kamaludin & Dharmayanti, (2019) berjudul “Aplikasi Monitoring Kurir Antar Jemput Hewan Peliharaan Pada *Petshop* Dengan Memanfaatkan *Websocket* dan *Flutter*” pada jurnal ini permasalahan yang didapat adalah perlunya pengawasan yang terhadap *staff* saat melakukan antar jemput hewan peliharaan hal ini berdampak kepada pelayanan ke pelanggan. Hasil akhir yang didapatkan adalah aplikasi *mobile* berbasis *flutter* dan menggunakan *websocket* yang memberikan informasi secara langsung dimana *staff* berada saat melakukan antar jemput dan juga layanan ke pelanggan agar bisa melakukan pemesanan penitipan hewan dan penjemputan langsung melalui aplikasi tersebut.
5. Geetha, dkk., (2020) Yang berjudul “*The Implementation of Fleet Monitoring Sistem using Mobile based Software*” permasalahan yang didapat pada jurnal ini adalah kebutuhan organisasi atau perusahaan untuk mengetahui dimana letak kendaraan milik perusahaan berada dan supir nya secara langsung setiap saat. Hasil yang didapat dalam jurnal ini adalah aplikasi *mobile* yang dapat memantau secara langsung dimana kendaraan itu berada menggunakan *GPS* yang yang dipasang di kendaraan dan juga siapa yang membawa kendaraan tersebut.

Tabel 1. Penelitian terkait

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1	Jaya, dkk., (2018)	Aplikasi Monitoring dan Evaluasi Kinerja Aparatur di Kejaksaan Negeri Mempawah	Aplikasi yang membantu kejaksaan dalam memantau kinerja aparatur dalam penanganan berkas perkara.
2	Tjandra & Chandra., (2020)	Pemanfaatan <i>Flutter</i> dan <i>Electron Framework</i> pada Aplikasi Inventori dan Pengaturan Pengiriman Barang	Aplikasi pemesanan air secara <i>online</i> untuk pelanggan dan laporan penjualan serta pencatatan stok barang untuk admin.
3	Ramadhani, dkk., (2021)	Pengembangan Aplikasi Monitoring Penyebaran Virus Covid-19 Berbasis <i>Mobile Area</i> Pekanbaru dengan <i>Prototyping</i>	Aplikasi <i>mobile</i> yang memberikan informasi secara real time ke masyarakat dan juga cek status kesehatan secara mandiri dan informasi status wilayah yang akan dikunjungi.
4	Kamaludin & Dharmayanti, (2019)	Aplikasi Monitoring Kurir Antar Jemput Hewan Peliharaan Pada <i>Petshop</i> Dengan Memanfaatkan <i>Websocket</i> dan <i>Flutter</i>	Aplikasi yang dapat memantau lokasi <i>staff</i> saat antar jemput berlangsung dan juga pemesanan penitipan dan antar jemput yang digunakan oleh pelanggan.
5	Geetha, dkk., (2020)	<i>The Implementation of Fleet Monitoring Sistem using Mobile based Software</i>	Aplikasi <i>mobile</i> yang dapat memantau lokasi dari kendaraan menggunakan <i>GPS</i> dan juga mengetahui supir dari kendaraan tersebut.

2.2. *Mobile application*

Mobile application adalah aplikasi perangkat lunak yang dapat digunakan di *smarthphone* dan *tablet*. Aplikasi *mobile* juga dikenal sebagai aplikasi yang dapat diunduh, memiliki fitur tertentu yang dapat ditambahkan ke perangkat seluler itu sendiri (Hasan dkk., 2019).

Mobile application lebih dipilih dikarenakan perkembangan dari *mobile application* semakin berkembang dan terdapat dampak positif secara global mengenai hal ini, *mobile application* dijalankan di alat yang kecil, mudah

digunakan, dan bisa dibawa juga diakses dimana saja kapan saja (Islam dkk., 2010). sekarang penggunaan *mobile application* lebih banyak digunakan oleh orang-orang untuk melakukan aktivitas mulai dari bekerja, bersosialisasi sampai mencari hiburan. dimana saja dan kapan saja pengguna bisa mengakses fasilitas yang telah disediakan, tidak hanya dari pengguna bahkan dari sisi bisnis juga mendapatkan keuntungan dari perkembangan dari *mobile application* ini. Kelebihan yang didapat menggunakan *mobile application* adalah sebagai berikut:

1. Mudah digunakan
2. Dapat digunakan dimana saja dan kapan saja
3. Memiliki performa yang cepat
4. Tampilan antar muka yang lebih bagus
5. Dapat mengakses fitur-fitur yang disediakan oleh hardware perangkat dari aplikasi tersebut.

Mobile application ini dapat dikembangkan di beberapa *platform*, terdapat 2 *platform* besar yang menguasai pasar *mobile application* ini yaitu *iOS* dan *Android*. Bahasa pengembangan untuk *platform* tersebut ada bermacam-macam contohnya untuk *iOS* terdapat bahasa *Swift* yang kemudian menggunakan *XCode IDE* sebagai tools pengembangannya, kemudian pada *Android* terdapat bahasa *Java* dan juga *Kotlin* yang bisa menggunakan *Android Studio* untuk tools pengembangannya. Terdapat juga bahasa pemrograman berbasis *Cross-Platform* yang dimana bahasa ini bisa membuat aplikasi di berbagai *platform web* maupun *mobile*, contohnya adalah *JavaScript*, *Dart*, *C#*, *C++*, dan *Python*. Dalam *mobile application* untuk melakukan pengujian dapat menggunakan metode-metode sebagai berikut:

1. Usability Testing

Adalah pengujian yang bertujuan mengetahui apakah aplikasi mudah digunakan oleh *user*, yang dimana *user* diberikan tugas untuk menggunakan aplikasi tanpa harus berpikir keras untuk menggunakannya (Lesaigne & Biers, 2016).

2. *Functional Testing*

Adalah tahap aplikasi diuji fungsionalitas apakah sudah sesuai dan diverifikasi secara benar dari seluruh fungsionalitasnya sudah berjalan dengan baik (Tramontana dkk., 2019).

3. *Performance Testing*

Merupakan tahap pengujian menganalisa seberapa efisien aplikasi merespon penggunaannya, tujuan utamanya menemukan kemacetan dalam sistem, dan juga memodelkan perilaku aplikasi yang diuji (Silva & Lopes, 2020).

2.3. *Monitoring*

Monitoring merupakan penilaian awal dan berkelanjutan terhadap kemajuan suatu pekerjaan. sedangkan menurut WHO, monitoring memberikan informasi dari pelaksanaan sebuah program, termasuk pemeriksaan secara berkala untuk melihat apakah kegiatan berjalan sesuai rencana dan jika ada masalah dapat diatasi (Jaya dkk., 2018).

2.4 *Application Programming Interface*

Application Programming Interface merupakan gabungan dari sejumlah fungsi atau perintah yang bisa digunakan oleh programmer saat membuat aplikasi. *API* juga memungkinkan perusahaan membuka data dan fungsionalitas aplikasi kepada pengembang pihak ketiga, mitra bisnis dan internal di dalam perusahaan itu sendiri (IBM Cloud Education, 2020). Hal ini memungkinkan aplikasi dapat berkomunikasi satu sama lain dan memanfaatkan fungsionalitasnya. *API* berada diantara aplikasi dan *server web*, bertindak sebagai lapisan perantara yang memproses data antar sistem.

Cara kerja *API* dijelaskan sebagai berikut

1. Klien memanggil *API* untuk mengambil informasi, dikenal juga sebagai *request*. *Request* diproses dari *server* melalui *Uniform Resource Identifier (URI) API*.
2. Setelah request valid, *API* melakukan panggilan ke program *server web*.
3. Server mengirim respon ke *API* dengan informasi diminta.

4. *API* mentransfer data ke aplikasi yang melakukan *request*.

Beberapa kelebihan menggunakan *API* seperti yang dijelaskan dalam di *website IBM* sebagai berikut:

1. Kolaborasi

API memungkinkan integrasi sehingga platform dan aplikasi dapat berkomunikasi dengan satu sama lain. Melalui integrasi ini *programmer* maupun perusahaan dapat meningkatkan kolaborasi dan optimasi alur kerja.

2. Keamanan

API menambahkan keamanan berdasarkan desain nya dikarenakan posisinya sebagai perantara antar 2 sistem. *API* memisahkan aplikasi dari infrastruktur yang menyediakan layanan.

2.5 *Representational State Transfer*

Representational State Transfer atau disebut *REST* merupakan arsitektur bagaimana menghubungkan jalur komunikasi antar mesin dan aplikasi melalui *HTTP* sederhana (Arianto dkk., 2016). Perintah *HTTP* yang biasa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*, kemudian hasil yang biasa dikirim berbentuk format *XML* atau *JSON* tanpa ada protokol pemaketan data, sehingga data dapat lebih mudah dibaca dan di *parsing* disisi *client* (Kurniawan, 2014). *REST* memiliki beberapa prinsip yang dijelaskan oleh (*IBM Cloud Education*, 2021) :

1. *Uniform Interface*

Semua *request* sumber daya yang sama akan terlihat sama dari manapun *request* berasal.

2. *Client-server decoupling*

Aplikasi *client* dan *server* harus terpisah dari satu sama lain, informasi yang diketahui oleh aplikasi klien hanyalah *URI* dari sumber daya yang diminta.

3. *Statelessness*

Dalam setiap *request* harus menyertakan semua informasi yang diperlukan untuk diproses. Aplikasi *server* tidak diizinkan untuk menyimpan data apa pun terkait request *client*.

4. *Cacheability*

Sumber daya harus dapat disimpan dalam *cache* di sisi *klien* atau *server*. Tujuannya agar dapat meningkatkan performa di sisi *klien*, sekaligus meningkatkan skalabilitas di sisi *server*.

2.6 *Dart*

Dart adalah bahasa pemrograman yang dikembangkan oleh *Google*. *Dart* awalnya dikembangkan sebagai pengganti dan penerus *JavaScript*, sehingga ia mengimplementasikan sebagian besar karakteristik penting dari standar *JavaScript* yang baru (ES7) (Wu, 2018). *Dart* dapat melakukan pemrograman asinkron. Ini adalah fungsi *non-blocking* yang memungkinkan program mengeksekusi kode lain bersamaan menunggu fungsi asinkron selesai, *Dart* juga dapat melakukan pemrograman *reactive*, yaitu pemrograman asinkron dengan *data stream* yang berarti dapat menerima data beberapa kali (Tjandra & Chandra, 2020). *Dart* juga merupakan bahasa pemrograman yang bersifat *open source* dan ditujukan untuk membuat aplikasi *multiplatform* seperti *mobile*, *desktop* dan *web*. *Dart* didesain mirip seperti bahasa pemrograman populer yang sudah ada seperti *JavaScript*, *Swift*, *Java*, *Kotlin*. Ini memungkinkan *developer* lebih familiar dan mempermudah untuk mempelajari bahasa *Dart*.

2.7 *Framework*

Framework merupakan kumpulan *function* atau prosedur dan *class* dengan tujuan tertentu yang siap digunakan sehingga mempermudah dan mempercepat pekerjaan seorang *programmer* tanpa harus membuat *function* dan *class* dari awal (Supardi & Hermawan, 2018). *Framework* ini berisi *library function* dan *class* yang mempermudah dalam proses pemrograman. beberapa fungsi fungsi menggunakan framework :

1. Meningkatkan keamanan

Platform *framework* sudah dilengkapi fitur keamanan tersendiri dari ancaman keamanan serangan siber.

2. Memungkinkan kolaborasi

Proses pemrograman didalam *framework* memungkinkan *programmer* mengerjakan proyek secara bersama tanpa kesulitan memahami bagian yang telah dikerjakan oleh rekan tim.

3. Koding yang lebih mudah dan efisien

Penulisan koding jadi lebih sederhana berkat *framework*, *programmer* hanya menggunakan *library* yang sudah disediakan didalamnya dan lebih menghemat waktu dan mengurangi resiko terjadinya *bug*.

2.8 Flutter

Flutter merupakan *cross-platform framework* atau kerangka kerja lintas platform, yang bertujuan untuk mengembangkan aplikasi *mobile application* yang berkinerja tinggi, tidak hanya dapat di jalankan pada *Android* dan *iOS* saja akan tetapi juga bisa dijalankan di *Fuschia* yang merupakan OS baru yang dikembangkan oleh Google. *Flutter* merender setiap tampilan menggunakan mesin rendering performa tinggi sendiri. sifat ini memberikan kemungkinan untuk membangun aplikasi berkinerja tinggi seperti aplikasi *native* (Wu, 2018). *Flutter* mempercepat proses menjalankan aplikasi dan menyesuaikan *widget*. Fitur yang paling penting adalah memungkinkan pengembang mendesain halaman pengguna dengan mudah untuk fleksibilitas yang lebih besar (Jaya dkk., 2018). Beberapa kelebihan menggunakan Flutter antara lain:

1. Fast

Flutter menggunakan mesin rendering sendiri yang berforma tinggi sehingga dapat membuat aplikasi berkinerja tinggi seperti aplikasi *native*.

2. Beautiful

Flutter menyediakan komponen *material design* yang berjalan baik pada *Android*, *iOS*, dan *web*. Desainer juga dapat berkreasi dengan bebas tanpa adanya batasan dari *framework*.

3. *Productive*

Fitur *hot-reload* yang dimiliki *Flutter* memungkinkan kompilasi secara *real-time*. dengan ini *programmer* dapat melihat perubahan kode pada perangkat tanpa perlu *restart*.

4. *Open Source*

Flutter bersifat *open source*, kode yang berada di *Flutter* berasal dari kontribusi *developer* dari seluruh dunia dengan banyak plugin yang sudah dibuat.

2.8.1 *Widget*

Widget merupakan elemen terpenting dalam *Flutter*. *Widget* harus menarik dan masuk akal karena pengguna melihat dan merasakan langsung. tidak hanya mengontrol dan mempengaruhi tampilan, tetapi juga menangani dan merespon tindakan pengguna (Wu, 2018). Konsep *widgets* pada *Flutter* sama seperti *component* yang berupa kumpulan blok kode untuk membangun aplikasi. *Widgets* pada *Flutter* mempunyai sebuah data yang dimiliki atau bisa disebut *state*, Berikut 2 jenis *widgets* yaitu :

1. *Stateful Widget*

Stateful widget adalah *widget* yang memiliki nilai *state* yang dapat berubah ubah, berarti hal ini membuat *widget* bersifat dinamis dan memiliki interaksi yang tidak terbatas.

2. *Stateless Widget*

Stateless Widget adalah *widget* yang memiliki nilai *state* tidak berubah maka ini membuat *widget* bersifat statis dan memiliki interaksi yang terbatas.

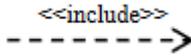
2.9 *Unified Modelling Language*

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. *UML* juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software (Touseef dkk., 2015).

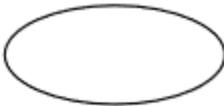
2.9.1 Use Case Diagram

Use case diagram adalah salah satu *object oriented diagram*. *use case* menunjukkan bagaimana sistem berinteraksi dengan entitas eksternal. Jadi, relatif jarang tentang perincian tentang bagaimana sistem berperilaku secara internal dan bagaimana lingkungan eksternal diatur (Aleryani, 2016). Berikut merupakan simbol simbol yang digunakan dalam use case diagram :

Tabel 2. Simbol Use Case.

Simbol (1)	Keterangan (2)
	<u>Use Case</u> Mengilustrasikan persyaratan fungsional yang diberikan oleh semua pemangku kepentingan
	<u>Actor</u> Pengguna sistem disebut aktor dan dapat berupa orang, perangkat, atau sistem lain.
	<u>Association</u> Merupakan Abstraksi dari penghubung antara peran orang dengan use case..
	<u>Include</u> Berada di antara dua use case di mana perilaku use case yang bergantung secara eksplisit disertakan di dalam use case dasar.

Tabel 2. Lanjutan Simbol Use Case.

Simbol	Keterangan
(1)	(2)
	<p><u>Use Case</u></p> <p>Mengilustrasikan persyaratan fungsional yang diberikan oleh semua pemangku kepentingan</p>

Sumber: (Sabharwal dkk., 2017)

2.9.2 Activity Diagram

Activity diagram adalah *diagram* perilaku yang menggambarkan perilaku internal operasi yang berbeda dari suatu program dengan bantuan *node* dan *edge*. digunakan untuk mewakili aktivitas yang berbeda, sub aktivitas, transisi, titik keputusan, kondisi penjagaan, aktivitas bersamaan, cabang, penggabungan, *swimlane*, *forking*, dll (Touseef dkk., 2015). Berikut merupakan simbol simbol yang digunakan dalam use case diagram (Touseef dkk., 2015):

Tabel 3. Simbol Activity Diagram

Simbol	Keterangan
(1)	(2)
	<p><u>Initial Node</u></p> <p>Merupakan awalan dari status.</p>
	<p><u>Activity</u></p> <p>Keterangan aktivitas.</p>
	<p><u>Decision Node</u></p> <p>Percabangan yang memiliki lebih dari 2 aktivitas.</p>

Tabel 3. Lanjutan Simbol Activity Diagram.

Simbol	Keterangan
(1)	(2)
	<u>Final Node</u> Status akhir dari sebuah aktivitas.
	<u>Swimlane</u> Memecah organisasi bisnis yang wajib terhadap aktivitas yang terjadi.

Sumber: (Touseef dkk., 2015)

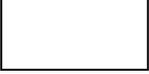
2.10 Mapping Chart

Mapping Chart merupakan diagram yang menyajikan alur dokumen dalam suatu sistem informasi dan menunjukkan pergerakan data dari satu lokasi ke lokasi lainnya (Ibnu Ridho, 2013). Simbol yang digunakan dalam *Mapping Chart* adalah sebagai berikut:

Tabel 4. *Mapping Chart*

Simbol	Keterangan
	Menunjukkan dokumen berupa dokumen input dan output pada proses manual maupun berbasis komputer
	Menunjukkan proses yang dilakukan secara manual
	Menunjukkan penyimpanan data non-komputer atau informasi pada proses manual. Dokumen dapat disimpan pada lemari, arsip, map file dan lain-lain
	Menunjukkan alir dokumen yang terputus atau terpisah pada halaman alir dokumen yang sama
	Garis alir dokumen
	Menunjukkan media penyimpanan data pada proses berbasis komputer

Tabel 4. Lanjutan Simbol *Mapping Chart*

Simbol	Keterangan
	Menunjukkan proses yang dilakukan secara komputerisasi
	Menunjukkan inout yang dimasukkan melalui keyboard
	Menunjukkan penyimpanan data secara manual

Sumber: (Ibnu Ridho, 2013)

2.11 *Rapid Application Development*

Rapid Application development merupakan salah satu model Sistem *Development Life Cycle*. dalam model *RAD* pengembangan sebuah aplikasi dapat dilakukan dalam waktu yang relatif singkat dalam kurun waktu 60-90 hari (Mishra & Dubey, 2013). Dalam model *RAD* hal yang menjadi dasarnya adalah pengembangan yang cepat dan berkualitas dengan investasi relatif rendah, mengurangi resiko dalam proyek dengan menjadikannya segmen yang lebih kecil memungkinkan proses perubahan selama pengembangan, keterlibatan pengguna secara aktif sangatlah penting (Lee, 2012). Dalam pengembangan *RAD* terdapat 3 tahapan yaitu:

1. **Rencana Kebutuhan (*Requirement/Planning*)**

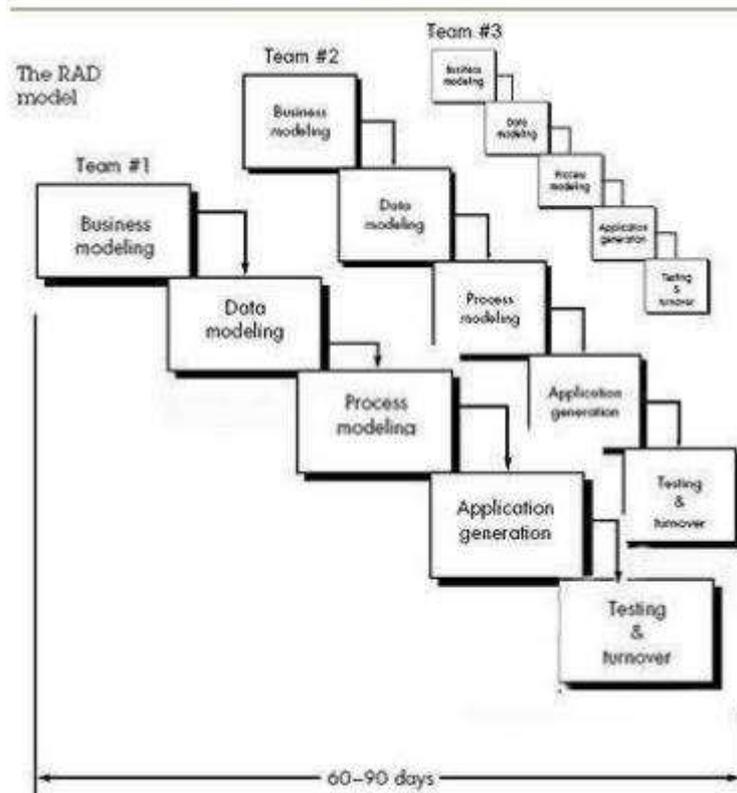
Di tahapan ini mengidentifikasi dan menganalisis layanan, Batasan, dan objeksitas dari sistem.

2. **Proses Desain Sistem (*System Design*)**

Dalam tahapan ini melakukan rancangan sistem berdasarkan tahap sebelumnya.pada tahapan planning dan desain terkait erat untuk mencapai rancangan yang dibutuhkan

3. **Implementasi (*Implementation*)**

Di tahap ini dilakukan pengecekan terhadap program apakah terdapat kesalahan atau tidak.jika proses pengecekan berhasil maka aplikasi telah berhasil



Gambar 2 *Rapid Application Development*

Sumber: (Mishra & Dubey, 2013)

2.12 *System Usability Scale*

System Usability Scale adalah metode yang digunakan untuk mengukur *usability* sebuah sistem menurut pandangan subjektif pengguna (Brooke, 2013). *System Usability Scale* merupakan kuesioner yang berisi 10 pertanyaan, responden diharapkan dapat memberikan penilaian yang subjektif berdasarkan 10 *item* pertanyaan yang diberikan. Cara perhitungan setiap pertanyaan bersifat positif atau bernomor ganjil. Nilai posisi skala dikurangi 1. Sedangkan untuk pertanyaan bersifat negatif atau bernomor genap, didapatkan nilai 5 dikurangi nilai posisi skala, kemudian tambahkan semua nilai yang didapat, selanjutnya dikalikan 2,5 untuk mendapatkan nilai keseluruhan *System usability scale*. Menurut (Ependi dkk., 2019) kelebihan yang didapat dalam menggunakan *System usability scale* diantaranya:

1. Skala penilaian yang diuji mudah dimengerti responden.
2. Dilakukan dengan sampel yang kecil dan dapat diandalkan.
3. Dilakukan secara efektif karena dapat membedakan perangkat lunak yang mampu ataupun tidak.