

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi internet saat ini yang terus meningkat sangat memudahkan manusia untuk melakukan segala aktivitasnya. Mulai dari aktivitas untuk kepentingan perdagangan, sosial, pendidikan hingga pekerjaan. Berdasarkan hasil survei yang dilaksanakan pada tahun 2016 oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) melaporkan bahwa jumlah pengguna internet meningkat menjadi 132,7 juta pengguna dibandingkan tahun sebelumnya. Angka tersebut menyatakan jumlah yang melampaui setengah dari total penduduk Indonesia, yaitu sekitar 51,7% (APJII, 2016).

Menurut Oktivasari & Sanjaya, (2015) *Internet Service Provider (ISP)* adalah perusahaan yang menawarkan jasa pelayanan kepada kita untuk berhubungan dengan internet. PT Lintas Data Multimedia merupakan salah satu ISP yang ada di Provinsi Lampung yang menyediakan layanan teknologi berbasis internet untuk perusahaan yang bermitra serta menyediakan internet untuk kebutuhan setiap masyarakat Indonesia. Perusahaan yang berlokasi di Jl. Gn. Galunggung Raya No. 13 Perumnas Way Halim ini bermitra bisnis dengan beberapa perusahaan seperti *Interlink*, Telkom Indonesia, Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), PT Media Akses Global Indo, *Indonesia Network Information Centre (IDNIC)*, Indosat, Pgncom, Telematika Media Solusi (TMS), dan *Greenet*.

PT Lintas Data Multimedia juga memiliki klien yang bermitra untuk membangun proyek pemasangan jaringan seperti instansi-instansi pemerintahan. Dalam pengerjaan proyek pemasangan jaringan tersebut dibagi menjadi dua tenaga kerja yaitu supervisor dan teknisi. Supervisor bertugas mengelola dan mengawasi alur kerja yang dilakukan teknisi. Sedangkan teknisi bertugas melakukan instalasi pemasangan jaringan, memantau serta memelihara sistem jaringan tersebut.

Mekanisme pemasangan jaringan yang sedang berjalan saat ini supervisor membuat jadwal dan *form* pemasangan jaringan di pagi hari yang kemudian

diserahkan ke teknisi, kemudian disiapkan kebutuhan barang sesuai *form* pemasangan yang sudah dibuat. Setelah pemasangan jaringan selesai teknisi memberikan bukti pemasangan berupa foto yang dikirimkan ke supervisor melalui *WhatsApp*, teknisi juga menyerahkan form pekerjaan ke supervisor untuk ditandatangani yang kemudian akan direkap dan dibuatkan berita acara untuk diserahkan ke direktur.

Disini supervisor dan teknisi masih menggunakan kertas *form* sebagai alat acuan dalam pemasangan jaringan, dan juga pada saat teknisi mengirimkan bukti pemasangan jaringan yang berupa foto masih dikirimkan melalui *WhatsApp*. Data-data tersebut belum tersimpan di *database*, sehingga rentan kehilangan data serta dapat memperlambat supervisor untuk penyusunan laporan pemasangan jaringan.

Berdasarkan permasalahan tersebut, maka dibutuhkan aplikasi monitoring pemasangan jaringan yang terdiri dari *frontend*, *backend*, dan *database*. Pada penulisan laporan tugas akhir ini penulis hanya memfokuskan pada pembuatan desain *database*.

1.2 Tujuan

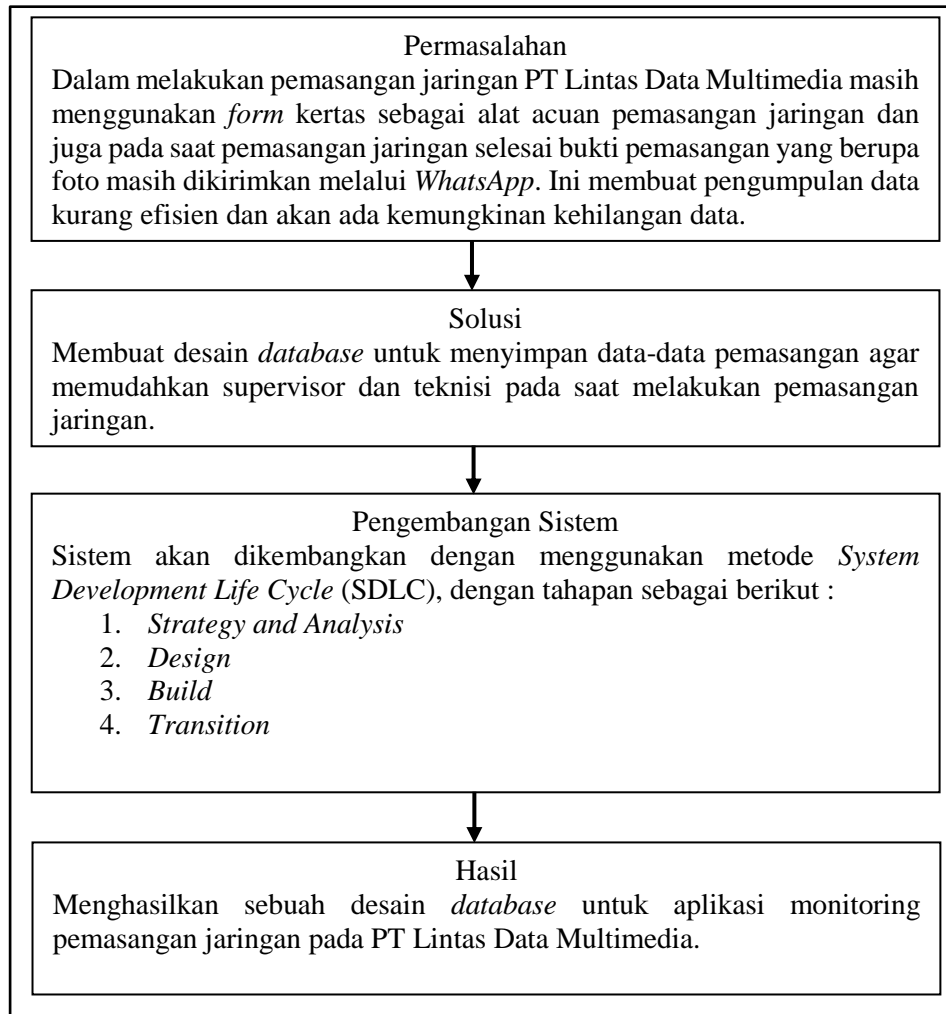
Tujuan dari penulisan tugas akhir ini adalah menghasilkan desain *database* untuk aplikasi monitoring pemasangan jaringan yang dapat digunakan oleh supervisor dan teknisi PT Lintas Data Multimedia saat melakukan pemasangan jaringan.

1.3 Kerangka Pemikiran

Semua proses yang dilakukan untuk pemasangan jaringan pada PT Lintas Data Multimedia saat ini masih kurang efisien, karena supervisor dan teknisi masih menggunakan kertas *form* sebagai alat acuan dalam pemasangan jaringan, dan juga pada saat teknisi mengirimkan bukti pemasangan jaringan yang berupa foto masih dikirimkan melalui *WhatsApp*. Semua data-data tersebut belum tersimpan di *database*, sehingga berkemungkinan besar data-data tersebut akan hilang.

Berdasarkan pemaparan masalah tersebut, maka diperlukan sebuah solusi untuk menyimpan data-data tersebut agar mempermudah supervisor dan teknisi dalam melakukan pemasangan jaringan, yaitu dengan membuat desain *database*

untuk aplikasi monitoring pemasangan jaringan pada PT Lintas Data Multimedia. Metode yang dipilih untuk pengembangan sistem ini ialah metode *System Development Life Cycle* (SDLC). Tahapan kerangka pemikiran dalam pembuatan desain *database* aplikasi ini disajikan pada gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Dengan adanya desain *database* aplikasi ini diharapkan dapat memberikan kontribusi kepada beberapa pihak antara lain :

1. Mempermudah untuk memonitoring serta mempermudah dalam pelaporan pekerjaan pemasangan jaringan.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait ialah teori dari banyak sumber yang dijadikan referensi penyusunan tugas akhir ini. Referensi yang diambil dari jurnal tersebut dapat berupa kesamaan studi kasus atau penggunaan metode. Dalam penyusunan tugas akhir ini, penulis mengambil beberapa kajian pustaka sebagai bahan acuan dalam pembuatan “Perancangan Desain *Database* pada Aplikasi Monitoring Pemasangan Jaringan pada PT Lintas Data Multimedia”, diantaranya jurnal dari :

1. Habibah dkk., (2017) yang berjudul “Desain *Database Daily Target* PE4K+BB pada PT CCAI Bandar Lampung”. Alasan dibuatnya desain *database* ini karena masih digunakannya sistem manual untuk mengolah data laporan *daily target* PE4K-BB di PT CCAI Bandar Lampung dan belum ada aplikasi untuk mengolah data laporan tersebut. Hasil dari penelitian ini yaitu dibuatnya desain basis data untuk membantu pembuatan aplikasi pengelolaan data laporan *daily target* PE4K+BB.
2. Martono dkk., (2017) yang berjudul “Desain Basis Data Sistem Informasi Akademik SMP XYZ Dengan Menggunakan *Oracle SQL Developer Data Modeler*”. Alasan dibuatnya desain *database* ini karena pelayanan sistem informasi akademik pada SMP XYZ masih menggunakan *Microsoft Office* dan sebagian juga masih menggunakan pembukuan sebagai sarana pengelolaan data. Hasil dari penelitian ini yaitu menghasilkan desain *database* yang diciptakan untuk mendukung pelayanan akademik pada SMP XYZ sehingga dapat memudahkan dalam pengelolaan data. Metode yang digunakan dalam penyelesaian *database* ini yaitu metode *System Development Life Cycle (SDLC)*.
3. Agung, (2016) yang berjudul “Perancangan *Database* untuk Aplikasi Sistem Kemahasiswaan pada *Mobile*”. Alasan dibuatnya perancangan *database* ini karena permasalahan kebutuhan akan penyampaian informasi perkuliahan yang terkadang selalu berubah-ubah setiap saat. Perancangan

database untuk aplikasi sistem kemahasiswaan ini merupakan wacana yang akan dikembangkan untuk mendukung aplikasi sistem kemahasiswaan pada *mobile* sebagai sebuah solusi untuk memenuhi kebutuhan akan penyampaian informasi. Hasil akhir dari perancangan *database* ini yaitu mampu mengimplementasikan sistem yang memenuhi kebutuhan untuk menampilkan data informasi perkuliahan berupa nilai semester, jadwal kuliah, jadwal ujian, dan informasi lain yang berhubungan dengan perkuliahan.

4. Khotijah, (2016) yang berjudul “Desain *Database* Sistem Informasi Akademik pada Lembaga Pendidikan Tinggi”. Alasan dibuatnya desain *database* ini karena pelayanan akademik yang terjadi pada lembaga pendidikan tinggi ini masih menggunakan sistem seadanya dengan cara yang manual, cara yang manual tersebut mengakibatkan terlambatnya informasi yang diterima oleh mahasiswa terutama informasi mengenai hasil studi mahasiswa. Hasil dari penelitian ini adalah menghasilkan sebuah desain *database* sistem informasi akademik pada lembaga pendidikan tinggi sehingga dapat menyelesaikan semua permasalahan terutama menyangkut hasil studi mahasiswa.
5. Yudo, (2018) yang berjudul “Perancangan Sistem Basis Data Online Monitoring Kualitas Air di Sungai Ciliwung”. Alasan dibuatnya penelitian ini karena terus bertambahnya pencemaran lingkungan khususnya pencemaran sungai Ciliwung. Salah satu upaya untuk mengendalikan pencemaran yaitu pemantauan kualitas air baik yang akan masuk ke sungai, atau di lokasi aliran sungai. Pemantauan ini diperlukan secara terus menerus dengan cara menganalisis kualitas air. Oleh karena itu, penulis melakukan sebuah penelitian dan menghasilkan sebuah perancangan sistem basis data pemantauan secara online dan *real-time* untuk menyimpan data kualitas air setiap saat yang terintegrasi dengan baik. Pemantauan ini dilakukan secara *real-time* dan online, dimana data kualitas air dikirim ke pusat data dan dinalisa menjadi informasi yang dapat ditampilkan dimana saja dan setiap saat melalui internet.

Tabel 1. Penelitian Terkait

Judul	Peneliti	Tahun	Hasil Penelitian
Desain <i>Database Daily</i> Target PE4K+BB pada PT CCAI Bandar Lampung	Habibah dkk.,	2017	Desain basis data untuk membantu pembuatan aplikasi pengelolaan data laporan <i>daily</i> target PE4K+BB.
Desain Basis Data Sistem Informasi Akademik SMP XYZ Dengan Menggunakan <i>Oracle SQL Developer</i> Data <i>Modeler</i>	Martono dkk.,	2017	Desain <i>database</i> untuk mendukung sistem pelayanan akademik pada SMP XYZ.
Perancangan <i>Database</i> untuk Aplikasi Sistem Kemahasiswaan pada <i>Mobile</i>	Agung,	2016	Perancangan <i>database</i> untuk aplikasi sistem kemahasiswaan.
Desain <i>Database</i> Sistem Informasi Akademik pada Lembaga Pendidikan Tinggi	Khotijah,	2016	Desain basis data sistem informasi akademik pada lembaga pendidikan tinggi.
Perancangan Sistem Basis Data Online Monitoring Kualitas Air di Sungai Ciliwung	Yudo,	2018	Perancangan sistem basis data pemantauan secara online kualitas air sungai ciliwung.

2.2 *Database*

Database merupakan sekumpulan *record* data yang tersimpan di dalam komputer secara terintegrasi agar dapat memenuhi informasi yang diperlukan oleh pengguna secara optimal (Nurmalina, 2017). Sebagai salah satu komponen penting yang mendasari sebuah sistem informasi, *database* diibaratkan sebuah wadah untuk menyimpan semua data yang diproses oleh sistem untuk mengeluarkan *output* laporan informasi yang diinginkan.

Data yang tersimpan di *database* tak hanya disimpan begitu saja, namun dikelompokkan, ditata sedemikian rupa, diberikan kode unik agar mampu diakses dengan cepat dan tepat sesuai kebutuhan. Data yang ada pun telah difilter agar sebisa mungkin tidak ada data yang sama. Semua itu dilakukan untuk menyiapkan data agar mampu menjadi informasi yang bermanfaat bagi semua pengguna sistem informasi. Agar dapat menjadi informasi, data yang ada pada *database* masih perlu dikelola menggunakan perangkat lunak yang biasa disebut DBMS (Bukhari, 2016).

Satu *database* menampilkan sekumpulan data, dimana data ini digunakan pada lingkup sebuah perusahaan ataupun instansi. Untuk menghasilkan laporan berisi informasi yang tepat dan rapi, maka data di *database* haruslah diolah terlebih dahulu. DBMS akan membantu user untuk mengolah data sehingga lebih mudah diurutkan dan disusun menjadi laporan yang diinginkan (Sovia & Febio, 2017). Menurut Akhmad & Ade Saputra, (2010) dalam penyusunan data yang berjumlah banyak atau besar, akan ada kemungkinan terjadinya masalah. Inilah kegunaan *database*, beberapa masalah yang dimaksud seperti :

1. Redudansi serta inkonsistensi data

Tanpa *database*, kemungkinan sebuah data untuk tersimpan lebih dari 1 kali di tempat atau *folder* berbeda sangatlah tinggi. Ini akan mengambil lebih banyak *space storage* dan menurunkan kecepatan akses. Menyimpan data yang berulang-ulang juga bisa menyebabkan inkonsistensi (tidak konsisten). Dengan adanya *database* diharapkan masalah ini dapat teratasi.

2. Kesulitan dalam pengaksesan data

Sebagai contoh misalnya untuk mengetahui sejumlah komponen mesin. Dengan adanya *database* permasalahan ini dapat diatasi. Data akan diberi kode unik, sehingga dapat memudahkan pengaksesan semua detail data tersebut dengan hanya menginputkan kode unik tersebut.

3. Isolasi data untuk standarisasi

Format *file* atau data yang berbeda dapat mempersulit penulisan program untuk menyimpan serta mengambil data. Sehingga untuk mengatasi masalah ini, data pada *database* dibuat dalam format yang sama sehingga pembuatan program lebih mudah.

4. *Multiple user*

Basis data umumnya dirancang untuk digunakan oleh banyak orang, jadi apabila ada kebutuhan *update* data, semua *user* dapat merubahnya. Dengan demikian ini dapat menghemat waktu dan mempercepat semua daya guna sistem.

5. Masalah keamanan (*security*)

Meskipun mendukung *multiple user*, *database* tetap dilengkapi keamanan untuk mengatur akses pada setiap data yang ada. Ini dikarenakan ada

beberapa data yang sifatnya pribadi dan hanya dapat diakses oleh divisi tertentu.

6. Masalah integritas (kesatuan)

Data dalam *database* haruslah berkaitan agar dapat menghasilkan informasi atau laporan. Untuk itu dibuatlah sebuah kode unik sebagai kunci untuk menyambungkan data yang ada. Dengan begitu data akan dapat menjadi kesatuan untuk menghasilkan informasi.

2.3 Perancangan

Menurut Triyono dkk., (2018) perancangan ialah proses perencanaan akan suatu hal. Perancangan ini merupakan gambaran yang dihasilkan berdasarkan bentuk-bentuk kreatif yang sudah direncanakan. Permulaan dari perancangan adalah ide atau gagasan pokok, kemudian diolah sedemikian rupa agar lebih teratur sehingga dapat menghasilkan fungsi serta kegunaan yang baik.

Salah satu langkah dalam pembuatan aplikasi yang utama adalah perancangan. Tujuannya adalah agar *programmer* mendapatkan gambaran keseluruhan sistem aplikasi dengan jelas. Perancangan sendiri adalah prosedur untuk merubah hasil analisa kebutuhan sistem kedalam bahasa pemrograman untuk dideskripsikan secara rinci dan diimplementasikan pada komponen sistem (Riko Rivanthio, 2018). Menurut Zuhri dkk., (2018) tujuan dari sebuah perancangan antara lain :

1. Pemenuhan spesifikasi secara fungsional.
2. Memenuhi batasan-batasan media sasaran implementasi dan sasaran sistem komputer.
3. Memenuhi kebutuhan-kebutuhan *implisit* serta *eksplisit* bersumber pada kinerja serta pemakaian sumber daya.
4. Memenuhi kebutuhan-kebutuhan *implisit* serta *eksplisit* bersumber pada wujud hasil rancangan yang dikehendaki.
5. Menyajikan keterbatasan proses perancangan seperti durasi pengerjaan dan biaya yang dibutuhkan.
6. Untuk memberikan rancang bangun kepada pemrogram dan juga teknik lain yang terlibat.

7. Untuk memastikan jika tujuan utama atau target sistem dapat terpenuhi dan masalah dapat diselesaikan.
8. Untuk memudahkan proses pembuatan *software* serta *control* dalam sistem yang akan dibangun.
9. Untuk mengoptimalkan solusi yang diusulkan melalui pengembangan sistem.
10. Agar dapat mengetahui segala aspek pendukung yang dibutuhkan untuk membangun sistem, seperti dukungan *hardware* ataupun *software*.

2.4 Perancangan Database

Menurut Abdillah, (2006) perancangan *database* ialah bagian dari *micro lifecycle*, dengan prosedur pengumpulan data serta analisis, perancangan konseptual, pemilihan DBMS, perancangan *logical*, perancangan *physical*, dan terakhir prosedur implementasi sistem *database*.

Ada tiga tahapan utama dari perancangan ini yaitu perancangan *database* secara konseptual, *logical*, dan *physical*. Tiga tahapan ini diharapkan akan dapat memenuhi beberapa tujuan, yakni :

1. Pemenuhan informasi kebutuhan *user* dan aplikasinya.
2. Mempermudah struktur informasi.
3. Untuk memberikan dukungan pada segala pemrosesan dan penampilan obyek (*response time*, *processing time*, dan *storage space*).

2.5 Database Management System (DBMS)

Database Management System (DBMS) merupakan aplikasi komputer yang digunakan untuk melakukan manajemen data berupa perubahan data, penambahan data, hapus data, dan *select* atau disebut juga dengan kegiatan memperoleh data atau informasi sesuai kebutuhan (Abhisena dkk., 2017). DBMS sebagai media manajemen data mempunyai beberapa keunggulan, antara lain :

1. Praktis

DBMS sebagai media manajemen data memberikan layanan penyimpanan permanen untuk banyak data, namun tetap dalam ukuran keseluruhan yang praktis.

2. *Up-to-date*

DBMS mampu menyajikan data dalam kondisi terbaru dan mendekati waktu *realtime*.

3. Cepat

Layanan DBMS yang ketiga adalah cepat, yang berarti DBMS ini mampu menampilkan atau menyajikan data secara cepat dan akurat.

Keutamaan DBMS sendiri adalah untuk menyediakan wadah yang mudah serta efisien untuk penarikan, penggunaan, dan penyimpanan informasi dan data. Pengelolaan yang ada pada manajemen basis data yaitu meliputi mendefinisikan struktur penyimpanan, menyediakan mekanisme untuk manipulasi data serta menyediakan keamanan dalam penarikan dan penyimpanan informasi dan data (Bukhari, 2016). Menurut Yanuardi & Permana, (2019) ada beberapa persyaratan minimal yang harus dipenuhi suatu sistem aplikasi agar dapat disebut DBMS :

1. Menyediakan sarana atau fasilitas untuk mengakses atau mengelola data.
2. Dapat menangani integritas data.
3. Dapat menangani akses data yang dilakukan.
4. Dapat menangani *backup* data.

Data menjadi komponen penting bagi semua perusahaan atau organisasi, karenanya mereka menggunakan DBMS untuk melakukan pengelolaan data yang lebih cepat, efisien, dan praktis. Pengelolaan DBMS ditangani oleh seorang tenaga ahli yang sudah spesialis menangani DBMS yang biasa disebut *database administrator*. Contoh-contoh dari DBMS adalah *PostgreSQL*, *SQL Server*, *MS Access*, *IBM DB2*, *Oracle*, *Dbase*, *FoxPro*, dsb (Nurhadi, 2018).

2.6 Aplikasi

Aplikasi adalah program yang sudah siap untuk dipakai dan dapat menjalankan serangkaian perintah atau prosedur dari pengguna aplikasi hingga menghasilkan informasi akurat. Aplikasi memiliki arti yaitu penyelesaian masalah yang dapat dilakukan dengan teknik pemrosesan data aplikasi yang berpacu dalam sebuah komputasi yang diinginkan (Widarma & Rahayu, 2017).

2.7 Monitoring

Menurut Widiastuti & Susanto, (2014) monitoring merupakan kegiatan mengumpulkan dan menganalisa informasi berdasarkan indikator yang telah disusun, dimana hasil yang didapatkan itu akan dievaluasi berulang dan dibuatlah sebuah keputusan untuk melakukan tindakan perbaikan. Kegiatan monitoring akan menghasilkan sebuah informasi yang dapat diukur untuk kemudian dievaluasi demi sebuah tujuan.

Proses monitoring merupakan proses mengumpulkan data dan mengukur kemajuan terhadap tujuan program. Berikut beberapa tujuan monitoring, yaitu :

1. Menilai kesesuaian hasil akhir kegiatan dengan rencana yang sebelumnya telah dibuat.
2. Guna mengidentifikasi kendala maupun masalah yang muncul yang kemudian segera dibuat solusinya.
3. Mengetahui hubungan antar kegiatan yang bertujuan memperoleh ukuran kemajuan.
4. Menilai apakah model pekerjaan dan manajemen yang digunakan sesuai untuk mencapai tujuan kegiatan.

2.8 Jaringan Internet

Menurut Oetomo, (2002) internet adalah serangkaian jaringan komputer yang besar dengan detail jaringan yang lebih kecil dimana jaringan itu berhubungan satu sama lainnya. Sedangkan menurut Allan, (2005) menjelaskan bahwa internet adalah kumpulan jaringan komputer yang secara fisik terhubung. Jaringan ini mampu membaca serta menguraikan protokol komunikasi yaitu *Internet Protocol (IP)* serta *Transmission Control Protocol (TCP)*.

Secara umum, jaringan internet sendiri dapat diartikan sebagai sebuah jaringan komputer yang tak terbatas yang mampu menghubungkan semua *user* dari segala penjuru dunia. Tak hanya menghubungkan, internet juga mampu menjembatani pertukaran informasi dan fasilitas layanan *surfing*. Istilah ini lebih dikenal dengan “*online*” di internet (Rohman, 2017).

2.9 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) merupakan tahapan yang dilakukan untuk menciptakan sebuah sistem sehingga dapat digunakan pada organisasi pemakai sistem tersebut. Dalam pembuatan desain *database* terdapat lima tahapan pada SDLC yang dapat diterapkan, yaitu :

1. *Strategy and Analysis*

Beberapa hal yang harus dilakukan di tahap *strategy and analysis* :

- a. Mempelajari serta menganalisis kebutuhan bisnis.
- b. Wawancara dengan pengguna sistem guna mengidentifikasi kebutuhan informasi.
- c. Menyatukan misi perusahaan dan aplikasi dengan spesifikasi sistem kedepannya.
- d. Membangun model konseptual sistem.
- e. Mengubah narasi bisnis menjadi representasi grafis kebutuhan bisnis.
- f. Memastikan serta memperbaiki pemodelan konseptual dengan analisis dan ahli.

2. *Design*

Beberapa hal yang harus dilakukan di tahap *design* :

- a. Mengonversi model yang dikembangkan pada tahap *strategy and analysis*.
- b. Memetakan entitas ke tabel, atribut ke kolom, relasi ke *foreign key*, dan aturan bisnis ke *constraints*.

3. *Build*

Beberapa hal yang harus dilakukan di tahap *build* :

- a. Menulis serta menjalankan perintah untuk membuat tabel dan juga objek pendukung untuk basis data.
- b. Mengisi tabel dengan data.
- c. Mengembangkan dokumentasi pengguna, teks bantuan, dan panduan operasi guna menunjang pemakaian serta pengoperasian sistem.

4. *Transition*

Hal yang dilakukan di tahap *transition* yaitu melakukan pengujian *database*.

2.10 Desain Database Konseptual

Menurut Umar dkk., (2019) desain *database* konseptual adalah proses pembentukan model yang didapatkan dari kebutuhan seluruh aspek data yang diperlukan untuk perancangan sistem basis data yang akan dibuat. Hasil dari perancangan desain *database* konseptual ini adalah skema konseptual dan model proses. Berikut ini langkah-langkah yang ada pada desain *database* konseptual menurut Widodo & Agustin, (2020) :

1. Identifikasi tipe *relationship*.
2. Identifikasi tipe *entity*.
3. Identifikasi dan hubungan atribut dengan tipe *entity* atau *relationship*.
4. Menentukan kandidat serta atribut-atribut *primary key*.
5. Menentukan atribut *domain*.

Pendekatan desain *database* konseptual menggunakan data relasional, yaitu dengan menggunakan teknik ERD (*Entity Relationship Diagram*) dan teknik normalisasi.

2.11 Desain Database Logis

Desain *database* logis adalah proses pemodelan dari informasi yang digunakan perusahaan berdasarkan model dan data tertentu. Hasil dari desain *database* logis ini yaitu hasil desain ERD pada DBMS yang digunakan (Umar dkk., 2019). Berikut ini langkah-langkah yang ada pada desain *database* logis menurut Widodo & Agustin, (2020) :

1. Memvalidasi dan membuat model data logis lokal untuk setiap bagian, memvalidasi dan membuat model data logis global.
2. Menggabungkan setiap model data logis lokal ke dalam satu model data logis global yang menggambarkan keseluruhan perusahaan.

Desain *database* logis menggunakan alat berupa relasi antar tabel. Relasi antar tabel adalah suatu bagian yang mendeskripsikan hubungan atau ketertarikan antara tabel-tabel yang digunakan dalam program aplikasi.

2.12 Desain Database Fisik

Desain *database* fisik merupakan tahapan mentransformasikan rancangan basis data logis ke dalam tabel-tabel basis data sesuai dengan jenis DBMS yang digunakan sehingga dapat disimpan pada media penyimpanan. *Firebase* merupakan salah satu opsi DBMS yang dapat digunakan untuk perancangan desain *database* (Umar dkk., 2019).

Desain *database* fisik menghasilkan penjabaran implementasi *database*, organisasi file, relasi dasar dan indeks untuk mengakses data serta semua kendala terkait keamanan (Handayani & Putri, 2018). Hasil dari desain *database* fisik dapat disimpan pada media penyimpanan eksternal yang sesuai dengan DBMS yang digunakan. Dalam hal ini, ada beberapa hal yang harus diperhatikan selama proses desain *database* fisik, antara lain :

1. Memperhitungkan kapasitas *disk* dan memastikan tipe dan ukuran atribut dalam entitas tidak melebihi kapasitas.
2. Merancang mekanisme keamanan data.
3. Merancang *user views*.
4. Menganalisis kemungkinan transaksi data yang terjadi antar entitas.
5. Mengendalikan kemungkinan redudansi pada entitas.

2.13 *Firebase Realtime Database*

Firebase adalah BaaS (*Backend as a Service*) milik Google. *Firebase* sendiri adalah solusi dari Google untuk memudahkan para *Mobile Apps Developer*. Dimana dengan adanya *Firebase*, para *apps developer* mampu focus pada bagian lain aplikasi tanpa menghabiskan waktu dalam pembuatan *backend* (Bachtiar, 2020).

Menurut Ramadhani dkk., (2021) *firebase realtime database* sendiri ialah *database* yang disimpan di *cloud* yang sudah disediakan oleh Google dan didukung oleh berbagai *platform* seperti *iOS*, *android* dan *web*. Data yang ada pada *Firebase* akan tersimpan dalam struktur JSON (*Java Script Object Notation*).

Firebase database dapat menggunakan *query NoSQL* untuk mengurutkan, mengambil serta memfilter data. Keunggulan teknologi *NoSQL* ini diantaranya yaitu jika infrastruktur pada *database* tersebut terdapat kerusakan, maka *database*

itu tidak akan hilang jadi *database* tersebut akan tetap tersedia. Berikut ini kemampuan pada *firebase* yang dipakai untuk pengembangan sistem :

1. Salah satu fitur dari *firebase* adalah *realtime database*. *Firestore* dapat memperbarui dan menyinkronkan data setiap kali ada perubahan data. Sinkronisasi yang dilakukan *firebase* dihitung dalam hitungan milidetik saja.
2. *Firestore* mempunyai sifat responsif meskipun sedang dalam keadaan *offline*. *Database realtime firebase* dilengkapi dengan SDK untuk menyimpan data ke *local disk*, sehingga meskipun sedang dalam keadaan *offline* pengguna akan tetap dapat menyimpan data ke aplikasi, dan akan secara otomatis melakukan *update* data setelah jaringan terhubung ke *server*.

2.14 *Firestore Cloud Firestore*

Cloud Firestore merupakan *database* terbaru dari *Firestore* melalui *Google Cloud Platform* untuk *mobile apps developers*. Dengan layanan *query* lebih lengkap dan cepat, *Cloud Firestore* ini mampu membuat data tetap terhubung dengan aplikasi yang dimiliki klien melalui *listener realtime*, juga menyediakan fasilitas saat klien dalam keadaan *offline*. Pembaharuan ini diharapkan dapat menghasilkan aplikasi yang lebih responsif tanpa bergantung dengan ada atau tidaknya sebuah jaringan internet (Ilham Firman Maulana, 2020).

Cloud Firestore adalah *NoSQL*, basis data berorientasi dokumen (*document-oriented database*). Tidak seperti *database SQL*, pada *Cloud Firestore* tidak ada tabel atau baris. Sebagai gantinya, data yang akan disimpan dalam sebuah dokumen, yang diatur dan dikelompokkan ke dalam koleksi. Setiap dokumen berisi satu set pasangan *key-value* (disebut sebagai *primary key* pada *database SQL*). *Cloud Firestore* dioptimalkan untuk menyimpan koleksi yang besar dalam dokumen yang berukuran kecil (Hakimah dkk., 2019). Adapun kemampuan utama dari *Cloud Firestore* menurut Aldy, (2019) adalah sebagai berikut :

1. Fleksibilitas

Cloud Firestore mendukung struktur data yang fleksibel. Penyimpanan data dapat disusun dalam beberapa *collection* dan sub-*collection* dengan obyek tingkatan yang kompleks.

2. Pembuatan *Query* yang Ekspresif

Pengguna diberi kebebasan untuk menggunakan *query* pada suatu bagian dokumen atau secara keseluruhan. Pengguna pun dapat membuat *query* yang berisi beberapa filter sekaligus dengan pengurutan atau *sorting*. Kueri juga diindeks secara default, sehingga performa kueri sebanding dengan ukuran kumpulan hasil, bukan kumpulan data.

3. *Update Realtime*

Seperti *Realtime Database*, *Cloud Firestore* menggunakan sinkronisasi data untuk mengupdate data pada semua *device* yang terhubung. Perbedaan dengan *Realtime Database* ialah *Cloud Firestore* dirancang mampu membuat kueri *select* yang sederhana dan efisien.

4. Dukungan *Offline*

Cloud Firestore akan terus menyimpan data secara aktif sehingga ketika aplikasi berubah *offline* akan tetap mampu menulis, membaca, mendeteksi, dan melakukan kueri data. Ketika perangkat telah tersambung dengan jaringan internet kembali, *Cloud Firestore* akan melakukan sinkronisasi data *local* kembali ke *Cloud Firestore*.

2.15 *NoSQL Database*

NoSQL adalah salah satu pendekatan yang lebih modern untuk teknologi basis data. Nama *NoSQL* dapat diartikan sebagai *Not Only SQL* yang artinya sebuah *database* yang didalamnya tidak terdapat perintah-perintah *SQL* dan bentuk penyimpanannya dapat menangani data yang terstruktur maupun tidak terstruktur hingga semi terstruktur. Relasi antar tabel pada *NoSQL* tidak diharuskan seperti layaknya pada *SQL* (Tangkuman, 2020).

Dengan menggunakan *NoSQL* para *developer* aplikasi dapat melakukan pengembangan tanpa mengonversi struktur yang ada menjadi relasional. Basis data

ini sering disebut basis data *cloud* untuk mengatasi masalah performa proses data tak struktur dalam *database* relasional. (Silalahi, 2018).


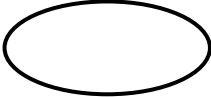
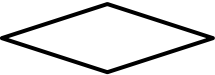
Menurut Bhugul, (2015) *NoSQL* menyimpan dan mengambil data dalam format-format yang berbeda. Terdapat 4 kategori basis data *NoSQL*, yaitu :

1. *Key value store*, ini adalah tipe simpan data yang paling sederhana dimana data yang diinputkan disimpan langsung tanpa peduli atau perlu mengetahui apa yang disimpan.
2. *Column-family*, ialah sistem *sparse matrix* yang menggunakan *row and column* sebagai kunci dalam menyimpan data.
3. *Graph databases*, adalah penyimpanan data yang menggunakan struktur secara grafik. Seperti *nodes*, *properties*, dan garis. Grafik-grafik ini menyimbolkan bagian-bagian *database* yang berbeda.
4. *Document stores*, tipe ini akan langsung menyimpan struktur data hierarki ke *database*.

2.16 Entity Relationship Diagram (ERD)

Menurut Nurmalina, (2017) ERD (*Entity Relationship Diagram*) merupakan diagram atau gambar yang menjelaskan informasi hubungan antar data dalam sistem bisnis dan menentukan hubungan relasi antar entitas. Simbol ERD disajikan pada tabel 2.

Tabel 2. Simbol ERD

Nama	Simbol	Keterangan
Entitas		Objek data yang akan disimpan atau diidentifikasi.
Atribut		Atribut merupakan informasi yang dibutuhkan dalam sebuah entitas.
Relasi		Menyatakan sebuah relasi yang menghubungkan antar entitas.

Tabel 3. Simbol ERD (Lanjutan)

Nama	Simbol	Keterangan
Link	—————	Menyatakan sebagai penghubung antara relasi entitas dengan atributnya.

Sumber: (Nurmalina, 2017)

2.17 Normalisasi

Agar *database* dapat digunakan secara efektif, maka pengelompokan yang rumit perlu dilakukan penyederhanaan sehingga dapat mengeliminasi data yang berlebih dan relasi yang salah.

Normalisasi ialah suatu teknik dalam pendesainan *database* untuk menghilangkan redudansi data sehingga struktur yang ada dalam *database* dapat terbentuk dengan baik. Normalisasi dibagi menjadi tiga tahap yaitu *First Normal Form*, *Second Normal Form*, dan *Third Normal Form* (Latief, 2010).

2.17.1 *First Normal Form* (1NF)

First Normal Form (1NF) yaitu proses menghilangkan ketergantungan sebagian. Apabila setiap baris dan kolom pada suatu entitas sudah bernilai tunggal, maka entitas tersebut sudah dikatakan dalam bentuk 1NF (Latief, 2010).

2.17.2 *Second Normal Form* (2NF)

Second Normal Form (2NF) yaitu proses menghilangkan ketergantungan transitif. Apabila suatu entitas telah memenuhi bentuk 1NF dan atribut yang bukan *key* sudah bergantung penuh terhadap *key*-nya, maka entitas tersebut dikatakan dalam bentuk 2NF. Dalam 2NF setiap entitas harus memiliki *primary key*, dimana *primary key* haruslah mewakili atribut lain yang menjadi anggotanya dan bersifat unik (Latief, 2010).





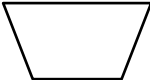
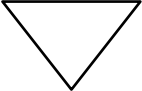
2.17.3 *Third Normal Form* (3NF)

Third Normal Form (3NF) yaitu proses mengeliminasi ketergantungan fungsional antar tabel. Suatu entitas dikatakan bentuk 3NF apabila entitas tersebut telah memenuhi bentuk 2NF dan atribut yang bukan *key* tidak bergantung transitif terhadap *key*-nya (Latief, 2010).

2.18 Mapping Chart

Menurut Liksha, (2018) *mapping chart* atau disebut juga diagram alir dokumen ialah bagan alir yang menunjukkan arus dari suatu sistem laporan. *Mapping chart* ini berfungsi untuk mempermudah melakukan analisa terhadap sistem dan sebagai sarana komunikasi antar pengguna dan analisis. Simbol *mapping chart* disajikan pada tabel 3.

Tabel 4. Simbol *mapping chart*

Simbol	Keterangan
	Dokumen.
	Dokumen beserta rangkapannya.
	Proses <i>input</i> atau <i>output</i> .
	Proses yang dilakukan oleh komputer.
	Pengolahan yang dilakukan secara manual.
	Pengarsipan.