

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada penelitian ini peneliti membuat judul dengan nama “Sistem Informasi Layanan Pelanggan Internet Service Provider PT Galunggung Access Solutions Berbasis Android”. Penelitian terdahulu yang peneliti cantumkan di bawah ini adalah sebagai acuan sehingga dapat memperkaya teori peneliti dalam mengkaji penelitian yang dilakukan. Berikut ini adalah penelitian yang peneliti kaji sebelumnya.

Penelitian yang dilakukan oleh Ira Purbo Asri dengan judul ‘Sistem Informasi Internet Service Provider Primanet Sragen Berbasis Web’ bertujuan membuat website sistem informasi ISP (*Internet Service Provider*) Primanet Sragen. Terdapat dua permasalahan yang dialami oleh ISP (*Internet Service Provider*) Primanet Sragen yaitu permasalahan yang menyangkut promosi produksi dan informasi status berlangganan seperti informasi tagihan internet dan *traffic*. Untuk mengatasi permasalahan tersebut dibuat website, penerapan metode website ini dapat dengan mudah diakses baik admin, *user*/ pelanggan maupun pengunjung sehingga dapat menarik minat masyarakat untuk berlangganan internet di Primanet Sragen dan dapat memberikan informasi status berlangganan kepada *user*/ pelanggan seperti tagihan internet setiap bulannya [4].

Selanjutnya, penelitian yang dilakukan oleh Abednego, Gilbert, and Humisar Hasugian dengan judul ‘Rancangan Sistem Informasi Administrasi Customer

Support Internet Service Provider Pada Globe Net' yang bertujuan untuk membuat sistem informasi administrasi customer support internet service provider pada Globe. Terdapat permasalahan pada Globe Net yaitu dalam pembuatan laporan dianggap masih kurang efektif karena masih menggunakan microsoft excel dan harus mencari data di email, selanjutnya sulit mengetahui teknisi yang sudah selesai mengerjakan aktivasi, troubleshoot atau terminate, sehingga sulit menentukan teknisi yang akan melakukan aktivasi dan troubleshoot berikutnya. Untuk mengatasi tersebut maka dibuatkan sistem informasi administrasi customer support internet service provider pada Globe Net untuk entry data pelanggan, entry registrasi, dan cetak surat tugas. Sehingga dapat memudahkan dalam hal administrasi bagi customer support Globe Net [5].

2.2 Teori-teori Dasar

Berikut ini adalah istilah-istilah atau teori-teori dasar dalam penelitian yang tentunya harus peneliti ketahui dan mengerti guna melakukan penelitian dengan baik.

2.2.1 Sistem Informasi

Menurut Sutabri (2012), Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasional organisasi yang bersifat manajerial dengan kegiatan strategi dari organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang yang diperlukan [6].

2.2.2 Internet

Menurut Sibero (2011:10) Internet adalah jaringan komputer yang menghubungkan antar jaringan secara global, internet juga dapat disebut jaringan alam suatu jaringan yang luas [7]. Seperti halnya jaringan komputer lokal maupun juga jaringan komputer area, internet juga menggunakan protokol komunikasi yang sama yaitu menggunakan TCP/IP (Transmission Control Protocol /Internet Protocol).

2.2.3 Internet Service Provider

ISP berasal dari kata internet yang berarti yaitu hubungan komputer dengan berbagai tipe yang membentuk sistem jaringan yang mencakup seluruh dunia, service yang berarti jasa atau layanan sehingga pengertian ISP adalah penyedia layanan internet penyedia jasa internet yakni suatu lembaga atau juga pengusaha yang menghubungkan komputer pengguna dengan internet (internet service provider) [8].

Pengertian ISP adalah perusahaan yang bergerak dalam jasa pelayanan internet. Jika kita ingin terhubung dengan ke jaringan internet tentu terlebih dahulu harus terhubung ke sebuah ISP tertentu dengan memenuhi syarat-syarat yang diberikan oleh ISP tersebut, mulai dari besarnya biaya yang dibebankan, kecepatan transfer data, dan batas waktu untuk mengakses internet.

2.2.4 Metode pengembangan *prototype*

Metode pengembangan perangkat lunak yang peneliti gunakan adalah *prototype*. *Prototype* ini metode pengembangan perangkat lunak yang berfokus untuk menghubungkan pemahaman kebutuhan pengguna dengan pengembangan

agar persepsi antara pengguna dan pengembang itu menjadi sama. Untuk dapat berinteraksi dalam proses kegiatan pengembangan maka metode *Prototype* akan menghasilkan sebuah *prototype* sistem yang akan dibuat sebagai tahapan awal dari sistem [9].

Tujuan peneliti menggunakan *prototype* ini agar pengembangan sistem dapat mengumpulkan informasi dari pengguna terlebih dahulu sehingga bisa dengan mudah menggambarkan model interaksi yang akan dikembangkan.

Untuk dapat menjalankan metode pengembangan *prototype* ini ada beberapa langkah-langkah yang akan dilakukan sebagai berikut:

1. Identifikasi Kebutuhan

Dalam mengumpulkan kebutuhan akan melibatkan pertemuan antara pengembang dengan pengguna untuk dapat menentukan apa yang menjadi tujuan dan kebutuhan dibuat sistem atau perangkat lunak ini.

2. Membuat Prototipe

Desain yang cepat ini mengarah ke pembangunan prototipe yang akan dihasilkan mencakup seperti apa *input*-nya, proses dan *output*-nya.

3. Menguji Prototipe

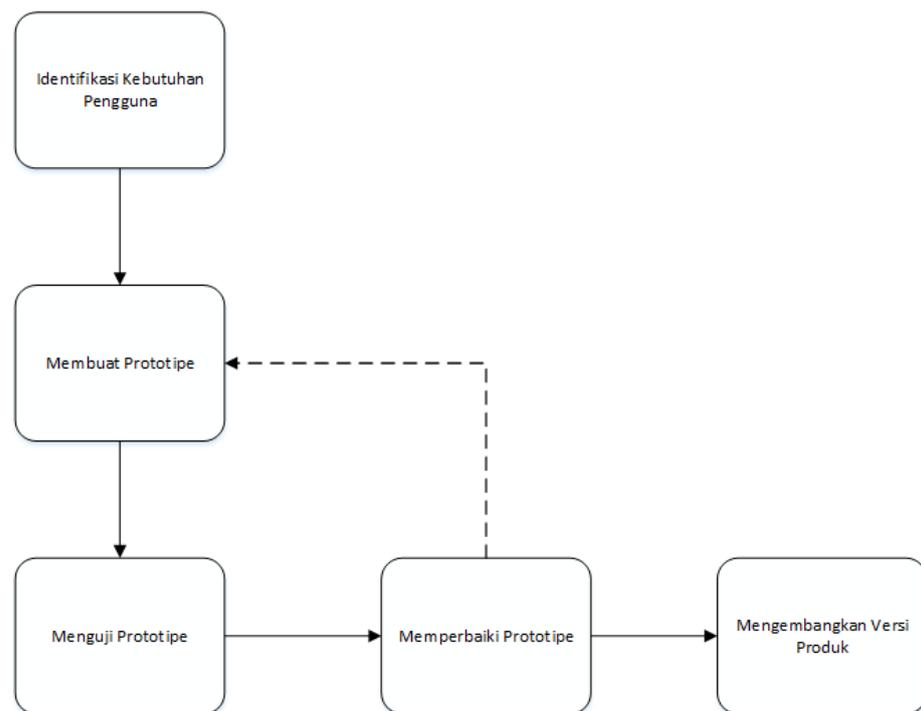
Dalam proses ini prototipe yang sudah dibangun akan di evaluasi bersama dengan pengguna dan pengembang untuk dapat menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. Pada tahap ini pengguna dan pengembang akan lebih jelas dan detail memahami apa yang perlu dilakukannya.

4. Memperbaiki Prototipe

Setelah beberapa tahapan sudah dilakukan selanjutnya membangun perangkat lunak dan kemudian melakukan pengujian dan perbaikan jika masih ada perubahan dalam sistem yang sedang dibangun.

5. Mengembangkan Versi Prototipe

Pada tahapan terakhir ini adalah mengembangkan versi prototipe yang bertujuan mengembangkan versi berikutnya sehingga menjadi produk nyata yang digunakan oleh pengguna.



Gambar 2. 1 Langkah Langkah Prototype

2.2.5 Unified Modeling Language (UML)

UML ini merupakan sebuah bahasa pemodelan sistem ataupun pemodelan di bidang lain yang membutuhkan sebuah pemodelan. Pemodelan disini

menggunakan model dalam bentuk grafis yang bertujuan untuk menspesifikasi, memvisualisasi dan membangun atau mendokumentasikan sebuah sistem sehingga mudah dipahami oleh manusia atau mesin. UML ini memakai konsep berorientasi *object* bertujuan untuk memudahkan manusia atau pengembang memahami sistem yang akan dibuat dalam bentuk grafis [10].

UML ini memiliki beberapa *diagram* yang digunakan untuk melakukan pemodelan data ataupun sistem. Berikut adalah *diagram*nya:

1. *Use Case Diagram*

Use Case Diagram merupakan salah satu *diagram* UML yang bertujuan untuk mendeskripsikan hubungan antara manusia atau sistem lain dengan sistem yang mau dirancang. Selain menggambarkan hubungan antara aktor dengan sistem, *use case* ini dapat menggambarkan sebuah fungsi dari sistem.

Mudahnya adalah apa yang dilakukan oleh sistem bukan bagaimana yang dilakukan oleh sistem.

2. *Use case Scenario*

Use case scenario adalah suatu gambaran model yang berbentuk skenario sehingga nantinya terdapat kemungkinan-kemungkinan tersebut nantinya akan dipecah menjadi beberapa skenario. Di dalam proses sebuah sistem nantinya akan terdapat sekumpulan proses, proses inilah yang akan menjadi beberapa kemungkinan-kemungkinan yang kemudian bisa disebut dengan skenario *use case*.

3. *Activity Diagram*

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang terdapat dalam perangkat lunak. Yang perlu dilakukan aktor.

Activity juga menggambarkan berbagai alir aktivitas pada sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi, dan bagaimana berakhir.

Diagram Aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan juga menggambarkan berbagai alir aktivitas pada sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi dan bagaimana berakhir.
- b. Urutan atau pengelompokan tampilan dari sistem/ user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilannya.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan dari kasus ujinya.
- d. Rancangan sistem yang ditampilkan pada perangkat lunak.

4. *Class Diagram*

Class diagram merupakan *diagram* yang menggambarkan struktur sistem secara spesifikasi yang didalamnya terdapat nama kelas, atribut dan metode. *Class diagram* ini akan menghasilkan berupa objek karena berhubungan dengan konsep berorientasi objek.

Area utama yang terdapat pada dalam *class diagram* terdapat tiga yaitu nama kelas, atribut dan metode. Atribut dan metode ini mempunyai beberapa sifat yaitu *private* atribut atau metode tidak bisa dipanggil dari luar *class* dapat dipanggil di dalam *class* yang sama. *Protected* yaitu ketika atribut dan metode hanya bisa dipanggil didalam *class* yang sama dan bisa mewariskan ke anak-anaknya, dan ketika atribut dan metode memiliki sifat *public* maka seluruh atribut dan metode baik yang ada di luar kelas atau di dalam *class* yang sama bisa dipanggil untuk menggambarkan *class diagram* adalah sebagai berikut:

- a. Asosiasi merupakan garis penghubung yang menghubungkan antar *class*. *Class* disini adalah *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui ekstensi *class* lain.
- b. Agregasi merupakan garis di dalam *class diagram* yang menghubungkan pernyataan bahwa *class* yang lain.
- c. Pewarisan merupakan *class* dapat diturunkan dari *class* lain sehingga bisa menambahkan fungsionalitas baru menjadi disebut anak dari *class* yang diwarisinya.
- d. Hubungan dinamis ini adalah satuan rangkaian pesan dikirim dari *class* pada *class* lain.

5. *Sequence Diagram*

Sequence Diagram merupakan *diagram* yang menggambarkan interaksi antar objek yang ada di dalam dan diluar sistem sehingga

menjadi dinamis. Kegunaan *diagram sequence* ini adalah untuk menunjukkan rangkaian pesan yang dikirim antar *object* atau antar sistem. Pesan ini digambarkan terhadap waktu. *Diagram* ini terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Untuk menghasilkan *output* tertentu dari sebuah sistem maka harus digambarkan melalui rangkaian langkah-langkah yang diawali dengan men-*trigger* aktivitas tertentu, kemudian apa saja proses yang berubah dan dihasilkan dari sistem yang dilakukan secara internal dan *output*.

2.2.6 Pengertian Implementasi

Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan juga terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap sempurna.

Menurut Nurdin Usman, implementasi adalah bermuara pada aktivitas, aksi tindakan atau adanya mekanisme suatu sistem, implementasi bukan sekedar aktivitas, tapi suatu kegiatan yang terencana untuk mencapai tujuan kegiatan [11].

2.3 Software yang Digunakan

Pada penelitian ini peneliti menggunakan beberapa *software* untuk menunjang terselesaikannya penelitian ini dengan baik. Berikut adalah *software* yang peneliti gunakan untuk membangun dan juga mengembangkan aplikasi ini.

2.3.1 Android Studio

Android studio adalah IDE (Integrated Developer Environment) resmi terkait akan bahasa pemrograman yang digunakan. Hal tersebut karena XML standar

yang digunakan oleh banyak perusahaan besar didunia yang digunakan untuk bertukar data, selain itu penggunaan SOAP menjadi metode-metode dari objek yang ada dalam sebuah web service dapat diakses dari aplikasi lain seperti aplikasi tersebut mengakses metode local [10].

2.3.2 Firebase

Firebase adalah sebuah platform untuk sebuah aplikasi realtime, ketika data ada yang berubah maka akan mengupdate secara langsung pada setiap device baik itu *website* maupun *mobile* [10]. *Firebase* mempunyai library yang lengkap, sebagian besar platform web dan juga mobile dapat digabung dengan berbagai framework lain seperti *node*, *java*, *javascript*, dan lainnya.

Firebase terdiri dari beberapa fitur yang telah disediakan yaitu sebagai berikut:

- a. *Analytics*, yaitu dapat mengamati tingkah laku pengguna dalam penggunaan aplikasi dan ditampilkan dalam satu *dashboard*.
- b. *Develop*, fitur ini terbagi menjadi beberapa fitur seperti *cloud messaging*, *authentication*, *realtime database*, *storage*, *hosting*, *testlab* dan *crash reporting*.
- c. *Grow*, yaitu digunakan untuk mempublish sebuah produk aplikasi.

Firestore Realtime Database ini merupakan basis data *online* yang dapat digunakan sebagai media untuk penyimpanan data dari aplikasi. Data yang disimpan dalam bentuk JSON dan dapat disinkronkan secara *relatime* pada setiap *client* yang terhubung, pada layanan ini memiliki 3 kemampuan inti yaitu:

- a. *Realtime*, yaitu ketika terdapat perubahan data pada *database*, maka seluruh *client* yang terhubung secara otomatis akan mendapatkan perubahan dengan cepat.
- b. *Offline*, pada pulikasi yang menggunakan fitur ini akan tetap responsif bahkan ketika dalam keadaan luring, karena *Firestore SDK (Software Developer Kit)* dapat mempertahankan data dan perubahannya pada media penyimpanannya. Kemudian pada saat *client* ke jaringan internet, *Firestore SDK* akan melakukan penyesuaian otomatis atas perubahan data yang disimpan pada median penyimpanan client dengan kondisi paling update dari *Firestore Server*.
- c. *Accessible from client device*. Pada laynana ini menawarkan kemudahan dalam mengakses *Firestore realtime database* ini secara langsung dari sebuah perangkat *mobile* atau sebuah web tanpa membutuhkan *server application*.

Firestore merupakan *database* yang bersifat non-relation atau NoSQL, dimana database ini merupakan *database* yang tidak menggunakan sistem tabel

dalam implementasinya serta tidak menyimpan data secara lokal pada perangkat tetapi pada awan. Selain itu pada *firebase* juga memiliki optimasi yang berbeda jika dibandingkan dengan basis data relasional.

2.3.3 PHP

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan juga berfungsi sebagai pengolah data pada sebuah server. [13]

PHP juga disebut sebagai bahasa pemrograman *scrip* karena kode programnya dapat disipkan ke dalam dokument HTML. *Serverside scripting* adalah bahasa pemrograman yang berjalan pada server, sedangkan *clientside scripting* bahasa pemrograman yang berjalan disisi client, contohnya seperti *Javascript*.

Dalam membangun aplikasi ini penulis menggunakan pemrograman PHP karena memiliki beberapa keunggulan diantaranya adalah mudah dipelajari, ringkas, pengembangannya cepat, *open source*, dan juga *maintenance* yang mudah.

2.3.4 Visual Studio Code

Visual Studio Code adalah aplikasi editor yang dapat berjalan di banyak *platform* perangkat lunak seperti OS X, Ubuntu dan windows. *Visual code studio* ini dibuat pada tahun 2015 oleh Microsoft dan sampai sekarang sudah banyak yang menggunakannya. Software ini keunggulannya yaitu bisa *support* dengan berbagai macam bahasa pemrograman.

2.3.5 Xampp

Dalam membangun aplikasi ini juga peneliti menggunakan XAMPP sebagai *local server* agar aplikasi ini berjalan dengan baik ketika dikembangkan dalam komputer peneliti.

Pada situs resminya XAMPP adalah *software opensource* yang digunakan untuk membuat *server* lokal pada komputer sendiri (Localhost) dan dapat di *download* secara gratis dari situs resminya dan *software* ini berjalan di semua sistem operasi.

Htdocs adalah folder yang ada dalam XAMPP yang berfungsi menyimpan seluruh file dari aplikasi yang akan dibuat. Kemudian folder Htdocs akan ditampilkan pada *browser* menggunakan bantuan *server localhost* sehingga aplikasi yang dibuat dapat tampil dalam browser.

2.3.6 Bootstrap

Bootstrap adalah *front-end framwork* yang digunakan untuk membuat suatu tampilan website menjadi lebih mudah, lebih bagus, dan menarik yang mengedepankan tampilan yang bisa dibuka dalam semua perangkat atau device baik itu *handphone*, atau komputer. Bootstrap menyediakan HTML, CSS dan juga *Javascript* untuk memudahkan dan mempercepat pengembangan sebuah *website* dan mudah digunakan untuk para pengembang *website*.

2.3.7 CSS

CSS (*Cascading style sheet*) adalah suatu rangkaian instruksi dalam membuat sebuah website yang menentukan bagaimana suatu text, *layout* dan bentuk dapat di tampilkan di halaman *website* tersebut.

Pada teknisnya CSS ini memisahkan antara konten utama dengan tampilan seperti, *font, color*, dan juga *layout*. Fungsi dari CSS ini adalah untuk mendasaain, merancang dan merubah bentuk secara keseluruhan tampilan dari suatu *website* dengan menempelkannya pada *tag-tag html*. Sehingga *tag-tag* tersebut dapat lebih fungsional dan juga lebih menarik. Cara kerja dari CSS ini juga sangat mudah, yaitu cukup menempelkan kelas yang telah dibuat pada *file* terpisah yang berformat CSS maka kita dapat menuliskannya pada *selector id dan class* dalam HTML.