

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berdasarkan penelitian Rini Junadah, Boy Abidin R. (2016) dengan judul “Model Aplikasi Layanan Informasi Tagihan Air Berbasis *Web*” Penelitian ini bertujuan tentang dengan adanya layanan informasi berbasis *web* dapat membantu petugas administrasi dalam menyampaikan informasi tagihan kepada pelanggan yang belum mengetahui besarnya jumlah tagihannya sehingga proses pembayaran tagihan bisa segera dilakukan demi kelancaran operasional usaha. layanan informasi tagihan ini bukanlah pertama kali dilakukan, sudah ada penelitian terlebih dahulu yang dilakukan oleh orang lain. Penelitian terdahulu yang relevan dengan penelitian ini adalah yang dilakukan oleh Kartikasari [1] mengenai bagaimana membangun suatu aplikasi tagihan listrik PLN yang dapat berjalan pada perangkat mobile seperti smartphone dan komputer tablet yang memiliki sistem operasi *android*, sehingga efektifitas pelayanan manajemen kepada pelanggan dalam menyampaikan informasi tagihan.

Berdasarkan penelitian Abdul Rohmad Basar, dan Novalita Anggraeni (2017) dengan judul “Rancang Bangun Aplikasi Retribusi Pembayaran Tagihan Sampah Berbasis *Web*” penelitian ini dilakukan PT. Sacor Mandiri Jaya. Proses Pelayanan kegiatan administrasi pembayaran tagihan sampah seperti pencatatan pembayaran, pemberian bukti pembayaran dan pencarian data pembayaran di PT.

Sacor Mandiri Jaya. Masih menggunakan system manual sehingga membutuhkan system informasi yang dapat membantu PT. Sacor Mandiri Jaya.

Penelitian ini bertujuan untuk membangun suatu aplikasi sistem informasi tagihan dan pembayaran pada perusahaan sampah di PT. Sacor Mandiri Jaya Berbasis *Web*. Aplikasi sistem-sistem informasi tagihan dan pembayaran sampah ini dibangun dengan bahasa pemrograman *PHP* dan *Database MySQL*. Metode pengumpulan data dengan menggunakan metode observasi, *interview*, dan kepustakaan. Tahap pengembangan aplikasi meliputi analisis, perancangan sistem, implementasi, dan pengujian. Aplikasi Retribusi Pembayaran tagihan sampah yang dibuat dapat digunakan untuk membantu perusahaan dalam melakukan administrasi tagihan dan pembayaran tagihan sampah yang lebih efektif dan efisien, sehingga dapat meminimalisir terjadinya kesalahan maupun kelalaian dalam Pengelolaan administrasi tagihan dan pembayaran tagihan sampah PT. Sacor Mandiri Jaya. Hasil pengujian sistem menunjukkan bahwa aplikasi ini layak dan dapat digunakan sebagai alat untuk membantu mengolah data tagihan dan pembayaran sampah.

Berdasarkan penelitian Aris, dkk (2017) dengan judul “Aplikasi Sistem Pelayanan data Pelanggan Berbasis *Web* pada PDAM Tirta Benteng Kota Tangerang ” tujuan dari penelitian ini adalah membutuhkan suatu system informasi yang menunjang dan memberikan pelayanan yang memuaskan untuk pelanggan karena system yang ada di PDAM Tirta Benteng Kota Tangerang ini masih dilakukan secara komputerisasi namun dirasakan kurang optimal dalam penggunaannya, memungkinkan pada saat proses berlangsung terjadi kesalahan

dalam pencatatan, kurang akuratnya laporan yang dibuat dan keterlambatannya dalam pencarian data data yang diperlukan. Untuk mengatasi hal ini PDAM Tirta Benteng Kota Tangerang membutuhkan system pelayanan yang terpadu dan terkomputerisasi, maka dengan membuat system pelayanan data pelanggan dengan bahasa pemograman *PHP* dan metode perancangan menggunakan *UML* untuk membangun aplikasi berbasis *web* pada PDAM Tirta Benteng Kota Tangerang diharapkan pelanggan tidak perlu lagi datang ke PDAM melainkan hanya membuka *website* untuk melakukan pendaftaran, mengecek info tagihan dan pembayaran.

II.2. Landasan Teori

II.2.1. Sistem

Sistem adalah kumpulan objek seperti orang, sumber daya, konsep, dan prosedur yang dimaksudkan untuk melakukan suatu fungsi yang dapat diidentifikasi atau untuk melayani suatu tujuan (Mikha Dayan Sinaga, 2018).

Sistem didefinisikan sebagai suatu kumpulan objek yang saling berkaitan dan saling bergantung secara tetap untuk mencapai tujuan bersama dalam suatu lingkungan yang kompleks. Banyak sistem yang dipelajari atau dijumpai dalam kehidupan sehari-hari seperti sistem tata surya, sistem informasi, sistem komunikasi, sistem antrian, dan lain-lain. Di dalam setiap sistem tersebut terdapat entitas yang saling berinteraksi (Siti Aminatunnisa, dkk, 2019).

II..2.2. Informasi

Informasi adalah data yang diolah menjadi bentuk yang berarti bagi penerima dan mempunyai suatu keputusan atau bakal keputusan (Fadiel Muhammad, dan Asri Mulyani, 2016). Informasi adalah data yang diolah menjadi bentuk yang berguna dan menjadi berarti bagi penerimanya. Kegunaan informasi adalah untuk mengurangi ketidak pastian di dalam proses pengambilan keputusan tentang suatu keadaan. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut (Priyo Sutopo, dkk, 2016).

II..2.3. Sistem Informasi

Sistem informasi adalah suatu sistem organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Fadiel Muhammad, dan Asri Mulyani, 2016).

Sistem informasi adalah sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik. Sistem informasi adalah kumpulan perangkat keras dan lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna (Kasman Rukun, dan B. Herawan Hayadi, 2018).

II.2.4. Website

Website adalah lokasi di internet yang menyajikan kumpulan informasi sehubungan dengan profil pemilik situs. *Website* adalah suatu halaman yang memuat situs-situs *web page* yang berada di internet yang berfungsi sebagai media penyampaian informasi, komunikasi, atau transaksi. *Web server* merupakan *software* yang memberikan layanan data yang berfungsi menerima permintaan *HTTP* atau *HTTPS* dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML* (Rulia Puji Hastanti, dkk, 2015).

II.2.5. PHP

PHP merupakan bahasa *web server-side* yang bersifat *open source*. Bahasa PHP menyatu dengan *script HTML* yang sepenuhnya dijalankan pada *server*. *File* yang hanya berisi kode HTML yang dirancang tidak mendukung pembuatan aplikasi yang melibatkan *database* karena HTML dirancang untuk menyajikan informasi yang bersifat statis (tampilan yang isinya tetap hingga *webmaster* atau penanggung jawab *web* melakukan perubahan isi). Oleh karena itu, muncul pemikiran untuk membuat suatu perantara yang memungkinkan aplikasi bisa menghasilkan sesuatu yang bersifat dinamis dan berinteraksi dengan *database*. Akhirnya lahirlah berbagai perantara seperti *PHP*, *ASP* dan *JSP* (Andy Pratama Nugraha, dkk , 2015).

PHP secara mendasar kemampuan *php* dapat mengerjakan semua yang dapat dikerjakan oleh program *CGI* (*Common Gateway Interface*), seperti

mendapat data dari *form*, menghasilkan isi halaman *web* yang dinamik, dan menerima *cookies*. *CGI* adalah spesifikasi standar modul yang ditambahkan kepada *server web*, agar *server web* dapat memiliki kemampuan untuk dapat memberikan layanan yang interaktif, tidak sekedar melayani permintaan dokumen *web (HTML)* saja. Antar muka pemakai (*user interface*) dari aplikasi yang dibuat menggunakan *PHP CLI*, pada dasarnya adalah menggunakan node teks (Teri Ade Putra, dan Aulia Fitrul Hadi, 2018).

II.2.6. MySQL

MySQL adalah *relational database management system (RDBMS)* yang didistribusikan secara gratis dibawah *license GPL (General Public License)*. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama yaitu *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian *database* terutama untuk pemilihan/seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah dan secara otomatis. Keandalan suatu sistem *database* dapat diketahui dari cara kerja *optimizer* nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* dapat sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase* (Rina Candra Noor Santi, dan Sri Eniyati, 2015).

II.2.7. UML

UML (Unified Modeling Language) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Aris, dkk, 2016).

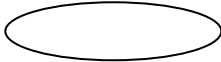
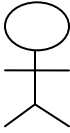
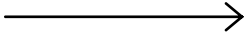
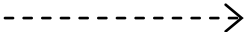
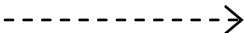
Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem . Perancangan desain sistem yang akan dibangun menggunakan pemodelan *Unified Modelling System (UML)*. Diagram-diagram yang digunakan *use case diagram, activity diagram, class diagram* dan *sequence diagram* (Dicky Juliawan, dkk, 2017).

Bagian-bagian dari *Unified Modelling System (UML)* adalah sebagai berikut :

a. Use Case Diagram

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk *diagram* yang dapat menjelaskan suatu alur proses sistem yang akan dibangun. Berikut ini merupakan simbol-simbol yang terdapat pada *Use Case Diagram*, dapat dilihat pada tabel II.1 sebagai berikut :






Tabel II.1 Simbol *Use Case Diagram*

| Simbol | Deskripsi |
|--|---|
| <i>Use Case</i>  | Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> . |
| <i>Aktor/Actor</i>  | Orang, proses atau <i>system</i> lain yang berinteraksi dengan <i>system</i> informasi yang akan dibuat diluar <i>system</i> yang akan dibuat itu sendiri. |
| <i>Relationship/Association</i>  | Dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| <i>Extensi/Extend</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. |
| <i>Include</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya. |

b. Activity Diagram

Activity diagram adalah *diagram* yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Berikut ini merupakan simbol-simbol yang terdapat pada *Activity Diagram*, dapat dilihat pada tabel II.2 sebagai berikut :





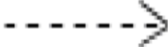

Tabel II.2 Simbol Activity Diagram

| No | Gambar | Nama | Keterangan |
|----|---|----------------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Action</i> | <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali. |
| 4 |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

c. *Class Diagram*

Class Diagram adalah deskripsi kelompok obyek -obyek dengan *property* perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan *global* atas sebuah *system*. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan yang lainnya. Berikut ini merupakan simbol-simbol yang terdapat pada *Class Diagram*, dapat dilihat pada tabel II.3 sebagai berikut :

Tabel II.3 Simbol *Class Diagram*

| No | Gambar | Nama | Keterangan |
|----|---|-------------------------|--|
| 1 |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 2 |  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 3 |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> . |
| 4 |  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 5 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

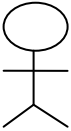
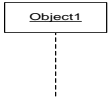


d. *Sequence Diagram*

Sequence Diagram merupakan salah satu *diagram interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Diagram* ini diatur berdasarkan waktu. Obyek-

obyek yang berkaitan dengan berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Berikut ini merupakan simbol-simbol yang terdapat pada *Sequence Diagram*, dapat dilihat pada tabel II.4 sebagai berikut :

Tabel II.4 Simbol *Sequence Diagram*

| Simbol | Deskripsi |
|---|--|
| <p>Aktor/<i>Actor</i></p>  | Menggambarkan orang yang berinteraksi dengan <i>system</i> |
| <p><i>Object Lifeline</i></p>  | Menggambarkan interaksi antara satu atau lebih <i>actor</i> dengan <i>system</i> memodelkan bagian dari <i>system</i> yang bergabung ada pihak lain disekitarnya |
| <p><i>Message</i> (Masukan)</p>  | Menyatakan bahwa sutau objek mengirimkan data atau masukan atau informasi ke objek lainnya |
| <p><i>Activation</i></p>  | Menyatakan objek dalam keadaan aktif dan berinteraksi, sebuah yang brhubungan dengan <i>activation</i> ini adalah sebuah tahapan yang dilakukan didalamnya |

| | |
|--|--|
| <p><i>Message</i> (Keluaran)</p> <p style="text-align: center;">←-----</p> | <p>Menyatakan bahwa suatu objek sudah menjalankan suatu operasi atau metode menghasilkan kembalian ke objek tertentu</p> |
|--|--|

III.2.8. Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data relasional yang mengelompokkan atribut dari suatu tabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redundansi data). Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh *base relations (file)*.
2. Untuk mengurangi kompleksitas.
3. Untuk mempermudah pemodifikasian data.

Adapun aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut :

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.

2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).

3. Tidak melanggar *Boyce-Codd Normal Form (BCNF)*.

1. Bentuk Normal Pertama / *1NF*, memiliki aturan sebagai berikut :

- a. Tidak adanya atribut *multi-value*, atribut komposit atau kombinasinya.
- b. Mendefinisikan atribut kunci.
- c. Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi lagi).

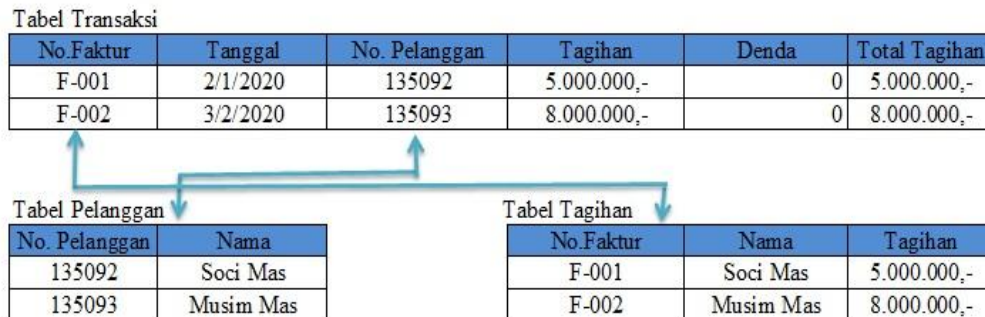
Tabel III.1 Tabel 1NF

Tabel Transaksi

| No. Faktur | No. Pelanggan | Nama | Golongan | Tanggal | Tagihan | Denda | Total Tagihan |
|------------|---------------|-----------|------------|----------|-------------|-------|---------------|
| F-001 | 135092 | Soci Mas | Perusahaan | 2/1/2020 | 5.000.000,- | 0 | 5.000.000,- |
| F-002 | 135093 | Musim Mas | Perusahaan | 3/2/2020 | 8.000.000,- | 0 | 8.000.000,- |

2. Bentuk Normal Kedua / *2NF*, memiliki aturan sebagai berikut :

- a. Sudah memenuhi dalam bentuk normal kesatu (*1NF*).
- b. Semua atribut bukan kunci hanya boleh bergantung (*functional dependency*) pada atribut kunci.
- c. Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.
- d. Perlu ada tabel penghubung ataupun kehadiran *foreign key* bagi atribut-atribut yang telah dipisah tadi.

Tabel III.2 Tabel 2NF

3. Bentuk Normal Ketiga / *3NF*, memiliki aturan sebagai berikut :

- a. Sudah memenuhi dalam bentuk normal ketiga (*3NF*).
- b. Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya) (Dwi Puspitasari, dkk, 2016).

Tabel III.3 Tabel 3NF