

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Sebagai referensi dalam melakukan penelitian, penulis melakukan studi pustaka menggunakan literatur penelitian terdahulu yang berhubungan dengan penelitian ini.

Penelitian yang dilakukan oleh Kusnita Yusmiarti (2021) dengan judul “Sistem Informasi Tagihan Rekening Air Berbasis Web”, sistem Informasi tagihan rekening air di Kecamatan Jarai saat ini masih menggunakan sistem manual, sehingga banyak sekali terdapat kekurangan yang terjadi, seperti memerlukan waktu yang cukup lama dalam memproses data, ketidak akuratan dari proses serta keterlambatan dalam memberikan informasi kepada pelanggan, berdasarkan hal inilah sehingga diperlukan sistem informasi yang dapat membantu PDAM Kecamatan Jarai dalam memberikan informasi tagihan rekening air pelanggan PDAM Kecamatan Jarai yang lebih efektif dan efisien dengan mengganti informasi tagihan rekening manual menjadi informasi tagihan rekening yang terkomputerisasi.

Penelitian yang dilakukan oleh Irma Suriyani (2020) dengan judul “SISTEM INFORMASI PEMBAYARAN REKENING AIR BERBASIS WEB PADA PAMSIMAS JORONG PANYALAI”, Program PAMSIMAS di Jorong Panyalai Nagari Cupak ini sudah dijalankan selama 2 tahun namun pelaksanaan program PAMSIMAS ini belum optimal terutama dibidang administrasi pembayaran rekening air perbulannya banyak pelanggan PAMSIMAS di Jorong Panyalai mengeluhkan dengan sistem pembayaran yang sedang berjalan, dikarenakan proses pembayaran rekening air pada PAMSIMAS Jorong Panyalai ini masih dengan cara manual. Untuk itu penulis menyarankan untuk dibagunnya sebuah sistem informasi pembayaran rekening air pada pamsimas jorong panyalai berbasis web dengan adanya sistem tersebut diharapkan dapat memudahkan petugas dalam melayani pelanggan dalam pembayaran rekening air dengan hasil

lebih efektif dan lebih akurat, selain itu dengan adanya sistem informasi dapat memudahkan dalam pengolahan data pelanggan dan penyajian laporan seperti laporan data pelanggan dan laporan pembayaran rekening air secara terkomputerisasi.

Penelitian yang dilakukan oleh Nofyat dkk (2018) dengan judul “Sistem Informasi Pengaduan Pelanggan Air Berbasis Website Pada Pdam Kota Ternate”, Perusahaan Daerah Air Minum (PDAM) Kota Ternate merupakan Badan Usaha Milik Daerah (BUMD) yang pelayanan pengaduan air kepada pelanggan masih bersifat konvensional dimana pelanggan yang ingin mengadukan keluhannya harus datang ke kantor unit wilayah PDAM Kota Ternate proses pengaduanpun hanya dapat dilakukan pada hari kerja yaitu Senin - Jumat mulai dari Pukul 07.30 - 15.00 wit. Diluar waktu tersebut pelanggan tidak dapat menyampaikan keluhannya sehingga dianggap kurang efektif. Sistem Informasi Pengaduan dapat mempermudah pelanggan dalam melakukan pengaduan di sistem tanpa harus datang ke kantor PDAM Kota ternate kapan saja pada sistem serta dapat meningkatkan kinerja dan pelayanan petugas informasi dalam memberikan informasi pengaduan pelanggan air.

Penelitian yang dilakukan oleh Khadhikal Amin dan Jarot Dian Susatyono (2018) dengan judul “Sistem Informasi Pencatatan Dan Perhitungan Biaya Penggunaan Air Bersih Berbasis Android”, untuk mempermudah dan mempercepat proses perhitungan dan pencatatan, maka di buatlah suatu dengan meng-*input*-kan angka pada meteran air yang merupakan angka besaran pemakaian penggunaan air kedalam form atau perintah memasukkan angka, input data angka tersebut kemudian dapat diproses secara otomatis perhitungan biayapenggunaan air berdasarkan angka dari meteran air, untuk penyimpanan datanya akan terkirim ke *server* dan tersimpan kedalam *database*.

Penelitian yang dilakukan oleh Fidy Arie Pratama dkk (2018) dengan judul “Sistem Informasi Monitoring Pajak Bumi Bangunan menggunakan Metode Rapid Application Development” pengembangan sistem informasi monitoring pajak bumi dan bangunan yang dirancang menggunakan metode pengembangan *RAD (Rapid Application Development)* dengan tahapan

Requirement Planning, Workshop Design dan Implementation, sehingga pengembangan sistem jadi lebih cepat dan efisien.

Dari beberapa kajian tersebut, dapat disimpulkan bahwa rata – rata sistem yang ada masih belum mengakomodir kebutuhan baik dari sisi petugas pengelola maupun pelanggan atau masyarakat umum. Pada penelitian ini maka akan dibuat suatu sistem yang dapat mengakomodir kebutuhan dari kedua sisi. Pada fitur pembayaran juga akan dilengkapi dengan sisa bayar yang dapat disimpan sebagai saldo untuk dikurangkan dengan tagihan bulan berikutnya dan dashboard untuk semua *role*. Selain itu sistem ini juga menyediakan kebutuhan mengenai komplain pelanggan dan memberikan informasi pengumuman dengan memanfaatkan metode pengembangan *RAD (Rapid Application Development)*.

2.2 Sistem Informasi

Menurut Sutabri (2012) Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajeral dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan – laporan yang di perlukan.

Menurut Anggraeni (2017) Sistem Informasi merupakan suatu kombinasi teratur dari orang – orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

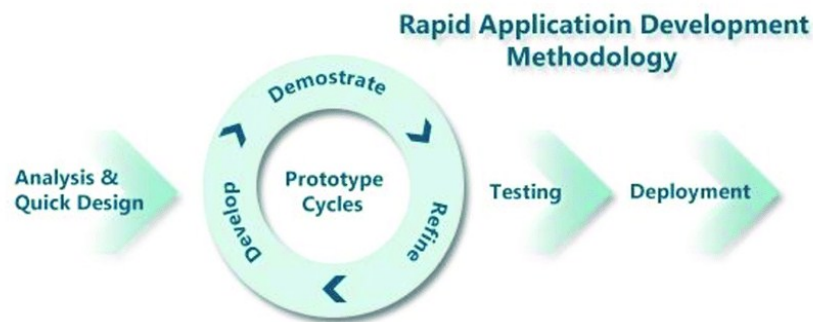
Berdasarkan teori tersebut, sistem informasi merupakan suatu kombinasi teratur dari orang – orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang dapat membantu suatu organisasi dalam mendukung fungsi operasi organisasi yang bersifat manajeral dengan mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi.

2.3 RAD (Rapid Application Development)

Menurut Whitten & Bentley (2007:98) *Rapid Application Development (RAD)* adalah sebuah strategi pengembangan sistem yang menekankan kecepatan

dalam pengembangan melalui keterlibatan pengguna dalam pembangunan secara cepat, iteratif, dan incremental dari suatu serangkaian prototype dari suatu sistem yang dapat berkembang menjadi suatu sistem akhir atau versi tertentu.

2.3.1 Tahapan RAD



Gambar 2.1 Metode RAD

(Sumber : <https://www.ramsoft.com.au/methodology.php>)

Berikut ini adalah tahapan – tahapan dari Metode RAD, antara lain:

1. Analysis & Quick Design

Tahapan ini merupakan tahap awal dalam pengembangan RAD, dimana pada tahap ini dilakukan identifikasi masalah, pengumpulan data, dan analisa data yang bertujuan untuk mengidentifikasi maksud akhir atau tujuan dari sistem dan kebutuhan informasi yang diinginkan.

2. Prototype Cycles

Di dalam tahap desain sistem, peneliti melibatkan narasumber dalam melakukan proses desain. Apabila masih terdapat ketidaksesuaian desain terhadap kebutuhan pengguna, maka akan dilakukan proses perbaikan desain. Desain yang dimaksudkan pada tahap ini adalah desain proses, desain antarmuka serta desain *database* dari sistem yang akan dibuat. Selain itu pada tahap ini desain sistem yang telah dibuat, dikembangkan ke dalam bentuk aplikasi. Pada tahapan ini juga peneliti juga melakukan kegiatan pengembangan dan integerasi dengan bagian-bagian lainnya sambil mempertimbangkan *feedback* dari narasumber. Jika proses berjalan lancar

maka dapat berlanjut ke tahapan berikutnya, sedangkan jika aplikasi yang dikembangkan belum memenuhi kebutuhan, maka akan kembali ke tahapan desain sistem.

3. Testing

Tahapan ini merupakan proses pengujian terhadap program untuk mendeteksi kesalahan yang ada pada sistem yang dikembangkan sebelum dilakukan *deployment*.

4. Deployment

Pada tahap ini peneliti melakukan *deploy* sistem informasi yang telah dikembangkan sebelumnya dan sudah melewati proses *testing*. Deployment sendiri adalah proses menyebarkan (*publish*) aplikasi yang telah dikerjakan oleh para pengembang agar dapat diakses oleh pengguna.

Metode ini juga memiliki kelebihan dan kekurangan, sebagai berikut:

a. Kelebihan RAD

1. Kebutuhan aplikasi bisa berubah sewaktu-waktu.
2. Aplikasi dikembangkan berdasarkan kebutuhan dan keinginan user.
3. Memperkecil kemungkinan error dan hal buruk lainnya.
4. Waktu pengembangan aplikasi bisa lebih cepat dan efektif.
5. Mempermudah proses integrasi.

b. Kekurangan RAD

1. Membutuhkan tim dengan *skill* teknis yang tinggi.
2. Memerlukan kolaborasi tim yang kuat.
3. Hanya cocok untuk proyek yang waktunya singkat.
4. Hanya cocok digunakan untuk mengembangkan aplikasi secara *modular* (fokus ke suatu fitur untuk dijadikan *module* terpisah).
5. Sulit diterapkan untuk mengembangkan aplikasi besar.

Berdasarkan pada penjelasan diatas, metode ini merupakan sebuah strategi pengembangan sistem yang menekankan kecepatan dan memiliki kelebihan dalam

pendefinisian kebutuhan aplikasi yang bisa berubah secara fleksibel dalam pengembangan sehingga sesuai digunakan untuk durasi penelitian yang singkat.

2.4 UML

Menurut Rosa Dan Shalahuddin (2015:133) “UML (Unified Modelling Language) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek.

Sedangkan menurut Ariani R. Sukamto dalam Taufik (2017) “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung”.

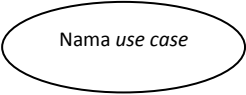


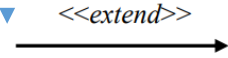
Pendapat lainnya menurut Fowler, M. dalam (B. O. Lubis, 2016) UML (Unified Modeling Language) adalah “Keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientansi objek (OO). Definisi ini merupakan definisi yang sederhana”.


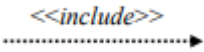

Unified Modelling Language (UML) merupakan gambaran umum aplikasi yang akan dibuat. Didalamnya terdapat berbagai macam diagram yang memiliki fungsi sendiri. UML digunakan karena pemodelan ini sudah merepresentasikan sesuai dengan program sebenarnya. Beberapa jenis diagram UML yang akan digunakan yaitu :

2.4.1 Use Case Diagram

Menurut Sukamto dan Shalahuddin (2016:155), Use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1 Simbol *Use Case* Diagram

No	Simbol	Deskripsi
1	<p data-bbox="507 315 868 360"><i>Use Case</i></p> 	<p data-bbox="890 315 1326 618">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
2	<p data-bbox="507 647 868 692">Aktor / <i>actor</i></p> 	<p data-bbox="890 647 1326 1173">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
3	<p data-bbox="507 1234 868 1279">Asosiasi / <i>association</i></p> 	<p data-bbox="890 1200 1326 1402">Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor.</p>
4	<p data-bbox="507 1581 868 1626">Ekstensi / <i>extend</i></p> 	<p data-bbox="890 1424 1326 1895">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; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; ditambahkan, missal arah panah mengarah pada <i>use case</i> yang</p>

		ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.
5	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan / <i>include</i> <i>/uses</i>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> .


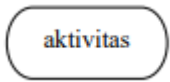



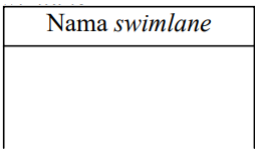
(Sumber : Sukamto dan Shalahuddin (2016:156-158))

2.4.2 Activity Diagram

Menurut Sukamto dan Shalahuddin (2016:161), “Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak”. Berikut ini simbol – simbol yang digunakan pada *activity* diagram:

Tabel 2.2 Simbol *Activity* Diagram

No	Simbol	Deskripsi
----	--------	-----------

1	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Sukamto dan Shalahuddin (2016:156-158))

2.4.3 Sequence Diagram


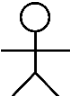

Menurut Sukamto dan Shalahuddin (2016:165) “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.




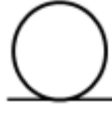

Menurut Adi Nugroho (2010), Sequence Diagram memperlihatkan interaksi sebagai diagram dua matra (dimensi). Matra vertikal adalah sumbu waktu, waktu bertambah dari atas ke bawah. Matra horizontal memperlihatkan

peran pengklasifikasian yang merepresentasikan objek - objek mandiri yang terlibat dalam kolaborasi. Masing – masing pengklasifikasian direpresentasikan sebagai kolom-kolom vertikal dalam sequence diagram yang sering disebut sebagai garis waktu (life line). Selama objek ada, peran digambarkan menggunakan garis tegas. Selama aktivitas prosedur pada objek aktif, garis waktu digambarkan sebagai garis ganda. Pesan-pesan digambarkan sebagai suatu tanda panah dari garis waktu suatu objek ke garis waktu objek lainnya. Panah-panah menggambarkan aliran pesan antar peran pengklasifikasian digambarkan dalam urutan waktu kejadiannya dari atas ke bawah.

Berikut selengkapnya notasi-notasi yang digunakan dalam sequence diagram :

Tabel 2.3 Simbol *Sequence* Diagram

No	Simbol	Deskripsi
1	Object (Partisipan) 	Objek atau biasa disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek di dalamnya yang diawali dengan titik koma.
2	<i>Actor</i> 	<i>Actor</i> juga dapat berkomunikasi dengan objek, maka actor juga dapat diurutkan sebagai kolom.
3	<i>Life line</i> 	<i>Life line</i> mengindikasikan keberadaan sebuah object dalam baris waktu. Notasi untuk <i>life line</i> adalah garis putus-putus vertikal yang ditarik dari sebuah object.
4	<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah

		kotak persegi empat yang digambarkan pada sebuah <i>life line</i> . <i>Activation</i> mengindikasikan sebuah object yang akan melakukan sebuah aktivasi.
5	<i>Self Message</i> 	<i>Self message</i> mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.
6	<i>Control</i> 	<i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan.
7	<i>Entity</i> 	<i>Entity</i> digunakan untuk menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data.
8	<i>Message</i> 	<i>Message</i> , digambarkan dengan anak panah <i>horizontal</i> antar <i>activation</i> . <i>Message</i> mengindikasikan komunikasi antara objek objek.

2.4.4 Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut pola dan metode atau operasi (Sukamto & Shalahuddin, 2013) :

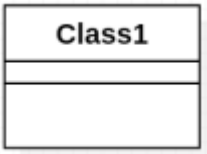

- a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas

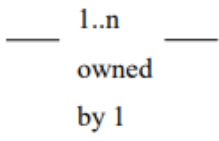

Dalam notasi UML, himpunan kelas kelas beserta hubungan / relasi / asosiasi antar kelas biasanya digambarkan menggunakan sebuah diagram UML yang dinamakan diagram kelas (class diagram). Jika kita perhatikan lebih jauh, sesungguhnya diagram kelas memiliki dua kegunaan / fungsi yang sangat penting, yaitu : (Nugroho, 2010)

- a. Mempresentasikan keadaan statis kelas-kelas yang terlibat dalam sistem. Kelas-kelas ini bisa saja merupakan kelas-kelas dalam bahasa pemrograman dan kelas-kelas persisten yang hadir dalam bentuk tabel-tabel yang ada di sistem basis data relasional.
- b. Hubungan antar kelas dalam sistem atau perangkat lunak yang sedang kita kembangkan dapat terlihat dengan mudah.

Berikut notasi simbol pada *class* diagram :

Tabel 2.4 Simbol *Class* Diagram

Simbol	Nama	Deskripsi
	<i>Class</i>	<i>Class</i> adalah balok-balok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi menjadi 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan atribut <i>class</i> . Bagian bawah mendefinisikan method dari sebuah <i>class</i> .
	<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus menjadi bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempatnya bergantung tersebut.
	<i>Assosiation</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang

		<p>menghubungkan antar 2 <i>class</i>. Garis ini dapat melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i>.</p>
	<p><i>Dependency</i></p>	<p>Kadang kala <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.</p>

2.5 Entity Relationship Diagram

Menurut Yanto (2016:32) “ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas”.

Berikut ini merupakan komponen – komponen dari ERD:

a. Entitas (*Entity*)

Kumpulan obyek atau sesuatu yang dapat dibedakan atau dapat diidentifikasi secara unik.

b. Relasi (*Relationship*)

Hubungan yang terjadi antara satu entitas atau lebih. Kumpulan *relationship* yang sejenis disebut *relationship set*.

c. Atribut

Karakteristik dalam *entity* atau *relationship* yang mengerjakan penjelasan detail tentang *entity* atau *relationship* atau dengan kata lain adalah kumpulan elemen data yang membentuk suatu entitas.

Derajat *relationship* ERD :

a. *Unary* (Derajat Satu)

Unary adalah satu buah *relationship* menghubungkan satu buah *entity*.

- b. *Binary* (Derajat Dua)
Binary adalah satu buah relationship yang menghubungkan dua buah *entity*.
- c. *Ternary* (Derajat Tiga)
Ternary adalah satu buah relationship menghubungkan tiga buah *entity*.

Rasio Kardinalitas :

- a. *One to One* (1:1)
One to One adalah perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding satu.
- b. *One to Many* (1:N)
One to Many adalah perbandingan antara *entity* pertama dengan *entity* kedua berbanding satu berbanding banyak.
- c. *Many to Many* (M:N)
Many to Many yaitu perbandingan antara *entity* pertama dengan *entity* kedua berbanding banyak berbanding banyak.

2.6 Web

Menurut Sibero (2013:11) *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan internet.

Cara Kerja Web adalah sebagai berikut :

1. Informasi *web* disimpan dalam dokumen dalam bentuk halaman-halaman *web* atau *web page*.
2. Halaman *web* tersebut disimpan dalam komputer server *web*.
3. Sementara dipihak pemakai ada komputer yang bertindak sebagai komputer klien dimana ditempatkan program untuk membaca halaman *web* yang ada di server *web* (*browser*).
4. *Browser* membaca halaman *web* yang ada di server *web*.

Unsur – Unsur Web :

a. *Domain name system*

Menurut Syafrizal (2005), DNS (Domain name system) dapat dianalogikan seperti pemakaian buku telepon, dimana orang kita kenali berdasarkan nama. Akan tetapi untuk menghubungi kita harus menelpon nomor telepon di ponsel atau di telpon rumah, yang dinamakan nomor tersebut dapat kita tulis dengan nama orang yang kita hubungi, dan akan terlihat di ponsel anda nama dan nomor orang tersebut.

Nama adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah website, atau dengan kata lain domain name adalah alamat yang digunakan untuk menemukan sebuah website pada dunia internet. Contohnya adalah <http://www.detik.com>. Nama domain diperjualbelikan secara bebas di internet dengan status sewa tahunan. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan website tersebut. Contoh nama domain ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Contoh nama domain ber-ekstensi lokasi Negara Indonesia adalah co.id (untuk nama domain website perusahaan), ac.id (nama domain website pendidikan), go.id (nama domain website instansi pemerintah), or.id (nama domain website organisasi).

b. *Web hosting*

Menurut Aliyun (2014:72) Hosting memiliki arti layanan berbasis internet sebagai tempat penyimpanan data atau tempat menjalankan aplikasi ditempat terpusat yang disebut dengan server dan dapat diakses melalui jaringan internet.

c. Bahasa program

Menurut Munir (2011:13) mengemukakan bahwa “Bahasa pemrograman adalah bahasa komputer yang digunakan dalam menulis program.” Untuk itu, bahasa pemrograman dibagi menjadi 4 (empat) tingkatan yaitu:

1. Bahasa Mesin (*Machine Language*)

Bahasa pemrograman yang hanya dapat dimengerti oleh mesin komputer yang didalamnya terdapat *Central Processing Unit* (CPU) yang hanya mengenal dua keadaan yang berlawanan, yaitu:

- a. Bila terjadi kontak atau ada arus bernilai 1.
- b. Bila tidak terjadi kontak atau arus bernilai 0.

2. Bahasa Tingkat Rendah (*Low Level Language*)

Karena banyak keterbatasan yang dimiliki bahasa mesin maka dibuatlah simbol yang mudah diingat yang disebut dengan *mnemonic* (pembantu untuk mengingat). Contoh: Bahasa Assembler, yang dapat menerjemahkan *mnemonic*.

3. Bahasa Tingkat Menengah (*Middle Level Language*)

Bahasa pemrograman yang menggunakan aturan-aturan gramatikal dalam penulisan pernyataannya, mudah untuk dipahami, dan memiliki instruksi-instruksi tertentu yang dapat langsung diakses oleh komputer. Contoh: Bahasa C.

4. Bahasa Tingkat Tinggi (*High Level Language*)

Bahasa pemrograman yang dalam penulisan pernyataannya mudah dipahami secara langsung.

- a. Bahasa Berorientasi pada Prosedur (*Procedure Oriented Language*)
Contoh: Algoritma, Fortran, Pascal, Basic, Cobol.
- b. Bahasa Berorientasi pada Masalah (*Problem Oriented Language*)
Contoh: Report Program Generator (RPG).

d. Desain *website*

Setelah melakukan penyewaan *domain name* dan *web hosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*. Desain sangat berpengaruh kepada penilaian pengunjung akan bagus tidaknya sebuah *website*.

Fungsi - fungsi Web:

- a. Fungsi komunikasi

Situs web yang mempunyai fungsi komunikasi pada umumnya adalah situs web dinamis. Karena dibuat menggunakan pemrograman web (*server side*) maka dilengkapi fasilitas yang memberikan fungsi-fungsi komunikasi, seperti web mail, form contact, chatting form, dan yang lainnya.

b. Fungsi informasi

Situs web yang memiliki fungsi informasi pada umumnya lebih menekankan pada kualitas bagian kontennya, karena tujuan situs tersebut adalah menyampaikan isinya.

c. Fungsi hiburan

Situs web juga dapat memiliki fungsi hiburan. Bila situs web kita berfungsi sebagai sarana hiburan maka penggunaan animasi gambar dan elemen bergerak dapat meningkatkan mutu presentasi desainnya, meski tetap harus mempertimbangkan kecepatan unduhannya. Beberapa fasilitas yang memberikan fungsi hiburan adalah *game online*, *film online*, *musik online*, dan sebagainya.

d. Fungsi transaksi

Situs web dapat dijadikan sarana transaksi bisnis, baik barang, jasa, atau lainnya. Situs web ini menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik. Pembayaran bisa menggunakan kartu kredit, transfer, atau dengan membayar secara langsung.

2.7 Database

Menurut Raharjo (2011:3), *Database* atau basis data adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.

Menurut Anhar (2010:45), *Database* atau basis data adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur file yang menyusun sebuah database adalah *Data Record* dan *Field*.

Menurut Andri Kristanto (2008) Database atau basis data adalah kumpulan data yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang bereleasi yang dapat tersimpan dalam magnetic disk, optical disk, magnetic drum, atau media penyimpanan sekunder lainnya. DBMS (Database Management System) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar.

Dari pengertian diatas penulis menyimpulkan *Database* adalah sekumpulan file yang saling berhubungan yang menyimpan data dan tersimpan dalam sebuah media penyimpanan.

2.8 MySQL

Menurut Raharjo (2011:21), “MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user”.

Menurut Arief (2011:152) MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya

Dari pengertian diatas penulis menyimpulkan MySql adalah adalah suatu software atau program yang digunakan untuk membuat sebuah database yang bersifat open source.

MySQL digunakan untuk membuat dan mengelola suatu database secara terstruktur dan otomatis menggunakan suatu Bahasa khusus. Namun lebih jelasnya lagi MySQL ini memberikan kemudahan bagi para pengguna yang ingin mengelola suatu data yang berisi informasi secara String (text based) dan dapat diakses secara pribadi maupun untuk umum dalam suatu web. Hampir semua host atau penyedia server web memberikan fasilitas MySQL untuk para developer web yang menginginkan pengelolaan database di websitenya.

2.9 PHP

Menurut Supono & Putratama (2018: 1) mengemukakan bahwa “PHP (PHP: hypertext preprocessor) adalah suatu bahasa pemrograman yang digunakan

untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang ditambahkan ke HTML”.

PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk Bahasa yang cross-platform, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”.

Sehingga dapat disimpulkan dari pengertian tersebut bahwa bahasa pemrograman PHP merupakan bahasa yang bisa digunakan untuk konten dinamis serta dapat terhubung dengan *database*.

2.10 MVC

Menurut Daqiqil (2011:5) “MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses”.

Dapat disimpulkan dari pengertian tersebut bahwa MVC merupakan teknik yang dapat mempermudah pengembang dalam membuat kode program agar lebih terstruktur, sehingga dokumentasinya juga lebih mudah.

2.11 Framework

Menurut Hakim (2010:3) menjelaskan bahwa, Framework adalah koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal.

Sehingga dapat disimpulkan bahwa dengan menggunakan framework, proses pengembangan aplikasi akan lebih cepat karena pengembang tidak perlu membuat kode dari awal.

2.12 Laravel

Menurut Aminudin (2015:1) Laravel adalah sebuah *Framework* PHP dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti *framework-framework* yang lain, Laravel dibangun dengan konsep MVC (*Model-Controller-View*), kemudian Laravel dilengkapi juga *command line tool* yang bernama “Artisan” yang bisa digunakan untuk *packaging bundle* dan *instalasi bundle* melalui *command prompt*.

Menurut Abdulloh (2017:3) mengatakan bahwa terdapat beberapa keunggulan yang dimiliki Laravel yaitu sebagai berikut.

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh framework lain.
2. Laravel merupakan framework PHP yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti sehingga programmer pemula sekalipun akan mudah memahami kegunaan dari suatu sintaks meskipun programmer tersebut belum mempelajarinya.
3. Laravel memiliki dokumentasi yang cukup lengkap, bahkan setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
4. Laravel digunakan oleh banyak programmer sehingga banyak library yang mendukung Laravel yang diciptakan para programmer pecinta Laravel.
5. Laravel didukung oleh Composer sehingga library-library diperoleh dengan mudah dari internet menggunakan Composer.
6. Laravel memiliki template engine tersendiri yang diberi nama blade yang memudahkan dalam menampilkan data pada template HTML.

Adapun fitur-fitur yang dimiliki Laravel antara lain sebagai berikut. (Aminudin, 2015:5).

- a. *Bundles* yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi.
- b. *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola *active record* menyediakan metode internal untuk mengatasi kendala hubungan antara objek database.
- c. *Application Logic* merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi *route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework* Sinatra.
- d. *Reverse Routing* mendefinisikan hubungan antara *link* dan *route*. Sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *link* yang dibuat dengan menggunakan nama-nama dari *route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
- e. *Restful Controller* memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f. *Class Auto Loading* menyediakan otomatis *loading* untuk kelas-kelas PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g. *View Composers* adalah kode unit *logical* yang dapat dijalankan ketika sebuah *view* di *load*.
- h. *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai *Singletons*.
- i. *Migrations* menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbaiki aplikasi.
- j. *Unit Testing* mempunyai peran penting dalam *framework* Laravel, dimana *unit testing* ini mempunyai banyak tes untuk mendeteksi dan mencegah

regresi. *Unit testing* dapat dijalankan melalui fitur “*artisan command-line*”.

- k. *Automatic Pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.13 Bootstrap

Menurut Rivaldi (2015:44) “Bootstrap merupakan Framework ataupun Tools untuk membuat aplikasi web ataupun situs *web responsive* secara cepat, mudah dan gratis”.

Sedangkan menurut Andriansyah (2017:26), “Bootstrap merupakan framework HTML, CSS, dan JavaScript paling populer yang digunakan untuk membangun *website responsive* dan berbasis *mobile*”.

Bootstrap dibuat oleh Mark Otto dan Jacob Thornton pertama kali dirilis pada tanggal 19 Agustus 2011 dan berlisensi open source yang artinya bebas digunakan tanpa harus melakukan pembayaran. Alamat *website* resmi dari *framework Bootstrap* adalah <http://getbootstrap.com>. Untuk mengunduh *framework Bootstrap* dapat dilakukan melalui *website* tersebut atau dapat melalui GitHub dengan alamat <https://github.com/twbs/bootstrap/>.

Sehingga dapat disimpulkan bahwa ketika membangun web menggunakan *Bootstrap* maka tampilan akan lebih interaktif karena sudah *responsive* dan berbasis *mobile*.

2.14 Blackbox Testing

Menurut Rosa dan Salahuddin (2015:275) “Blackbox testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pada hasil penelitian ini akan dilakukan pengujian menggunakan blackbox testing yang menekankan pada fungsionalitasnya sesuai dengan tujuan penelitian ini.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

3.1.1 Identifikasi Masalah

Permasalahan yang timbul pada sistem yang ada saat ini adalah kurangnya efektif dan efisien dalam hal pelayanan serta sistem kurang menunjang untuk kecepatan kinerja lembaga. Beberapa masalah yang timbul sebagai berikut:

1. Keterlambatan pembuatan tagihan dan laporan akibat proses pengelolaan data melalui banyak tahapan. Sistem yang digunakan untuk pembuatan tagihan hingga menjadi sebuah laporan data pada saat ini yaitu petugas lapangan melakukan pendataan meteran pada buku. Admin menunggu petugas lapangan menyerahkan semua data, kemudian melakukan input data untuk pembuatan daftar tagihan. Setelah itu admin akan menyerahkan daftar tagihan kepada petugas pembayaran, kemudian petugas pembayaran melakukan pendataan pembayaran pelanggan pada buku. Dengan jumlah data yang banyak dan banyak tahapan inilah yang menyebabkan keterlambatan pembuatan tagihan dan laporan.
2. Ketika ada pengaduan pelanggan mengenai tagihan admin tidak mengetahui harus mengecek pada data yang mana, sehingga untuk perubahan data hanya diperkirakan seperti tagihan sebelumnya saja.
3. Tidak tersedianya informasi jadwal perbaikan kerusakan dan pengumuman yang berhubungan dengan pengelolaan air bersih.

3.1.2 Pemecahan Masalah

Berdasarkan permasalahan diatas, maka dilakukan analisis untuk menemukan solusi dari permasalahan dengan menggunakan metode sebab akibat.

Tabel 3.1 Analisa Sebab Akibat

Permasalahan	Solusi	Manfaat
Keterlambatan pembuatan tagihan dan laporan akibat proses pengelolaan data melalui banyak tahapan.	Fitur pencatatan untuk petugas lapangan, perekapan pembayaran untuk petugas pembayaran yang tersinkronisasi dalam sebuah <i>database</i> .	Memudahkan kinerja admin dalam melakukan pembuatan tagihan dan laporan, karena tidak perlu menunggu petugas menyerahkan data dan <i>input</i> ulang.
Pengaduan pelanggan mengenai tagihan admin tidak mengetahui harus mengecek pada data yang mana, sehingga untuk perubahan data hanya diperkirakan seperti tagihan sebelumnya saja.	Pada fitur pencatatan meteran dilengkapi dengan unggah foto meteran.	Pelanggan memiliki acuan dalam melakukan pengaduan dan admin juga dapat mencocokkan dengan hasil unggahan foto dari petugas.
Tidak tersedianya informasi jadwal perbaikan kerusakan.	Pada fitur <i>feedback</i> pengaduan admin dilengkapi dengan jadwal dan tertera nomor telepon bisa dihubungi oleh pelanggan.	Pelanggan dapat mengonfirmasi ketersediaan pada jadwal tersebut kepada admin.
Tidak ada	Fitur untuk admin	Masyarakat dapat

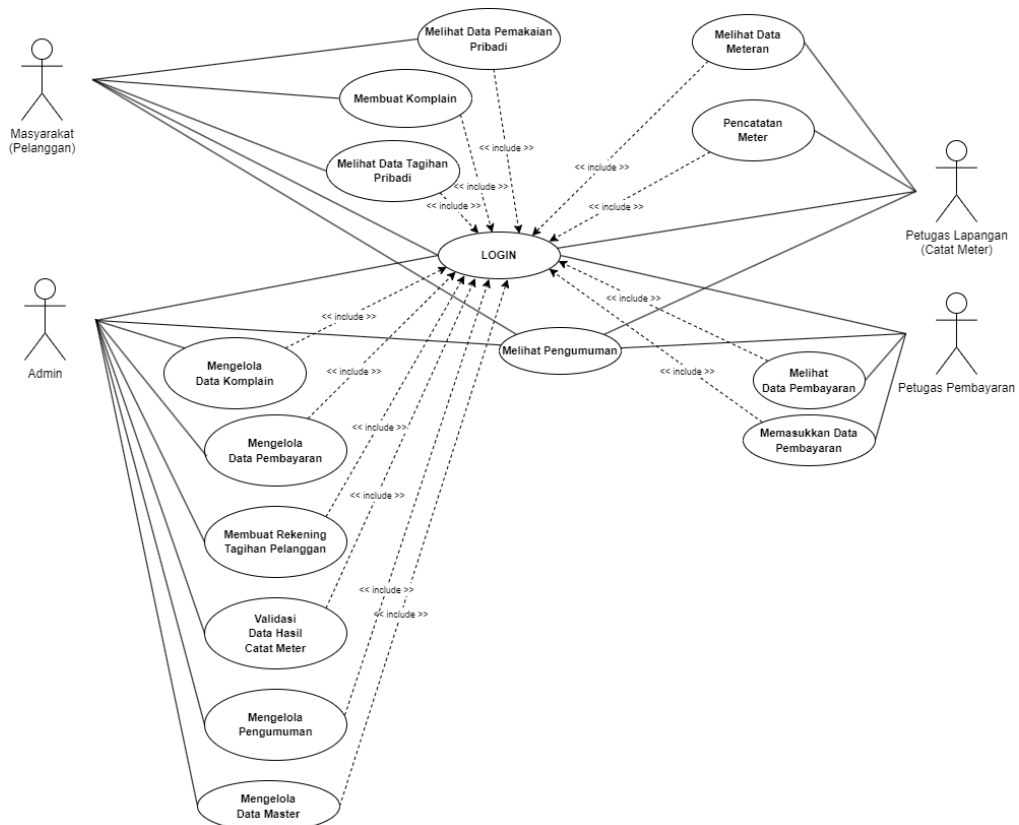
pengumuman yang berhubungan dengan pengelolaan air bersih.	membuat pengumuman.	mengetahui informasi terbaru terkait pengelolaan air bersih.
--	---------------------	--

3.2 Perancangan

3.2.1 Perancangan Sistem

Berdasarkan analisa permasalahan tersebut, maka akan dibuat gambaran umum aplikasi menggunakan *Use Case Diagram* untuk mendefinisikan hak akses pengguna atau mendeskripsikan sebuah interaksi antara satu atau lebih aktor (pengguna) serta *Activity Diagram* untuk menggambarkan proses bisnis aplikasi.

3.2.1.1 Use Case Diagram



Gambar 3.1 Rancangan *Use Case Diagram*

a. Keterangan gambar untuk Masyarakat (Pelanggan) :

1. Melihat Pengumuman
Pelanggan dapat melihat pengumuman.
 2. Melihat Data Pemakaian Pribadi
Pelanggan dapat melihat data pemakaian pribadi termasuk data pembayaran dengan melakukan *login* dahulu.
 3. Membuat Pengaduan
Pelanggan dapat membuat pengaduan dan melihat *feedback* dari admin dengan melakukan *login* dahulu.
- b. Keterangan gambar untuk Petugas Lapangan (Catat Meter) :
1. Melihat Pengumuman
Petugas Lapangan dapat melihat pengumuman.
 2. Melihat Data Meteran
Petugas Lapangan dapat melihat data meteran milik pelanggan yang sesuai areanya dengan melakukan *login* dahulu.
 3. Pencatatan Meteran
Petugas Lapangan dapat melakukan pencatatan meteran milik pelanggan yang sesuai areanya dengan melakukan *login* dahulu.
- c. Keterangan gambar untuk Petugas Pembayaran :
1. Melihat Pengumuman
Petugas Pembayaran dapat melihat pengumuman.
 2. Melihat Data Pembayaran
Petugas Pembayaran dapat melihat data pembayaran milik pelanggan yang sesuai areanya dengan melakukan *login* dahulu.
 3. Merekap Pembayaran
Petugas Pembayaran dapat melakukan perekapan pembayaran pelanggan yang sesuai areanya dengan melakukan *login* dahulu.
- d. Keterangan gambar untuk Admin
1. Melihat Pengumuman
Admin dapat melihat pengumuman.
 2. Mengelola Data Pembayaran

Admin dapat mengelola data pembayaran apabila ada kesalahan input dari petugas, maka admin dapat melakukan revisi dengan *login* dahulu.

3. Membuat Rekening Tagihan Pelanggan

Admin dapat membuat rekening tagihan pelanggan untuk dikirimkan ke petugas pembayaran dengan *login* dahulu.

4. Mengelola Pengumuman

Admin dapat mengelola pengumuman, mulai dari membuat, mengubah dan menghapus dengan *login* dahulu.

5. Mengelola Data Laporan

Admin dapat mengelola data laporan yang bisa di *export* dalam bentuk Excel dengan *login* dahulu, data laporan meliputi Laporan Catat Meteran dan Laporan Pembayaran.

6. Mengelola Data Hasil Catat Meteran

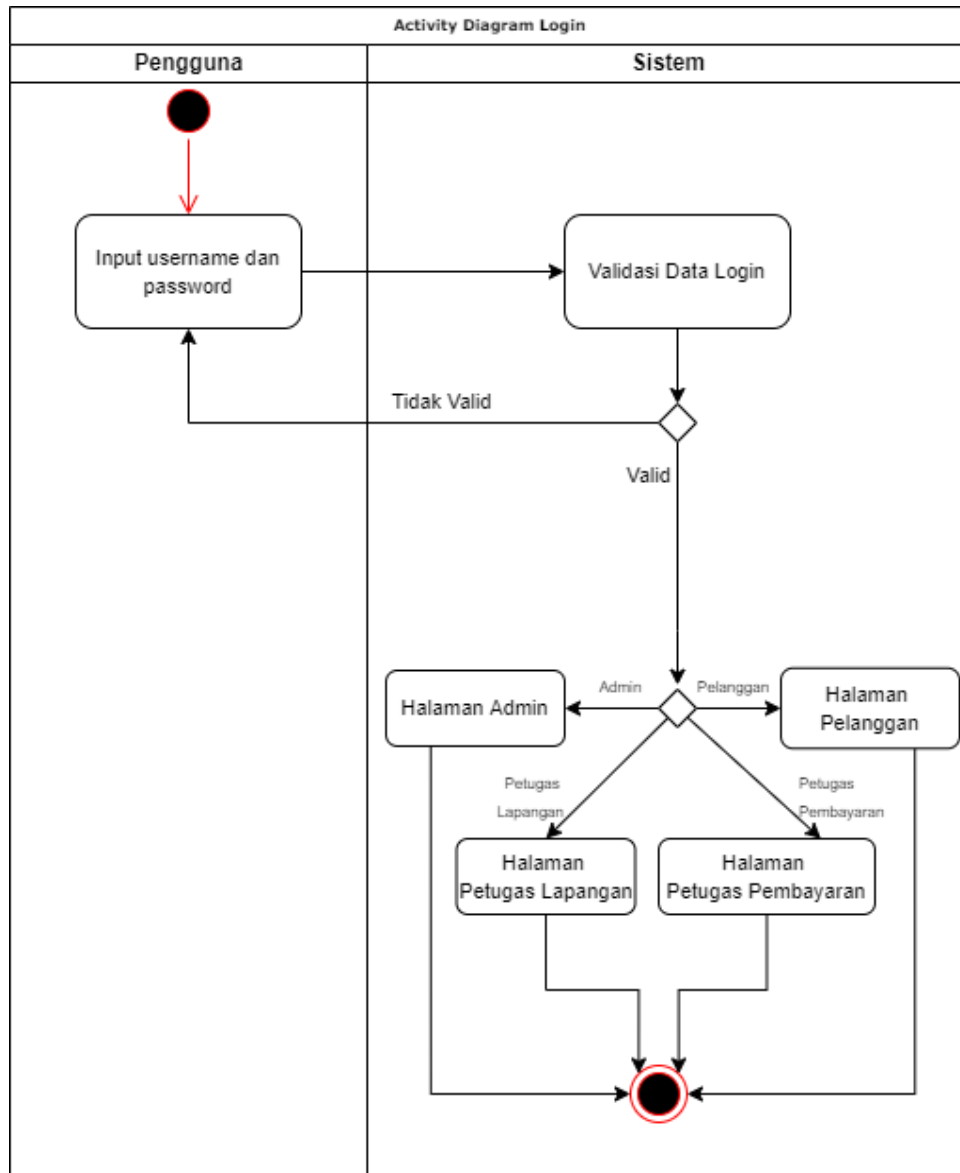
Admin dapat mengelola data laporan yang bisa di *export* dalam bentuk Excel dengan *login* dahulu, data laporan meliputi Laporan Catat Meteran dan Laporan Pembayaran.

7. Mengelola Data Master

Admin dapat mengelola data master, mulai dari membuat, mengubah dan menghapus dengan *login* dahulu, data master meliputi Data Pelanggan, Data Petugas, Data Meteran, Data Wilayah, Data RT, Data RW, Data RW, Data Dusun.

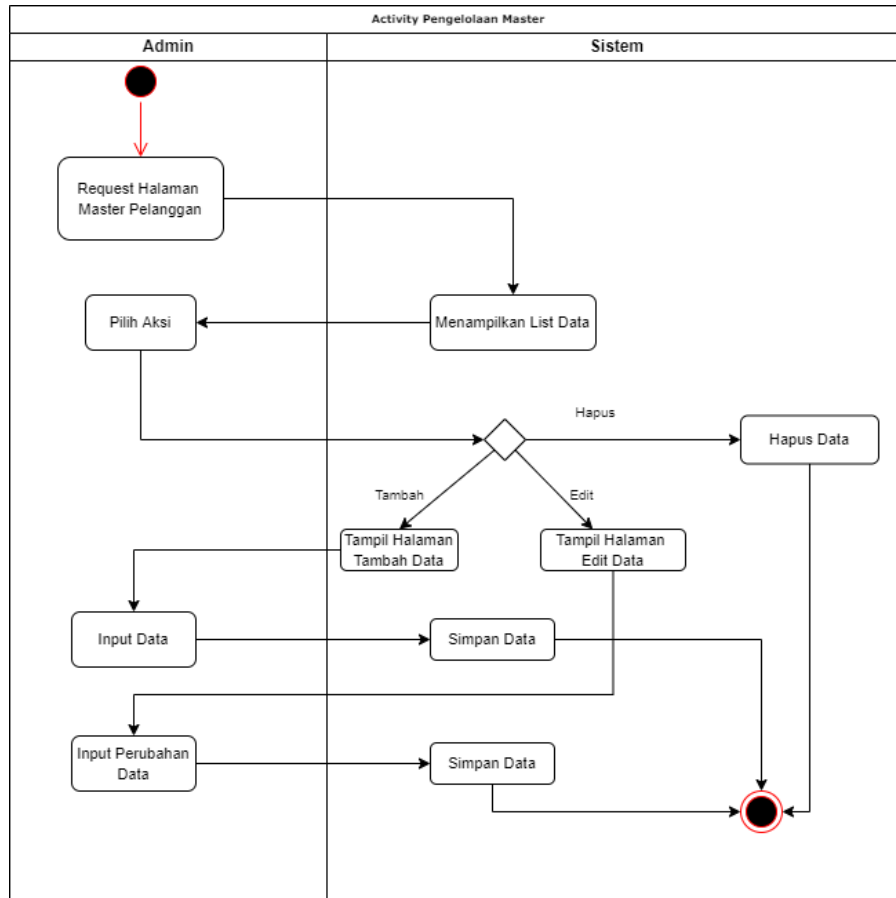
3.2.1.2 Activity Diagram

a. Login



Gambar 3.2 Rancangan Activity Diagram Login

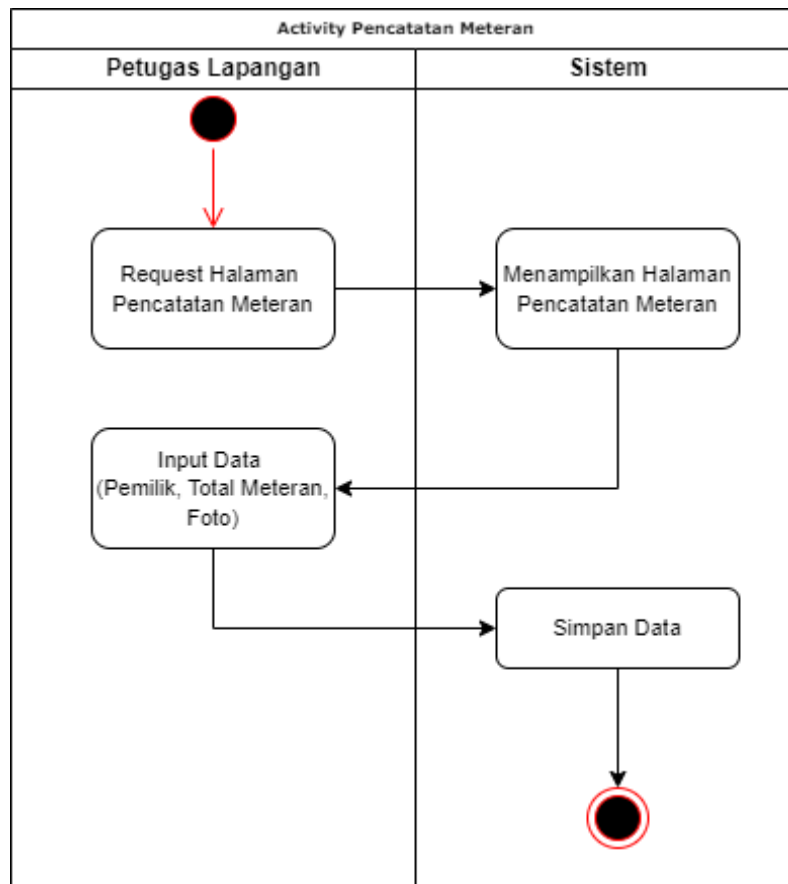
b. Pengelolaan Master



Gambar 3.3 Rancangan *Activity* Diagram Pengelolaan Master

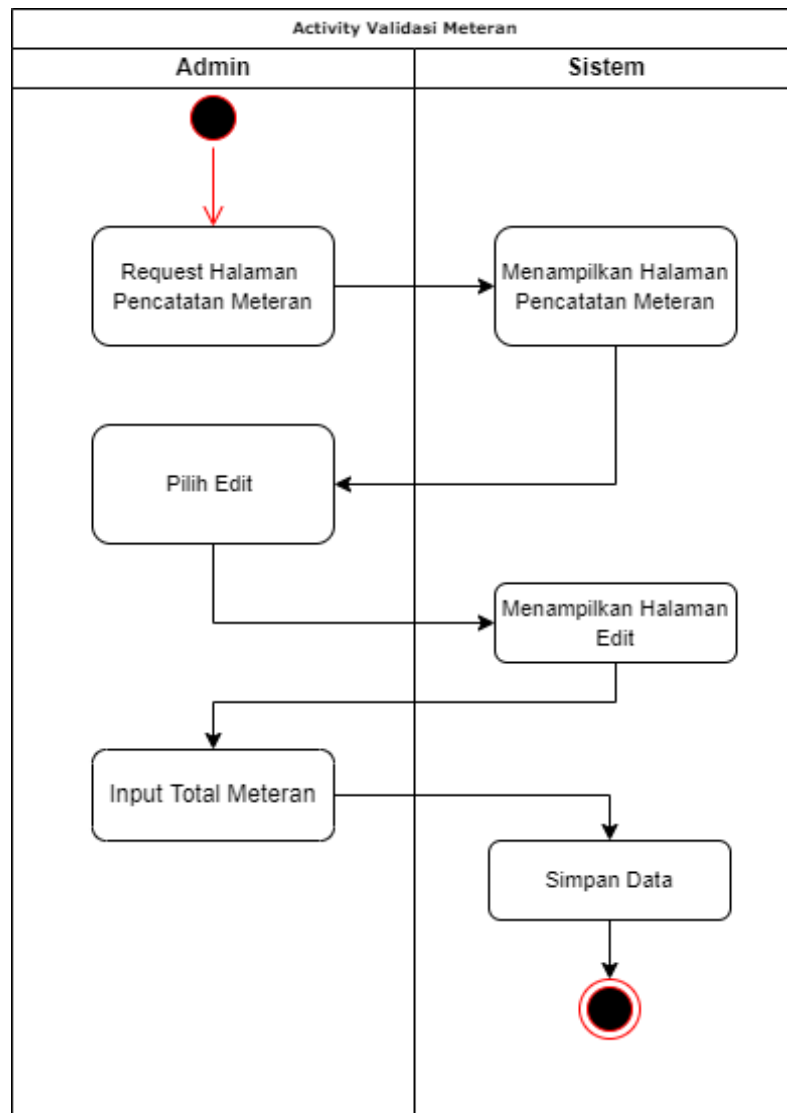
Data Master meliputi Data Pelanggan, Data Petugas, Data Meteran, Data Wilayah, Data RT, Data RW, Data RW, Data Dusun.

c. Pencatatan Meteran



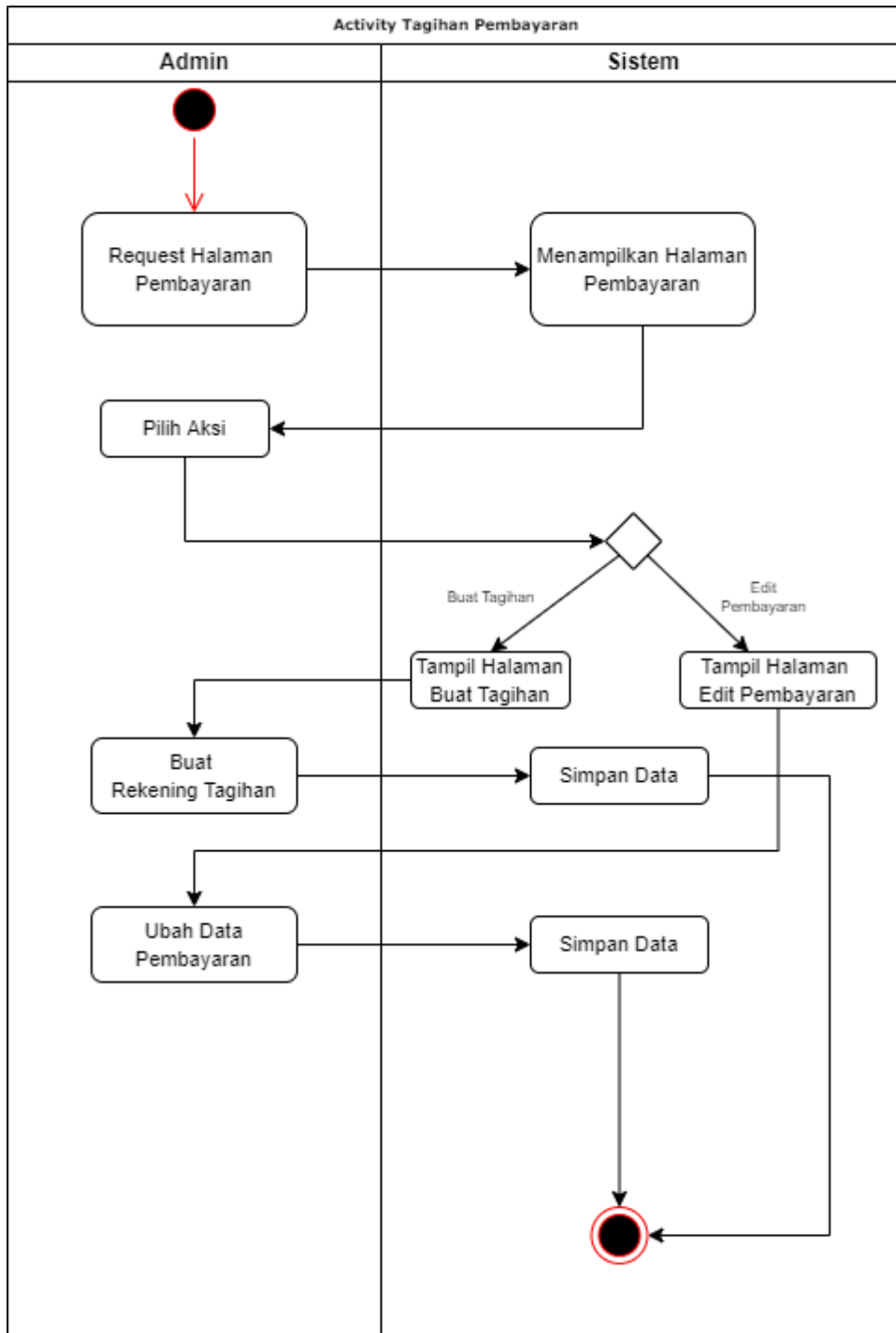
Gambar 3.4 Rancangan *Activity* Diagram Pencatatan Meteran

d. Validasi Data Hasil Catat Meteran



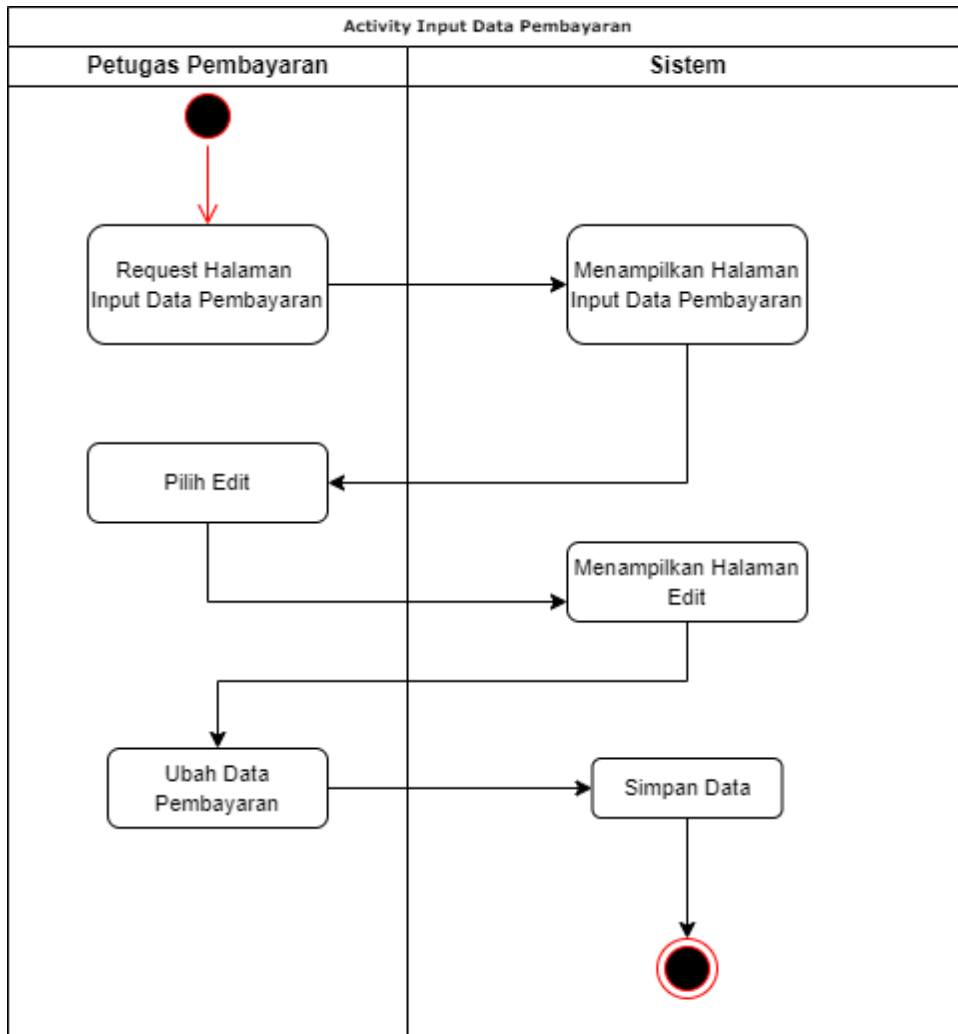
Gambar 3.5 Rancangan *Activity* Diagram Validasi Data Hasil Catat Meteran

e. Tagihan Pembayaran



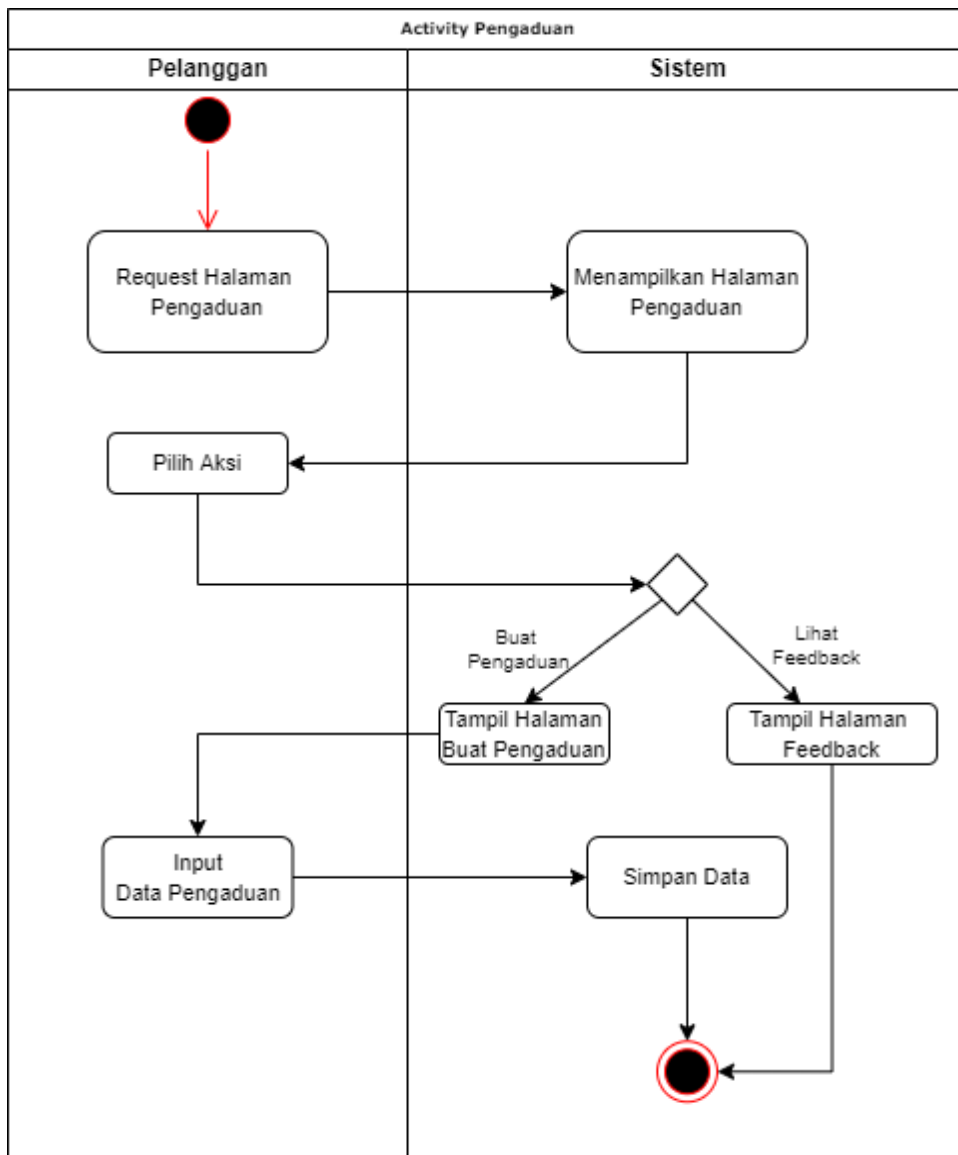
Gambar 3.6 Rancangan *Activity Diagram* Tagihan Pembayaran

f. Input Data Pembayaran



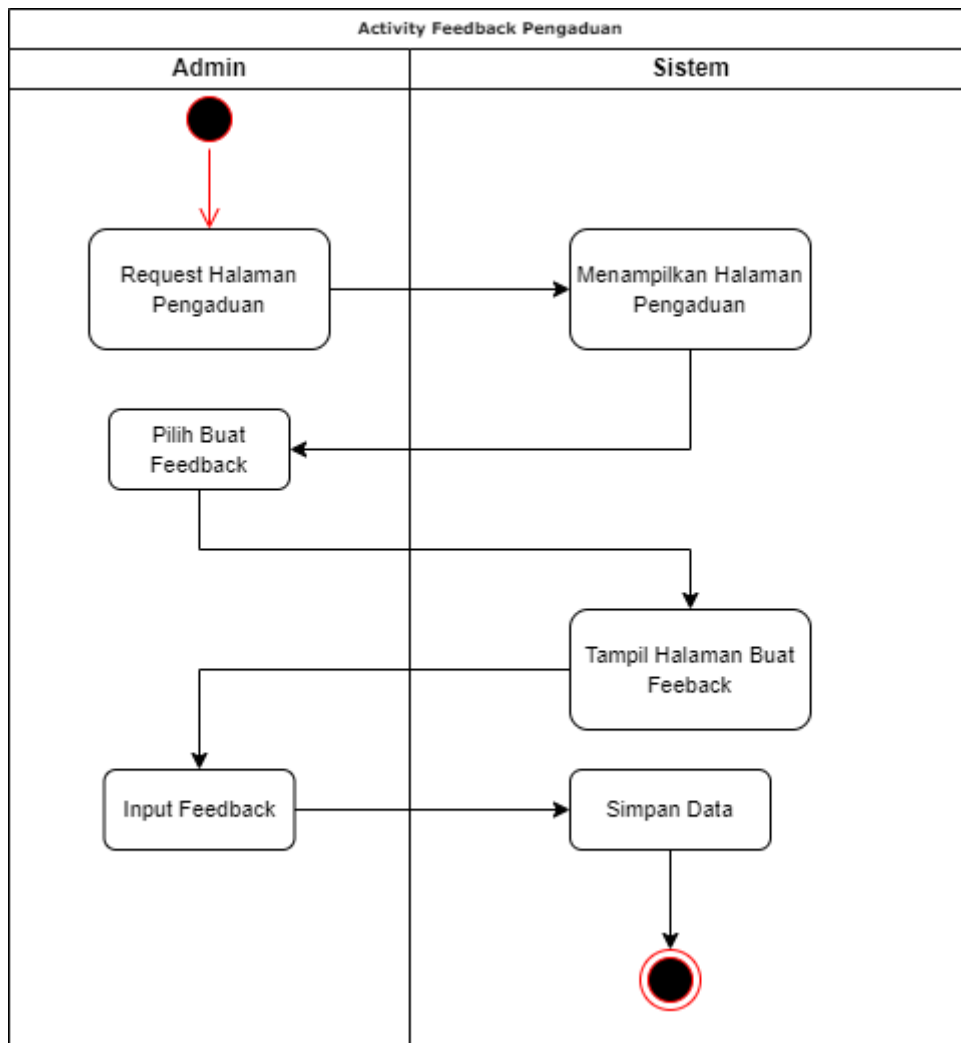
Gambar 3.7 Rancangan *Activity* Diagram Input Data Pembayaran

g. Pengaduan (Komplain)



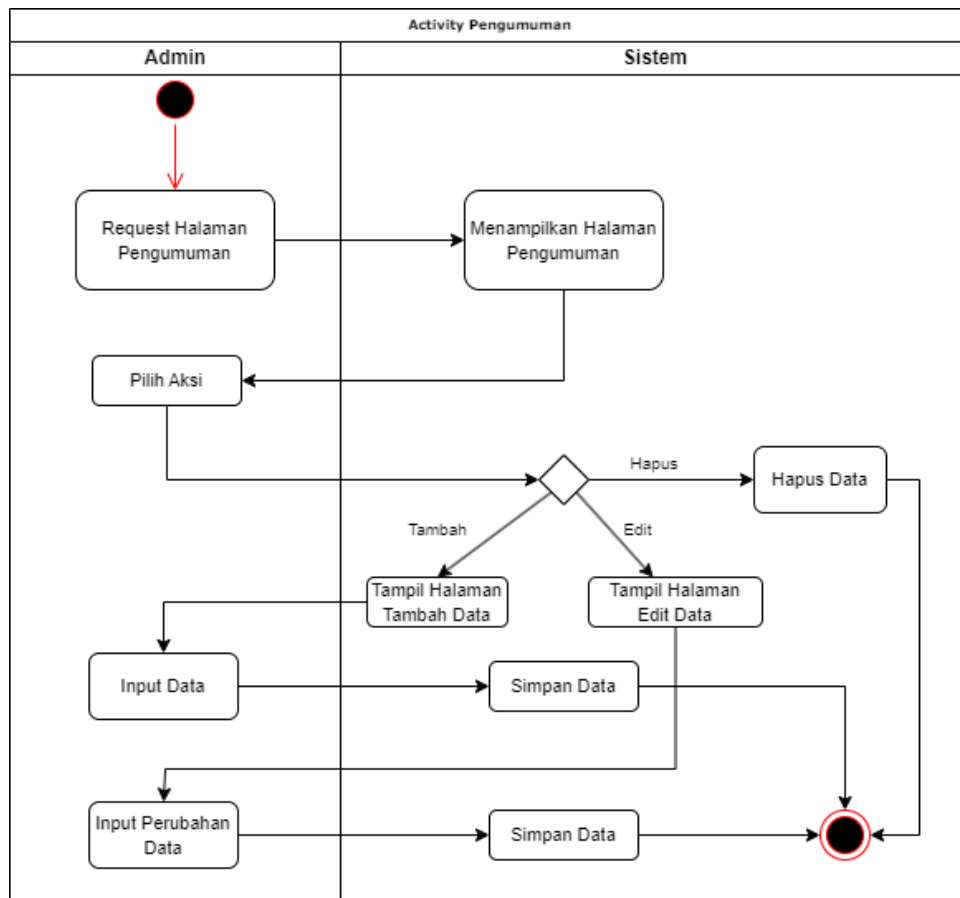
Gambar 3.8 Rancangan *Activity* Diagram Pengaduan (Komplain)

h. *Feedback* Pengaduan (Komplain)



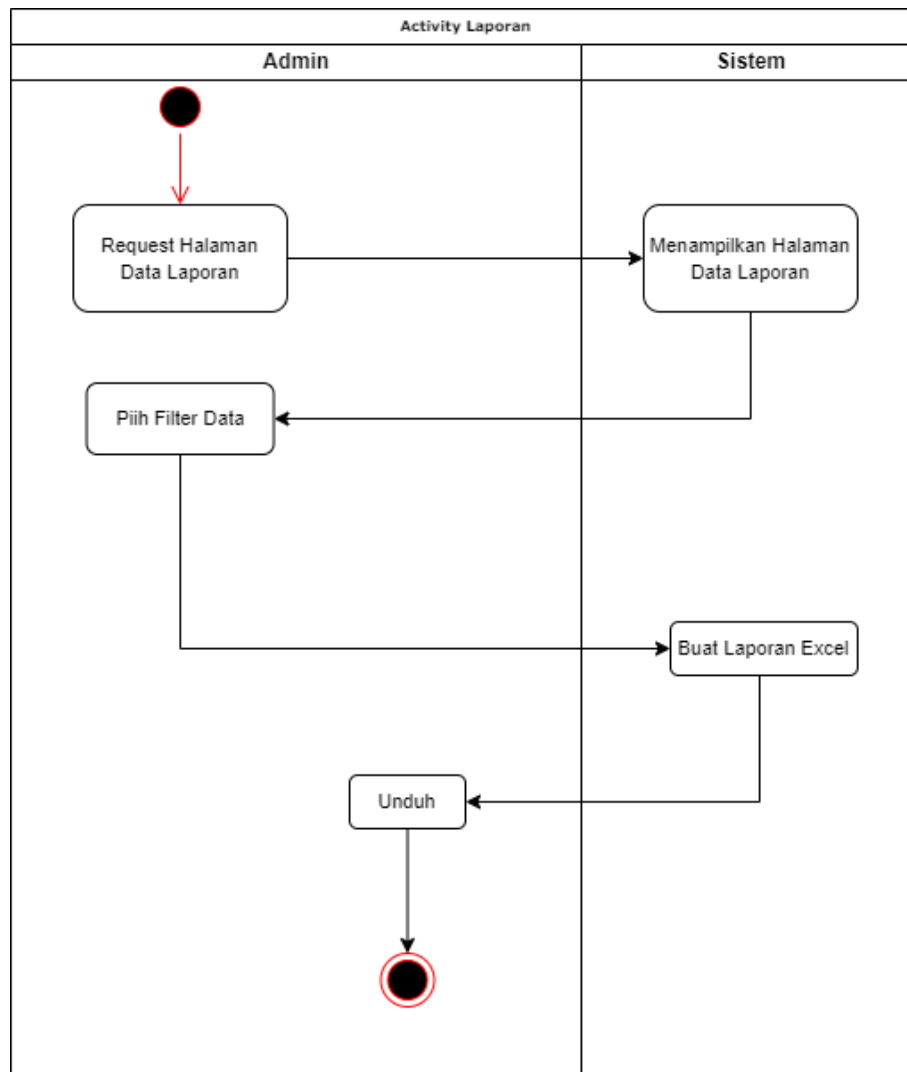
Gambar 3.9 Rancangan *Activity Diagram* *Feedback* Pengaduan (Komplain)

i. Pengumuman



Gambar 3.10 Rancangan *Activity* Diagram Pengumuman

j. Laporan

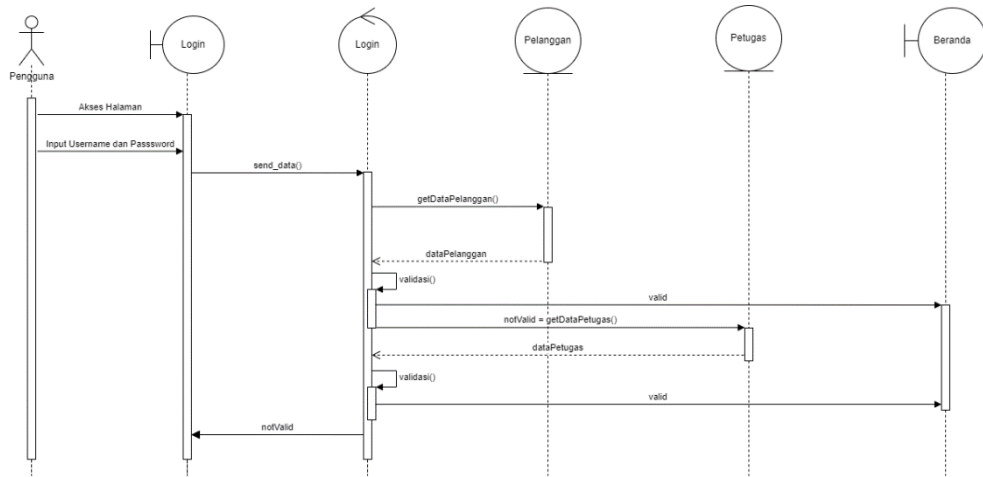


Gambar 3.11 Rancangan *Activity Diagram* Laporan

Pada *Activity Diagram* Laporan digunakan pada laporan Hasil Pencatatan Meteran dan Pembayaran.

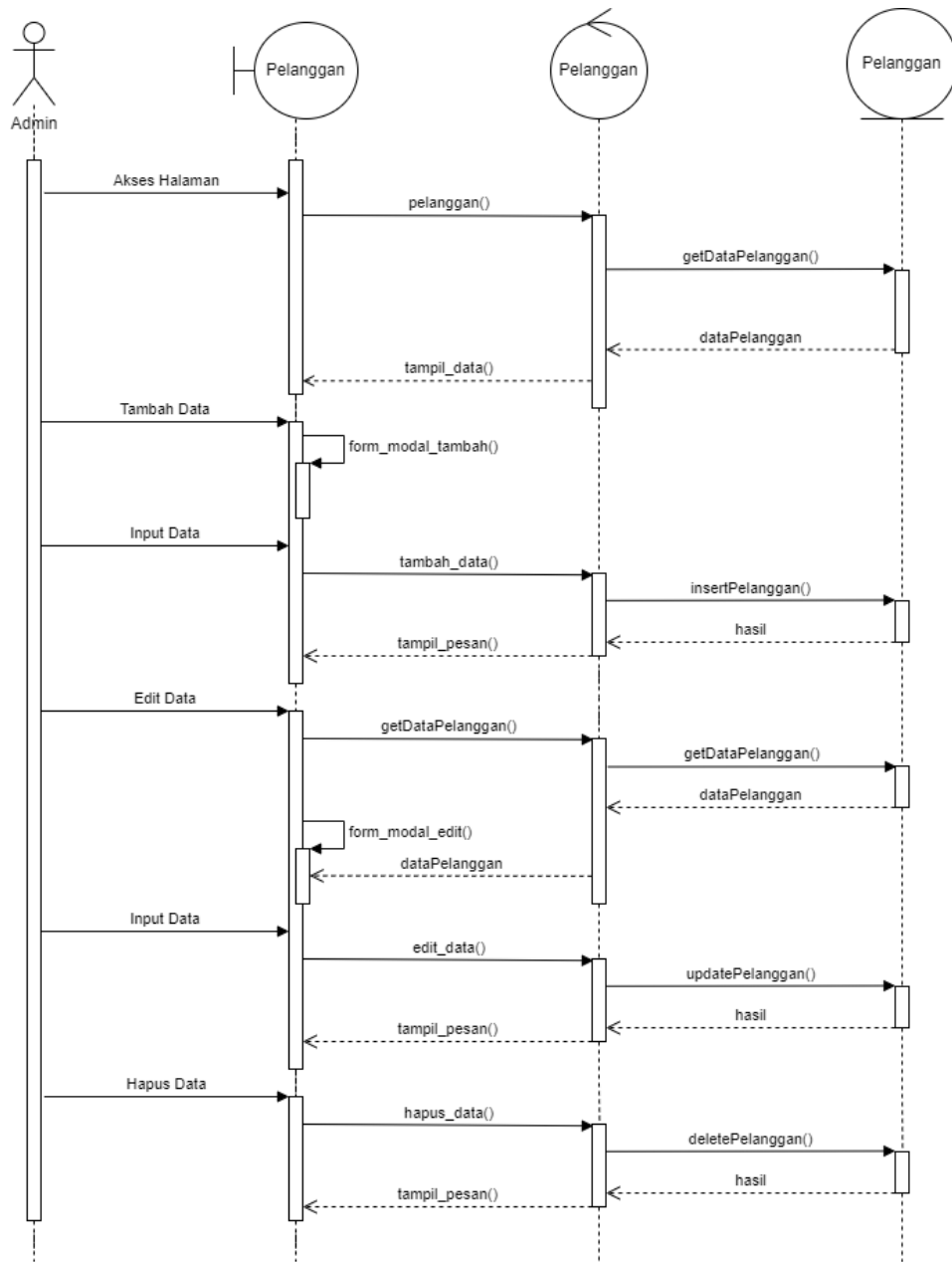
3.2.1.3 Sequence Diagram

a. Login



Gambar 3.12 Rancangan Sequence Diagram Login

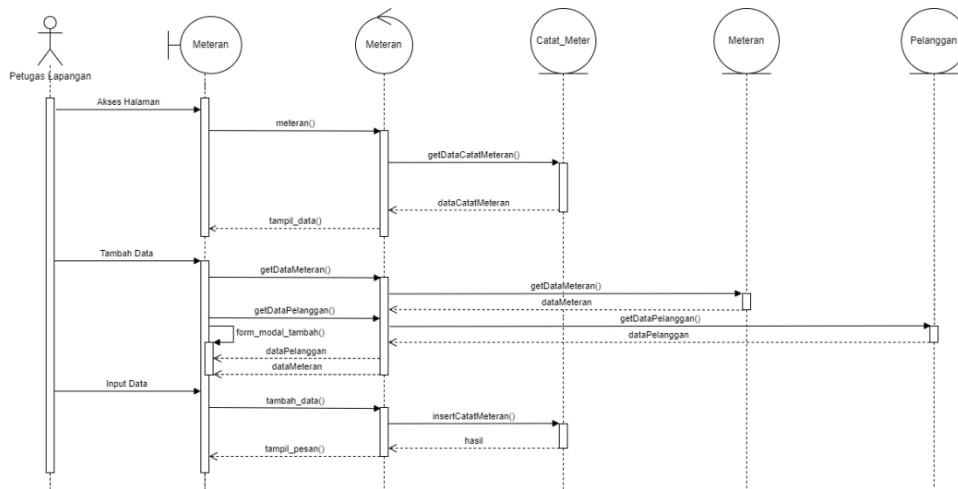
b. Pengelolaan Data Master



Gambar 3.13 Rancangan *Sequence Diagram* Pengelolaan Data Master

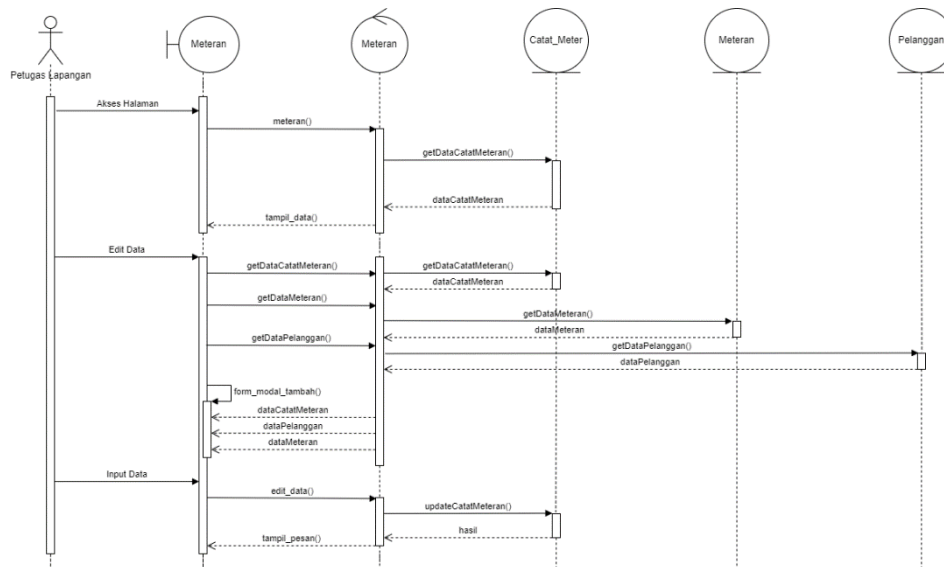
Data Master meliputi Data Pelanggan, Data Petugas, Data Meteran, Data Wilayah, Data RT, Data RW, Data RW, Data Dusun.

c. Pencatatan Meteran



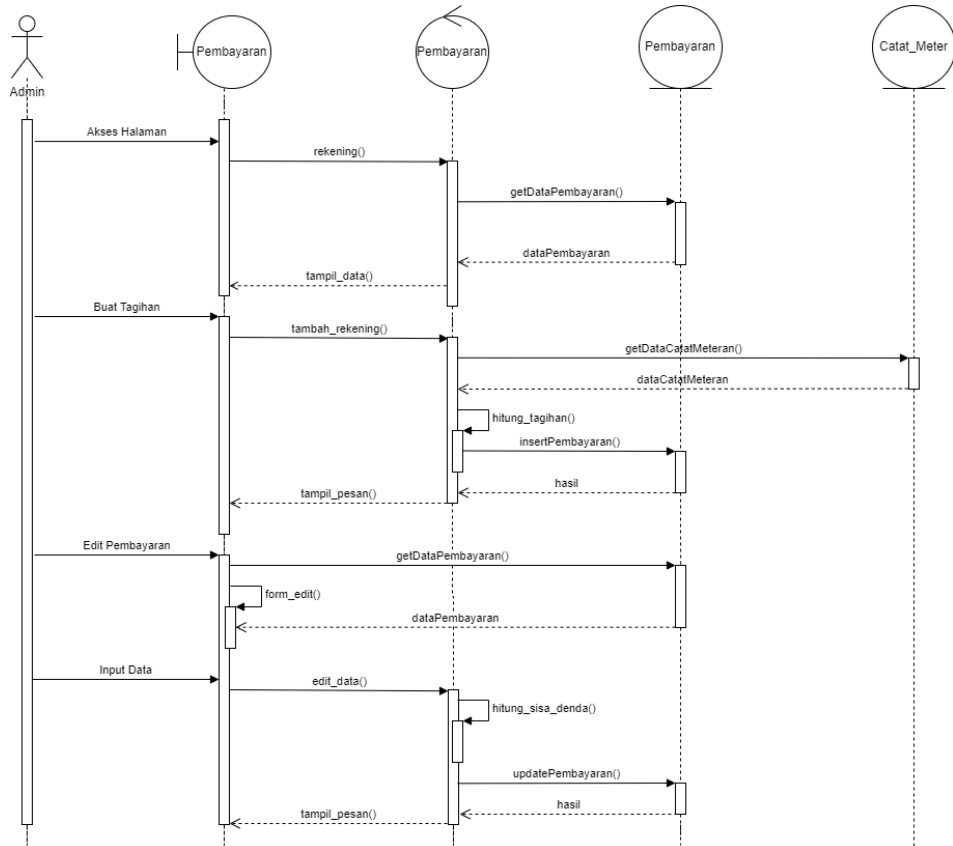
Gambar 3.14 Rancangan *Sequence Diagram* Pencatatan Meteran

d. Validasi Data Hasil Catat Meteran



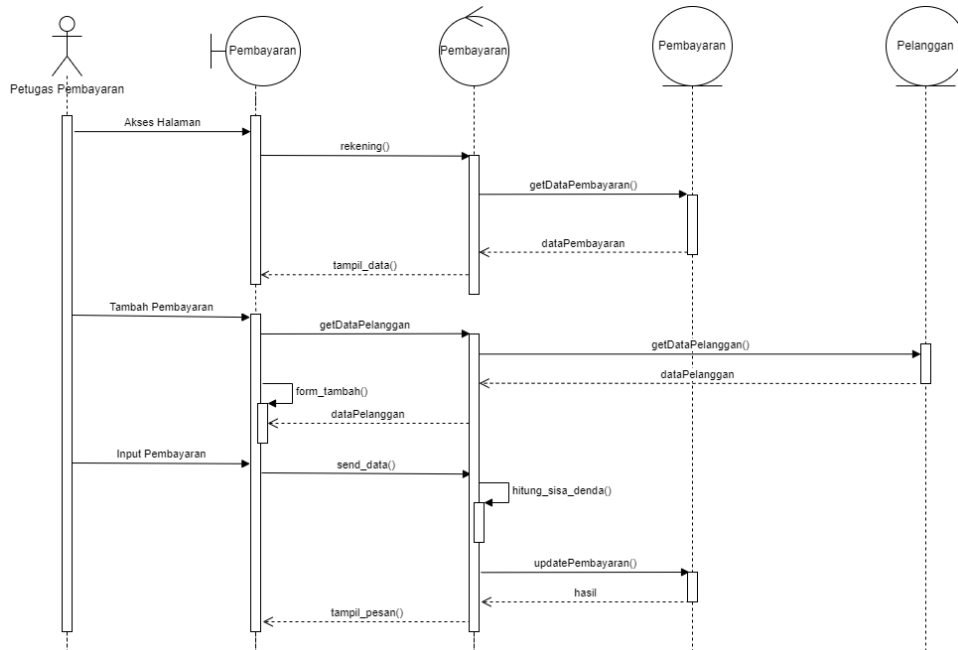
Gambar 3.15 Rancangan *Sequence Diagram* Validasi Data Hasil Catat Meteran

e. Tagihan Pembayaran



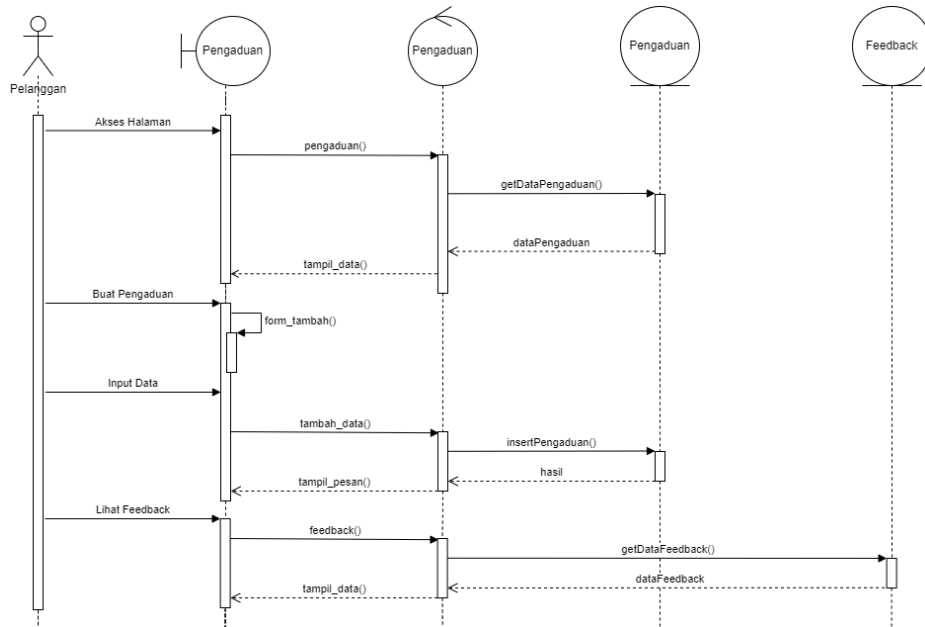
Gambar 3.16 Rancangan *Sequence* Diagram Tagihan Pembayaran

f. Input Data Pembayaran



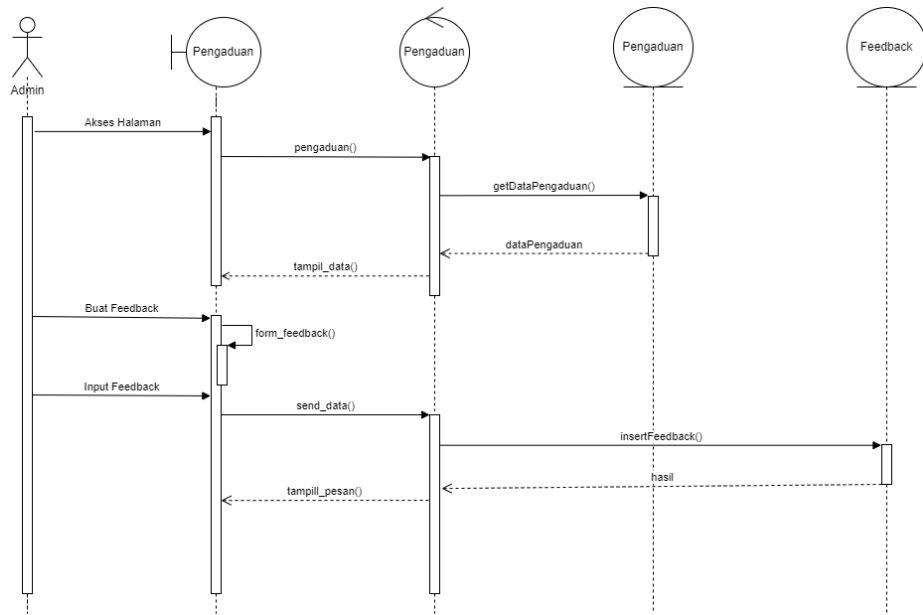
Gambar 3.17 Rancangan *Sequence Diagram* Input Data Pembayaran

g. Pengaduan



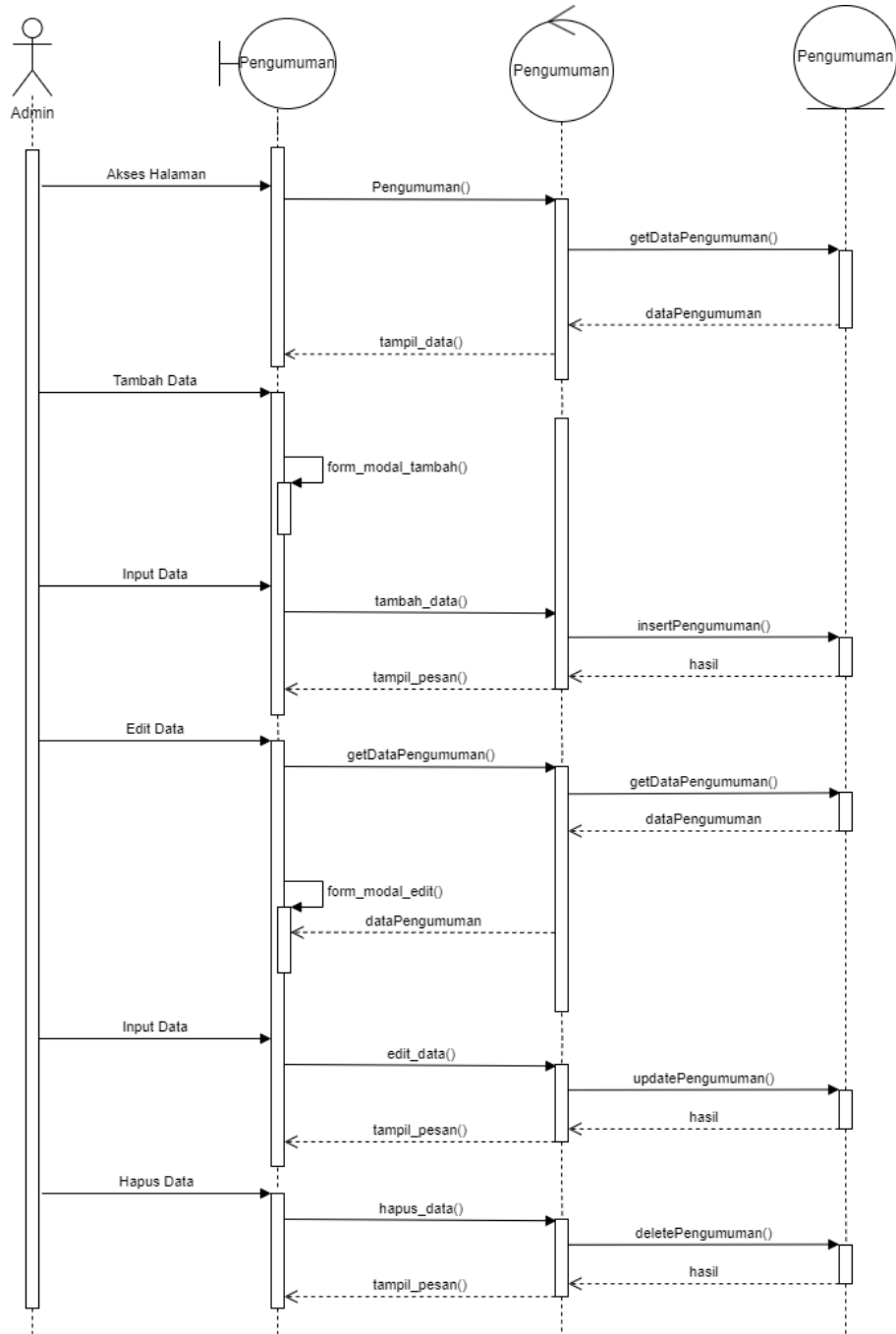
Gambar 3.18 Rancangan *Sequence Diagram* Pengaduan

h. *Feedback Pengaduan*



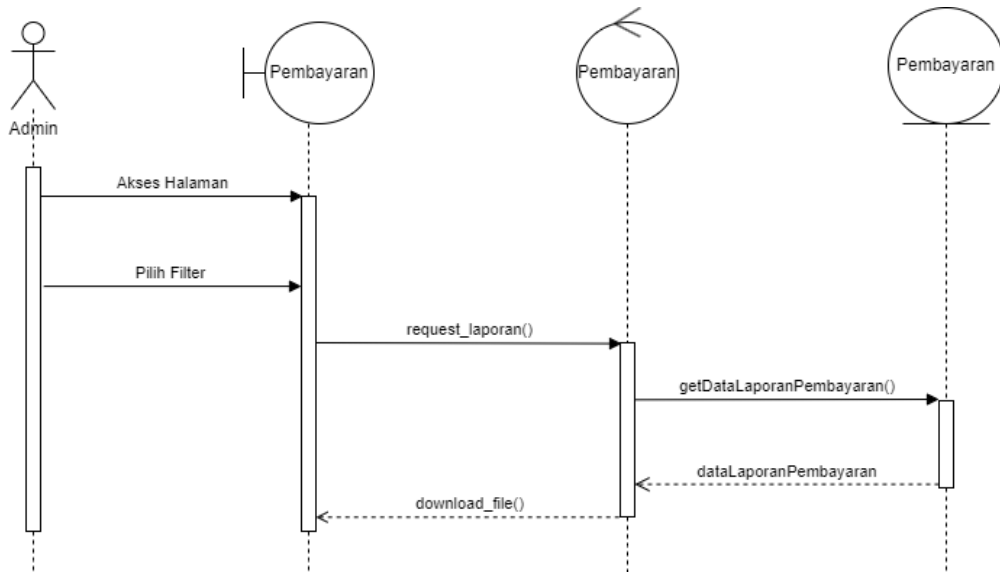
Gambar 3.19 Rancangan *Sequence Diagram Feedback Pengaduan*

i. Pengumuman



Gambar 3.20 Rancangan *Sequence Diagram* Pengumuman

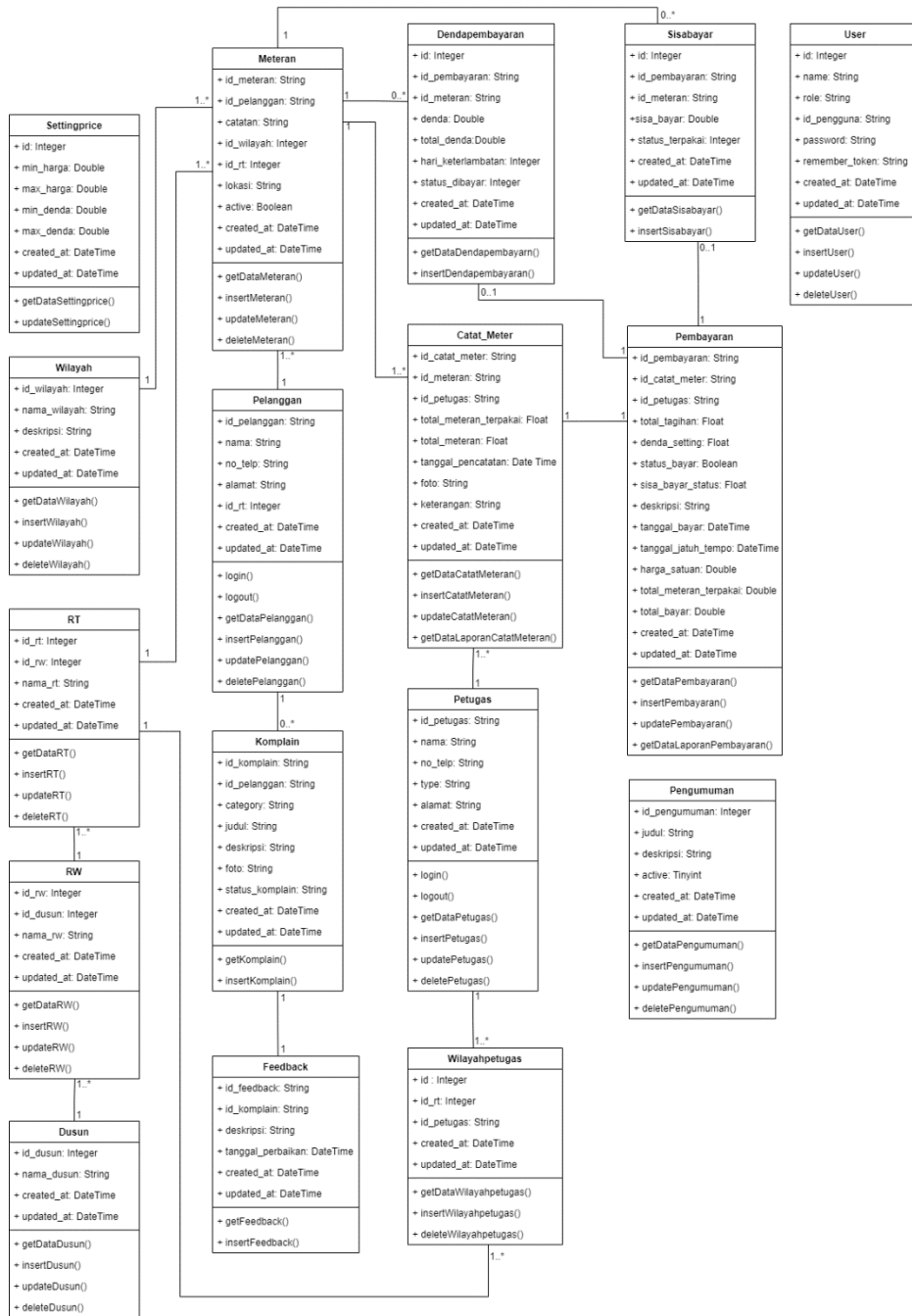
j. Laporan



Gambar 3.21 Rancangan *Sequence* Diagram Laporan

Pada *sequence* diagram laporan akan digunakan pada laporan Hasil Pencatatan Meteran dan Pembayaran.

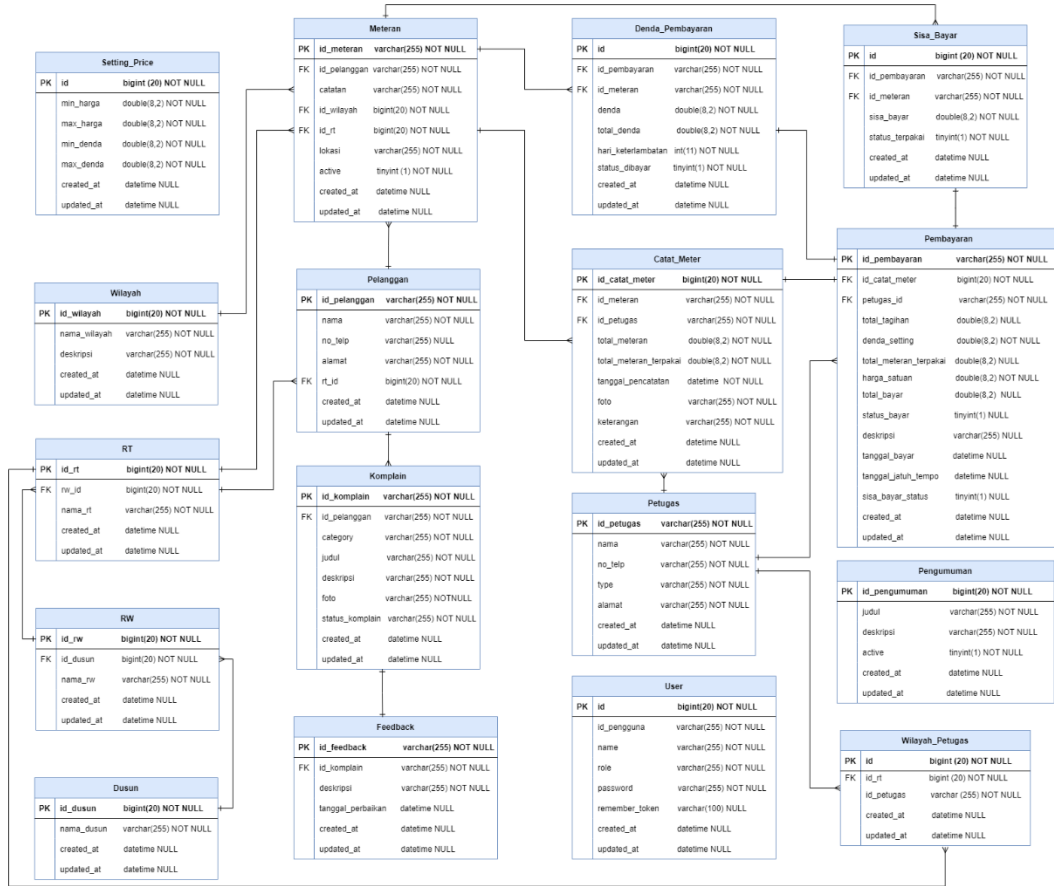
3.2.1.4 Class Diagram



Gambar 3.22 Rancangan *Class Diagram*

3.2.2 Perancangan Data

Setelah penggambaran perancangan sistem, maka selanjutnya menggunakan ERD untuk untuk menggambarkan desain dari database.



Gambar 3.23 Rancangan ERD

Berdasarkan pada Gambar 3.23 tersebut berikut ini penjelasan detail pada ERD:

a. Tabel Meteran

Tabel Meteran menyimpan data meteran milik pelanggan.

Tabel 3.2 Tabel Meteran

Nama Field	Type Data	Keterangan
id_meteran	varchar (255) NOT NULL	Primary Key, berisi nomor ID dari meteran
id_pelanggan	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari pemilik meteran
catatan	varchar (255) NOT NULL	Catatan dari meteran
wilayah_id	bigint (20) NOT NULL	Foreign Key, berisi nomor ID dari wilayah dimana meteran terletak
rt_id	bigint (20) NOT NULL	Foreign Key, berisi nomor ID dari RT dimana meteran terletak
lokasi	varchar(255) NOT NULL	Lokasi atau alamat meteran
active	tinyint (1) NOT NULL	Status meteran aktif atau tidak
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

b. Tabel Pelanggan

Tabel Pelanggan menyimpan data pelanggan.

Tabel 3.3 Tabel Pelanggan

Nama Field	Type Data	Keterangan
id_pelanggan	varchar (255) NOT NULL	Primary Key, berisi nomor ID dari pelanggan
nama	varchar (255) NOT NULL	Nama lengkap dari pelanggan
no_telp	varchar (255) NULL	Nomor telepon dari pelanggan
alamat	varchar (255) NOT NULL	Alamat lengkap dari pelanggan
rt_id	bigint (20) NOT NULL	Foreign Key, berisi nomor ID dari RT pelanggan
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

c. Tabel RT

Tabel RT menyimpan data RT.

Tabel 3.4 Tabel RT

Nama Field	Tipe Data	Keterangan
id_rt	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari RT
rw_id	bigint (20) NOT NULL	Foreign key, berisi nomor ID dari RW
nama_rt	varchar (255) NOT NULL	Berisi RT
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

d. Tabel RW

Tabel RW menyimpan data RW.

Tabel 3.5 Tabel RW

Nama Field	Tipe Data	Keterangan
id_rw	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari RW
dusun_id	bigint(20) NOT NULL	Foreign Key, berisi nomor ID dari dusun
nama_rw	varchar(255) NOT NULL	Berisi RW
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

e. Tabel Dusun

Tabel Dusun menyimpan data Dusun.

Tabel 3.6 Tabel Dusun

Nama Field	Tipe Data	Keterangan
id_dusun	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari dusun
nama_dusun	varchar (255) NOT NULL	Nama dusun

created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

f. Tabel Komplain

Tabel Komplain menyimpan data Komplain dari pelanggan.

Tabel 3.7 Tabel Komplain

Nama Field	Type Data	Keterangan
id_komplain	varchar (255) NOT NULL	Primary Key, berisi nomor ID dari komplain
pelanggan_id	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari RW
category	varchar (255) NOT NULL	Kategori komplain (kerusakan,tagihan,lain-lain)
judul	varchar (255) NOT NULL	Judul komplain
deskripsi	varchar (255) NOT NULL	Penjelasan komplain
foto	varchar (255) NOT NULL	Bukti foto komplain
status_pengaduan	varchar (255) NOT NULL	Status komplain (baru,dalam proses,selesai)
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

g. Tabel Petugas

Tabel Petugas menyimpan data Petugas.

Tabel 3.8 Tabel Petugas

Nama Field	Type Data	Keterangan
id_petugas	varchar (255) NOT NULL	Primary key, berisi nomor ID dari petugas
nama	varchar (255) NOT NULL	Nama lengkap petugas
no_telp	varchar (255) NULL	Nomor telepon petugas
type	varchar (255) NOT NULL	Divisi dari petugas (lapangan,pembayaran,admin)
alamat	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari RT

created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	varchar(255) NULL	Tanggal dan waktu saat data diubah

h. Tabel *Feedback*

Tabel *Feedback* menyimpan *Feedback* pengaduan (komplain) pelanggan dari admin.

Tabel 3.9 Tabel *Feedback*

Nama Field	Tipe Data	Keterangan
id_feedback	varchar (255) NOT NULL	Primary Key, berisi nomor ID dari feedback berjumlah 10 digit
komplain_id	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari pengaduan
deskripsi	archar(255) NOT NULL	<i>Feedback</i> dari admin untuk pengaduan pelanggan
tanggal_perbaikan	datetime NULL	Tanggal perbaikan apabila kategori pengaduan kerusakan
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

i. Tabel *Catat_Meter*

Tabel *Catat_Meter* menyimpan hasil pencatatan meteran.

Tabel 3.10 Tabel *Catat_Meter*

Nama Field	Tipe Data	Keterangan
id_catat_meter	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari pencatatan meteran berjumlah 10 digit
meteran_id	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari meteran
petugas_id	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari petugas lapangan (pencatat meteran)
tanggal_pencatatan	datetime NOT NULL	Tanggal petugas melakukan pencatatan meteran
total_meteran	double(8,2) NOT NULL	Total meteran dari pelanggan
total_meteran_terpakai	double(8,2) NOT NULL	Total meteran bulan saat ini dikurangkan dengan bulan sebelumnya

foto	varchar(255) NOT NULL	Foto meteran pelanggan
keterangan	varchar(255) NOT NULL	Keterangan apabila meteran tidak terbaca atau tidak terjangkau
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

j. Tabel Pembayaran

Tabel Pembayaran menyimpan data Pembayaran pelanggan.

Tabel 3.11 Tabel Pembayaran

Nama Field	Tipe Data	Keterangan
id_pembayaran	varchar (255) NOT NULL	Primary Key, berisi nomor ID dari pembayaran berjumlah 10 digit
catat_meter_id	varchar (255) NULL	Foreign Key, berisi nomor ID dari hasil pencatatan meteran
petugas_id	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari petugas pembayaran
status_bayar	tinyint(1) NULL	Status pembayaran (lunas/ belum lunas)
sisa_bayar_status	tinyint(1) NULL	Uang sisa dari pelanggan saat membayar
harga_satuan	double(8,2) NOT NULL	Harga per m ³
denda_setting	double(8,2) NOT NULL	Denda apabila terlambat membayar
total_meteran_terpakai	double(8,2) NULL	Total meteran bulan saat ini dikurangkan dengan bulan sebelumnya
total_bayar	double(8,2) NULL	Total pembayaran yang dilakukan oleh pelanggan
total_tagihan	double(8,2) NULL	Total tagihan pelanggan (dikurangi dengan sisa dan ditambahkan dengan denda bulan sebelumnya)
deskripsi	varchar (255) NOT NULL	Catatan dari petugas
tanggal_bayar	datetime NULL	Tanggal pembayaran pelanggan
tanggal_jatuh_tempo	datetime NULL	Tanggal jatuh tempo pembayaran
created_at	datetime NULL	Tanggal dan waktu saat data

		dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

k. Tabel Wilayah

Tabel Wilayah menyimpan data Wilayah.

Tabel 3.12 Tabel Wilayah

Nama Field	Tipe Data	Keterangan
id_wilayah	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari wilayah
nama_wilayah	varchar (255) NOT NULL	Nama wilayah
deskripsi	varchar(255) NOT NULL	Catatan
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

l. Tabel Sisa Bayar

Tabel Sisa Bayar menyimpan sisa bayar dari pelanggan saat melakukan pembayaran jika ingin digunakan untuk tagihan bulan berikutnya.

Tabel 3.13 Tabel Sisa Bayar

Nama Field	Tipe Data	Keterangan
id	bigint(20) NOT NULL	Primary Key, berisi nomor ID dari sisa bayar
id_pembayaran	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari pembayaran
id_meteran	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari meteran
sisa_bayar	double(8,2) NOT NULL	Kembalian dari pembayaran pelanggan
status_terpakai	tinyint(1) NOT NULL	Status jika sisa bayar di pakai untuk tagihan bulan berikutnya atau tidak
created_at	datetime NULL	Tanggal dan waktu saat data dibuat

update_at	datetime NULL	Tanggal dan waktu saat data diubah
-----------	---------------	------------------------------------

m. Tabel Setting Price

Tabel Setting Price menyimpan harga dan denda untuk pembayaran pelanggan.

Tabel 3.14 Tabel Setting Price

Nama Field	Tipe Data	Keterangan
id	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari setting price
max_harga	double (8,2) NOT NULL	Harga pemakaian per m ³ jika pemakaian lebih dari 30 m ³
min_harga	double (8,2) NOT NULL	Harga pemakaian per m ³ jika pemakaian kurang dari 30 m ³
max_denda	double (8,2) NOT NULL	Denda maksimal akumulasi per hari keterlambatan pembayaran
min_denda	double (8,2) NOT NULL	Denda per hari keterlambatan pembayaran
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

n. Tabel Denda Pembayaran

Tabel Denda Pembayaran menyimpan denda pelanggan ketika melakukan pembayaran untuk diakumulasi ke tagihan berikutnya.

Tabel 3.15 Tabel Denda Pembayaran

Nama Field	Tipe Data	Keterangan
id	bigint(20) NOT NULL	Primary Key, berisi nomor ID dari denda pembayaran
id_pembayaran	varchar (255) NOT NULL	Foreign Key, berisi nomor ID dari pembayaran
id_meteran	varchar (255) NOT	Foreign Key, berisi nomor ID dari

	NULL	meteran
denda	double (8,2) NOT NULL	Denda per hari keterlambatan pembayaran
total_denda	double (8,2) NOT NULL	Total denda per hari keterlambatan pembayaran
hari_keterlambatan	int(11) NULL	Total hari keterlambatan pembayaran
status_terpakai	tinyint(1) NULL	Status sisa bayar dipakai atau tidak
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
update_at	datetime NULL	Tanggal dan waktu saat data diubah

o. Tabel Pengumuman

Tabel Pengumuman menyimpan data pengumuman.

Tabel 3.16 Tabel Pengumuman

Nama Field	Type Data	Keterangan
id_pengumuman	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari pengumuman
judul	varchar (255) NOT NULL	Judul pengumuman
deskripsi	varchar (255) NOT NULL	Deskripsi pengumuman
active	tinyint (1) NOT NULL	Status pengumuman ditampilkan atau tidak
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

p. Tabel User

Tabel Pengumuman menyimpan data user yang digunakan untuk login.

Tabel 3.17 Tabel User

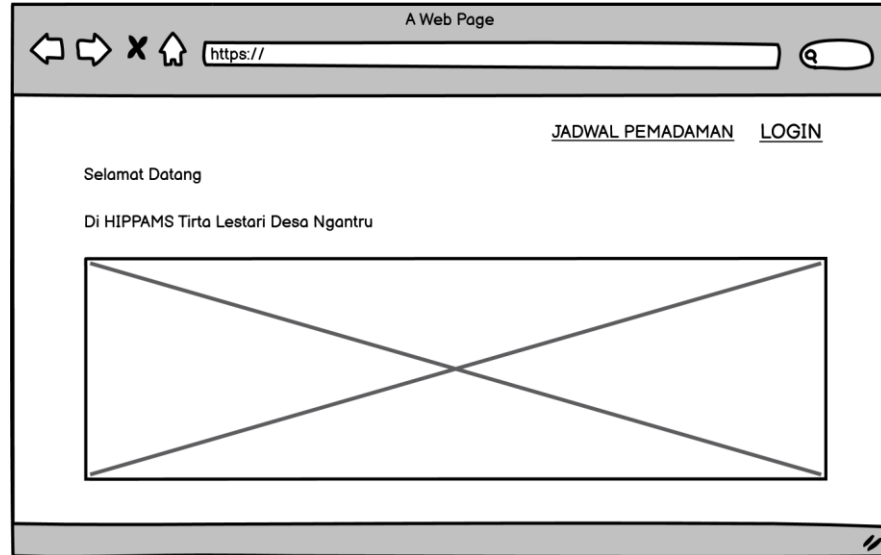
Nama Field	Tipe Data	Keterangan
id	bigint (20) NOT NULL	Primary Key, berisi nomor ID dari user
id_pengguna	varchar (255) NOT NULL	ID dari user admin, pelanggan, dan petugas yang digunakan untuk login
name	varchar (255) NOT NULL	Nama dari user admin, pelanggan, dan petugas
role	varchar (255) NOT NULL	Role user
password	varchar (255) NOT NULL	Password user
remember_token	varchar (100) NULL	Remember token user
created_at	datetime NULL	Tanggal dan waktu saat data dibuat
updated_at	datetime NULL	Tanggal dan waktu saat data diubah

3.2.3 Perancangan *User Interface / Mockup* Aplikasi

Pada tahap perancangan *user interface / mockup* aplikasi ini penulis menggunakan *wireframe*. Berikut ini rancangan *wireframe* aplikasi:

a. Tampilan tanpa autentikasi yang bisa diakses semua pengguna.

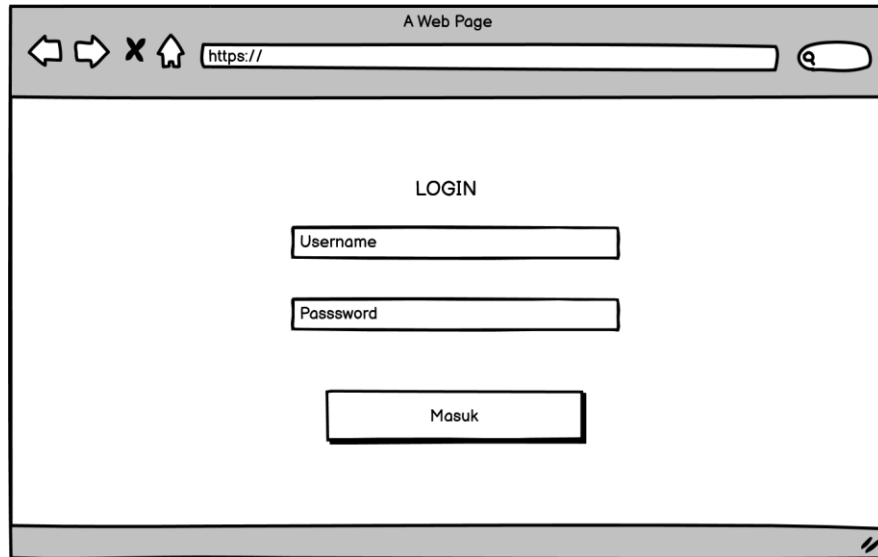
1. *Landing Page*



Gambar 3.24 Rancangan *Mockup Landing Page*

Halaman *Landing Page* berisi pesan mengenai fitur pada aplikasi dan navigasi menuju halaman *login* serta pengumuman.

2. Login

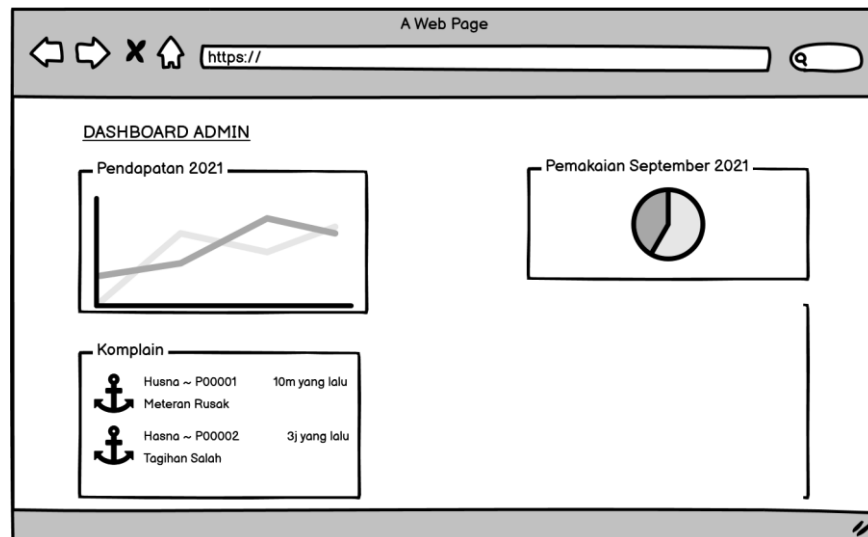


Gambar 3.25 Rancangan *Mockup Login*

Halaman *login* menginputkan username dan password yang valid agar bisa masuk ke aplikasi.

b. Tampilan Admin

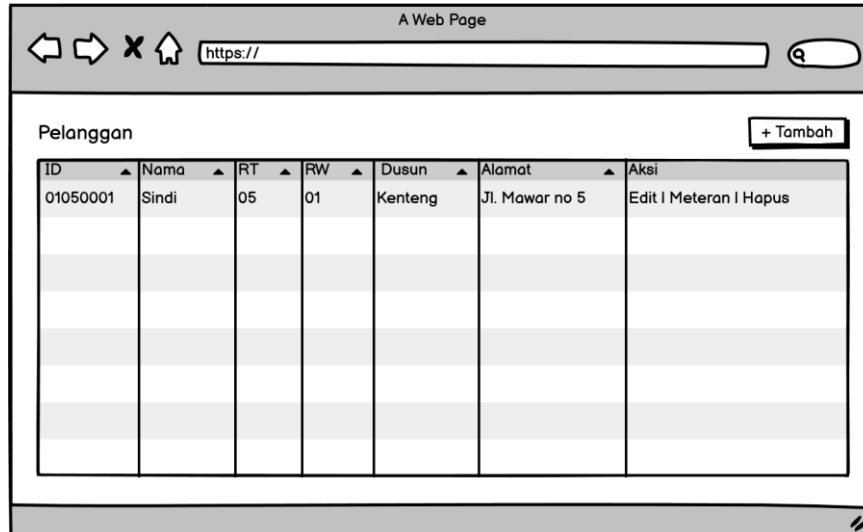
1. *Dashboard Admin*



Gambar 3.26 Rancangan *Mockup Dashboard Admin*

Halaman *Dashboard* Admin menampilkan pendapatan tahun saat ini berdasarkan bulan, presentase pemakaian air bulan sebelumnya berdasarkan wilayah, pengaduan (komplain) yang belum ditangani admin.

2. *List* Pelanggan

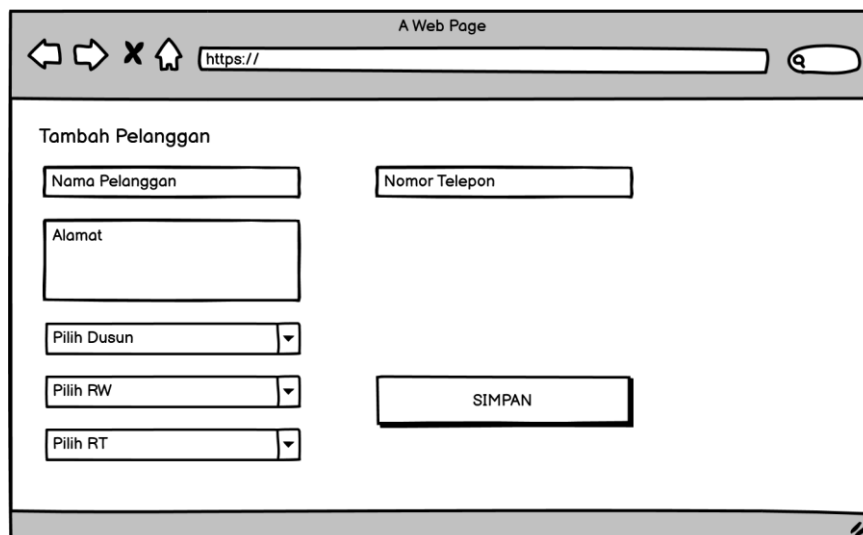


ID	Nama	RT	RW	Dusun	Alamat	Aksi
01050001	Sindi	05	01	Kenteng	Jl. Mawar no 5	Edit Meteran Hapus

Gambar 3.27 Rancangan *Mockup List* Pelanggan

Halaman *List* Pelanggan menampilkan data – data pelanggan.

3. Tambah Pelanggan



Tambah Pelanggan

Nama Pelanggan Nomor Telepon

Alamat

Pilih Dusun

Pilih RW

Pilih RT

SIMPAN

Gambar 3.28 Rancangan *Mockup* Tambah Pelanggan