



RANCANG BANGUN APLIKASI *DEBT COLLECTING* PADA PT.
TELEKOMUNIKASI INDONESIA DIVISI REGIONAL V

KERJA PRAKTIK

Program Studi

S1 Sistem Informasi

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Oleh :

MOH. FACHRUL ARIFIN

15.41010.0090

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

**RANCANG BANGUN APLIKASI *DEBT COLLECTING* PADA PT.
TELEKOMUNIKASI INDONESIA DIVISI REGIONAL V**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

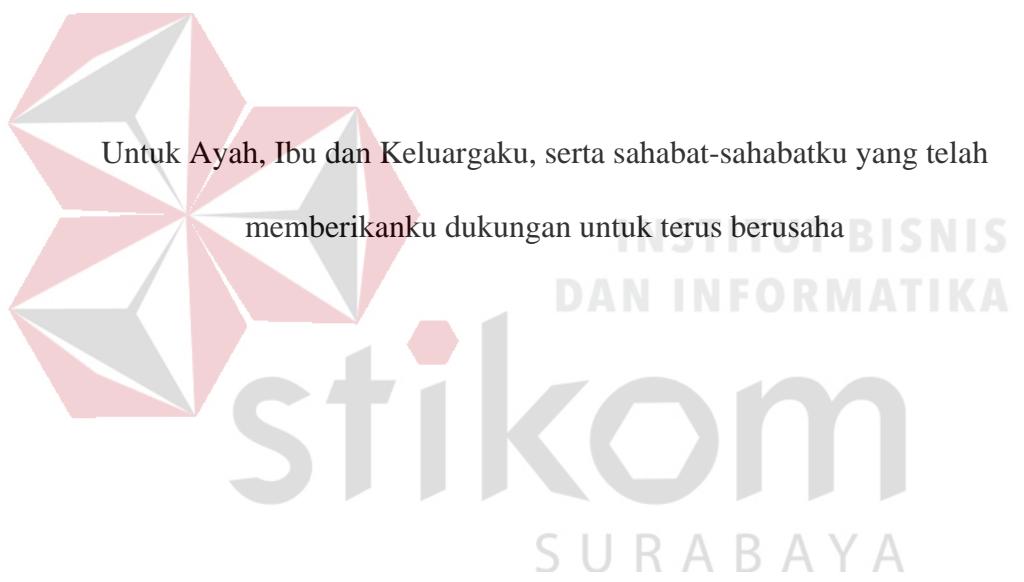
2019



“Live life to the fullest, don’t let people opinion get you”

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA



Untuk Ayah, Ibu dan Keluargaku, serta sahabat-sahabatku yang telah
memberikanku dukungan untuk terus berusaha

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI *DEBT COLLECTING* PADA PT. TELEKOMUNIKASI INDONESIA DIVISI REGIONAL V

Laporan Kerja Praktik Oleh

Moh. Fachrul Arifin

NIM : 15410100090

Telah diperiksa, diuji dan disetujui



Ir. Henry Bambang Setyawan, M.M.

NIDN : 0725055701

Surabaya, 21 Januari 2019

Disetujui:

INSTITUT BISNIS
DAN INFORMATIKA

I Gede Sudarmawan
Penyelia
Manager Digital Service and
Wifi

 Mengetahui:

Ketua Program Studi S1 Sistem Informasi



Dr. Anjik Sukmaali, S.Kom., M.Eng.

NIDN : 0731057301

SURAT PERNYATAAN

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Moh. Fachrul Arifin

NIM : 15410100090

Program Studi : S1 Sistem Informasi

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Kerja Praktik

Judul Karya : **RANCANG BANGUN APLIKASI DEBT COLLECTING
PADA PT. TELEKOMUNIKASI INDONESIA DIVISI
REGIONAL V**

menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmedianakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 7 Januari 2019



Moh. Fachrul Arifin
NIM : 15410100090

ABSTRAK

PT. Telekomunikasi Indonesia Regional V merupakan perpanjangan dari PT. Telekomunikasi Indonesia Jawa Timur yang beralamat JL. Raya Ketintang No.156, Ketintang, Gayungan, Kota SBY, Jawa Timur. PT. Telekomunikasi Indonesia Divisi Regional V (Telkom) memiliki cukup banyak masalah piutang. Proses penagihan piutang pada PT. Telkom masih dilakukan secara manual oleh bagian *Collector (Debt Collector)* sehingga mengalami kesulitan untuk menyelesaikan proses tersebut secara cepat.

Kondisi inilah yang membuat PT. Telekomunikasi Indonesia Regional V menginginkan untuk memiliki aplikasi yang dapat membantu proses dalam penagihan tanpa harus membawa banyak dokumen ke *customer*. Melalui aplikasi *debt collecting* ini, maka *debt collector* memiliki akses terhadap menu yang ditawarkan secara *realtime* yang berisi data *customer* dan utang yang dimilikinya.

Berdasarkan hasil uji coba aplikasi yang telah dibuat, aplikasi ini mampu menampilkan data secara *realtime*. Aplikasi yang dibangun membutuhkan koneksi untuk mengakses *database* yang telah disediakan dan pengguna tidak dapat mengakses aplikasi yang dibangun dikarenakan data-data sudah tersimpan pada *cloud* untuk setiap proses yang dilakukannya. Dengan demikian aplikasi ini mampu membantu pengguna dalam memantau informasi yang berkaitan.

Kata Kunci: *Debt collecting, Debt collector, Realtime.*

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa, karena berkat dan rahmat yang diberikan, penulis dapat menyelesaikan Laporan yang berjudul “Rancang Bangun Aplikasi *Debt Collecting* Pada PT. Telekomunikasi Indonesia Divisi Regional V”. Laporan ini disusun berdasarkan hasil studi dalam pelaksanaan kerja praktik di Telekomunikasi Indoensia Divisi Regional V yang dilakukan selama satu bulan. Penulis juga menyampaikan terima kasih kepada:

1. Kedua orang tua sekeluarga dan teman-teman yang selalu mendoakan dan memberikan dukungan untuk menyelesaikan penyusunan laporan ini.
2. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng. selaku Ketua Program Studi Sistem Informasi Institut Bisnis dan Informatika Stikom Surabaya.
3. Bapak Ir. Henry Bambang Setyawan, M.M. selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan bimbingan dan arahan selama pelaksanaan Kerja Praktik
4. Ibu Jayanti Sukma Maulani selaku penyelia yang telah memberikan ijin kepada penulis dalam melaksanakan kerja praktik.

Penulis menyadari dalam penulisan Laporan Kerja Praktik ini masih terdapat banyak celah dan kekurangan. Oleh karena itu, saran dan kritik sangat diharapkan oleh penulis untuk mengembangkan kemampuan penulis di masa yang akan datang.

Surabaya, 7 Januari 2019

Penulis

DAFTAR ISI

Halaman

| | |
|--|----------|
| ABSTRAK | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI..... | ix |
| DAFTAR GAMBAR | xii |
| DAFTAR TABEL..... | xiv |
| DAFTAR LAMPIRAN | xv |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang Masalah | 1 |
| 1.2 Perumusan Masalah..... | 2 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan..... | 3 |
| 1.5 Manfaat..... | 3 |
| 1.6 Sistematika Penulisan..... | 3 |
| BAB II GAMBARAN UMUM PERUSAHAAN..... | 5 |
| 2.1 Profile PT. Telekomunikasi Indonesia (Persero)..... | 5 |
| 2.2 Visi | 6 |
| 2.3 Misi..... | 6 |
| 2.4 Logo Perusahaan | 6 |

| | | |
|----------------------------------|-----------------------------------|----|
| 2.5 | Struktur Organisasi..... | 7 |
| 2.6 | Lokasi Perusahaan..... | 7 |
| BAB III LANDASAN TEORI..... | | 8 |
| 3.1 | Piutang..... | 8 |
| 3.2 | Android..... | 8 |
| 3.3 | Android Studio | 9 |
| 3.4 | Java | 10 |
| 3.5 | <i>Database</i> | 10 |
| 3.6 | <i>Firebase</i> | 11 |
| BAB IV DESKRIPSI PEKERJAAN | | 12 |
| 4.1 | Identifikasi Masalah | 12 |
| 4.2 | Analisis Bisnis | 12 |
| 4.3 | Analisis Sistem | 13 |
| 4.3.1 | Observasi..... | 13 |
| 4.3.2 | Wawancara..... | 13 |
| 4.3.3 | Analisis proses bisnis | 13 |
| 4.3.4 | Analisis Kebutuhan Pengguna | 14 |
| 4.3.5 | Kebutuhan Fungsional | 14 |
| 4.3.6 | Kebutuhan Non-Fungsional | 15 |
| 4.4 | Perancangan Sistem..... | 15 |
| 4.4.1 | <i>Activity Diagram</i> | 16 |

| | | |
|----------------------|---|----|
| 4.4.2 | <i>Flow Of Document</i> | 24 |
| 4.4.3 | <i>Sequence Diagram</i> | 31 |
| 4.4.4 | <i>Class Diagram</i> | 38 |
| 4.5 | Desain Implementasi Sistem | 40 |
| 4.5.1 | Halaman <i>Login</i> | 40 |
| 4.5.2 | Halaman Awal Admin..... | 41 |
| 4.5.3 | Halaman <i>Collector</i> | 42 |
| 4.5.4 | Halaman <i>Customer</i> | 44 |
| 4.5.5 | Halaman <i>Debt</i> | 46 |
| 4.5.6 | Halaman Menu Utama <i>Collector</i> | 48 |
| 4.5.7 | Halaman Konfirmasi | 51 |
| BAB V PENUTUP..... | | 52 |
| 5.1 | Simpulan..... | 52 |
| 5.2 | Saran..... | 52 |
| DAFTAR PUSTAKA | | 53 |
| LAMPIRAN | | 54 |

DAFTAR GAMBAR

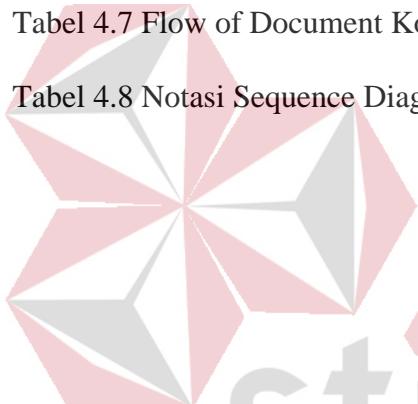
| | Halaman |
|---|----------------|
| Gambar 2.1 Logo PT. Telekomunikasi Indonesia (Persero)..... | 6 |
| Gambar 2.2 Struktur Organisasi Perusahaan | 7 |
| Gambar 2.3 Lokasi Perusahaan..... | 7 |
| Gambar 4.1 Use case Diagram aplikasi | 15 |
| Gambar 4.2 Activity Diagram Login | 18 |
| Gambar 4.3 Activity Diagram Mengelolah data Collector | 19 |
| Gambar 4.4 Activity Diagram Mengelolah data Customer | 20 |
| Gambar 4.5 Activity Diagram Menambahkan Piutang..... | 21 |
| Gambar 4.6 Activity Diagram Update Status Customer..... | 22 |
| Gambar 4.7 Activity Diagram Konfirmasi Piutang | 23 |
| Gambar 4.8 Sequence Diagram Login | 33 |
| Gambar 4.9 Sequence Diagram Mengelolah data Collector..... | 34 |
| Gambar 4.10 Sequence Diagram Mengelolah data Customer | 35 |
| Gambar 4.11 Sequence Diagram Menambahkan Piutang..... | 36 |
| Gambar 4.12 Sequence Diagram Update Status Customer..... | 37 |
| Gambar 4.13 Sequence Diagram Konfirmasi Piutang | 38 |
| Gambar 4.14 Class Diagram Debt Collect..... | 39 |
| Gambar 4.15 Implementasi Menu Login | 40 |
| Gambar 4.16 Implementasi Menu Admin..... | 41 |
| Gambar 4.17 Implementasi halaman collector..... | 42 |
| Gambar 4.18 Implementasi tambah collector | 43 |
| Gambar 4.19 Implementasi halaman customer | 44 |

| | |
|---|----|
| Gambar 4.20 Implementasi tambah Customer..... | 45 |
| Gambar 4.21 Implementasi menu Debt | 46 |
| Gambar 4.22 Dialog memasukkan biaya | 47 |
| Gambar 4.23 Implementasi menu utama Collector..... | 48 |
| Gambar 4.24 Rincian data Customer | 49 |
| Gambar 4.25 Konfirmasi perubahan status | 50 |
| Gambar 4.26 Implementasi halaman konfirmasi Admin | 51 |



DAFTAR TABEL

| | Halaman |
|---|----------------|
| Tabel 4.1 Notasi Activity Diagram | 16 |
| Tabel 4.2 Flow of Document Login..... | 24 |
| Tabel 4.3 Flow of Document Mengelola Master Customer..... | 25 |
| Tabel 4.4 Flow of Document Mengelola Master Collector | 27 |
| Tabel 4.5 Flow of Document Menambahkan Piutang | 28 |
| Tabel 4.6 Flow of Document Update Status Customer..... | 30 |
| Tabel 4.7 Flow of Document Konfirmasi Piutang | 30 |
| Tabel 4.8 Notasi Sequence Diagram..... | 32 |



INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

DAFTAR LAMPIRAN

Halaman

| | |
|---|----|
| Lampiran 1, Surat Balasan Institusi | 54 |
| Lampiran 2, Form KP-5 | 55 |
| Lampiran 3, Form KP-6 | 57 |
| Lampiran 4, Form KP-7 | 58 |
| Lampiran 5, Kartu Bimbingan KP | 59 |
| Lampiran 6, Biodata Penulis | 60 |



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan bisnis di Indonesia semakin berkembang. Hal ini, membuat perusahaan-perusahaan yang ada di Indonesia saling bersaing untuk merebut perhatian konsumen. Setiap perusahaan atau organisasi akan melakukan berbagai aktivitas untuk mencapai tujuan yang telah ditetapkan perusahaan yaitu untuk mencapai laba sebesar-besarnya, oleh karena itu diperlukan berbagai strategi dan kebijakan dalam mengambil keputusan yang membuat perusahaan dapat bersaing dan bertahan dalam setiap situasi.

Salah satu cara untuk memperoleh laba sebesar-besarnya adalah dengan pemberian jasa kredit yang dapat menimbulkan piutang. Salah satu penyebab kegagalan sebuah perusahaan adalah kurang baiknya manajemen yang dilaksanakan oleh perusahaan yang bersangkutan dalam mengelola perusahaan. Untuk mengantisipasi kemungkinan tersebut, maka fungsi-fungsi manajemen seperti pengendalian, perencanaan, pengorganisasian, dan pengarahan yang merupakan satu kesatuan yang tidak dapat dipisahkan. Dengan tertatanya pengarahan suatu manajemen dapat menyelesaikan masalah piutang dengan cepat.

Ada beberapa cara untuk menyelesaikan piutang salah satunya adalah penagihan. Penagihan adalah proses atau kegiatan melakukan tagihan kepada *customer* atau seseorang yang telah melakukan proses utang. Bagian *Debt Collector* (Seseorang yang menagih) harus melakukan monitor piutang terhadap para *customernya*. Hal ini tidak cukup hanya dengan cara menyampaikan tagihan saja,

tetapi juga bertugas untuk mengumpulkan seluruh piutang para pelanggan. Prosedur penagihan diawali dengan pengiriman surat peringatan, jika tidak ada *respond* maka diingatkan kembali melalui via telepon. Apabila masih juga belum ada *respond*, barulah bagian *Debt Collection* perlu melakukan negosiasi dan kesepakatan pembayaran terhadap pelanggan tersebut.

PT. Telekomunikasi Indonesia Divisi Regional V (Telkom) memiliki cukup banyak masalah piutang. Proses penagihan pada PT. Telkom masih dilakukan secara manual oleh bagian *Collector (Debt Collector)*. Dalam proses penagihan, bagian *Debt Collector* memiliki tugas mulai dari mengatur berkas daftar *customer* yang memiliki piutang, menentukan tanggal penagihan, menugaskan proses penagihan,dll. Dalam melakukan proses penagihan sering terjadinya masalah seperti pihak *customer* bahwa mengatakan mereka sudah membayar tetapi pihak *Debt Collector* mengatakan sebaliknya, terlambatnya proses penyetoran kepada bagian *Debt Collector*, kesalahan *customer* yang ditagih, dll. Oleh karena itu, dibutuhkannya aplikasi yang dapat memudahkan melaksanakan proses penagihan piutang customer.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan yang ada, maka dapat dirumuskan permasalahan adalah:

“Bagaimana membuat rancang bangun aplikasi penagihan piutang pada PT. Telekomunikasi Indonesia Divisi Regional V.”

1.3 Batasan Masalah

Dalam sistem ini, agar tidak menyimpang dari tujuan yang akan dicapai maka pembahasan masalah dibatasi pada hal-hal sebagai berikut:

1. Aplikasi *android* hanya menampilkan data *customer* yang memiliki utang
2. Aplikasi *android* tidak memiliki akses untuk mengubah data *customer* kecuali Status
3. Penambahan, pengurangan, atau perubahan data *customer* dilakukan lewat aplikasi berbasis *android* oleh *Admin*

1.4 Tujuan

Adapun tujuan dari pelaksanaan Kerja Praktik ini adalah terbentuknya suatu rancang bangun aplikasi penagihan piutang yang dapat membantu bagian *Debt Collector* dalam melakukan proses penagihan pada *customer*.

1.5 Manfaat

Beberapa manfaat yang dapat diperoleh dari kegiatan Kerja Praktik di PT. Telkom Ketintang antara lain:

Bagi Penulis:

Mendapatkan pengalaman bekerja dalam bidang teknologi informasi.

Bagi PT. Telkom:

Mempermudah proses penagihan piutang pada *customer*

Memberi kemudahan dalam mengatur kegiatan penagihan

1.6 Sistematika Penulisan

Sistematika penulisan laporan Kerja Praktik adalah sebagai berikut:

BAB I: PENDAHULUAN

Bab ini membahas tentang latar belakang masalah, perumusan masalah, Batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II: GAMBARAN UMUM PERUSAHAAN

Pada bab ini membahas tentang sejarah perusahaan, lokasi perusahaan, visi dan misi perusahaan, dan struktur organisasi perusahaan.

BAB III: LANDASAN TEORI

Pada bab ini menjelaskan teori-teori dan literature yang berkaitan dan mendukung penyelesaian laporan Kerja Praktik, yaitu berisi penjelasan tentang piutang, utang, penagihan, bahasa pemrograman, PHP, *Database*, MySql, Firebase, JavaScript, Html.

BAB IV: DEKSRIPSI PEKERJAAN

Berisi penjelasan mengenai perancangan sistem yang akan digunakan, implementasi dari rancangan sistem yang telah dibuat, dan evaluasi terhadap sistem yang telah diuji coba.

BAB V: PENUTUP

Pada bab ini dibahas mengenai kesimpulan dari perancangan dan pembuatan aplikasi penagihan piutang terkait dengan tujuan dan permasalahan yang ada, serta saran untuk pengembangan aplikasi di masa mendatang.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Profile PT. Telekomunikasi Indonesia (Persero)

PT. Telekomunikasi Indonesia (Persero) Tbk (Telkom) adalah Badan Usaha Milik Negara (BUMN) yang bergerak di bidang jasa layanan teknologi informasi dan komunikasi (TIK) serta penyedia jasa jaringan telekomunikasi di Indonesia. Pemegang saham mayoritas adalah Pemerintah sebesar 52,09% sedangkan 47,91% sisanya dikuasai publik. Dalam pembagian divisi regional pada PT. Telkom Indonesia terdapat tujuh wilayah, sebagai berikut:

- 
1. Divisi Regional I Sumatera
 2. Divisi Regional II Jakarta dan sekitarnya
 3. Divisi Regional III Jawa Barat
 4. Divisi Regional IV Jawa Tengah dan Daerah Istimewa Yogyakarta
 5. Divisi Regional V Jawa Timur, Bali, dan Nusa Tenggara
 6. Divisi Regional VI Kalimantan
 7. Divisi Regional VII kawasan Timur yang meliputi: Sulawesi , Timor-Timur, Maluku dan Irian Jaya.

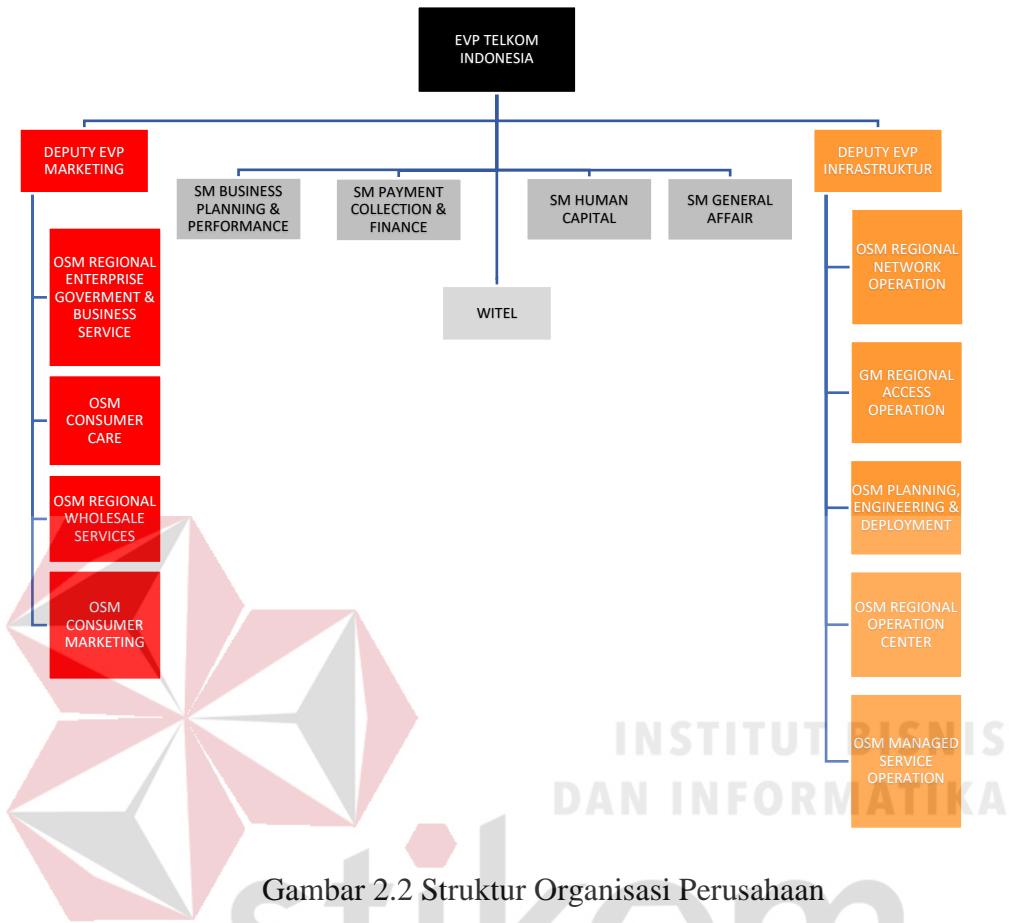
Kegiatan usaha TelkomGroup bertumbuh dan berubah seiring dengan perkembangan teknologi, informasi dan digitalisasi, namun masih dalam koridor industri telekomunikasi dan informasi. Hal ini terlihat dari lini bisnis yang terus berkembang melengkapi *legacy* yang sudah ada sebelumnya.

Dalam upaya bertransformasi menjadi *digital telecommunication company*, TelkomGroup mengimplementasikan strategi bisnis dan operasional perusahaan yang berorientasi kepada pelanggan (*customer-oriented*). Transformasi tersebut akan membuat organisasi TelkomGroup menjadi lebih *lean* (ramping) dan *agile* (lincah) dalam beradaptasi dengan perubahan industri telekomunikasi yang berlangsung sangat cepat. Organisasi yang baru juga diharapkan dapat meningkatkan efisiensi dan efektivitas dalam menciptakan *customer experience* yang berkualitas. Saat ini TelkomGroup mengelola 6 produk portofolio yang melayani empat segmen konsumen, yaitu korporat, perumahan, perorangan dan segmen konsumen lainnya



Gambar 2.1 Logo PT. Telekomunikasi Indonesia (Persero)

2.5 Struktur Organisasi



Gambar 2.2 Struktur Organisasi Perusahaan

2.6 Lokasi Perusahaan

Jl. Raya Ketintang No.156, Ketintang, Gayungan, Kota SBY, Jawa Timur.



Gambar 2.3 Lokasi Perusahaan

BAB III

LANDASAN TEORI

Untuk mendukung pembuatan laporan ini, maka perlu dikemukakan hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan ini.

3.1 Piutang

Piutang adalah aktiva atau kekayaan perusahaan yang timbul sebagai akibat dari dilaksanakannya kebijakan penjualan kredit, piutang adalah sumber dana operasi perusahaan jika piutangnya tak tertagih akan menimbulkan dana operasi perusahaan tidak berjalan dengan baik.

3.2 Android

Menurut (Susanto, 2011), Android merupakan *Operating System (OS) Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka.

Menurut (Safaat, 2014), Android adalah *system operasi* untuk telepon seluler yang berbasis Linux. Android menyediakan platform yang bersifat open source bagi para pengembang untuk menciptakan sebuah *aplikasi*. Awalnya, Google Inc. Mengakuisi Android Inc. yang mengembangkan *software* untuk 8

ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, yaitu konsorsium dari 34 perusahaan *hardware*, *software*, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

3.3 Android Studio

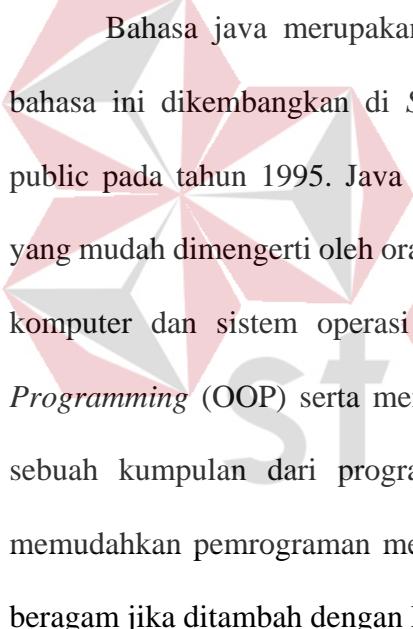
Menurut (Studio, 2018), Android Studio adalah Lingkungan Pengembangan Terpadu – *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdata guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas anda saat membuat aplikasi Android, misalnya:

- a. Sistem versi berbasis *Gradle* yang fleksibel
- b. Emulator yang cepat dan kaya fitur
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- d. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- f. Alat pengujian dan kerangka kerja yang ekstensif

- g. Alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.

3.4 Java

Menurut (Kadir, 2014), Java adalah nama sebuah bahasa pemrograman yang sangat terkenal dan dapat digunakan untuk menulis program. Program merupakan kumpulan intruksi yang ditujukan untuk *computer* agar dapat melaksanakan tugas sesuai dengan perintah *programmer* atau orang yang membuat program.



Bahasa java merupakan Bahasa pemrograman yang berorientasi objek, bahasa ini dikembangkan di *Sun Microsystems* dan mulai diperkenalkan pada public pada tahun 1995. Java bersifat universal menggunakan perintah-perintah yang mudah dimengerti oleh orang. Java dapat dijalankan dalam beberapa platform komputer dan sistem operasi yang berbeda. Java termasuk *Object Oriented Programming* (OOP) serta memiliki *library* yang lengkap, Librari disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas Java.

3.5 Database

Menurut (Connolly & Begg, 2010), basis data adalah suatu kumpulan *logical data* yang berhubungan dan deskripsi dari data tersebut yang dirancang untuk kebutuhan informasi suatu organisasi. Istilah lain basis data adalah kumpulan *file* yang saling terkait satu sama lain. Basis data tidak hanya merupakan kumpulan

file, record pada setiap *file* harus memperbolehkan hubungan-hubungan untuk menyimpan *file* lain.

Database adalah suatu susunan atau kumpulan data operasional lengkap dari suatu organisasi atau perusahaan yang terorganisir dan disimpan secara bersamaan dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya. Penyusunan satu *database* digunakan untuk mengatasi masalah – masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, *multiple user, security*, dan *integrase*.

3.6 *Firebase*

Firebase adalah suatu *database server* disediakan oleh google secara gratis untuk pengembangan aplikasi *mobile* atau *web application* yang terilis pada tahun 2012. Layanan yang disediakan meliputi *Analytics, Cloud Messaging, Authentication, Realtime Database, Storage, Hosting, Config, Test Lab, Crash Reporting, Notifications, App Indexing, Dynamic Links, Invites*.

BAB IV

DESKRIPSI PEKERJAAN

4.1 Identifikasi Masalah

PT. Telekomunikasi Indonesia Divisi Regional V (Telkom) memiliki prosedur sistem yang masih manual. Manual disini diartikan bahwa belum adanya sistem yang dapat membantu Telkom sendiri dalam mengelola data-data *customer* yang memiliki keterlambatan dalam pembayaran biaya layanan internet yang telah disediakan. Permasalahan yang terjadi ketika melakukan kegiatan *Debt Collecting* adalah tidak seringnya keterlambatan dalam melakukan penyetoran pembayaran yang telah ditagih oleh *Debt Collector*.

Untuk mengatasi permasalahan di atas, dilakukannya beberapa kegiatan dalam pembuatan aplikasi sebagai berikut:

- a. Menganalisis sistem
- b. Merancang sistem
- c. Mengimplementasikan sistem
- d. Melakukan pembahasan terhadap implementasi sistem.

4.2 Analisis Bisnis

Tahapan ini adalah tahapan dimana dilakukannya proses wawancara dan *survey* terkait proses yang berjalan saat ini dari perusahaan terkait bagaimana proses dari pada sistem penagihan PT. Telkom. Pihak divisi akuntansi akan melakukan proses pengelompokan konsumen yang tidak melakukan pembayaran dalam waktunya. Setelah terkumpulnya data data konsumen tersebut diberikan kepada pihak *Debt Collecting*. Kemudian dilakukannya proses penagihan, dimana penagih

mendatangi rumah konsumen untuk diadakannya negosiasi atau bagaimana proses utang tersebut diselesaikan. Jika piutang sudah diterima, *Collector* akan memberikan dana tersebut pada pihak akuntansi untuk dilakukannya proses konfirmasi dana dari customer mana yang telah melunasi utangnya.

4.3 Analisis Sistem

Pengumpulan data-data untuk penyelesaian kerja praktik ini diperlukannya kegiatan memperoleh data, merancang dan membangun program. Tahapan analisis *system* adalah dengan melakukan observasi, wawancara, analisis proses bisnis, analisis kebutuhan pengguna, analisis kebutuhan fungsional dan non-fungsional

4.3.1 Observasi

Observasi dalam penelitian ini adalah melihat dan mengamati secara langsung bagaimana kegiatan bisnis terjadi pada PT. Telkom kemudian mencatatnya untuk dilakukannya kegiatan analisis.

4.3.2 Wawancara

Kegiatan ini dilakukan untuk memperoleh informasi lebih jelas tentang kegiatan pihak PT. Telkom untuk aplikasi yang ingin dibangun

4.3.3 Analisis proses bisnis

Analisis proses bisnis dilakukan dengan tujuan untuk mengetahui bagaimana alur dari proses bisnis yang akan diterapkan dalam aplikasi yang akan dibuat. Berdasarkan hasil dari wawancara dan observasi proses bisnis, alur dari proses bisnis yaitu:

1. Admin memasukkan data *customer* dan biaya yang diperlukan
2. *Collector* mengetahui data tersebut kemudian melakukan proses penagihan

3. *Collector* menerima uang tersebut kemudian menyetorkannya kepada Admin.
4. Admin mengkonfirmasi uang tersebut

4.3.4 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan dengan tujuan untuk mengetahui data dan informasi yang digunakan ataupun dibutuhkan oleh pengguna aplikasi yang akan dibuat. Selain itu juga untuk menganalisis *output* yang diperoleh dari pengguna tersebut. Berdasarkan hasil analisis proses bisnis dari aplikasi yang akan dibuat adalah sebagai berikut:

1. Sistem dapat menampilkan rincian data *customer* yang memiliki utang
2. Sistem dapat menghubungi pelanggan
3. Sistem dapat menampilkan data *customer*

4.3.5 Kebutuhan Fungsional

Berdasarkan hasil: observasi, wawancara, analisis proses bisnis, dan analisis kebutuhan pengguna maka dapat dilakukan identifikasi kebutuhan fungsional untuk aplikasi yang akan dibuat, yaitu:

1. Sistem dapat menambahkan data *customer* yang memiliki utang
2. Sistem dapat menampilkan data *customer*
3. Sistem dapat menampilkan data *collector*
4. Sistem dapat menambahkan atau mengubah data *collector*
5. Sistem dapat mengkonfirmasi pembayaran yang telah disetorkan
6. Sistem dapat menampilkan data yang telah dikonfirmasi
7. Aplikasi dapat mengakses *database* yang disediakan

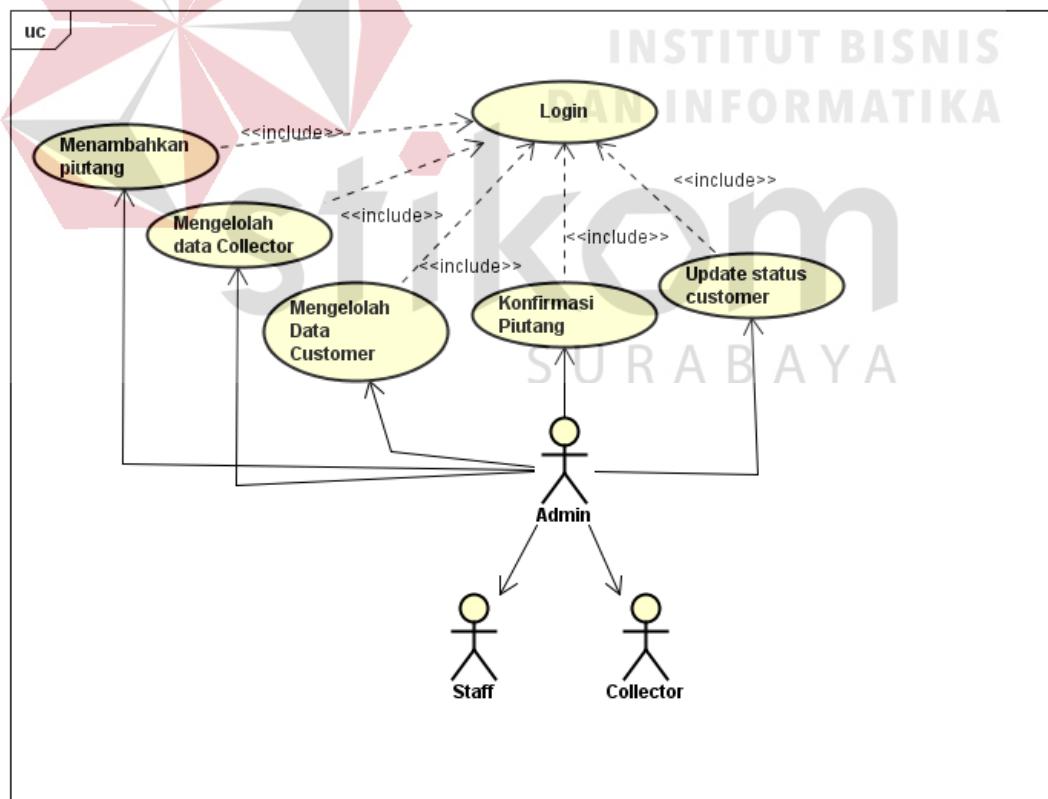
4.3.6 Kebutuhan Non-Fungsional

Berdasarkan wawancara dapat dilakukan identifikasi bahwa kebutuhan non-fungsional untuk aplikasi yang akan dibuat, yaitu:

1. Keamanan: Sistem aplikasi dan *database* dilengkapi *password*
2. Keefektifan: Sistem ini bertujuan untuk mempermudah penagih atau *collector* dalam proses penagihan

4.4 Perancangan Sistem

Berikut merupakan *use case* aplikasi *Debt Collecting*. Pada *use case* ini terdapat 2 *actor* yaitu *Staff* dan *Collector* di generelasikan menjadi sebuah *Admin*. Terdapat 6 proses bisnis yang terdapat pada *use case* tersebut.



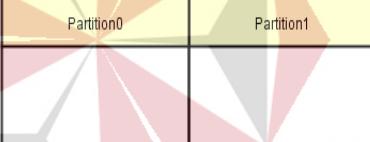
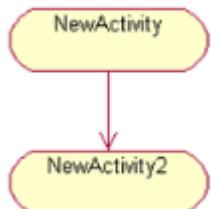
Gambar 4.1 *Use case Diagram* aplikasi

4.4.1 Activity Diagram

Diagram aktivitas (*Activity Diagram*) adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari use case bisnis dalam bentuk grafik. Diagram aktivitas ini menunjukkan langkah-langkah dalam aliran kerja, titik keputusan dalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja.

Berikut adalah simbol-simbol yang terdapat pada aktivitas diagram:

Tabel 4.1 Notasi *Activity Diagram*

| Simbol | Keterangan |
|---|--|
|  | <i>Swinlane</i> : Peranan yang melakukan aktivitas |
|  | Keadaan awal: Menunjukkan dimana aliran kerja dimulai |
|  | Aktivitas: Kegiatan dalam aliran kerja Transisi: Bagaimana aliran kerja berjalan dari satu aktivitas ke aktivitas lainnya |
|  | <i>Decision</i> : Menunjukkan dimana keputusan harus dibuat. |

| Simbol | Keterangan |
|--------|--|
| | Sinkronisasi: Beberapa aktivitas berjalan secara bersamaan |
| | Keadaan akhir: Menunjukkan dimana aliran kerja diakhiri |

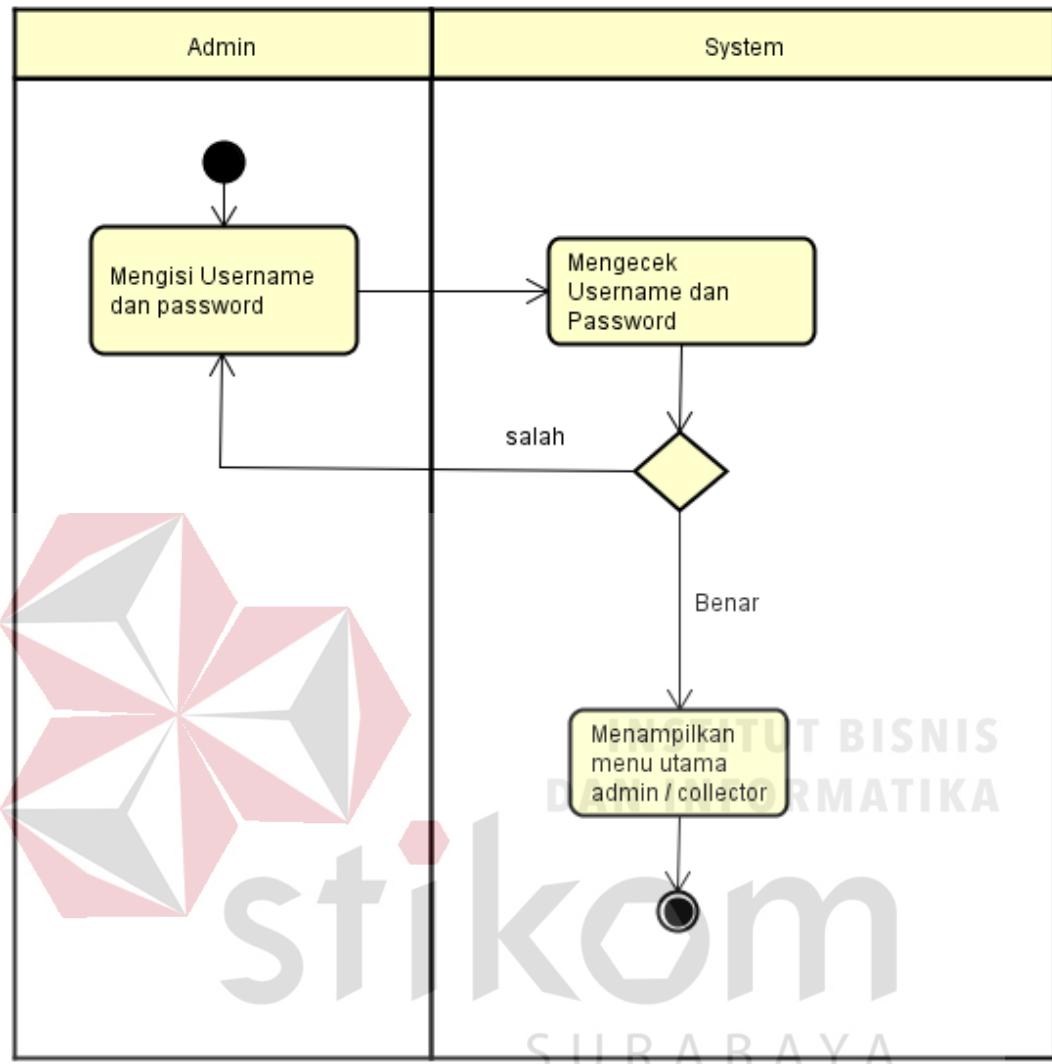
Terdapat 6 tabel *Activity Diagram* dalam pembuatan aplikasi ini yaitu:

1. *Login*
2. *Mengelola Master Customer*
3. *Mengelola Master Collector*
4. *Menambahkan Piutang*
5. *Update Status Customer*
6. *Konfirmasi Piutang.*

Berikut adalah gambar dari *Activity Diagram* untuk aplikasi *Debt Collecting*

Berbasis *Android* pada PT. Telkom Regional V:

A. Login

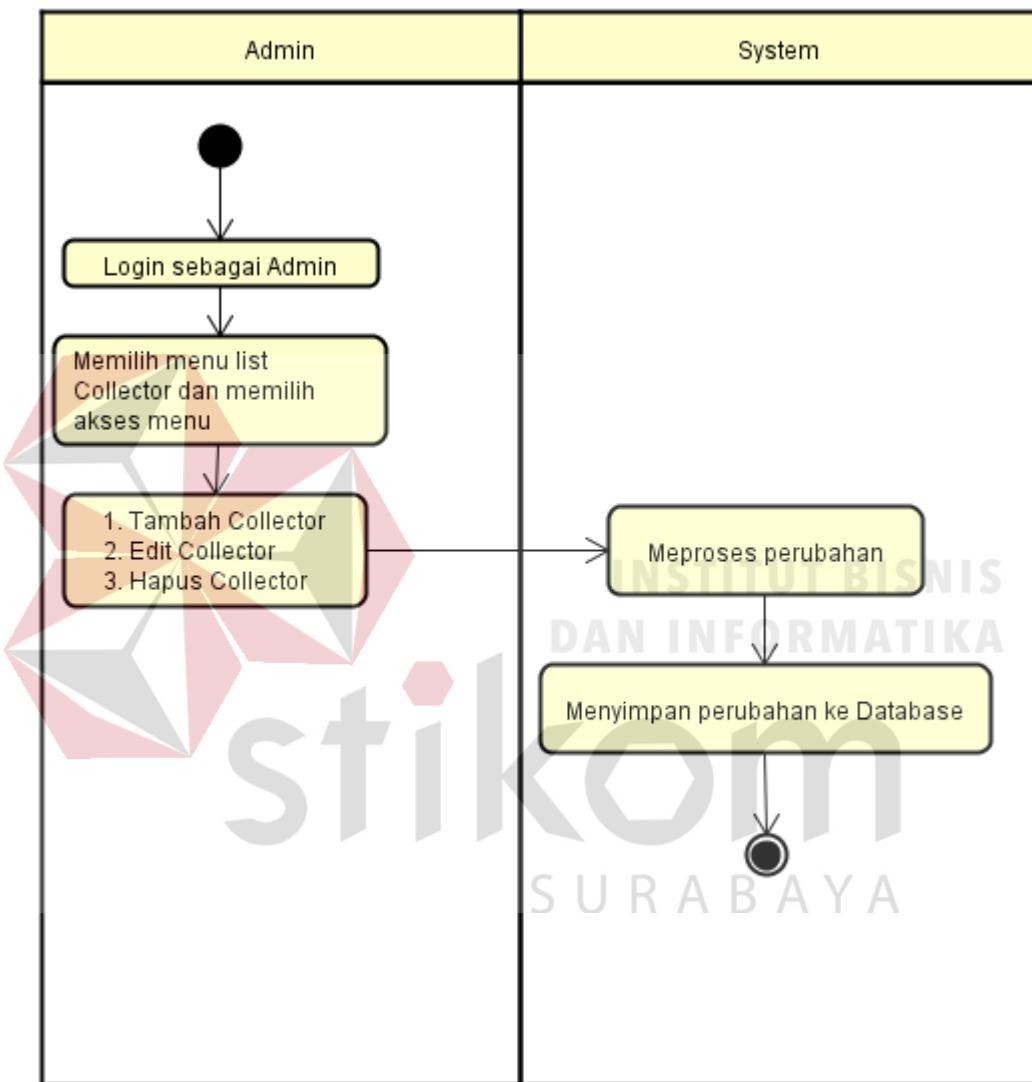


Gambar 4.2 Activity Diagram Login

Gambar 4.2 menggambarkan proses aktivitas login pada aplikasi *Debt Collecting*. Gambar tersebut menjelaskan bahwa terdapat validasi *username* dan *password*. Sistem akan mengecek apakah *username* tersebut termasuk milik admin atau *collector*. Jika admin, sistem akan menampilkan menu utama untuk admin. Begitupun juga untuk *collect*, jika *Collector* maka *system* akan menampilkan

halaman utama untuk *collector*. Sedangkan jika keduanya salah akan kembali lagi ke halaman *login* untuk menginputkan ulang *username* dan *password*.

B. Mengelolah data *Collector*

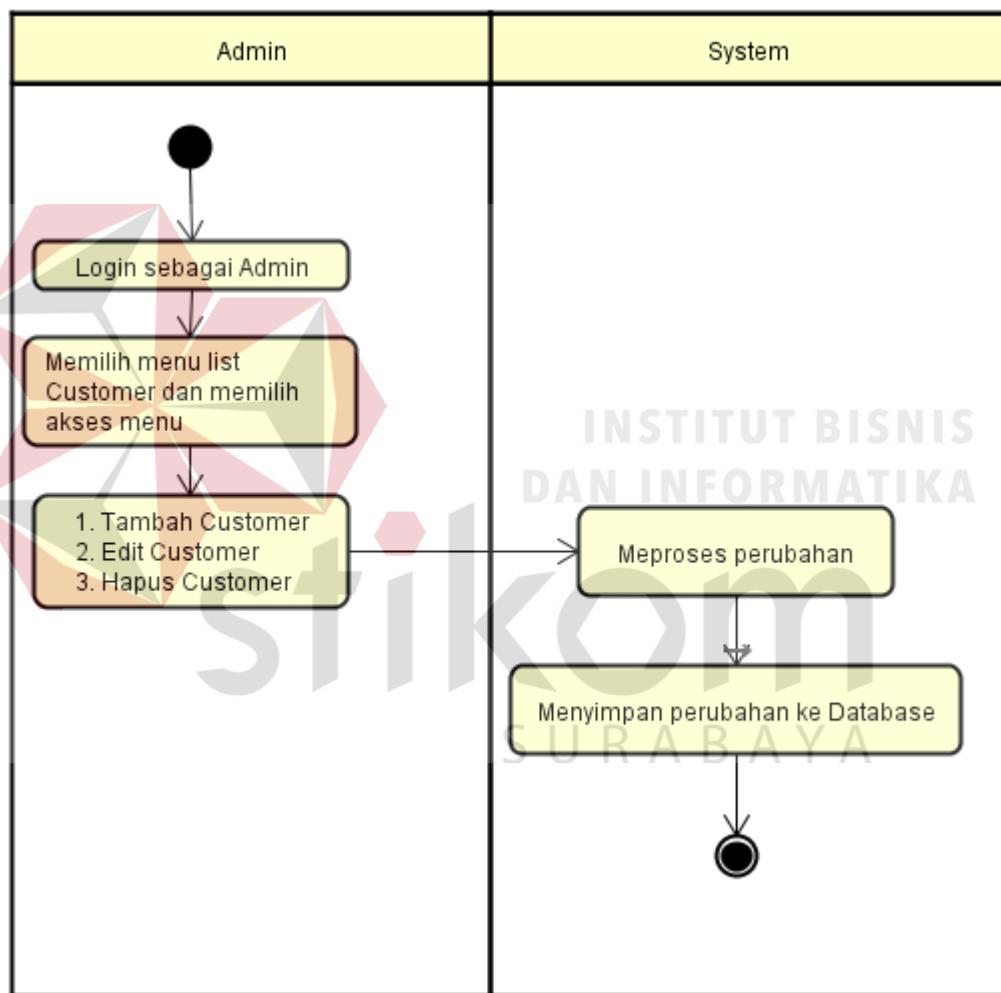


Gambar 4.3 *Activity Diagram* Mengelolah data *Collector*

Gambar 4.3 menjelaskan bahwa jika ingin mengubah, menambah, atau menghapus harus melalui dari *login* admin terlebih dahulu. Kemudian dilanjutkan dengan memilih menu yang diinginkan, jika ingin mengubah cukup memilih dari *list* tersebut kemudian menginputkan data yang ingin diubah kemudian tekan

tombol “Simpah perubahan”. Jika ingin menghapus memilih list data *collector* yang ingin dihapus kemudian tekan tombol hapus, akan muncul notifikasi kemudian menekan tombol “OK”. Sedangkan jika ingin menambah cukup tekan tombol tambah kemudian menginputkan data *collector* baru yang ingin ditambahkan.

C. Mengelolah data *Customer*

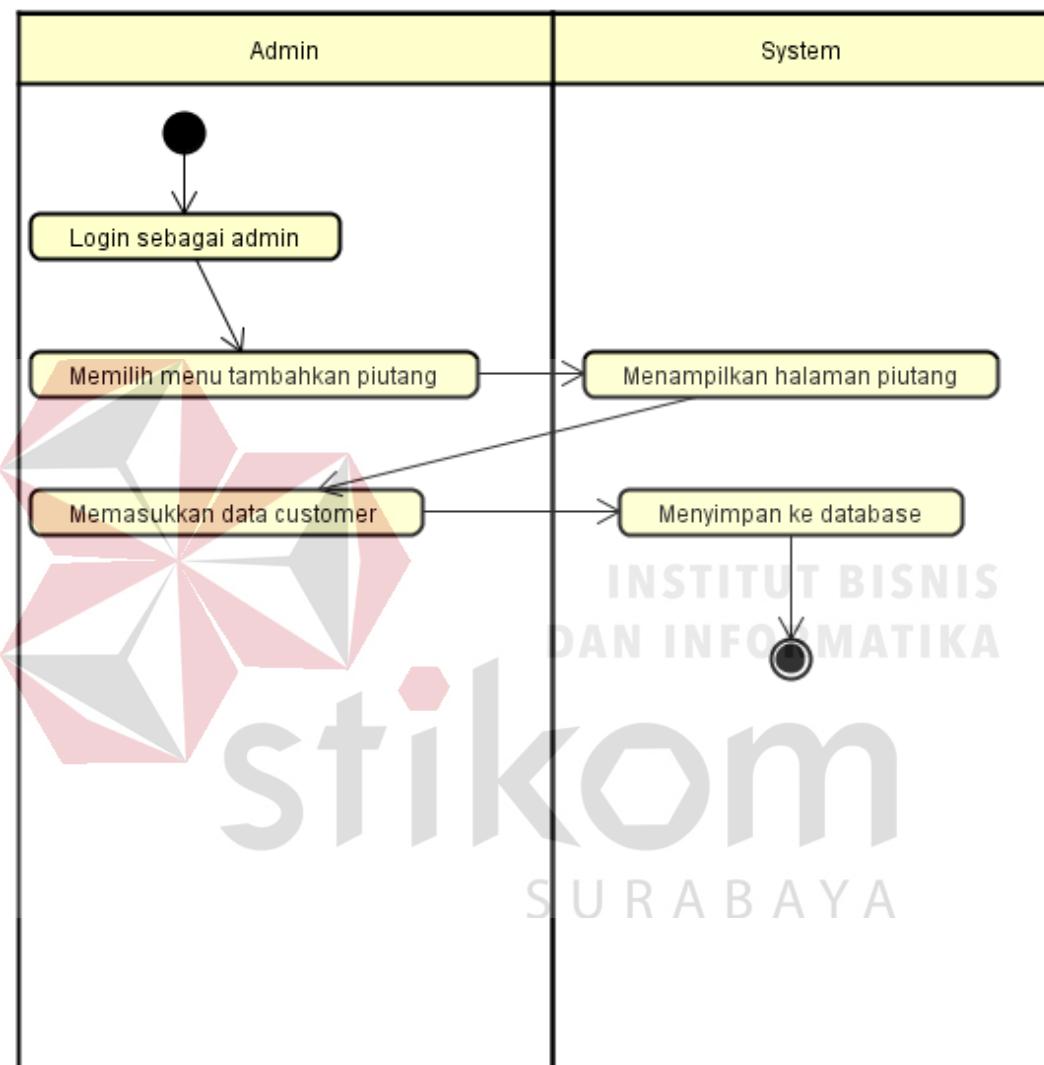


Gambar 4.4 *Activity Diagram Mengelolah data Customer*

Activity Diagram yang terdapat pada Gambar 4. 4 adalah proses untuk mengelolah data *Customer*. Untuk prosesnya hampir sama seperti mengelolah data *Collector*, hanya saja setelah melakukan *login* admin menu yang dipilih adalah

menu *Customer*. Disini admin dapat melakukan pengubahan, penambahan, ataupun menghapus data *Customer* tersebut.

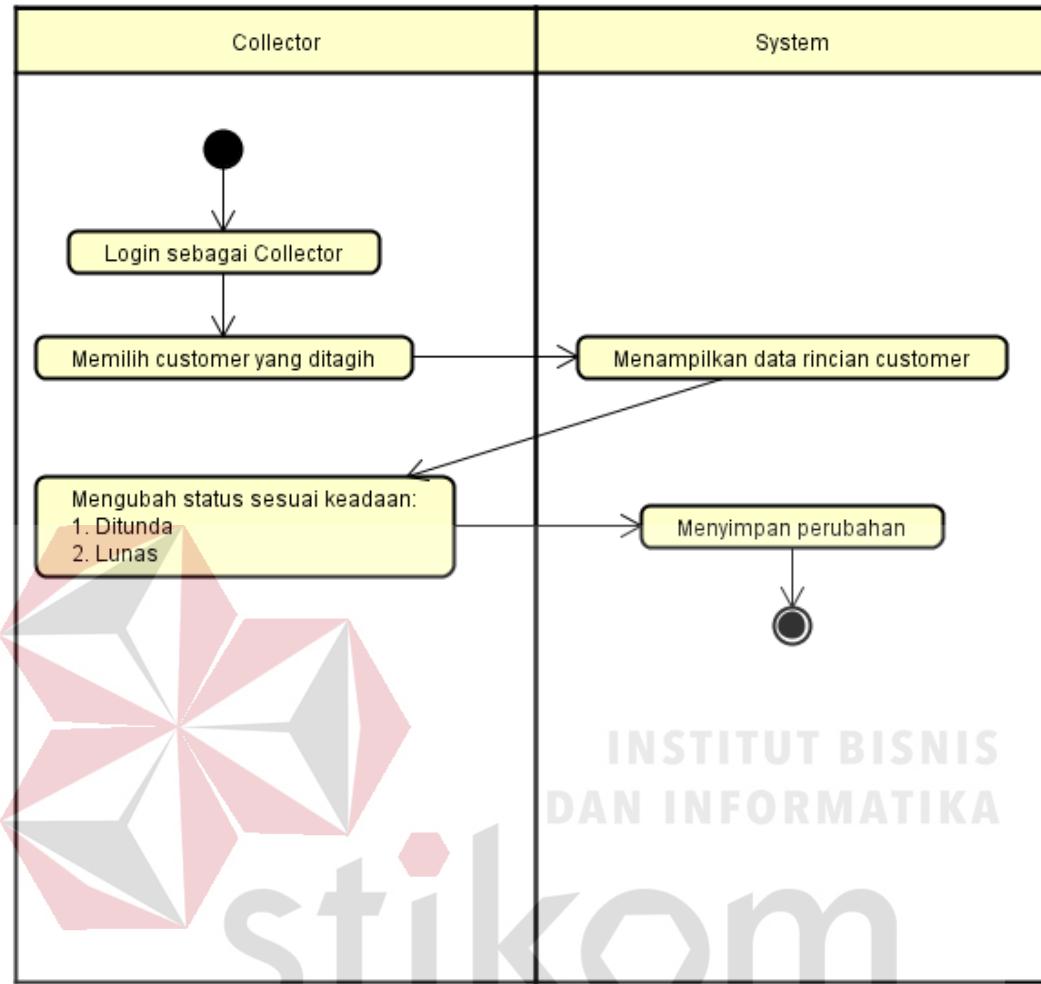
D. Menambahkan piutang



Gambar 4.5 *Activity Diagram* Menambahkan Piutang

Pada Gambar 4.5, Admin melakukan proses *login* kemudian memilih menu tambahkan piutang. Sistem kemudian akan menampilkan halaman tersebut. Pada halaman ini admin memasukkan data *customer* yang memiliki utang, tanggal, dana yang diperlukan untuk pelunasan. Setelah itu system menyimpan data tersebut.

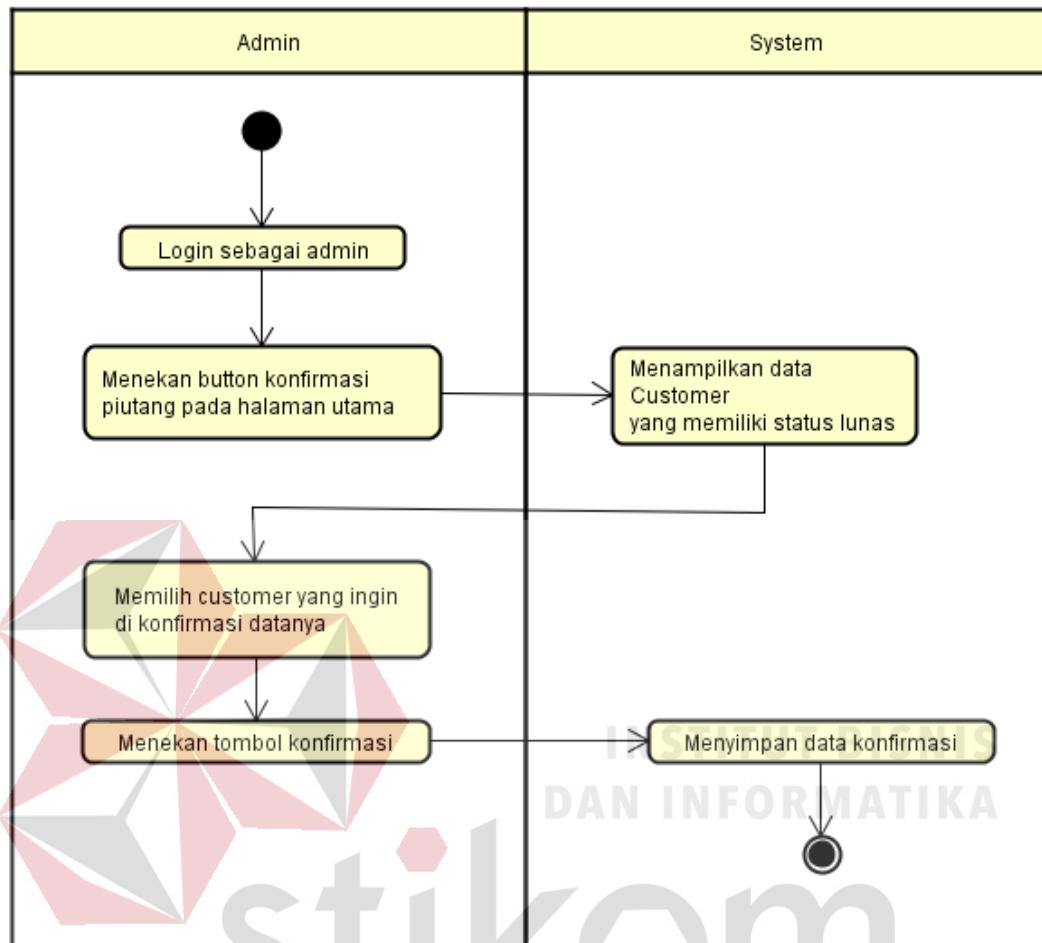
E. Update Status Customer



Gambar 4.6 Activity Diagram Update Status Customer

Pada *Activity* diagram di atas, *Collector* melakukan *login* kemudian menekan *customer* yang ditagih. Sistem akan menampilkan data rincian *customer* kemudian *collector* bisa mengubah status *customer* tersebut sesuai keadaan. Bisa saja ditunda ataupun lunas. *System* kemudian menyimpan perubahan tersebut.

F. Konfirmasi Piutang



Gambar 4.7 Activity Diagram Konfirmasi Piutang

Pada *Activity* diagram di atas, admin melakukan *login* kemudian pada halaman utama memilih menu konfirmasi piutang. Setelah itu sistem akan menampilkan data *customer* yang memiliki status lunas. Admin dapat memilih *customer* yang datanya ingin dikonfirmasi kemudian menekan tombol konfirmasi yang tersedia. Setelah itu data konfirmasi akan disimpan pada *database* yang tersedia.

4.4.2 Flow Of Document

Flow of Document disini adalah sebuah tabel yang akan menjelaskan tentang alur untuk diterapkan pada *Sequence Diagram* yang akan dibuat. Terdiri dari nama, deskripsi, prasyarat, *actor*, kondisi akhir, dan alur. Alur disini dibagi menjadi 3, yaitu normal, *alternative*, dan *error*. Normal adalah alur utama untuk *system* yang akan dibuat, *Alternative* adalah alur yang bisa dilakukan oleh *actor* ataupun *system* ketika alur normal tidak bisa dijalankan ataupun tidak sesuai. Sedangkan alur *Error* akan dijalankan ketika kedua alur tersebut tidak bisa dijalankan ataupun terdapat kesalahan pada *system*.



Berikut adalah *Flow of Document* untuk aplikasi *Debt Collecting* Berbasis *Android* pada PT. Telkom Regional V:

A. *Login*

Tabel 4.2 *Flow of Document Login*

| Nama | Login |
|-----------|--|
| Deksripsi | Staff Admin atau <i>Collector</i> melakukan proses <i>login</i> agar bisa mengakses fitur aplikasi |

| Nama | Login | |
|---------------|--|---|
| Prasyarat | Memiliki akun yang sudah terdaftar pada <i>database</i> aplikasi | |
| Aktor | Admin | |
| Kondisi Akhir | Aplikasi menampilkan halaman utama sesuai user yang melakukan login | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | <p>1. Admin membuka aplikasi</p> <p>3. Admin memasukkan data untuk login berupa <i>Username & Password</i></p> | <p>2. Sistem menampilkan halaman <i>login</i></p> <p>4. Sistem melakukan pengecekan pada <i>database</i> apakah <i>username</i> dan <i>password</i> tersedia.</p> |
| Alternatif | <i>Username</i> tidak terdapat pada <i>database</i> maka menampilkan text “Gagal Login” | |
| Error | Sistem tidak bisa melakukan pengecekan <i>database</i> Firebase karena koneksi gagal | |

B. Mengelola *Master Customer*

Tabel 4.3 *Flow of Document* Mengelola *Master Customer*

| Nama | Mengelola <i>Master Customer</i> |
|-----------|--|
| Deksripsi | <i>Staff</i> admin melakukan proses tambah, <i>delete</i> , maupun <i>edit</i> pada proses mengelola data customer |
| Prasyarat | Sudah melakukan Login sebagai Admin dan memiliki hak akses |

| Nama | Mengelola Master <i>Customer</i> | |
|---------------|--|--|
| Aktor | <i>Staff</i> (Admin) | |
| Kondisi Akhir | Data master berhasil ditambahkan, dihapus, ataupun diubah dan tersimpan pada <i>database</i> Firebase | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | <p>1. Admin memilih menu <i>customer</i></p> <p>4. Admin dapat memilih untuk menambahkan <i>customer</i>, mengedit <i>customer</i>, atau menghapus <i>customer</i>. Untuk menambahkan <i>customer</i> tekan tombol “tambah”. Untuk mengedit <i>customer</i> mengeklik <i>customer</i> tersebut kemudian tekan tombol “edit”. Untuk menghapus <i>customer</i>, tekan tombol <i>delete</i> di bawah data tersebut.</p> | <p>2. Sistem menampilkan halaman <i>customer</i></p> <p>3. Sistem memanggil dan menampilkan data-data <i>customer</i> pada <i>database</i> Firebase di halaman <i>customer</i></p> |
| | | <p>5. Jika admin menekan tombol tambah akan memunculkan halaman berisi <i>Edit Text</i> dan tombol simpan.</p> |
| | | <p>6. Setelah tombol sesuai proses di tekan <i>system</i> akan mengirim data tersebut ke <i>database</i> Firebase untuk mengubah, menambah, ataupun menghapusnya.</p> |
| Alur | Aksi Aktor | Reaksi Sistem |
| Alternatif | | |
| Error | Sistem tidak bisa melakukan penyimpanan proses ataupun mengubah karena koneksi gagal | |

C. Mengelola Master Collector

Tabel 4.4 *Flow of Document Mengelola Master Collector*

| Nama | Mengelola Master Collector | |
|---------------|--|--|
| Deksripsi | <i>Staff</i> admin melakukan proses tambah, <i>delete</i> , maupun <i>edit</i> pada proses mengelola data <i>collector</i> | |
| Prasyarat | Sudah melakukan <i>Login</i> sebagai Admin dan memiliki hak akses | |
| Aktor | <i>Staff</i> (Admin) | |
| Kondisi Akhir | Data master berhasil ditambahkan, dihapus, ataupun diubah dan tersimpan pada <i>database</i> Firebase | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | 1. Admin memilih menu <i>collector</i> Untuk menambahkan <i>collector</i> tekan tombol “tambah”. Untuk mengedit <i>collector</i> mengeklik <i>collector</i> tersebut kemudian tekan tombol “edit”. Untuk menghapus <i>collector</i> , | 2. Sistem menampilkan halaman <i>collector</i> 3. Sistem memanggil dan menampilkan data-data <i>collector</i> pada <i>database</i> Firebase di halaman <i>collector</i> |

| | | |
|------------|--|---|
| Nama | Mengelola Master <i>Collector</i> | |
| | tekan tombol delete di bawah data tersebut | |
| | | 5. Jika admin menekan tombol tambah akan memunculkan halaman berisi <i>Edit Text</i> dan tombol simpan. |
| | 6. Setelah tombol sesuai proses di tekan <i>system</i> akan mengirim data tersebut ke <i>database</i> Firebase untuk mengubah, menambah, ataupun menghapusnya. | |
| Alternatif | | |
| Error | Sistem tidak bisa melakukan penyimpanan proses ataupun mengubah dikarenakan tidak adanya koneksi | |

D. Menambahkan Piutang

Tabel 4.5 *Flow of Document* Menambahkan Piutang

| | |
|-----------|---|
| Nama | Menambahkan Piutang |
| Deksripsi | Admin melakukan proses tambah data piutang yang berhubungan dengan data <i>customer</i> |
| Prasyarat | Sudah melakukan <i>Login</i> sebagai Admin dan memiliki hak akses |
| Aktor | <i>Staff</i> (Admin) |

| Nama | Menambahkan Piutang | |
|---------------|---|--|
| Kondisi Akhir | Data piutang berhasil ditambahkan dan dapat muncul pada halaman piutang | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | <p>1. Admin memilih menu piutang</p> <p>4. Admin dapat memilih untuk menambahkan piutang dengan memilih <i>customer</i> yang memiliki utang kemudian memasukkan dana tersebut. Untuk menu <i>edit</i> hanya bisa mengubah dana tersebut. Dan untuk menghapus hanya dengan menekan tombol hapus.</p> | <p>2. Sistem menampilkan halaman piutang</p> <p>3. Sistem memanggil dan menampilkan data-data <i>customer</i> dan data piutang pada <i>database Firebase</i> di halaman piutang</p> |
| | | <p>5. Jika menekan tombol tambah akan memunculkan list <i>customer</i>. Setelah tombol sesuai proses di tekan system akan mengirim data tersebut ke <i>database Firebase</i> untuk mengubah, menambah, ataupun menghapusnya.</p> |
| | | <p>6. Muncul notifikasi kemudian mengembalikan <i>user</i> kehalaman utama</p> |
| Alternatif | | |
| Error | Sistem tidak bisa melakukan pengecekan ataupun mengubah karena koneksi gagal | |

E. Update Status Customer

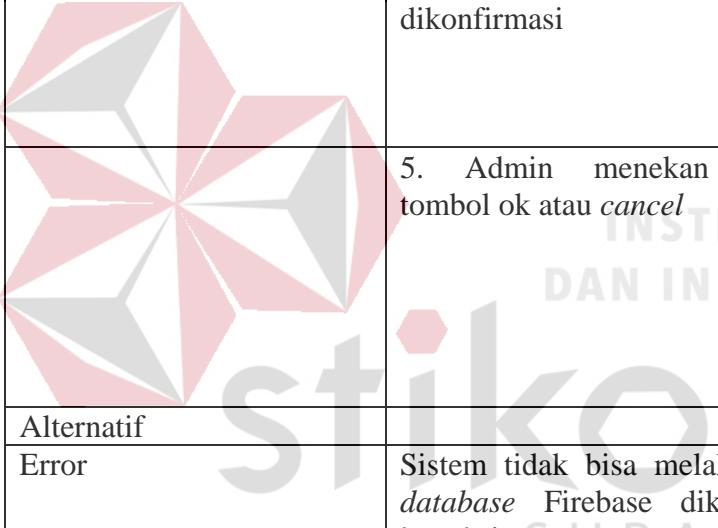
Tabel 4.6 Flow of Document Update Status Customer

| Nama | Update Status Customer | |
|---------------|--|--|
| Deksripsi | <i>Collector</i> melakukan perubahan status <i>customer</i> yang sesuai dengan keadaan status. | |
| Prasyarat | Sudah melakukan <i>Login</i> sebagai <i>collector</i> dan memiliki hak akses | |
| Aktor | <i>Collector</i> (Admin) | |
| Kondisi Akhir | Data master berhasil ditambahkan, dihapus, ataupun diubah | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | 1. <i>Collector</i> menekan <i>customer</i> yang ingin diubah statusnya 3. <i>Collector</i> menekan tombol “ubah” status. Untuk mengubah status terdapat pilihan : Lunas ataupun ditunda. <i>Collector</i> dapat memilih status sesuai keadaan yang terjadi | 2. Sistem menampilkan data rincian <i>customer</i> 4. Sistem menyimpan data yang telah diubah oleh <i>collector</i> |
| Alternatif | | |
| Error | Sistem tidak bisa melakukan pengecekan ataupun mengubah karena koneksi gagal | |

F. Konfirmasi Piutang

Tabel 4.7 Flow of Document Konfirmasi Piutang

| Nama | Konfirmasi Piutang |
|-----------|---|
| Deksripsi | <i>Staff</i> melakukan proses konfirmasi piutang <i>customer</i> yang telah memiliki status lunas |

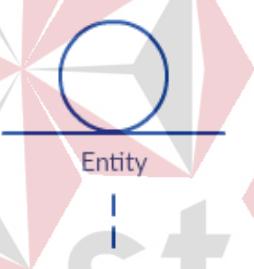
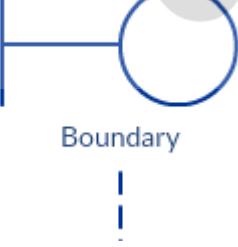
| Nama | Konfirmasi Piutang | |
|---------------|---|---|
| Prasyarat | Sudah melakukan <i>Login</i> sebagai <i>staff</i> dan punya hak akses | |
| Aktor | <i>Staff</i> (Admin) | |
| Kondisi Akhir | Data piutang berhasil tersimpan ke data piutang yang lunas. | |
| Alur | Aksi Aktor | Reaksi Sistem |
| Normal | 1. Admin memilih menu konfirmasi piutang  3. Admin menekan <i>customer</i> yang ingin dikonfirmasi | 2. Sistem menampilkan halaman konfirmasi piutang yang berisi data piutang yang memiliki status lunas 4. Sistem menampilkan dialog yang berisikan “Apakah dana sudah diterima dan siap dikonfirmasi.” |
| Alternatif | 5. Admin menekan tombol ok atau <i>cancel</i> | 6. Jika admin menekan tombol ok, data tersebut akan di simpan ke data lunas jika <i>cancel</i> akan kembali pada halaman konfirmasi piutang |
| Error | Sistem tidak bisa melakukan pemanggilan data <i>database</i> Firebase dikarenakan tidak adanya koneksi | |

4.4.3 Sequence Diagram

Model *Sequence Diagram* ini menggambarkan rancangan interaksi antar objek di dalam dan sekitar *system* (termasuk pengguna, *display*, dan sebagainya) berupa *message* atau pesan juga menggambarkan terhadap waktu, scenario atau langkah yang dilakukan sebagai respon dari *client* untuk menghasilkan *output*. *Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Terdapat beberapa notasi dalam *Sequence Diagram*. Notasi tersebut dapat dilihat pada tabel di bawah ini:

Tabel 4.8 Notasi *Sequence Diagram*

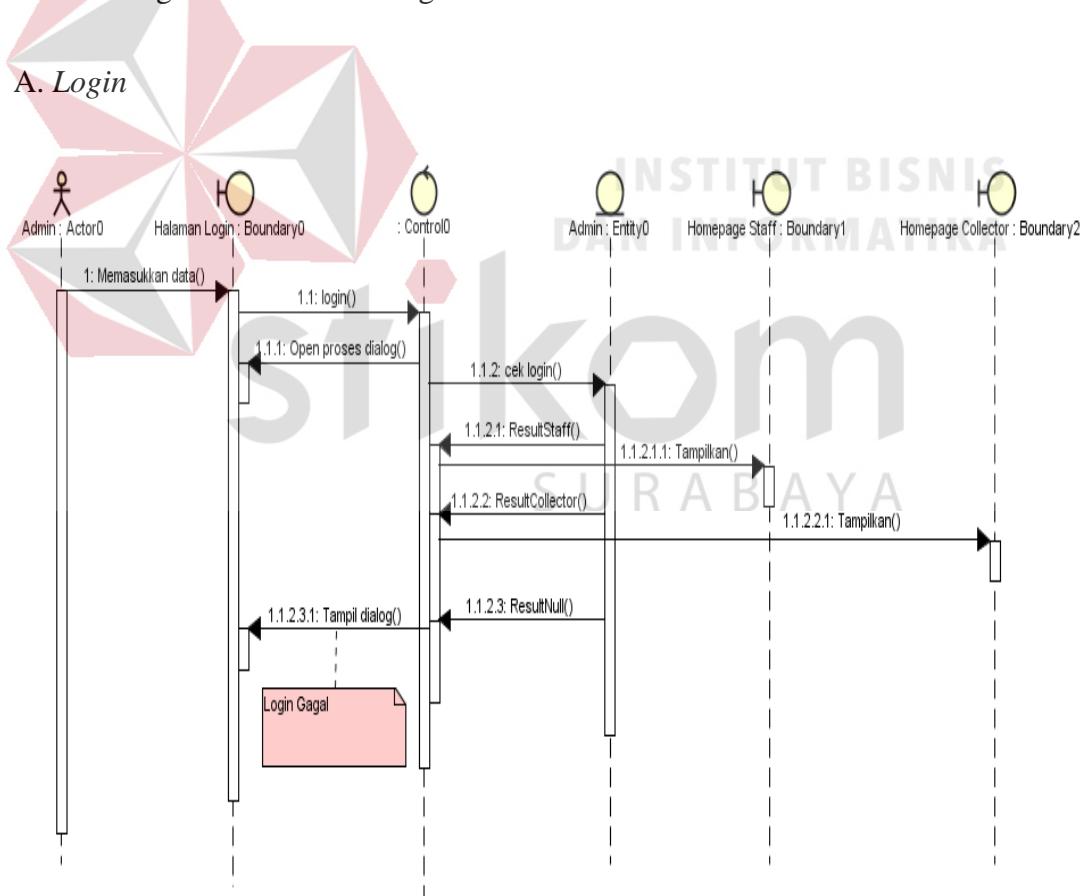
| Gambar | Penjelasan |
|---|--|
|  | Notasi dengan <i>actor element symbol</i> digunakan ketika sebuah <i>sequence diagram</i> dimiliki sebuah <i>usecase</i> . |
|  | Sebuah notasi yang memiliki <i>element entity</i> sebagai penyebutan <i>system data</i> . Contoh sebuah <i>Database Customer</i> . |
|  | <i>Boundary</i> menunjukkan sebuah perbatasan antara element <i>system</i> dengan tampilan. Contoh: <i>GUI</i> atau tampilan yang dapat diakses. |
|  | Notasi <i>control</i> mengatur sebuah <i>entity</i> . Notasi ini mengorganisir dan menjelaskan interaksi antara <i>boundary</i> dengan <i>entity</i> sebagai pihak penghubung. |

Terdapat 6 *Sequence Diagram* dalam pembuatan aplikasi ini yaitu:

1. *Login*
2. Mengelola *Master Customer*
3. Mengelola *Master Collector*
4. Menambahkan Piutang
5. *Update Status Customer*
6. Konfirmasi Piutang.

Sequence Diagram dari aplikasi *Debt Collecting* berbasis *Android* pada PT.

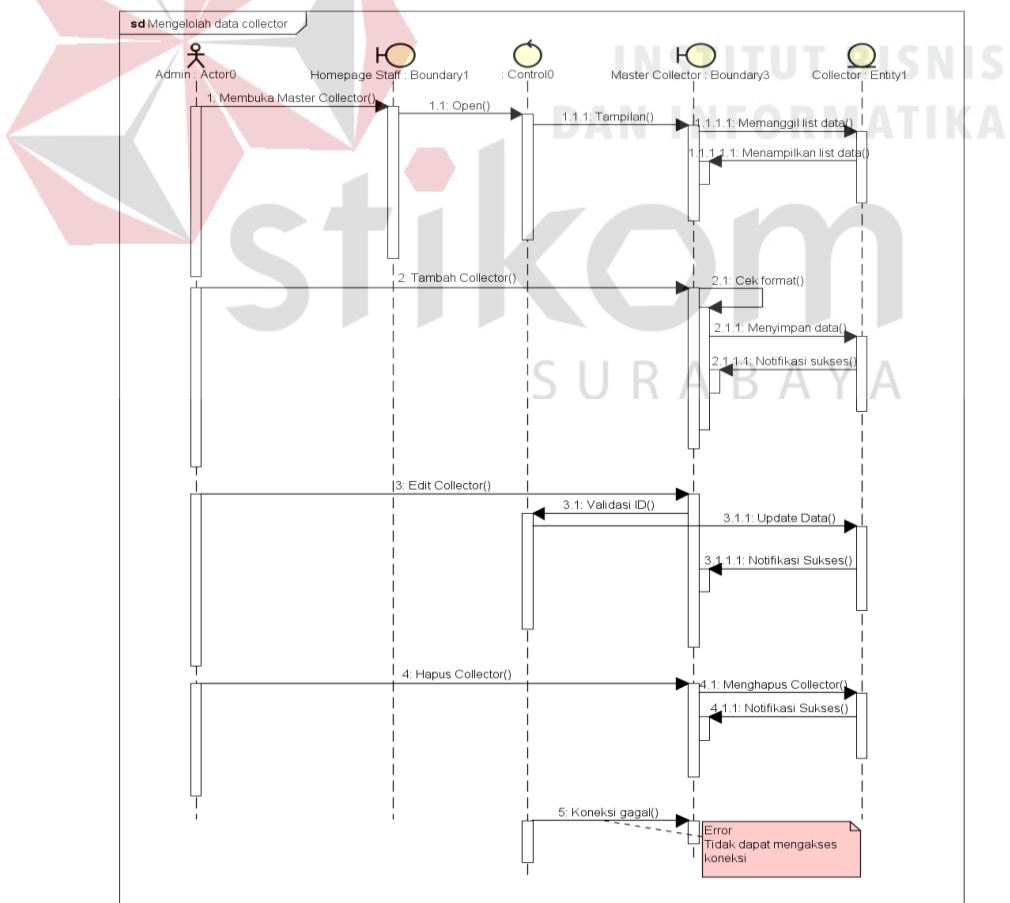
Telkom Regional V terdiri sebagai berikut:



Gambar 4.8 *Sequence Diagram Login*

Pada *Sequence Diagram login*, dapat dilihat *actor* berupa admin memasukkan data berupa *username* dan *password* pada Halaman *Login* (*boundary*). Kemudian ketika ditekannya tombol *login*, *boundary* tersebut meminta kepada *control* untuk melakukan proses *login* dan memanggil *open dialog* yaitu *loading*. *Control* melakukan validasi ke *database* dengan fungsi cek *login()*. Jika hasil admin maka *tampilkan()* disini akan menampilkan *homepage* admin, jika *tampilkan()* *value collector* maka akan menampilkan *homepage collector*. Jika tidak ada keduanya maka *value* yang dikembalikan adalah *null* atau *error*. Kemudian *control* menampilkan dialog berisi “*Login Gagal*”.

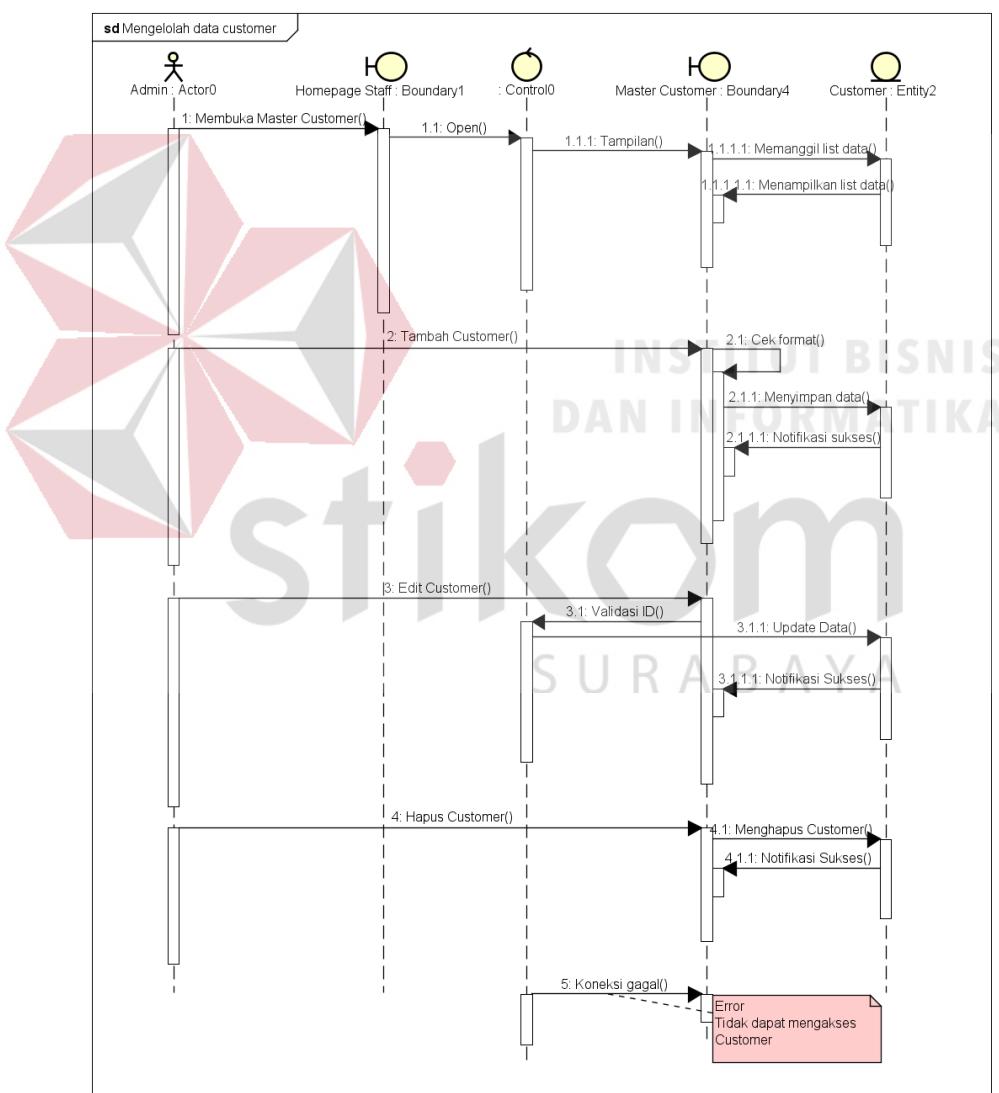
B. Mengelolah data *Collector*



Gambar 4.9 *Sequence Diagram Mengelolah data Collector*

Sequence Diagram ini dimulai dari Admin (*Actor*) mengakses halaman *master collector* kemudian *system control* akan melakukan fungsi *tampilan()* yang memanggil *entity collector* dan menampilkan data-datanya. Kemudian langkah selanjutnya bebas dilakukan oleh *actor*, entah itu *add*, *edit*, maupun *delete*. Fungsi notifikasi gagal akan dilakukan ketika tidak dapat mengakses koneksi.

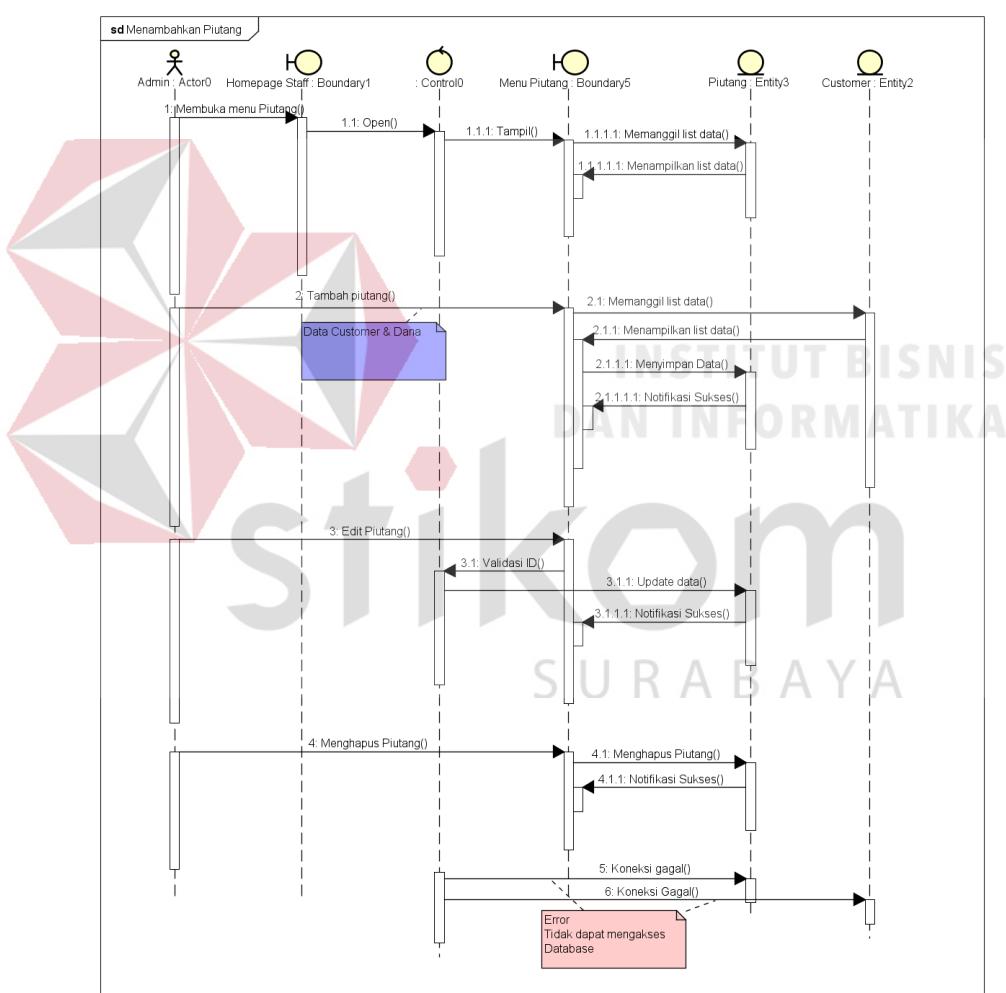
C. Mengelolah data *Customer*



Gambar 4.10 *Sequence Diagram* Mengelolah data *Customer*

Sequence Diagram ini dimulai dari Admin (*Actor*) mengakses halaman *master customer* kemudian *system control* akan melakukan fungsi *tampilan()* yang memanggil *entity customer* dan menampilkan data-datanya. Kemudian langkah selanjutnya bebas dilakukan oleh *actor*, entah itu *add*, *edit*, maupun *delete*. Fungsi notifikasi gagal akan dilakukan ketika tidak dapat mengakses koneksi.

D. Menambahkan Piutang

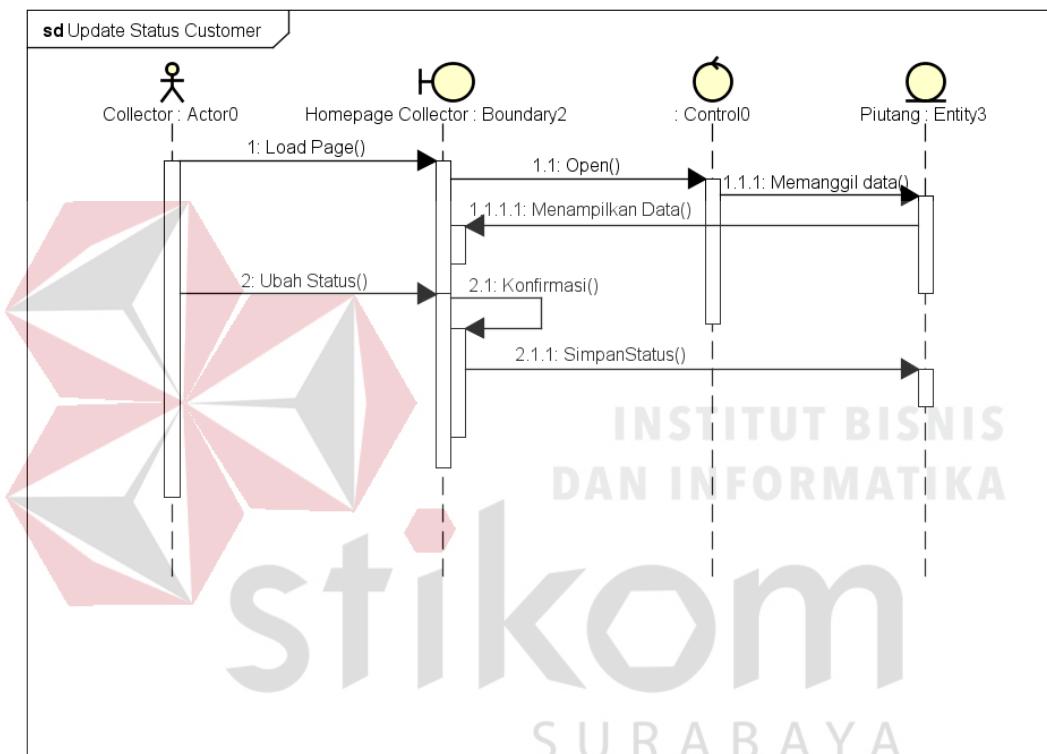


Gambar 4.11 *Sequence Diagram* Menambahkan Piutang

Sequence Diagram ini dimulai dari Admin (*Actor*) mengakses halaman *master piutang* kemudian *system control* akan melakukan fungsi *tampilan()* yang

memanggil *entity piutang* dan menampilkan data-datanya. Kemudian langkah selanjutnya bebas dilakukan oleh *actor*, entah itu *add*, *edit*, maupun *delete*. Untuk piutang terdapat attribute baru yaitu berupa dana atau uang untuk yang ditagih. Fungsi notifikasi gagal akan dilakukan ketika tidak dapat mengakses koneksi.

E. Update Status Customer

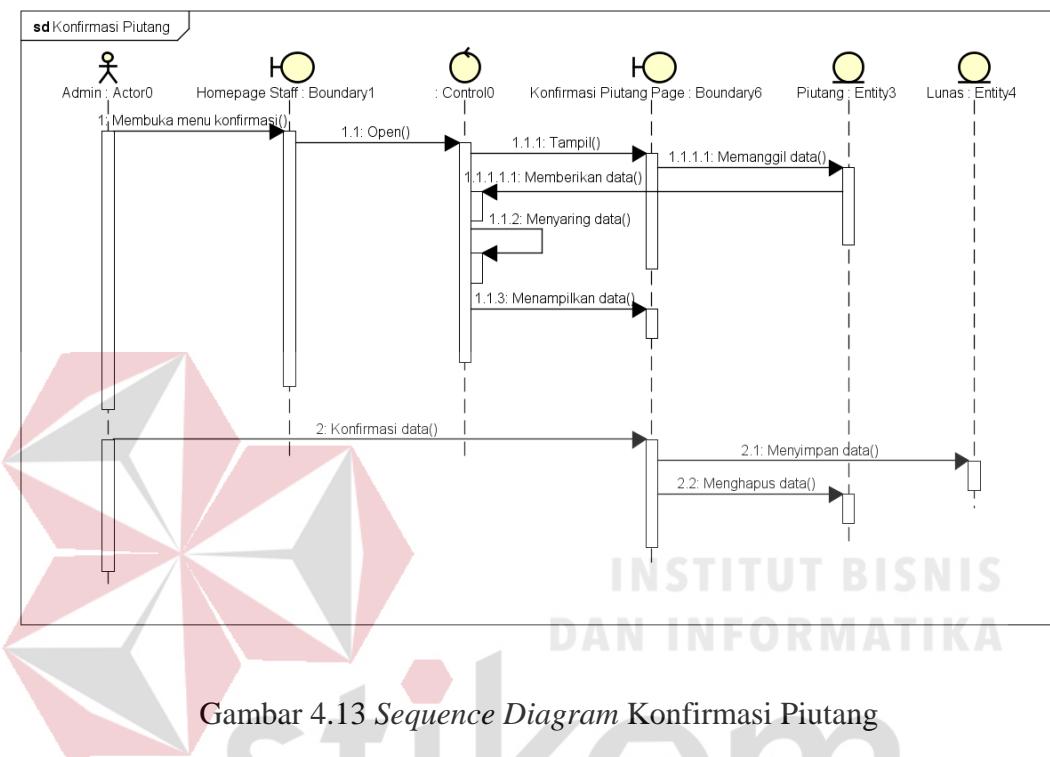


Gambar 4.12 Sequence Diagram Update Status Customer

Sequence Diagram update status customer disini dimulai dari *Collector* (Actor) membuka *Homepage Collector*, pada halaman ini dilakukannya proses *loadpage* yang memanggil *system control* untuk menghubungkan dengan *database* piutang dan menampilkan data tersebut ke boundary *Homepage Collector*. Setelah itu *Actor* dapat melakukan perubahan status dengan memilih data *control* juga akan

menanyakan konfirmasi kepada *Actor*. Kemudian *control* menyimpan data perubahan tersebut ke *entity* piutang.

F. Konfirmasi Piutang



Gambar 4.13 Sequence Diagram Konfirmasi Piutang

Pada *Sequence Diagram* ini Admin (*Actor*) membuka menu konfirmasi kemudian *system control* akan menampilkan halaman konfirmasi piutang (*Boundary*). Ketika terbukanya halaman ini, *control* memanggil *entity* piutang untuk memberikan data-data tersebut lalu dimasukkannya kedalam sebuah *list* pada halaman konfirmasi piutang. Admin kemudian dapat memilih data mana yang ingin dijadikan lunas. Control kemudian melaksanakan fungsi konfirmasi data yaitu menyimpan data dari *entity* piutang ke *entity* lunas.

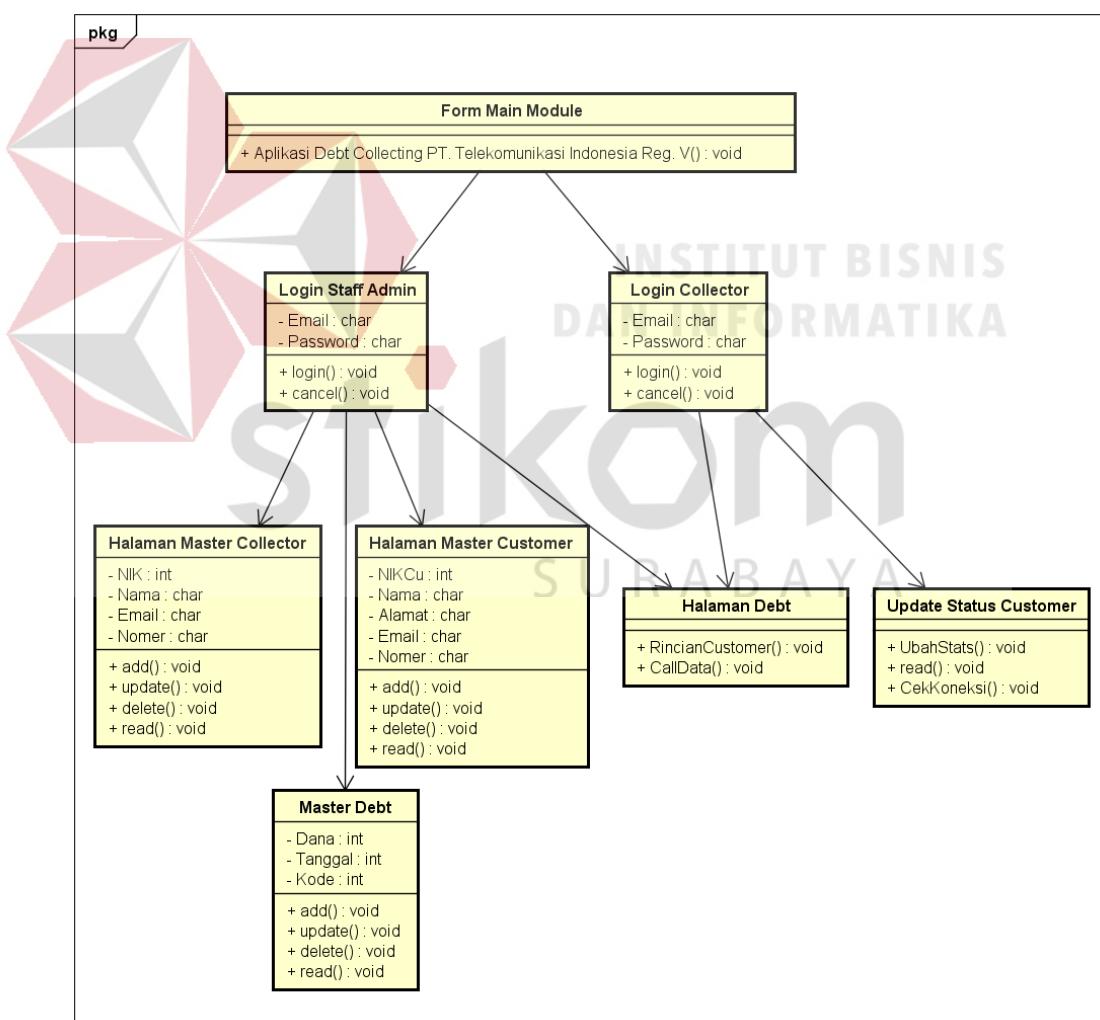
4.4.4 Class Diagram

Class Diagram yaitu sebuah spesifikasi yang jika diinstalasi menghasilkan objek dan inti dari pengembangan desain berorientasi objek. *Class Diagram*

digunakan untuk mengimplementasikan interface dari sebuah aplikasi yang akan dibuat dan diimplementasikan.

Class diagram dalam *system debt collecting* pada PT. Telkom Regional V ini menggambarkan rancangan *interface* atau rancangan antar muka berupa modul-modul Aplikasi piutang yaitu *class diagram debt collect*. Berikut *class diagram* dari aplikasi *debt collecting* PT. Telkom Regional V.

A. Class diagram debt collect



Gambar 4.14 Class Diagram Debt Collect

4.5 Desain Implementasi Sistem

Fase desain implementasi *system* menjelaskan mengenai tampilan tatap muka pengguna (*User Interface*) dari aplikasi yang telah selesai dibuat, beserta fungsinya.

4.5.1 Halaman Login

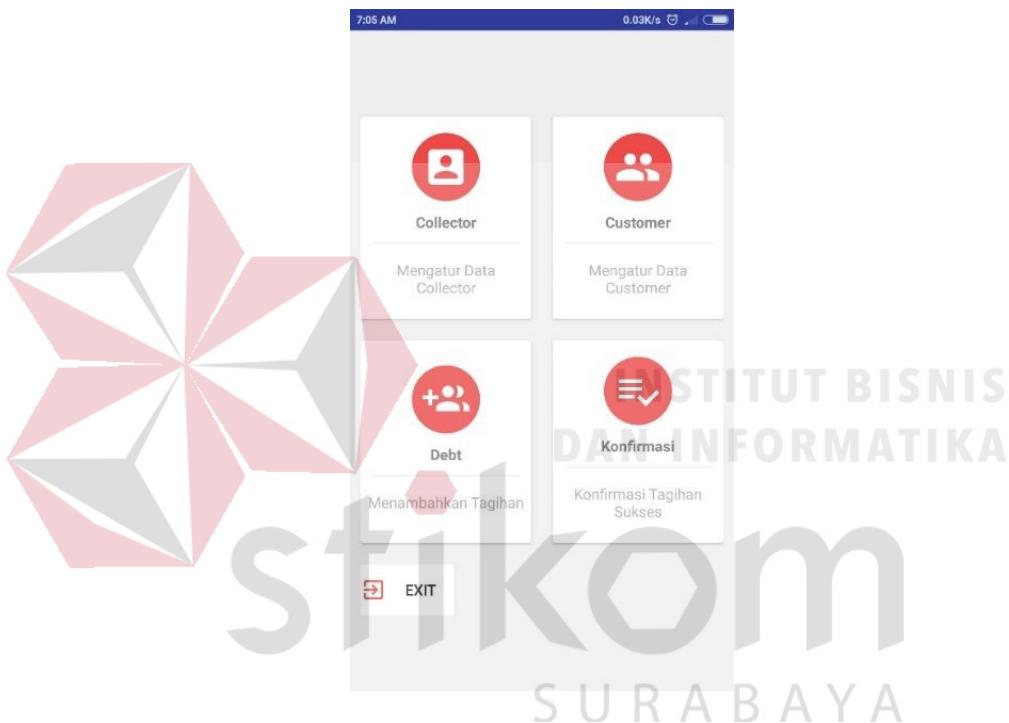


Gambar 4.15 Implementasi Menu *Login*

Fungsi dari *Form Login* adalah sebagai menu awal untuk mengakses aplikasi. Seperti yang terlihat pada Gambar 4.15 terdapat 2 *Text box* yang dapat diisi yaitu berupa *username* dan *password*. Sebelum mengakses halaman utama yang diinginkan, *user* harus memasukkan sesuai data *username* dan *password* yang telah diketahui dan terdapat pada *database*. Jika dalam memasukkan data diri *user* salah maka akan muncul notifikasi “*username* atau *password* anda salah”.

4.5.2 Halaman Awal Admin

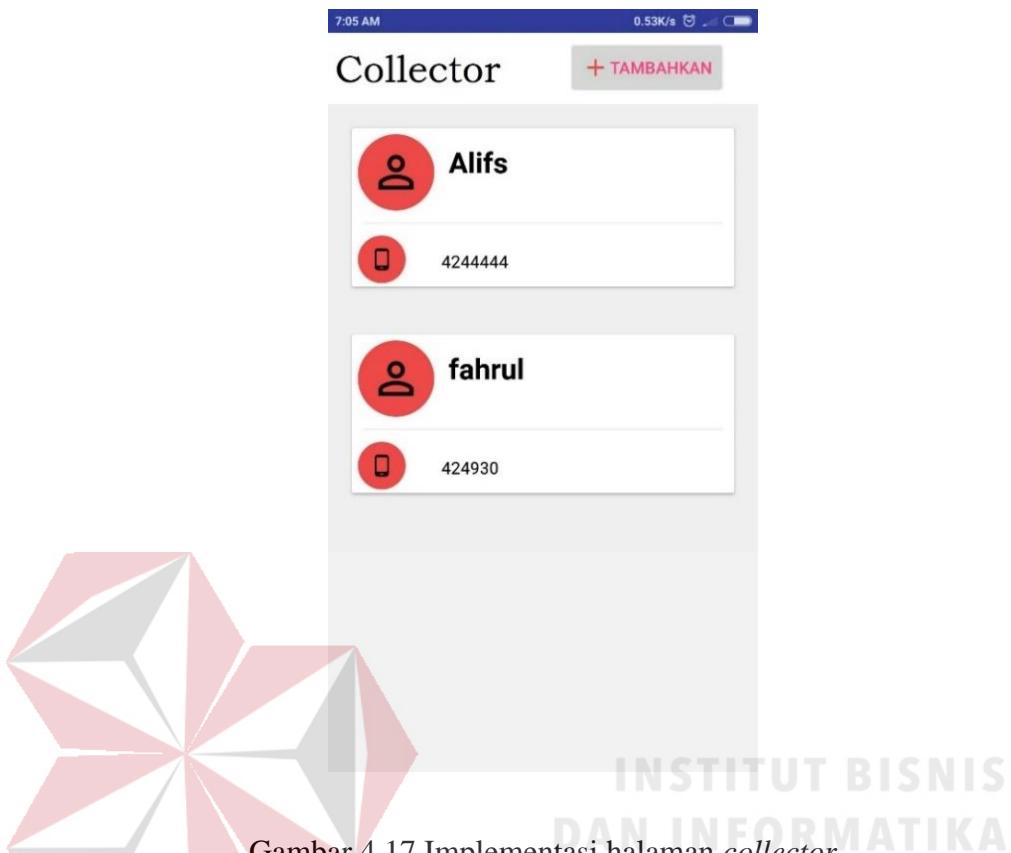
Fungsi dari *Form* Admin disini hanya bisa diakses setelah *user* telah melakukan proses *Login* dan data ID tersebut termasuk milik Admin. Pada halaman ini terdapat beberapa *button collector*, *customer*, *debt*, dan konfirmasi yang masing-masing dari tombol tersebut akan memindahkan *user* pada halaman yang telah dipilih selanjutnya.



Gambar 4.16 Implementasi Menu Admin

Pada Gambar 4.16 *Collector* dan *Customer* adalah sebuah *button* yang apabila ditekan akan menampilkan tampilan halaman ke *Master* data yang dipilih. *Debt* adalah *button* menu untuk menampilkan data utang.

4.5.3 Halaman *Collector*



Gambar 4.17 Implementasi halaman *collector*

Pada Gambar 4.17 adalah halaman utama yang ditampilkan ketika *user* menekan tombol *Collector* pada menu utama. Didalam halaman tersebut terdapat *list*, dan tombol tambah. *List* tersebut berisi daftar *Collector* yang telah ada di *database*. Sedangkan pada Gambar 4.18 adalah tampilan ketika admin menekan tombol tambah.



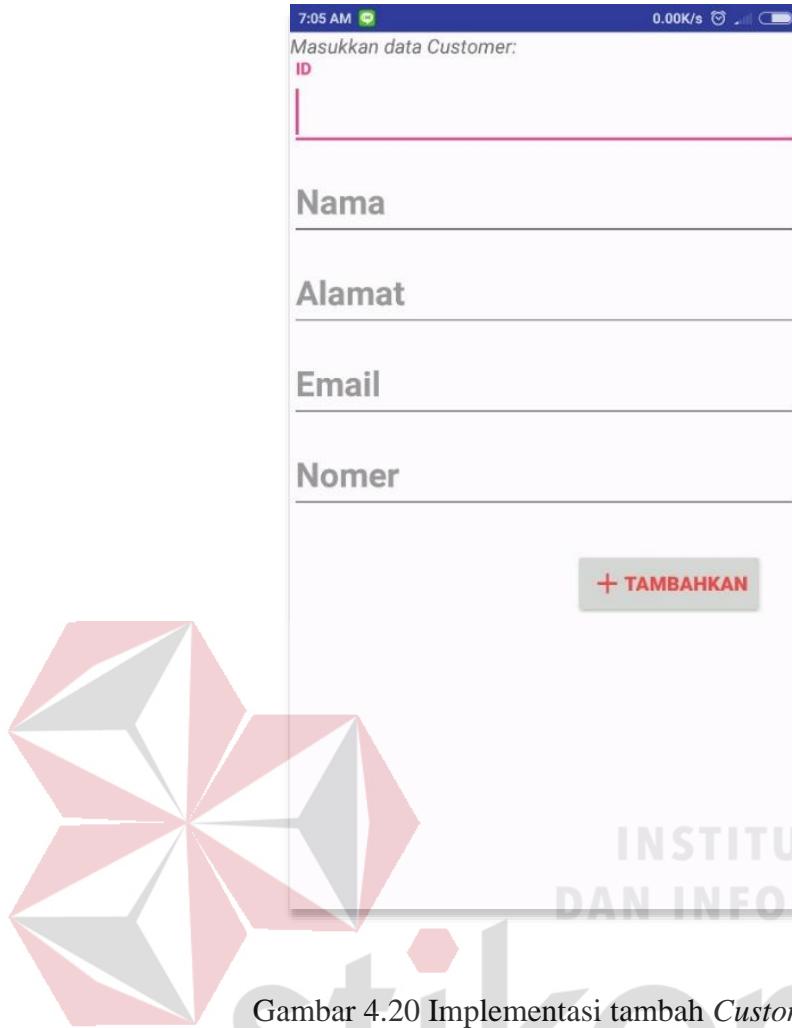
Gambar 4.18 Implementasi tambah *collector*

Variabel yang dibutuhkan pada halaman tambah *collector* adalah ID, Nama, Alamat, *Email*, dan Nomor. Setelah mengisi data tersebut maka data akan tersimpan ke *database firebase*. Untuk update data *collector* cukup menekan data yang ingin diubah pada *list* yang terdapat pada Gambar 4.17.

4.5.4 Halaman Customer



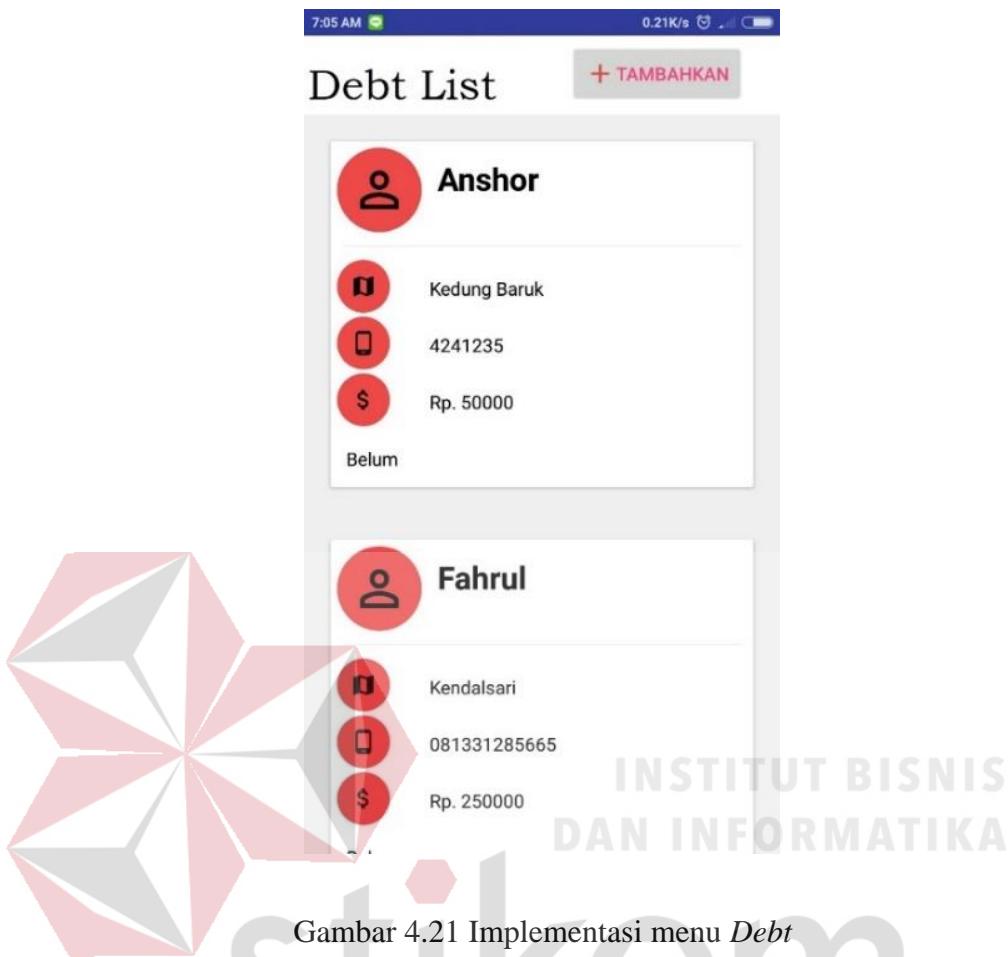
Pada Gambar 4.19 adalah halaman utama yang ditampilkan ketika user menekan tombol *Customer* pada menu utama. Didalam halaman tersebut terdapat *list*, dan tombol tambah. *List* tersebut berisi daftar *Customer* yang telah ada di *database*. Sedangkan pada Gambar 4.20 adalah tampilan ketika admin menekan tombol tambah.



Gambar 4.20 Implementasi tambah *Customer*

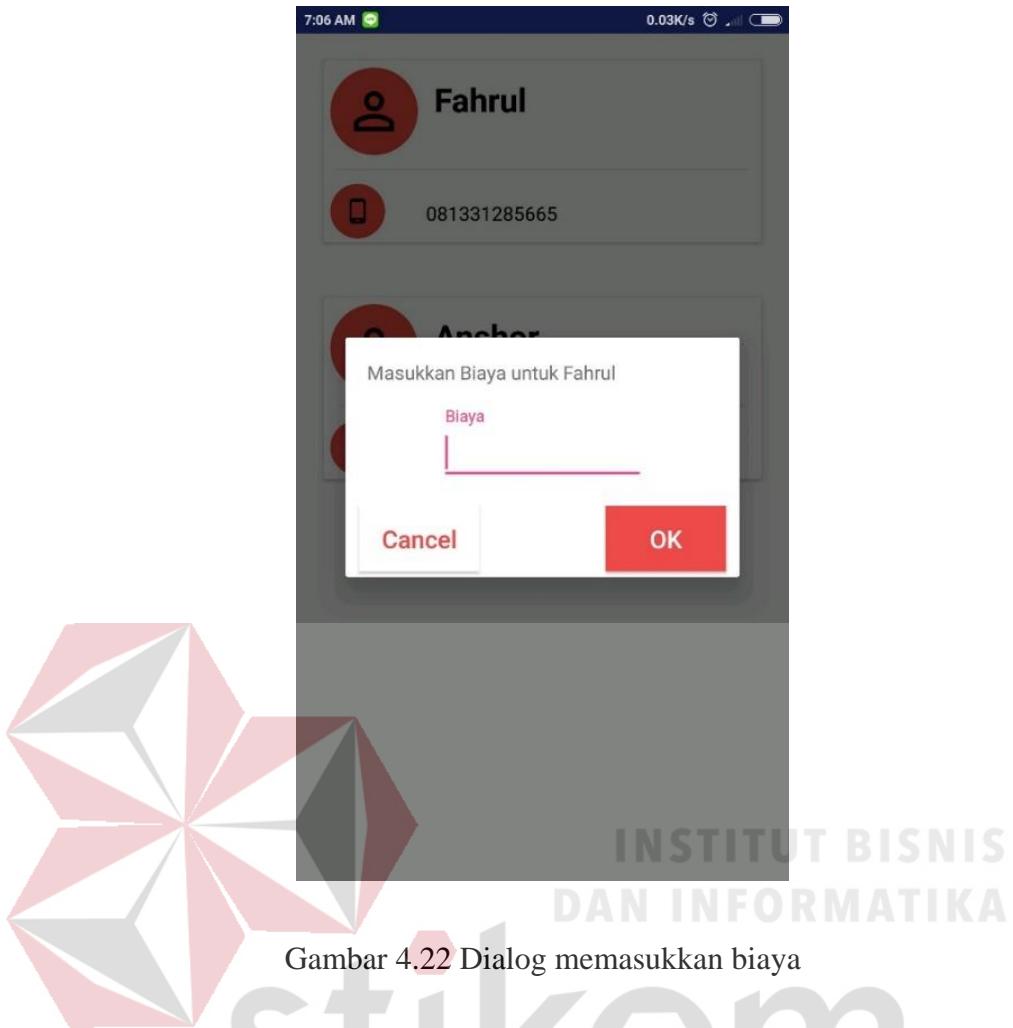
Variabel yang dibutuhkan pada halaman tambah *collector* adalah ID, Nama, Alamat, *Email*, dan Nomor. Setelah mengisi data tersebut maka data akan tersimpan ke *database firebase*. Untuk *update* data *collector* cukup menekan data yang ingin diubah pada *list* yang terdapat pada Gambar 4.17.

4.5.5 Halaman *Debt*



Gambar 4.21 Implementasi menu *Debt*

Pada Gambar 4.21 adalah gambar tampilan awal yang menampilkan *list Customer* yang memiliki utang dan terdapat tombol tambah pada *headbar* aplikasi. Dalam menu ini admin dapat melakukan proses pengubahan data sama seperti menu *Customer* ataupun menu *Collector*. Ketika tombol tambahkan di tekan, maka tampilan akan berubah menjadi sebuah *list customer* yang ada pada *database*. Untuk proses pengubahan cukup menekan data pada *list* yang ingin diubah. Kemudian mengisi data yang ingin diubah kemudian tekan simpan perubahan.



Gambar 4.22 Dialog memasukkan biaya

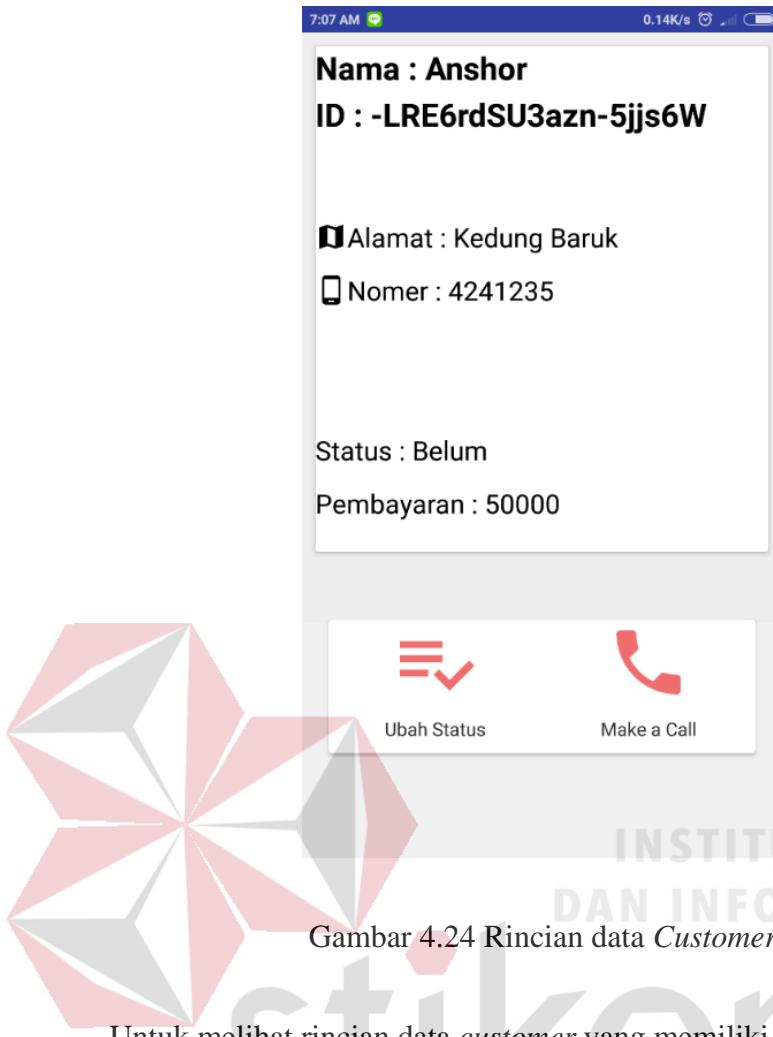
Pada Gambar 4.22 ditampilkan data *customer* yang telah terdaftar. Kemudian untuk menambahkan utang tersebut tekan data *customer* yang ingin ditambahkan. Masukan biaya sesuai data yang ada dan tekan tombol OK untuk menyimpan. Data tersebut akan tersimpan dan dapat dilihat pada *Debt List*. Untuk menghapus data *Debt List* cukup menenkan data tersebut lalu tekan tombolhapus data.

4.5.6 Halaman Menu Utama *Collector*



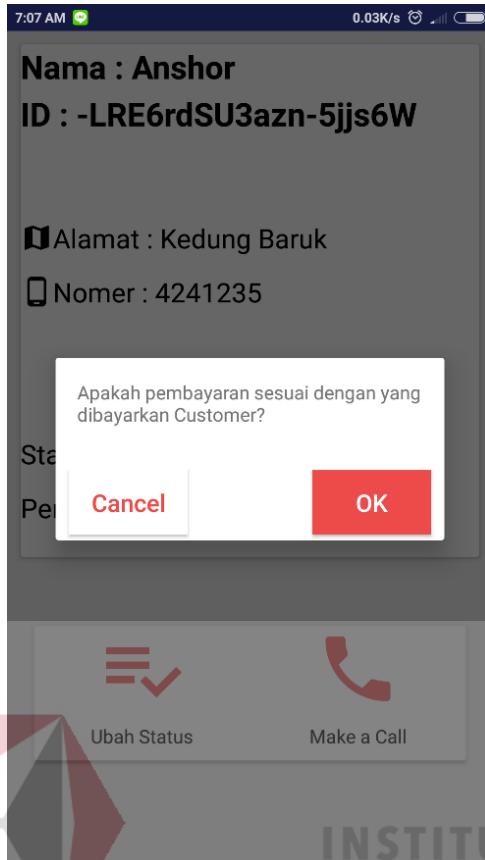
Gambar 4.23 Implementasi menu utama *Collector*

Pada Gambar 4.23 adalah tampilan ketika dilakukannya proses *Login* dan akun yang terdaftar terdeteksi milik *Collector*. Dalam halaman ini terdapat *list Customer* yang memiliki utang, beserta alamat dan nomer telepon milik *customer*. *User interface* selanjutnya dapat dilihat pada Gambar 4.24 dibawah ini.



Gambar 4.24 Rincian data *Customer*

Untuk melihat rincian data *customer* yang memiliki utang cukup tekan data tersebut maka halaman yang ditampilkan akan seperti pada Gambar 4.24. Pada halaman ini terdapat *variable* Nama, ID, Alamat, Nomor, Status, Pembayaran. ID disini akan *digenerate random* oleh *system* ketika pembuatan data *Customer* oleh admin. Terdapat 2 *button* yang bisa ditekan pada halaman ini yaitu Ubah Status, dan *Make a Call*. *Make a Call* disini akan otomatis menghubungi data nomer yang tercantum.



Gambar 4.25 Konfirmasi perubahan *status*

Ketika *button* Ubah Status ditekan maka akan muncul dialog seperti

Gambar 4.25. Proses Ubah Status ditekan apabila *Collector* telah menerima biaya uang sesuai data yang tercantum. Ketika pembayaran sesuai *Collector* bisa menekan tombol OK dan data akan tersimpan sebagai data dengan status LUNAS.

4.5.7 Halaman Konfirmasi



BAB V

PENUTUP

5.1 Simpulan

Berdasarkan dari hasil perancangan, pembuatan, dan implementasi Aplikasi *Debt Collecting* pada PT. Telekomunikasi Indonesia Regional V maka dapat dibuat kesimpulan sebagai berikut:

1. Aplikasi yang dibangun membutuhkan koneksi untuk dapat mengakses *database*. *Collector* tidak dapat mengakses aplikasi tanpa koneksi
2. Data diakses secara *real time*. Jika data berubah dalam satu *device* maka semua *device* yang sedang mengakses aplikasi yang dibangun juga akan ikut berubah.
3. Aplikasi yang dibangun membantu dalam menyajikan informasi untuk proses penagihan.

5.2 Saran

Dalam Aplikasi *Debt Collecting* pada PT. Telekomunikasi Indonesia Regional V pada PT. Telekomunikasi Regional V tentunya memiliki kekurangan. Oleh sebab itu, untuk pengembangan aplikasi ini agar menjadi lebih baik, maka disarankan adanya petugas untuk melakukan *maintenance* program ataupun pengembangan berlanjut. Aplikasi ini dapat dikembangkan ke versi selanjutnya dengan memperbaiki ataupun membangun hal-hal berikut:

1. Desain UI aplikasi diperbaiki dan dikembangkan.
2. Aplikasi dikembangkan ke iOS.

DAFTAR PUSTAKA

- Connolly, T., & Begg, C. (2010). *Database Systems A Practical Approach to Design, Implementation, and Management Fifth Edition*. Boston: Perason Education.
- Kadir, A. (2014). *Buku Pertama Belajar Pemrograman Untuk Pemula*. Yogyakarta: Mediakom.
- Safaat, N. (2014). *Pemrograman Aplikasi Mobile Smarthphone dan Tablet berbasis Android*. Bandung: BI-Obses.

Studio, A. (2018, April 25). *Mengenal Android Studio*. Retrieved from [deverloper.android.com](http://developer.android.com)

Susanto, S. H. (2011). *Mudah Membuat Aplikasi Android*. Yogyakarta: CV. ANDI OFFSET.

