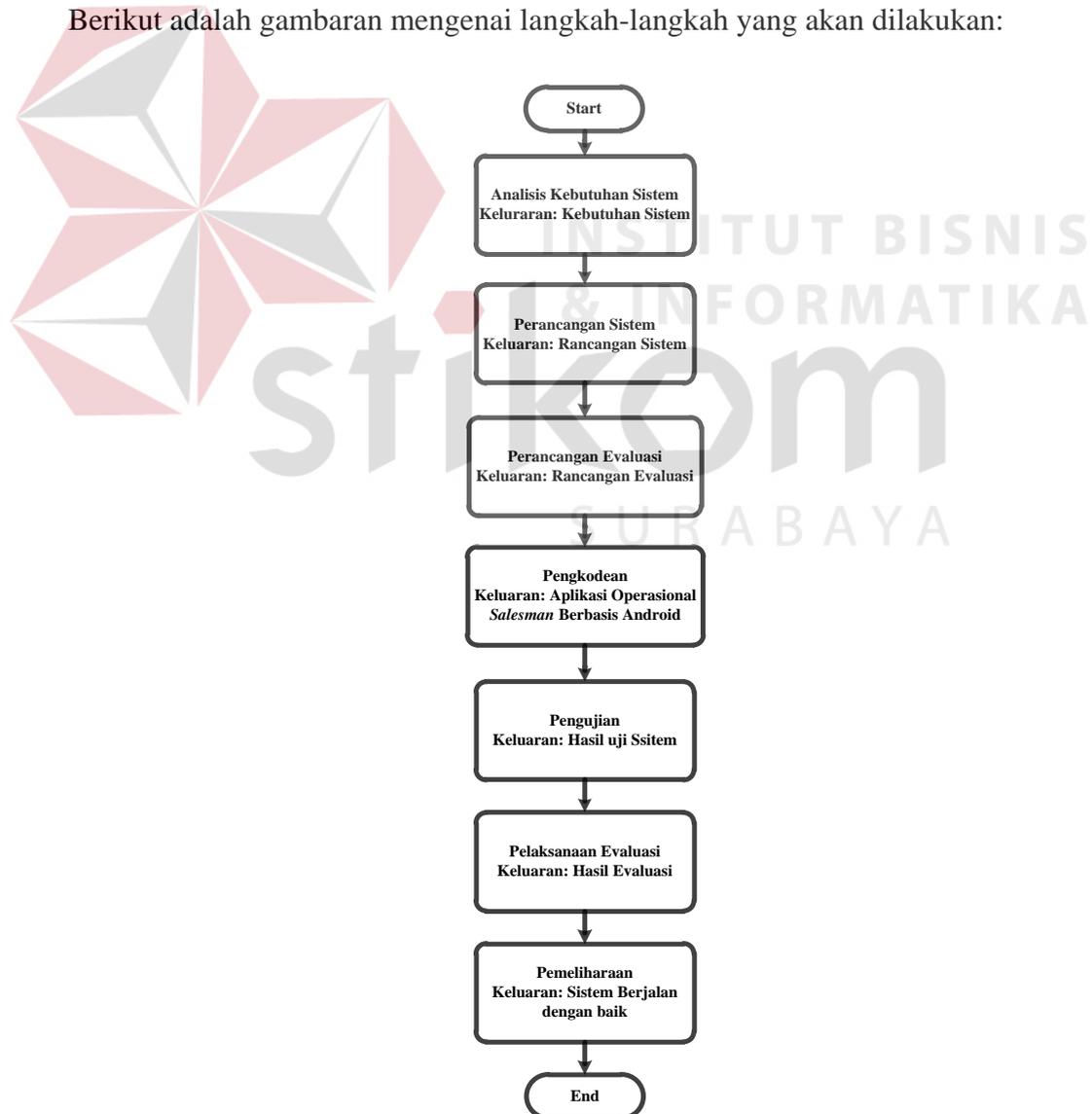


BAB III

ANALISIS DAN PERANCANGAN SISTEM

Sebelum melakukan analisis dan perancangan sistem diperlukan langkah-langkah untuk penelitian tugas akhir ini. Dalam penelitian tugas akhir ini menggunakan *System Development Life Cycle* (SDLC) dengan model *waterfall*. Model ini dipilih karena merupakan metode pengembangan yang terstruktur dan dalam penelitian tugas akhir ini dapat dilakukan secara sistematis dan terarah.

Berikut adalah gambaran mengenai langkah-langkah yang akan dilakukan:



Gambar 3.1 Langkah-Langkah Penelitian Tugas Akhir

Berikut penjelasannya:

a. Langkah analisis kebutuhan sistem

Analisis kebutuhan sistem adalah menganalisis kebutuhan pengguna dari sisi sistem. Analisis kebutuhan sistem didapatkan dari mengidentifikasi masalah. Adapun tujuannya untuk menganalisis kebutuhan sistem adalah untuk memahami kebutuhan apa saja yang diperlukan pengguna yaitu *salesman* dari sisi sistem yang akan dibangun. Output dari analisis kebutuhan sistem adalah menghasilkan kebutuhan sistem yang diperlukan oleh *salesman*.

b. Langkah perancangan sistem

Perancangan sistem adalah suatu langkah untuk melakukan perancangan sistem. Tujuan dalam perancangan sistem tugas akhir ini adalah untuk menggambarkan rancangan sistem berupa desain arsitektur, diagram *input-process-output*, *document flow*, *system flow*, *data flow diagram*, *entity relationship diagram*, struktur basis data, dan desain antar muka pada aplikasi operasional *salesman* berbasis android. Output dari perancangan sistem ini adalah menghasilkan sebuah rancangan sistem aplikasi operasional *salesman* berbasis android

c. Langkah perancangan evaluasi

Perancangan evaluasi digubakan untuk merancang apa saja yang akan dilakukan untuk mengevaluasi apakah tugas akhir ini sudah sesuai dengan tujuan yang sudah ditetapkan, yaitu menghasilkan aplikasi operasional *salesman* berbasis android untuk membantu kinerja *salesman* CV. Dwi

Star dan perusahaan CV. Dwi Star. Output dari perancangan evaluasi ini adalah rancangan evaluasi.

d. Langkah pengkodean

Pengkodean dilakukan untuk menghasilkan aplikasi operasional *salesman* berbasis android. Output dari pengkodean adalah aplikasi operasional *salesman* berbasis android.

e. Langkah pengujian

Pengujian dilakukan untuk menguji dari sisi sistem aplikasi operasional *salesman* berbasis android. Tujuan dari pengujian pada tugas akhir ini adalah untuk mengetahui apakah *input-output* dari sistem aplikasi operasional *salesman* berbasis android sudah berjalan sesuai dengan harapan. Output dari pengujian ini adalah hasil uji atau kesesuaian sistem aplikasi operasional *salesman* berbasis android dengan tujuan yang sudah ditetapkan.

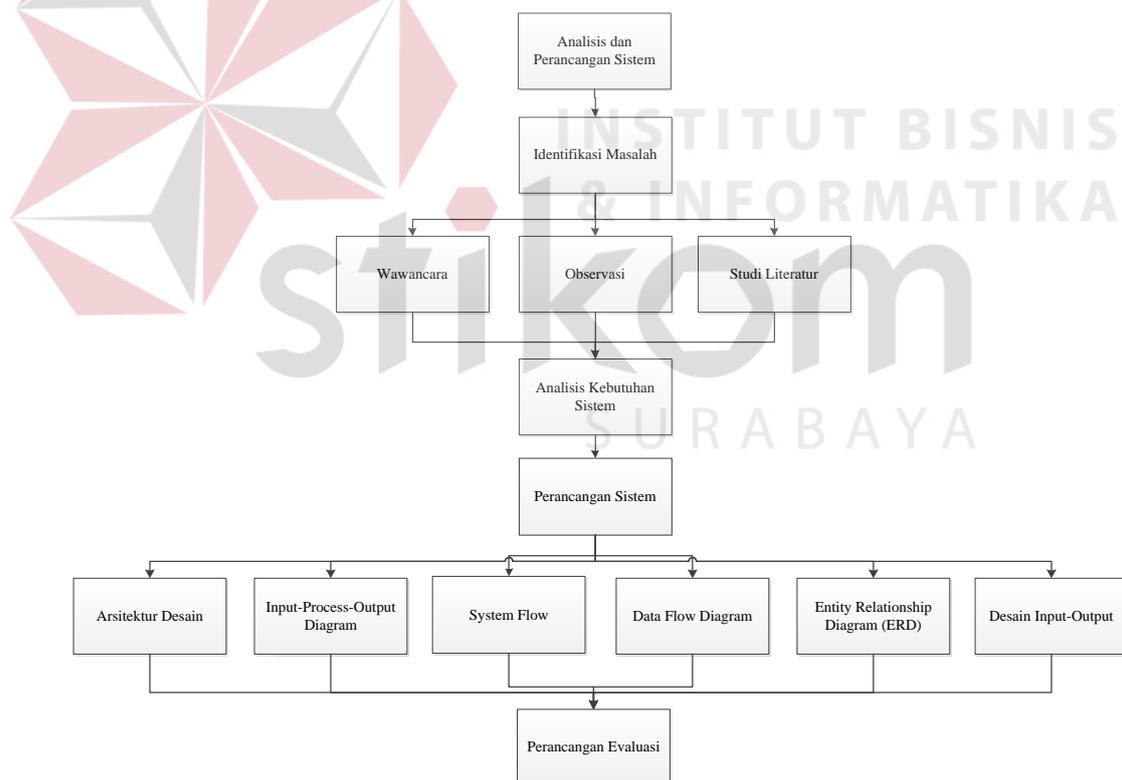
f. Langkah pelaksanaan evaluasi

Setelah melakukan perancangan evaluasi tahap yang dilakukan adalah melakukan pelaksanaan evaluasi. Pelaksanaan evaluasi adalah proses untuk mengevaluasi apakah tujuan dari tugas akhir ini sudah sesuai atau belum. Tujuan dari pelaksanaan evaluasi adalah menghasilkan aplikasi operasional *salesman* berbasis android untuk membantu kinerja *salesman* CV. Dwi Star dan perusahaan CV. Dwi Star. Output dari pelaksanaan evaluasi adalah hasil evaluasi.

g. Langkah pemeliharaan

Langkah pemeliharaan adalah langkah untuk memelihara atau mempertahankan kinerja dari sistem aplikasi operasional *salesman* berbasis android dapat berjalan dengan baik. Output dari pemeliharaan ini adalah menghasilkan sistem aplikasi operasional *salesman* berbasis android berjalan dengan baik.

Setelah menggambar dan menjelaskan langkah-langkah dalam penelitian tugas akhir ini, maka selanjutnya adalah mendetailkan langkah analisis dan perancangan sistem. Berikut adalah gambaran mengenai tahapan analisis sistem, perancangan sistem, dan perancangan evaluasi:



Gambar 3.2 Tahapan Analisis dan Perancangan Sistem

3.1 Analisis Sistem

Analisis sistem merupakan tahap yang akan digunakan untuk menganalisis sistem dengan metode *waterfall*. Dalam analisis sistem terdapat dua proses didalamnya, yaitu mengidentifikasi masalah dan menganalisis kebutuhan sistem. Berikut adalah penjabarannya:

3.1.1 Identifikasi Masalah

Mengidentifikasi permasalahan merupakan tahap awal untuk melakukan analisis sistem. Sebelum mengidentifikasi masalah pada tugas akhir ini, dilakukanlah wawancara, observasi, dan studi literatur mengenai kondisi saat ini pada CV. Dwi Star, berikut penjelasannya:

1. Wawancara

Dalam penelitian ini dilakukan proses wawancara selama 4 minggu. Proses wawancara dilakukan untuk mengetahui proses bisnis perusahaan CV. Dwi Star. Selanjutnya mencari permasalahan yang dihadapi oleh perusahaan CV. Dwi Star. Wawancara yang dilakukan untuk menggali informasi dilakukan langsung dari owner perusahaan dan beberapa karyawan CV. Dwi Star. Dari wawancara yang telah dilakukan diperoleh berbagai informasi, seperti profil perusahaan, proses bisnis, visi, misi, tujuan, struktur organisasi beserta tugas dan kewajibannya, dan harapan kedepan perusahaan.

2. Observasi

Penelitian ini tidak hanya melakukan wawancara, namun juga melakukan observasi yang dimana dilakukan pengamatan secara langsung terkait dengan proses yang berlangsung pada bagian *salesman*. Proses bisnis yang diamati meliputi pekerjaan, tanggung jawab, peran, dan kewajiban seorang *salesman*.

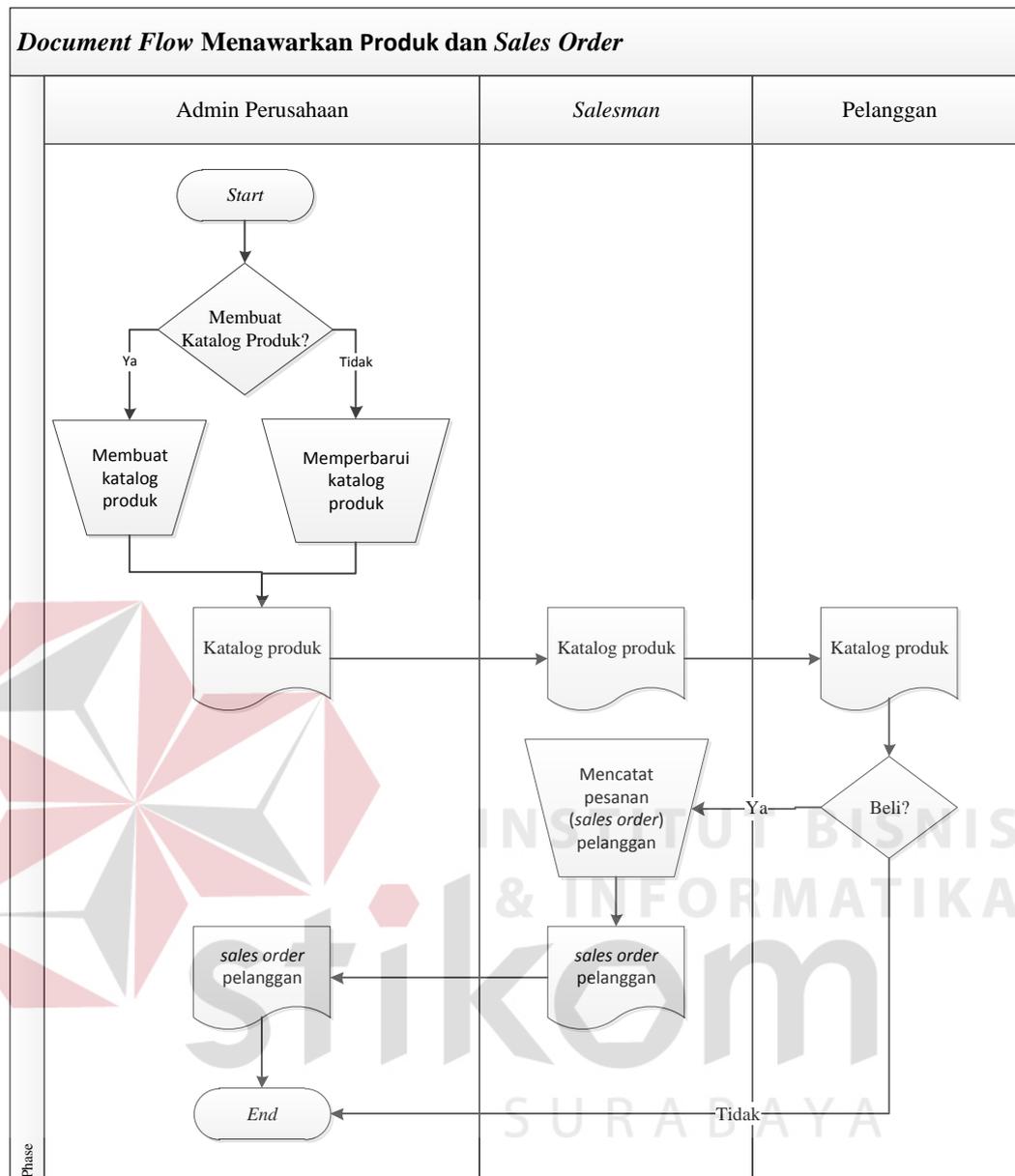
Adapun pekerjaan *salesman* adalah menawarkan produk kepada pelanggan, membuat orderan (*sales order*), melakukan proses penagihan kepada pelanggan, dan mencari pelanggan baru. Selain itu, pada saat dilakukannya observasi juga didapatkan beberapa dokumen yang dibutuhkan untuk proses analisa, seperti daftar produk yang meliputi kode, nama, dan spesifikasi produk.

3. Studi Literatur

Studi literatur yang dilakukan pada saat penelitian ini berlangsung dengan cara mencari informasi yang berkaitan dengan topik penelitian di perpustakaan, toko buku, dan melalui situs internet. Hal ini bertujuan untuk memperoleh data dan informasi yang lebih mengenai aplikasi yang dibuat, seperti yang dijelaskan dibawah ini:

- a. Mengenai aplikasi berbasis android.
- b. Mengenai peran, tugas, dan kewajiban *salesman*
- c. Mengenai *database* android.

Setelah melakukan wawancara, observasi, dan studi literatur mengenai proses bisnis CV. Dwi Star terutama mengenai peran dan tanggungjawab seorang *salesman* CV. Dwi Star. Maka peran dan tanggungjawab seorang *salesman* CV. Dwi Star dapat digambarkan dalam sebuah *document flow* seperti pada gambar 3.3 dan gambar 3.4. Gambar 3.3 menjelaskan proses seorang *salesman* dalam menawarkan produk dan membuat *sales order* pelanggan. Sedangkan Gambar 3.4 menjelaskan mengenai proses penagihan hutang yang dilakukan oleh *salesman*.

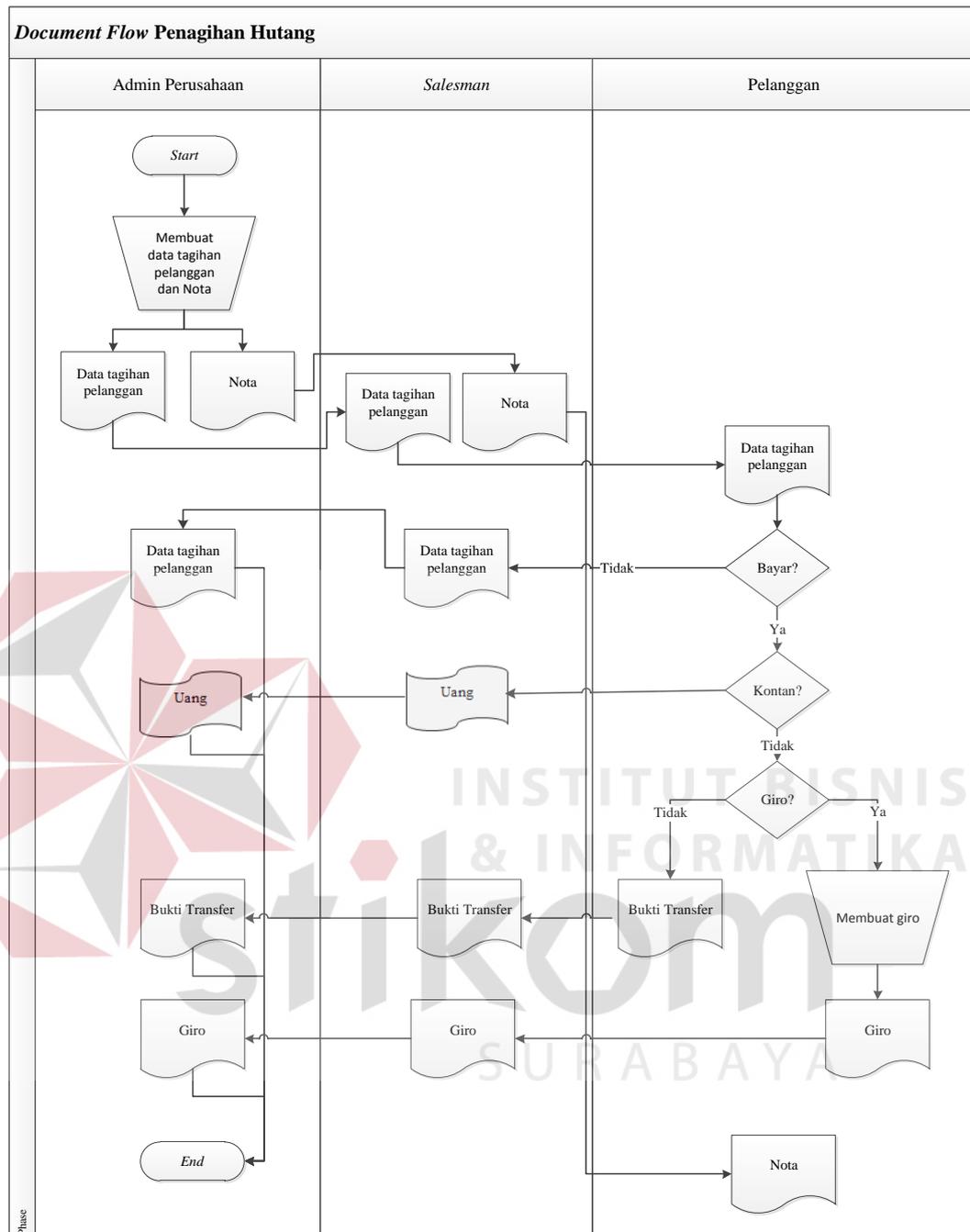


Gambar 3.3 Document flow menawarkan produk dan sales order

Gambar 3.3 merupakan alur dari menawarkan produk dan melakukan sales order. Sebelum seorang salesman menawarkan produk, admin perusahaan harus membuat katalog produk. Katalog produk ini berfungsi untuk membantu salesman dalam menawarkan produk dan dalam katalog produk ini berisikan data produk (kode produk, nama, harga, jenis, dan keterangan prduk). Katalog produk ini akan mengalami pembaruan jika terdapat perubahan, misal terjadi perubahan dalam

harga produk. Pembuatan dan pembaruan katalog produk dilakukan oleh admin perusahaan. Jika katalog produk telah siap, maka akan diberikan kepada seorang *salesman*. *Salesman* inilah yang akan menawarkan produk kepada pelanggan. Jika pelanggan berminat untuk membeli produknya, maka *salesman* akan mencatat pesanan (*sales order*) pelanggan. Setelah itu *sales order* akan diberikan kepada admin perusahaan untuk diproses.

Proses awal penagihan hutang pelanggan ini adalah admin perusahaan membuat data tagihan pelanggan dan nota seperti pada gambar 3.4. Data tagihan pelanggan ini berguna untuk membantu *salesman* untuk menagih kepada pelanggan, sedangkan nota ini berfungsi jika pelanggan telah membayar tagihannya. Data tagihan pelanggan dan nota akan diberikan kepada *salesman*, kemudian *salesman* memberikan data tagihan tersebut kepada pelanggan. Jika pelanggan membayar tagihan tersebut maka pelanggan mempunyai tiga cara untuk melakukan pembayaran, yaitu secara kontan, transfer, dan giro. Dalam melakukan proses pembayaran secara transfer atau giro, pelanggan akan memberikan buktinya kepada *salesman*. Kemudian *salesman* akan memberikan nota sebagai bukti pelunasan atau pembayaran kepada pelanggan. Sedangkan jika pelanggan tidak melakukan pembayaran, maka data tagihan akan dikembalikan kepada *salesman*. Data tagihan dan bukti pembayaran (transfer/giro) atau uang (kontan) akan diberikan kepada admin perusahaan. *Document flow* penagihan hutang pada gambar 3.4:



Gambar 3.4 Document flow penagihan hutang

Dari penjabaran dan uraian mengenai proses bisnis kinerja dan tanggungjawab *salesman* CV. Dwi Star, maka dapat diidentifikasi permasalahannya beserta dampak yang ditimbulkan yang diuraikan pada tabel 3.1:

Tabel 3.1 Permasalahan dan Dampak

Permasalahan	Dampak
<p><i>Salesman</i> tidak dapat memastikan harga produk yang sudah tertera pada katalog, karena harga produk yang tertera pada katalog sewaktu-waktu dapat berubah.</p>	<ol style="list-style-type: none"> 1. Membuat admin perusahaan harus selalu memberitahukan kepada <i>salesman</i> bahwa terdapat perubahan harga produk. 2. Membuat admin perusahaan harus selalu mengganti halaman katalog yang telah diperbarui harganya. 3. Membuat biaya tambahan (biaya komunikasi, biaya cetak). 4. Membuat pelanggan kecewa dengan harga yang sewaktu-waktu dapat berubah.
<p>Ada 2 cara yang dilakukan salesman dalam melakukan pemesanan produk:</p> <ol style="list-style-type: none"> 1. <i>Salesman</i> melakukan pemesanan saat itu juga dengan menghubungi (via telepon) admin perusahaan. 2. <i>Salesman</i> mengumpulkan pemesanan dalam kurun waktu tertentu (saat salesman kembali ke perusahaan). 	<ol style="list-style-type: none"> 1. Membuat proses pemesanan produk menjadi rumit karena salesman harus mencatat nama produk sesuai dengan spesifikasinya, nama pelanggan, dan jumlah produk yang dipesan. 2. Membuat proses pengiriman produk (oderan pelanggan) tertunda, karena <i>salesman</i> tidak langsung memberikan pemesanan produk tersebut kepada admin perusahaan melainkan <i>salesman</i> mengumpulkan pemesanan produk terlebih dahulu. 3. Membutuhkan biaya untuk melakukan proses komunikasi. Sedangkan seorang salesman tidak

Permasalahan	Dampak
	hanya melayani 1 pelanggan dalam sehari, tetapi dapat melayani hingga lebih dari 3 pelanggan.
<p><i>Salesman</i> harus mencatat data pelanggan baru dan <i>salesman</i> harus menghubungi admin perusahaan bahwa terdapat pelanggan baru yang ingin memesan produk saat proses keliling.</p>	<ol style="list-style-type: none"> 1. Membuat proses pemesanan produk tertunda, karena harus menunggu <i>salesman</i> menghubungi admin perusahaan bahwa ada pelanggan baru yang ingin bergabung menjadi pelanggan CV. Dwi Star. 2. Membuat biaya tambahan dalam berkomunikasi.
<p>Admin perusahaan harus menghubungi <i>salesman</i> jika terdapat penambahan tagihan pelanggan.</p>	<ol style="list-style-type: none"> 1. Membuat biaya tambahan dalam berkomunikasi. 2. Membuat kinerja dari <i>salesman</i> tidak efisien, karena <i>salesman</i> harus kembali ke wilayah tertentu untuk melakukan penagihan.

3.1.2 Analisis Kebutuhan Sistem

Dengan adanya permasalahan dan dampak yang ada maka dibuatlah aplikasi mengenai operasional *salesman* yang berguna untuk membantu kinerja dari *salesman* sendiri. Adapun analisis kebutuhan sistem *salesman* beserta penjelasannya diuraikan pada tabel 3.2:

Tabel 3.2 Analisis Kebutuhan Sistem

Analisis Kebutuhan Sistem	Data	Penjelasan
Informasi katalog produk	Data produk	Katalog produk dibutuhkan untuk membantu <i>salesman</i> dalam menawarkan produk dan <i>salesman</i> membutuhkan kepastian harga yang <i>up to date</i> pada katalog produk. Harga yang <i>up to date</i> ini berguna untuk membantu <i>salesman</i> dapat mengambil keputusan mengenai harga produk yang diberikan kepada pelanggan tanpa perlu bertanya kembali kepada admin perusahaan
Informasi pembuatan <i>sales order</i>	Data produk Data Pelanggan	Berguna untuk membantu <i>salesman</i> dapat melakukan pemesanan saat itu juga, tanpa perlu mengumpulkan pemesanan dalam kurun waktu tertentu
Informasi pelanggan baru	Data pelanggan baru	Berguna untuk membantu <i>salesman</i> dalam memproses pemesanan produk yang dilakukan oleh pelanggan baru saat itu juga.
Informasi penagihan hutang pelanggan	Data tagihan Data pelanggan	Berguna untuk membantu <i>salesman</i> dalam melakukan penagihan pelanggan dengan jumlah tagihannya, apabila admin perusahaan tiba-tiba memberikan data tagihan pelanggan baru.

3.2 Perancangan Sistem

Setelah melakukan analisis kebutuhan sistem, maka dibuatlah Rancang Bangun Sistem Informasi Operasional Salesman Berbasis Android Pada CV. Dwi Star. Perancangan sistem yang akan dibuat akan digambarkan pada gambaran arsitektur sistem, *Input-Process-Output (IPO)* diagram, *system flow*, *Data Flow Diagram (DFD)*, rancangan *database* berupa *Entity Relationship Diagram (ERD)*, struktur tabel, serta desain I/O dari sistem informasi yang akan dibuat. Berikut gambaran dan penjelasannya:

3.2.1 Arsitektur sistem

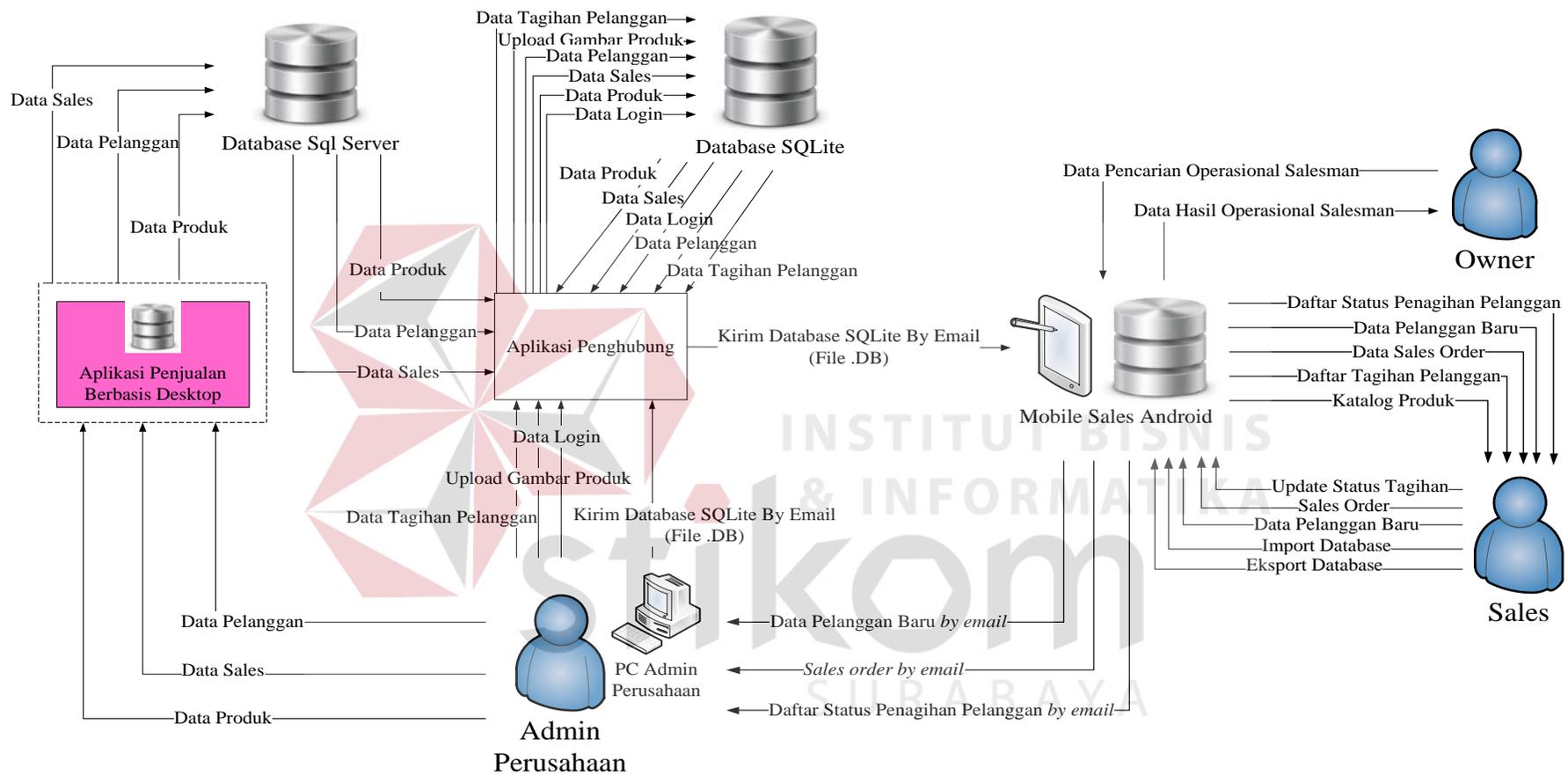
Arsitektur sistem pada penelitian ini menggambarkan mengenai sistem informasi operasional salesman berbasis android pada CV. Dwi Star yang akan dibangun. Dari arsitektur sistem pada gambar 3.5 dapat dijelaskan bahwa admin perusahaan menginputkan data pelanggan, sales, dan produk ke aplikasi penjualan berbasis desktop. Aplikasi penjualan berbasis desktop merupakan aplikasi di luar area tugas akhir ini atau aplikasi tersebut sudah ada di perusahaan CV. Dwi Star.

Selanjutnya data pelanggan, sales, dan produk akan diinputkan otomatis ke *database* Sql Server yang dibangun pada penelitian ini. Fungsi dari *database* Sql Server adalah untuk menampung ketiga data tersebut. Kemudian admin perusahaan akan mensinkronisasi *database* Sql Server menjadi *database* SQLite. Admin perusahaan juga menginputkan data tagihan pelanggan, data login, dan mengupload gambar produk. Semua data tagihan pelanggan, data login, dan gambar produk ini disimpan langsung pada *database* SQLite. Selanjutnya admin perusahaan hanya perlu mengirimkan *database* SQLite *by email* ke aplikasi operasional *salesman* berbasis android atau kepada *salesman*.

Salesman akan menerima *database SQLite by email*, lalu mengimport *database SQLite* tersebut ke dalam aplikasi operasional *salesman*. Setelah *database SQLite* diimportkan, maka aplikasi akan memproses semua data yang terdapat pada *database SQLite*. Sehingga aplikasi operasional *salesman* dapat membaca informasi data tersebut yang disebut dengan katalog produk, dan daftar tagihan pelanggan. Selanjutnya *salesman* akan melakukan pemesanan order pelanggan (*sales order*). Data *sales order* yang telah diinputkan oleh *salesman* akan dikirimkan *by email* kepada admin perusahaan. Admin perusahaan akan menerima data *sales order* melalui *email*.

Selanjutnya *salesman* melakukan penagihan pelanggan dan melakukan proses pembayaran hutang pelanggan. *Salesman* menginputkan data pembayaran hutang pelanggan pada aplikasi operasional *salesman* berbasis android. Kemudian data pembayaran hutang akan dikirimkan *by email*. Admin perusahaan juga akan mendapatkan data pembayaran hutang pelanggan yang sudah melunasi hutangnya (melalui transfer/*salesman*) atau belum melunasi.

Salesman juga dapat menambahkan data pelanggan baru pada aplikasi operasional *salesman* berbasis android. Data pelanggan baru yang telah diinputkan oleh *salesman*, akan dikirimkan *by email*. Data pelanggan baru yang telah dikirimkan *by email* akan diterima oleh admin perusahaan. Selanjutnya pelanggan baru dapat melakukan *sales order* pada *salesman*.

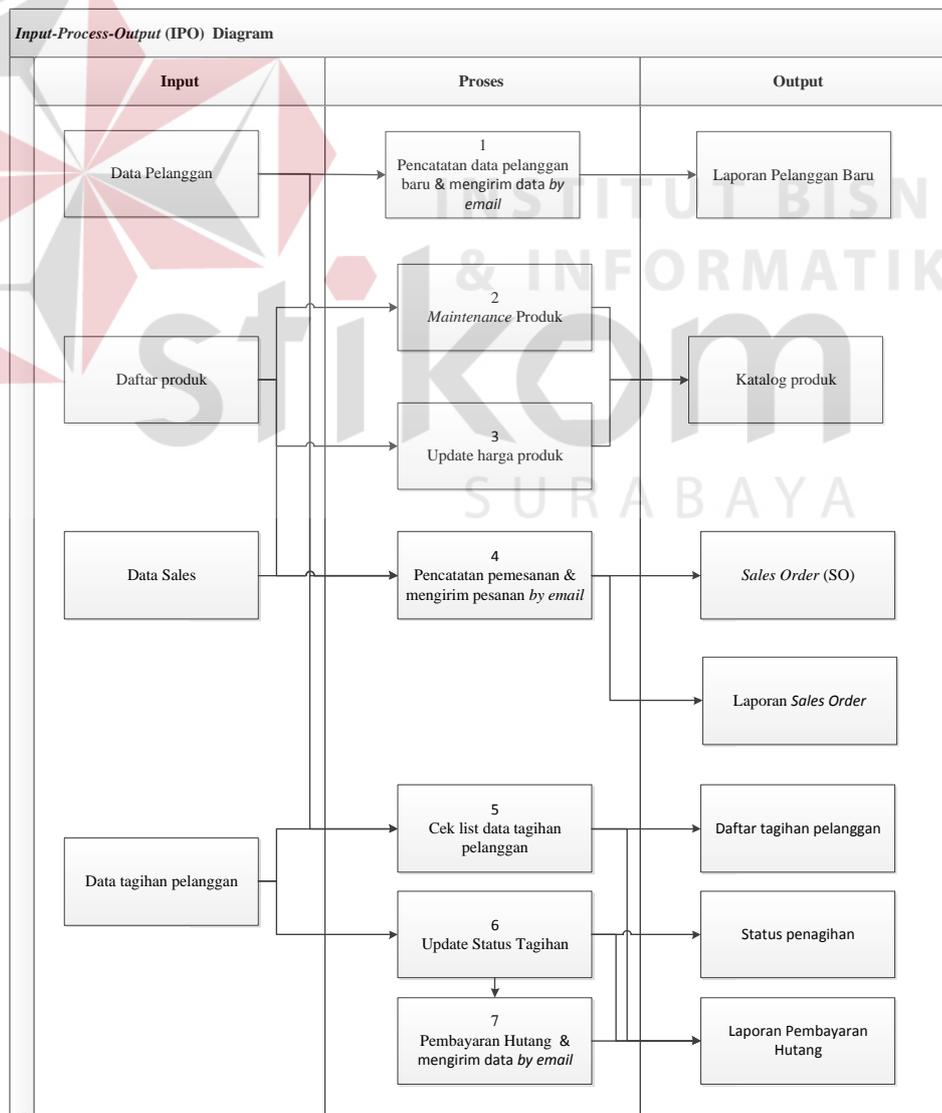


Gambar 3.5 Arsitektur sistem

Keterangan: Aplikasi penjualan berbasis desktop merupakan aplikasi di luar area tugas akhir ini atau aplikasi tersebut sudah ada di perusahaan CV. Dwi Star.

3.2.2 Input-Process-Output (IPO) Diagram

Rancang Bangun Sistem Informasi Operasional Salesman Berbasis Android Pada CV. Dwi Star ini bertujuan untuk mempermudah kinerja dari segi kecepatan pencarian barang, harga barang yang *up to date*, dan interaksi antar *salesman*-pelanggan-admin perusahaan. *Input-Process-Output (IPO)* Diagram menggambarkan mengenai aplikasi *salesman* android yang akan dibangun dengan mengetahui input yang dibutuhkan, kemudian mengolah data tersebut (proses) menjadi output. Berikut adalah IPO diagram Rancang Bangun Sistem Informasi Operasional Salesman Berbasis Android Pada CV. Dwi Star:



Gambar 3.6 *Input-Process-Output (IPO)* Diagram

Berikut adalah penjabaran dari proses IPO Diagram:

1. Pencatatan Data Pelanggan Baru & Mengirim Data By Email

Pencatatan data pelanggan baru & mengirim data *by email* merupakan proses untuk mencatat data pelanggan baru dan mengirimkan data pelanggan baru *by email*. Hasil dari pencatatan data pelanggan baru & mengirim data *by email* adalah laporan pelanggan baru.

2. Maintenance Produk

Maintenance produk merupakan proses dari daftar produk yang berguna untuk mengisi data daftar produk dan memperbarui daftar produk. Hasil dari *maintenance* produk ini adalah katalog produk.

3. Pengupdatean Harga Produk

Pengupdatean harga produk berfungsi untuk memperbarui harga produk yang berasal dari daftar produk. Hasil dari pengupdatean harga produk ini sama dengan proses *maintenance* produk yaitu menghasilkan sebuah katalog produk.

4. Pencatatan Pemesanan dan Mengirim Pesanan By Email

Proses dari pencatatan pemesanan dan mengirim pesanan *by email* ini berasal dari data pelanggan, daftar produk, dan data sales. Pencatatan pemesanan berguna untuk mencatat pesanan produk dari pelanggan dengan menyertakan data sales yang kemudian data tersebut dikirim melalui aplikasi operasional *salesman* dan akan diterima oleh admin perusahaan berupa *email*. Hasil akhir dari proses ini adalah menghasilkan *sales order* (SO) dan laporan

kinerja sales. Maksud dari kinerja sales adalah untuk mengetahui berapa banyak penjualan atau orderan yang dilakukan sales.

5. Cek List Data Tagihan Pelanggan

Cek list data tagihan pelanggan ini berisikan data tagihan pelanggan yang diinputkan oleh admin perusahaan berupa nama pelanggan, alamat pelanggan dan jumlah tagihan. Proses cek list data tagihan pelanggan berguna untuk membantu *salesman* dalam menagih pelanggan. Hasil akhir dari proses ini adalah daftar tagihan pelanggan.

6. Update Status Tagihan

Setelah mendaftarkan data tagihan pelanggan, maka *salesman* harus melakukan proses update status tagihan. Update status tagihan ini digunakan untuk memproses data tagihan pelanggan. Hasil akhir ini akan memberikan informasi mengenai status penagihan. Status penagihan ini sendiri berisikan mengenai tagihan yang belum dibayar atau sudah dibayar dan status yang sudah dibayar dengan kontan/transfer/ giro

7. Pembayaran Hutang & Mengirim Data *By Email*

Pembayaran hutang ini berisikan data pelanggan yang sudah melakukan proses pembayaran melalui *salesman*. Data pembayaran ini akan dikirimkan melalui *email*. Hasil dari pembayaran hutang ini adalah laporan pembayaran hutang.

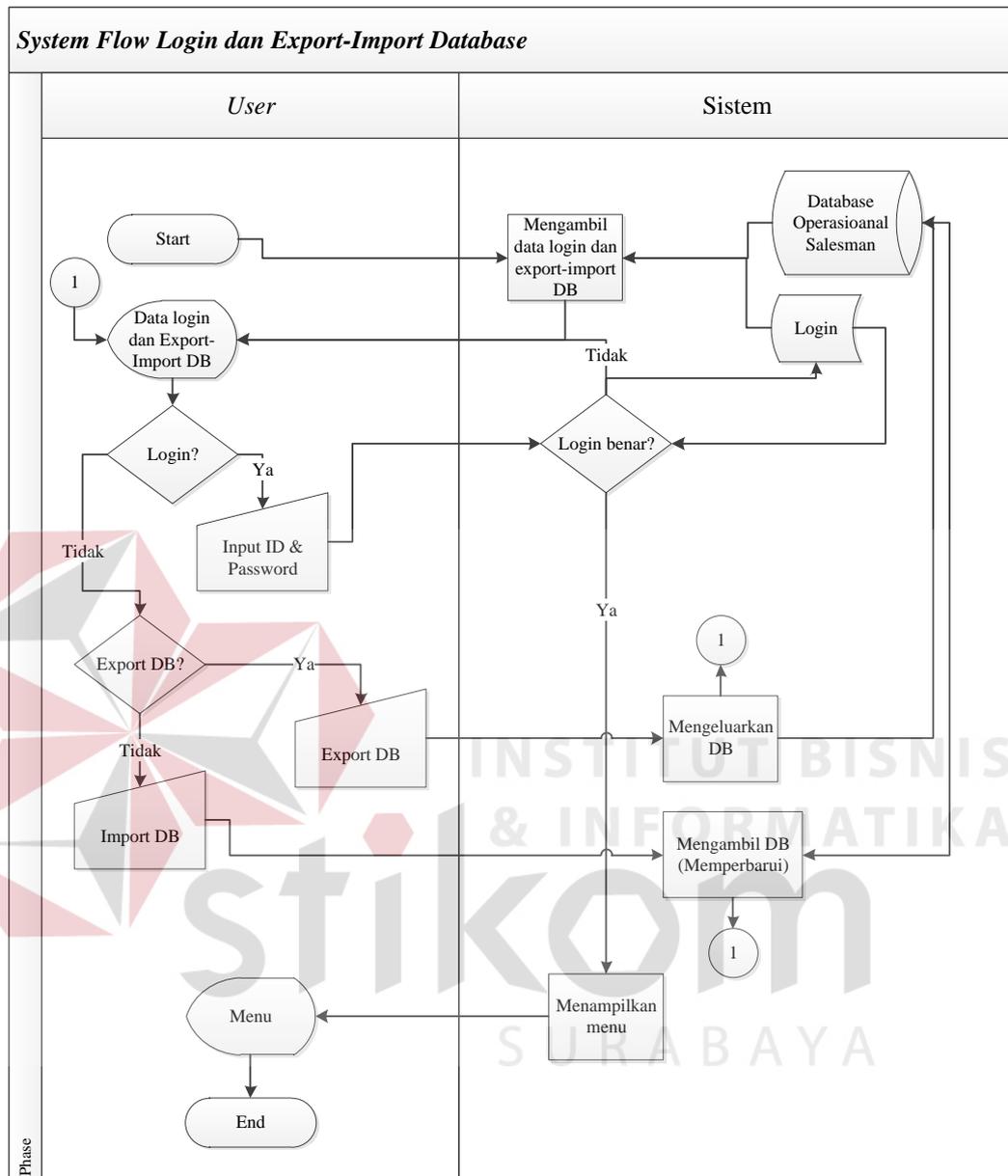
3.2.3 System Flow

System flow merupakan bagan yang menunjukkan alur kerja yang sedang dikerjakan di dalam sistem secara keseluruhan dengan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Pada laporan tugas akhir Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android ini juga menggunakan *system flow* untuk menunjukkan alur kerja pada sistem aplikasi ini, berikut adalah alur kerja dari setiap sistem:

A. System Flow Login dan Export-Import Database

System flow login dan *export-import database* ini digunakan untuk mengatur hak akses dari pengguna dan untuk memperbarui *database* operasional *salesman*. Hak akses yang diperlukan untuk mengakses proses *login* berupa ID dan *Password* yang sudah dibuat oleh admin perusahaan. Sistem disini berguna untuk melakukan proses pengecekan antara ID dan *Password* yang dimasukkan benar atau salah. Jika ID dan *Password* dinyatakan benar oleh sistem, maka *user* dapat melakukan proses selanjutnya atau menampilkan menu dari aplikasi (sesuai hak akses untuk mengakses menu pada aplikasi). Tetapi jika ID dan *Password* yang dimasukkan oleh *user* salah, maka proses untuk meminta ID dan *Password* akan berulang kembali.

Sementara untuk proses *export database* berfungsi untuk mengeluarkan *database* operasional *salesman*, sedangkan *import database* berfungsi untuk mengambil dan memperbarui *database* operasional *salesman*. Berikut adalah gambaran dari *system flow login* dan *export-import database* operasional *salesman*:

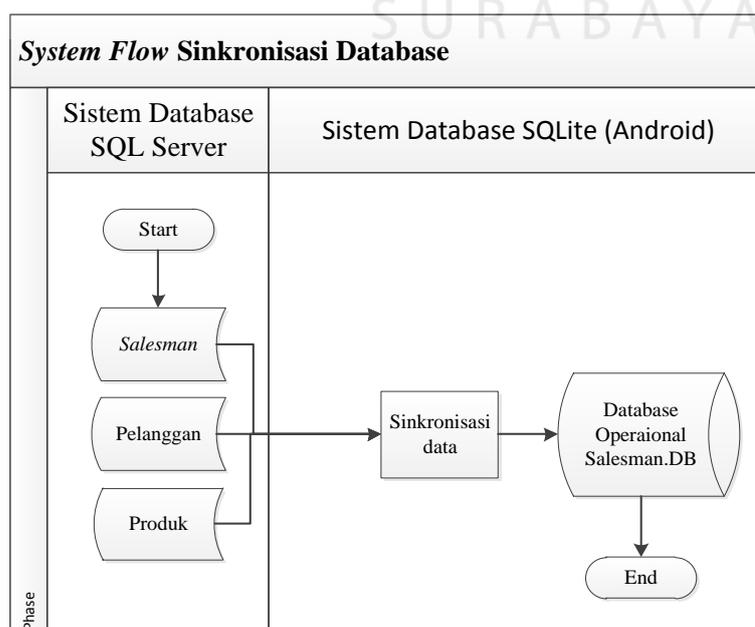


Gambar 3.7 System Flow Login

System flow master login ini digunakan untuk membuat ID dan *Password* untuk *user* (*user* disini adalah *salesman*) beserta untuk mengatur hak akses aplikasi. Admin perusahaan adalah pemegang kendali dari master login ini. Admin perusahaan dapat membuat, memperbarui, dan menghapus ID dan *Password salesman*. Dalam membuat master login ini, admin menginputkan ID, *password*, dan nama *salesman*. Admin perusahaan juga dapat mencari ID dan *Password user* dengan mengetikan kata kuncinya. Berikut adalah gambaran dari *system flow* master login:

C. System Flow Sinkronisasi Database

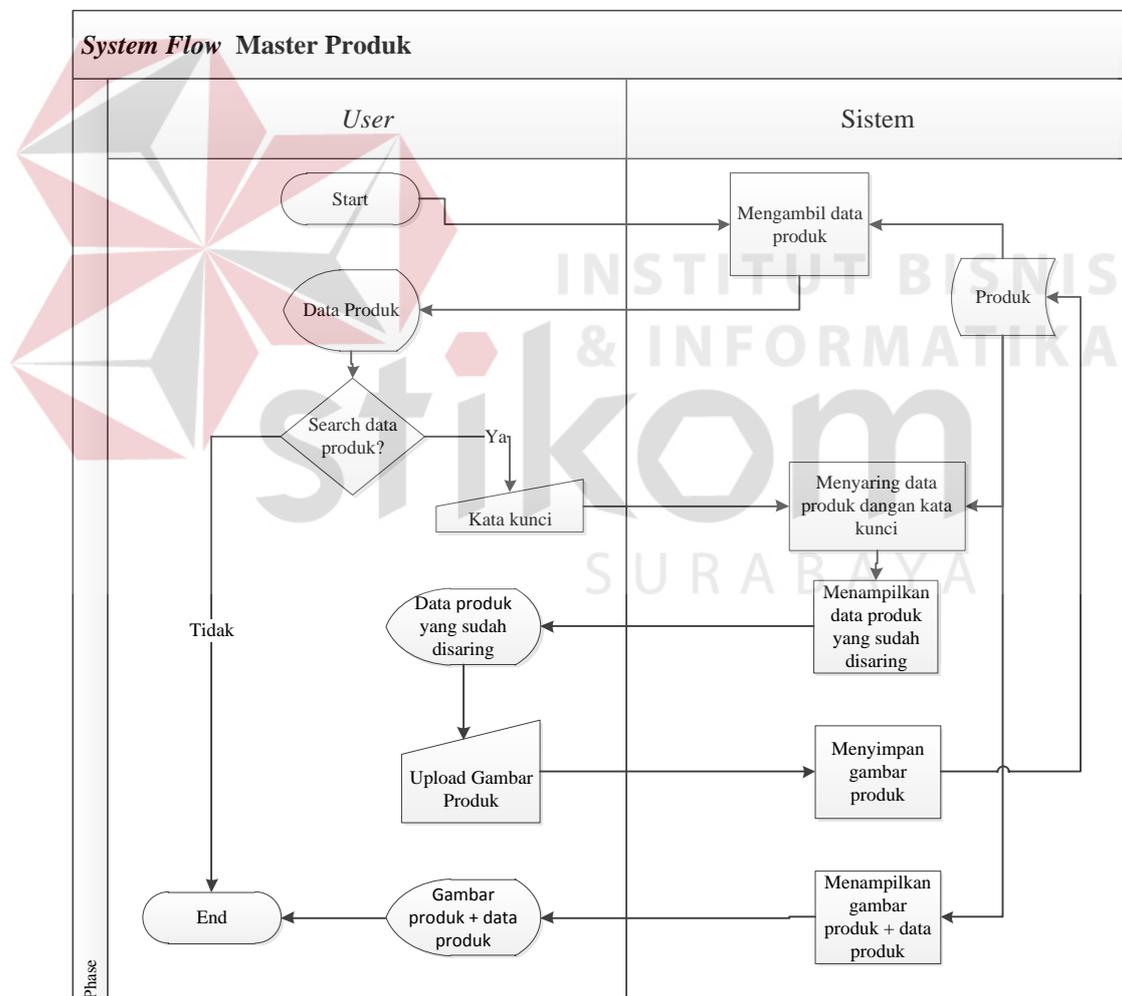
System flow sinkronisasi *database* ini berguna untuk mensinkronisasikan *database* Sql server menjadi *database* SQLite. Sinkronisasi *database* ini mensinkronisasikan tiga tabel (*data*), yaitu data pelanggan, *salesman*, dan produk. Jika ketiga data tersebut telah disinkronisasi menjadai *database* SQLite, maka ketiga data tersebut siap untuk digunakan pada aplikasi operasional *salesman* berbasis android. Berikut adalah gambaran dari *system flow* sinkronisasi *database*:



Gambar 3.9 System Flow Sinkronisasi Database

D. System Flow Master Produk

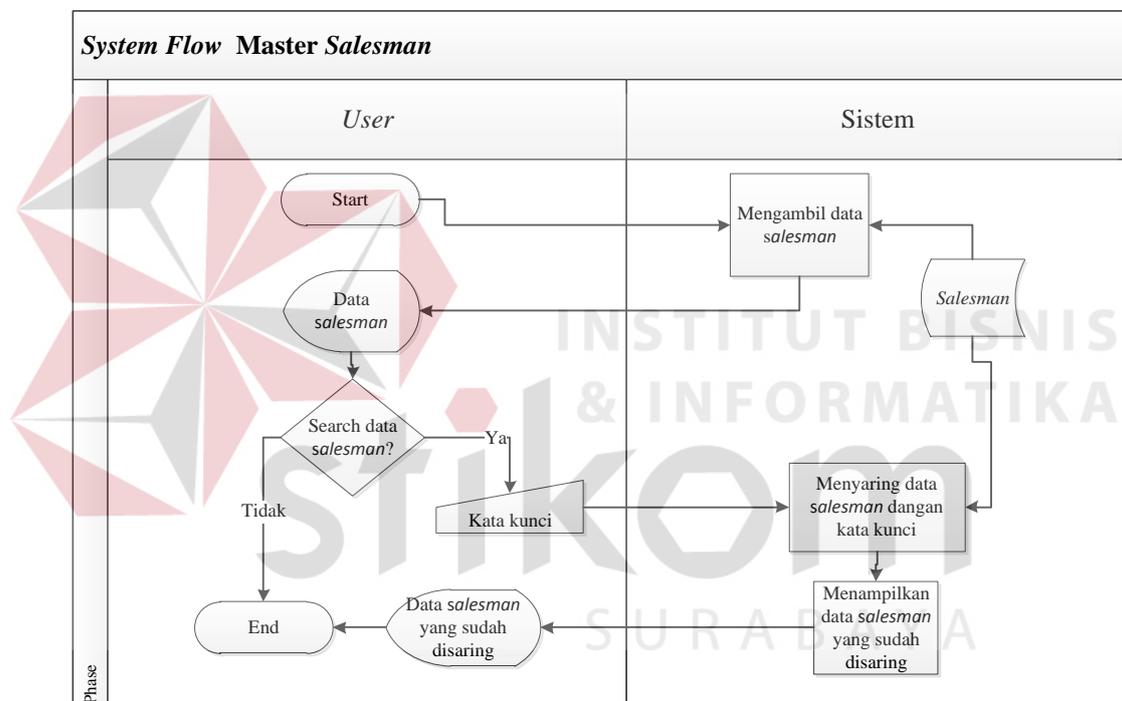
System flow master produk ini berguna untuk membaca data produk yang akan ditampilkan pada menu katalog. Data dari tabel produk pada *system flow* ini didapatkan dari proses sebelumnya, yaitu pada proses *system flow* sinkronisasi *database*. Pada master katalog ini dilengkapi dengan pencarian data sesuai dengan kata kunci yang diinginkan. Pada master produk ini juga terdapat penguploadan gambar produk. Berikut adalah gambaran dari *system flow* master katalog:



Gambar 3.10 System Flow Maintenance Master Produk

E. System Flow Master Salesman

System flow master *salesman* ini berguna untuk membaca data *salesman* yang akan ditampilkan pada menu *salesman*. Data dari tabel *salesman* pada *system flow* ini didapatkan dari proses sebelumnya, yaitu pada proses *system flow* aplikasi sinkronisasi *database*. Pada master *salesman* ini dilengkapi dengan pencarian data sesuai dengan kata kunci yang diinginkan. Berikut adalah gambaran dari *system flow* master *salesman*:

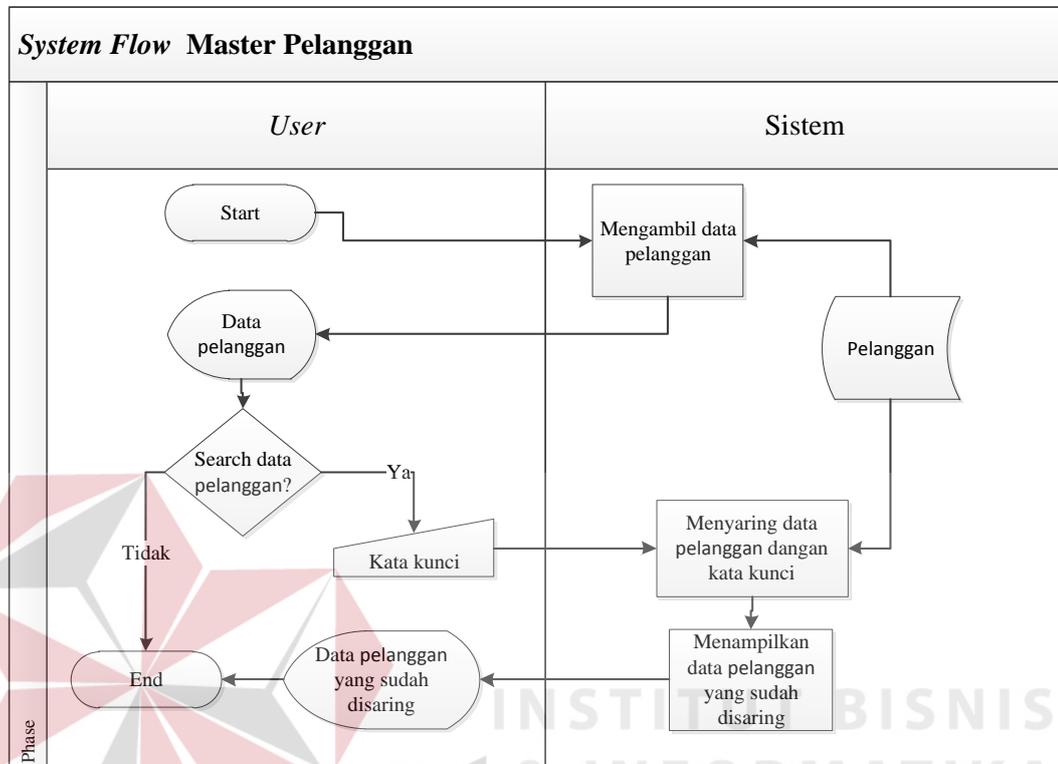


Gambar 3.11 System Flow Maintenance Master Salesman

F. System Flow Master Pelanggan

System flow master pelanggan ini berguna untuk membaca data pelanggan yang akan ditampilkan pada menu pelanggan. Data dari tabel pelanggan pada *system flow* ini didapatkan dari proses sebelumnya, yaitu pada proses *system flow* aplikasi penghubung pelanggan. Pada master pelanggan ini dilengkapi dengan

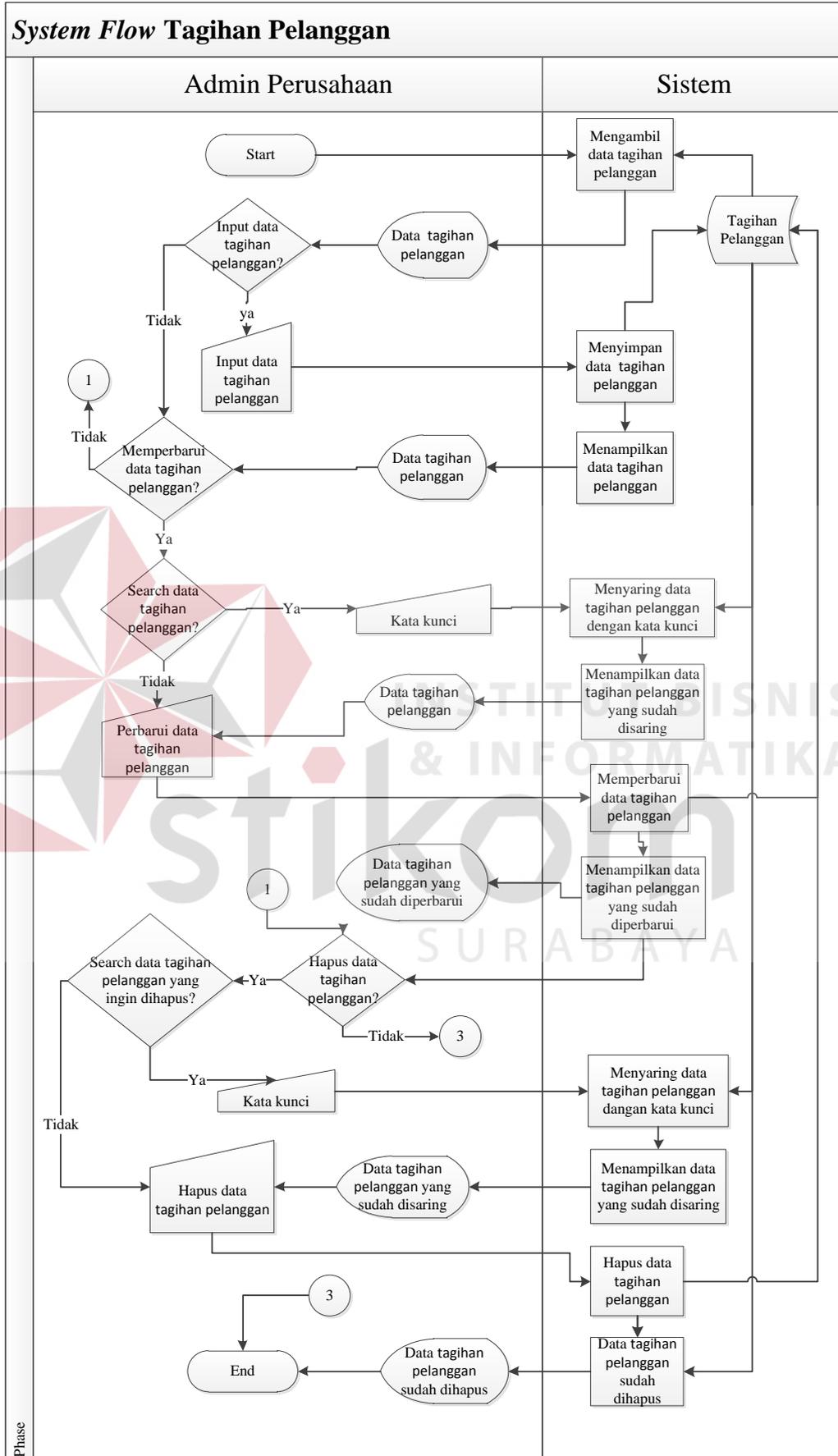
pencaharian data sesuai dengan kata kunci yang diinginkan. Berikut adalah gambaran dari *system flow* master pelanggan:



Gambar 3.12 *System Flow* Master Pelanggan

G. *System Flow* Master Tagihan Pelanggan

System flow master tagihan pelanggan ini berguna untuk membuat daftar tagihan pelanggan. Admin perusahaan yang akan membuat, memperbarui, menghapus, dan mencetak daftar tagihan pelanggan. Daftar tagihan pelanggan ini berguna untuk membantu *salesman* dalam proses menagih kepada pelanggan. Berikut adalah gambaran dari *system flow* master tagihan pelanggan:

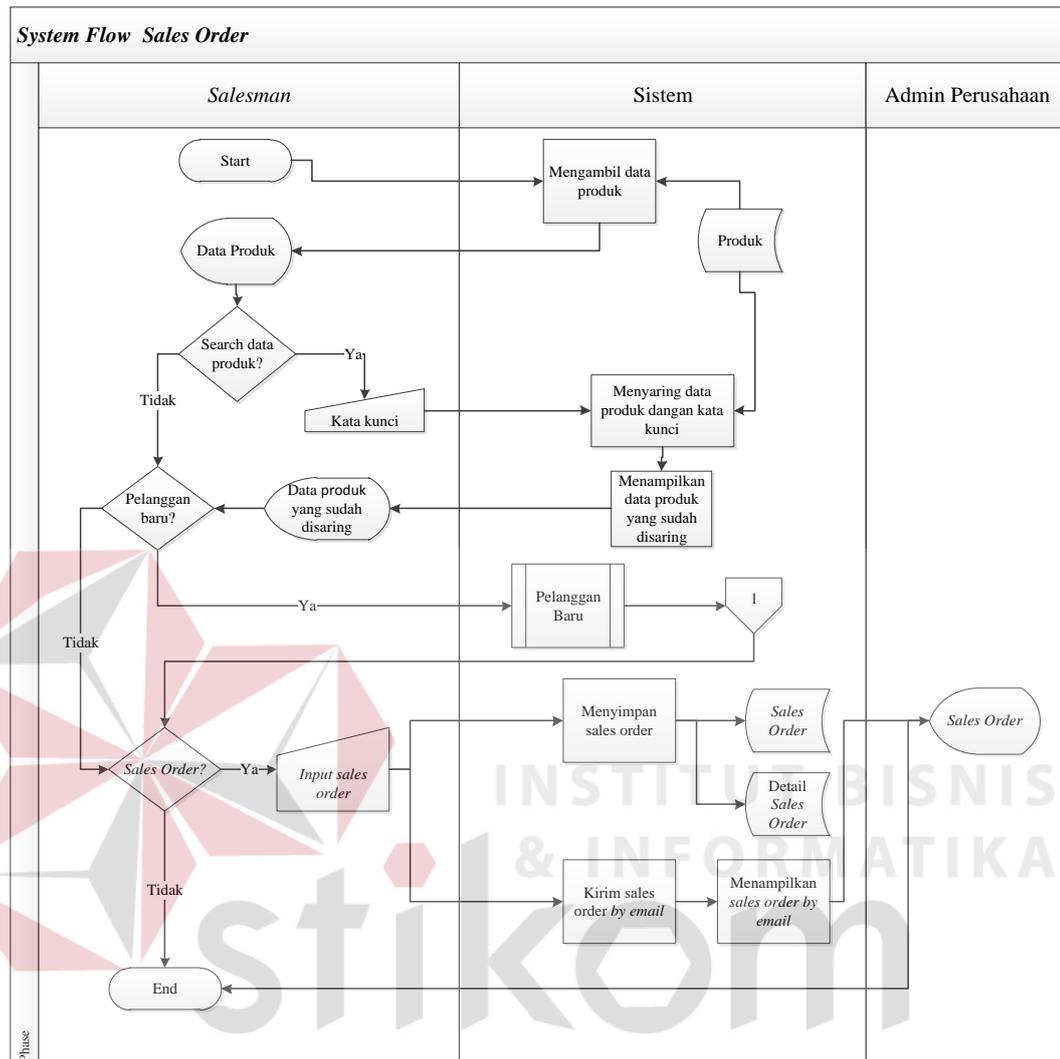


Gambar 3.13 System Flow Tagihan Pelanggan

H. System Flow Sales Order

System flow sales order ini berguna untuk membantu tugas operasional dari *salesman* dalam melakukan menawarkan produk dan membuat *sales order*. *Salesman* akan menawarkan produknya kepada pelanggan. *Salesman* dapat mencari data produk dengan memasukkan kata kunci yang berhubungan dengan data produk. Saat pelanggan menyetujui pembelian produk kepada *salesman*, maka tugas *salesman* adalah melakukan *sales order*. *Salesman* dapat membuat data pelanggan baru, yaitu pada sub-proses pelanggan baru. Sedangkan untuk pelanggan yang sudah terdaftar, proses *sales order* dapat dilakukan langsung tanpa perlu memasukan data pelanggan terlebih dahulu. Pengiriman *sales order* untuk pelanggan yang terdaftar juga sama dengan pelanggan baru, yaitu *sales order* akan dikirim melalui *email*. *Email* akan dikirimkan kepada admin perusahaan untuk dilakukan proses selanjutnya.

Proses *sales order* akan tetap dijalankan jika terdapat pelanggan baru, tetapi untuk dapat melakukan proses selanjutnya yaitu pengiriman barang tergantung oleh persetujuan owner dan terdapat kebijakan yang sudah diterapkan pada perusahaan CV. Dwi Star untuk menjadikan pelanggan tetap. Berikut adalah gambaran dari *system flow sales order*:



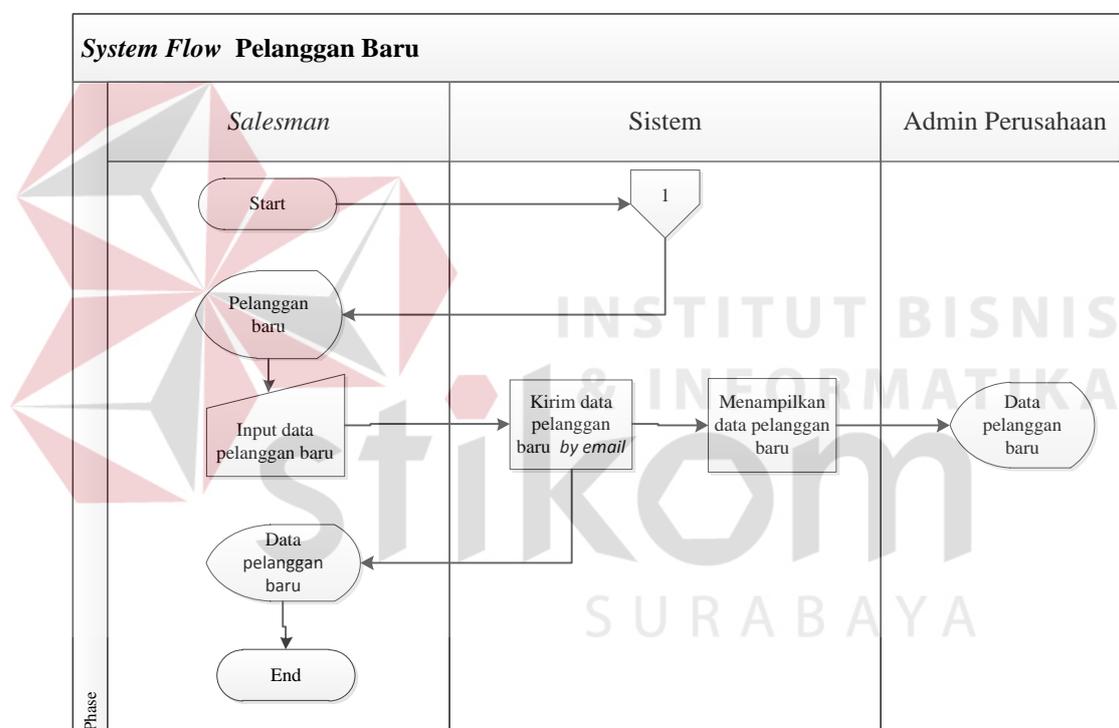
Gambar 3.14 System Flow Sales Order

I. System Flow Pelanggan Baru

System flow pelanggan baru ini berguna untuk membantu tugas operasional dari *salesman* dalam pembuatan *sales order* yang harus mencantumkan biodata pelanggan. Sehingga jika terdapat pelanggan baru, *salesman* dapat membuat pelanggan baru pada aplikasi operasional *salesman* dan *salesman* dapat melanjutkan proses *sales order*. Jika data pelanggan baru telah

dibuat maka data tersebut akan dikirimkan kepada admin perusahaan *by email*.

Pengiriman data pelanggan baru dilakukan secara *by email* dikarenakan dibutuhkannya persetujuan dari owner CV. Dwi Star dan dibutuhkannya pengecekan pelanggan baru. Proses sales order akan tetap dijalankan untuk pelanggan baru, tetapi untuk dapat melakukan proses selanjutnya yaitu pengiriman barang tergantung oleh persetujuan owner. Berikut adalah gambaran dari *system flow* pelanggan baru:

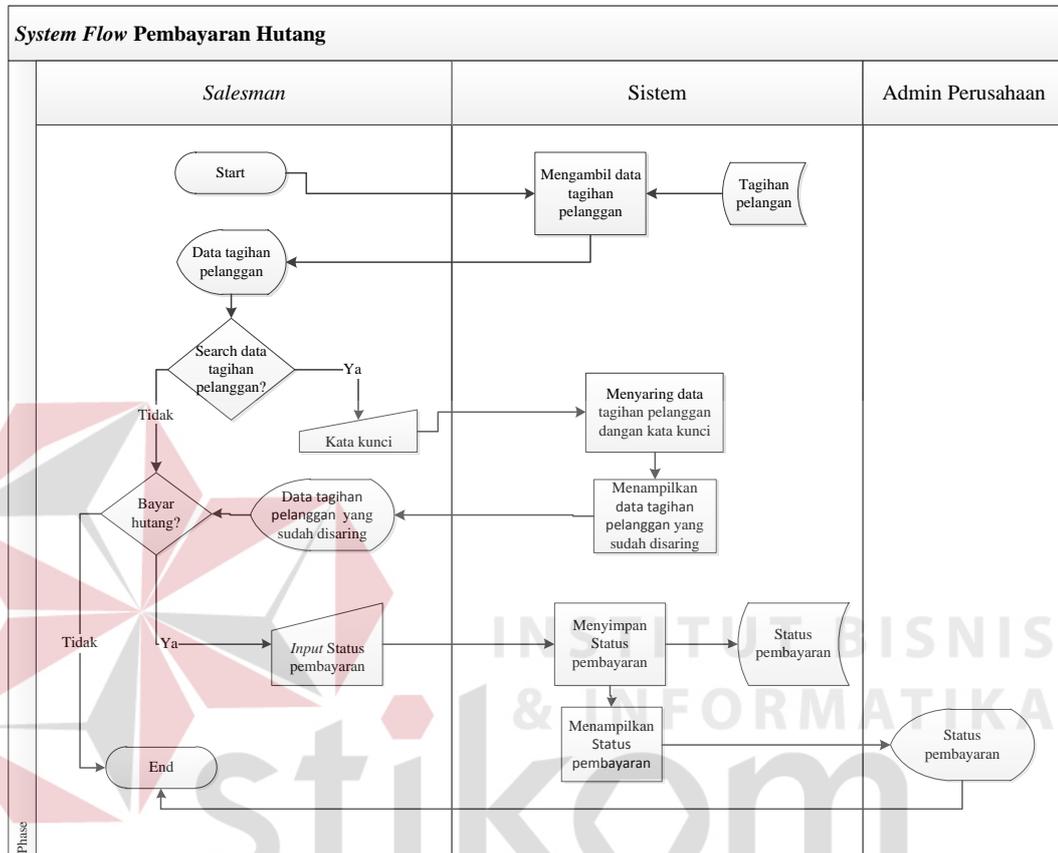


Gambar 3.15 *System Flow* Pelanggan Baru

J. *System Flow* Pembayaran Hutang

System flow pembayaran hutang ini berguna untuk membantu tugas operasional dari *salesman* dalam melakukan penagihan hutang kepada pelanggan. *Salesman* akan memperbarui status hutang. Status hutang adalah perubahan status

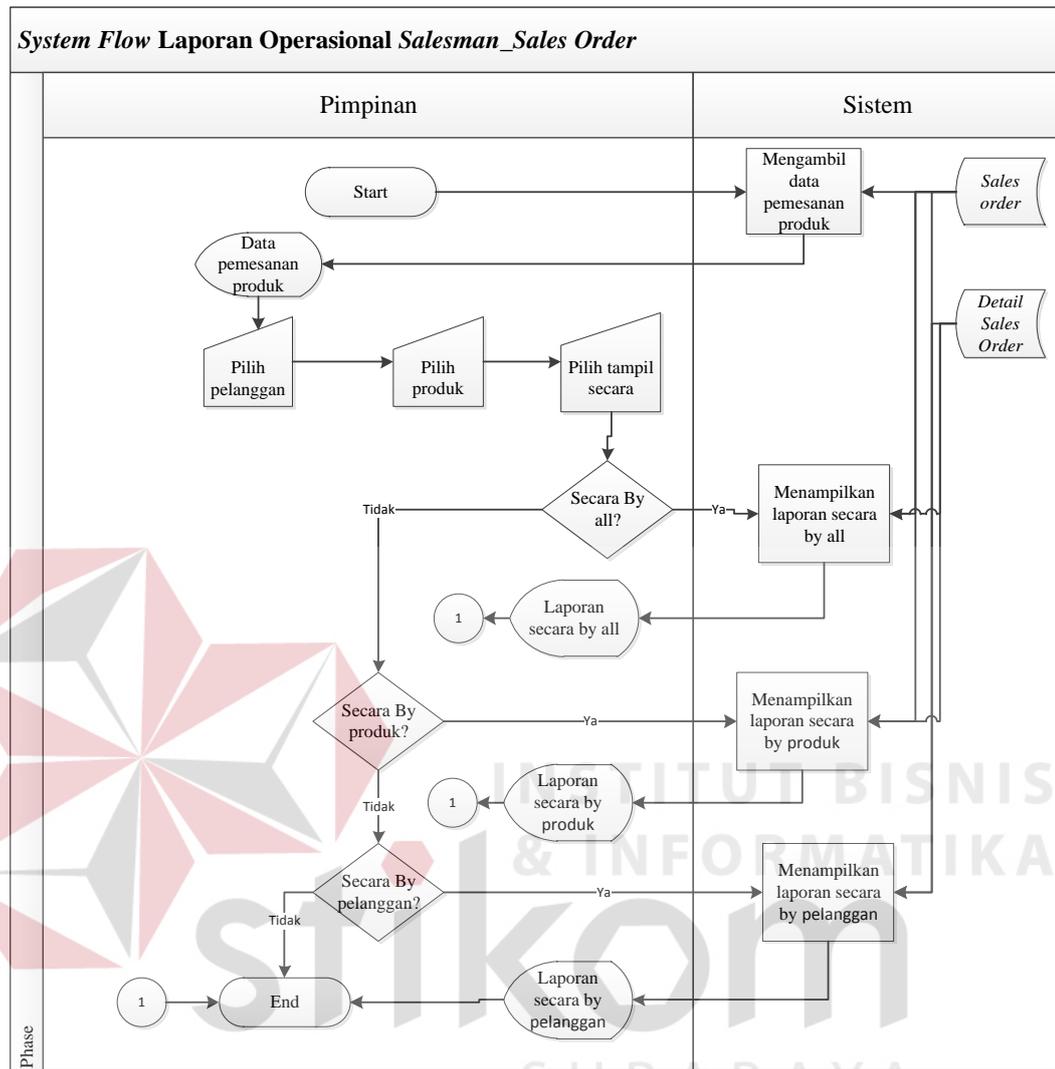
pelanggan yang membayar hutang atau tidak membayar hutang. Berikut adalah gambaran dari *system flow* pembayaran hutang:



Gambar 3.16 *System Flow* Penagihan Hutang

K. *System Flow* Laporan Operasional Salesman_Sales Order

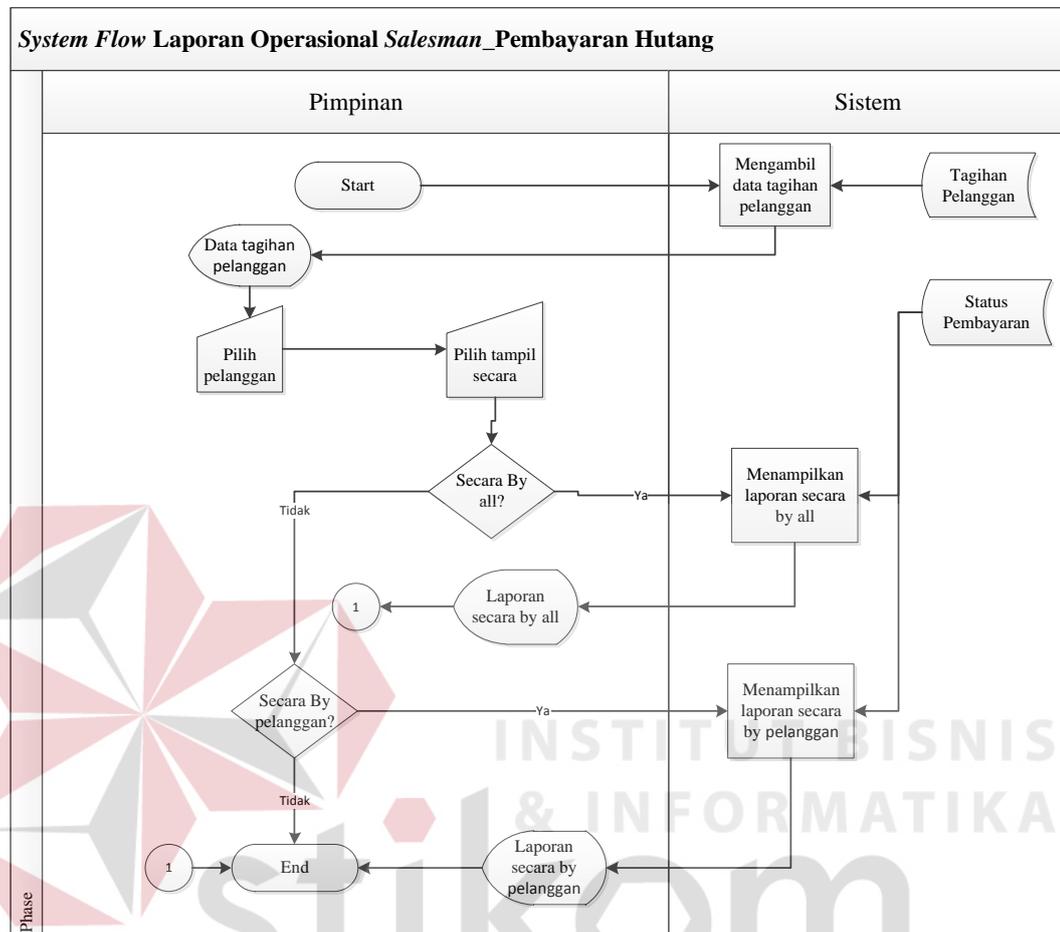
System flow laporan operasional *salesman* ini berguna untuk membantu *salesman* dan *owner* dalam melihat laporan operasional *salesman* dalam *sales order*. *Salesman* dan *owner* dapat memilih periode, nama pelanggan, dan nama produk untuk melihat laporan *sales order*. Berikut adalah gambaran dari *system flow* laporan operasional *salesman sales order*:



Gambar 3.17 System Flow Laporan Operasional Salesman_Sales Order

L. System Flow Laporan Operasional Salesman_Pembayaran Hutang

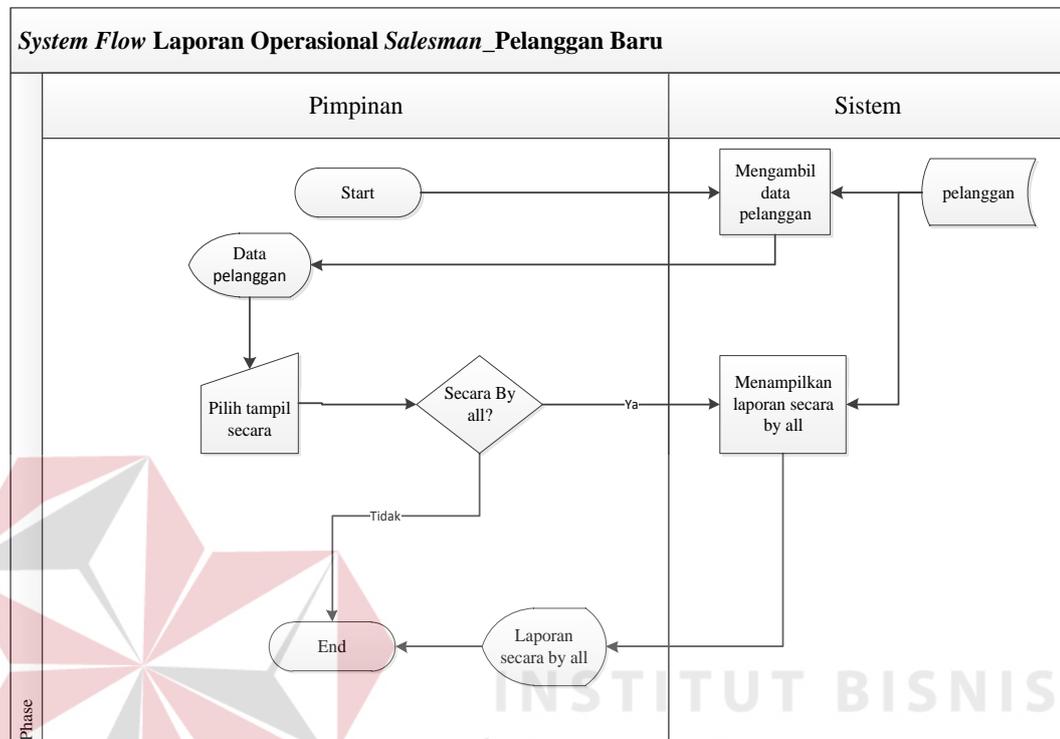
System flow laporan operasional salesman ini berguna untuk membantu salesman dan owner dalam melihat laporan operasional salesman dalam pembayaran hutang. Salesman dan owner dapat memilih periode dan nama pelanggan untuk melihat laporan pembayaran hutang. Berikut adalah gambaran dari system flow laporan operasional salesman pembayaran hutang:



Gambar 3.18 System Flow Laporan Operasional Salesman_Pembayaran Hutang

M. System Flow Laporan Operasional Salesman_Pelanggan Baru

System flow laporan operasional salesman ini berguna untuk membantu salesman dan owner dalam melihat laporan operasional salesman dalam pelanggan baru. Salesman dan owner dapat memilih periode untuk melihat laporan pelanggan baru. Berikut adalah gambaran dari system flow laporan operasional salesman pelanggan baru:



Gambar 3.19 System Flow Laporan Operasional Salesman_Pelanggan Baru

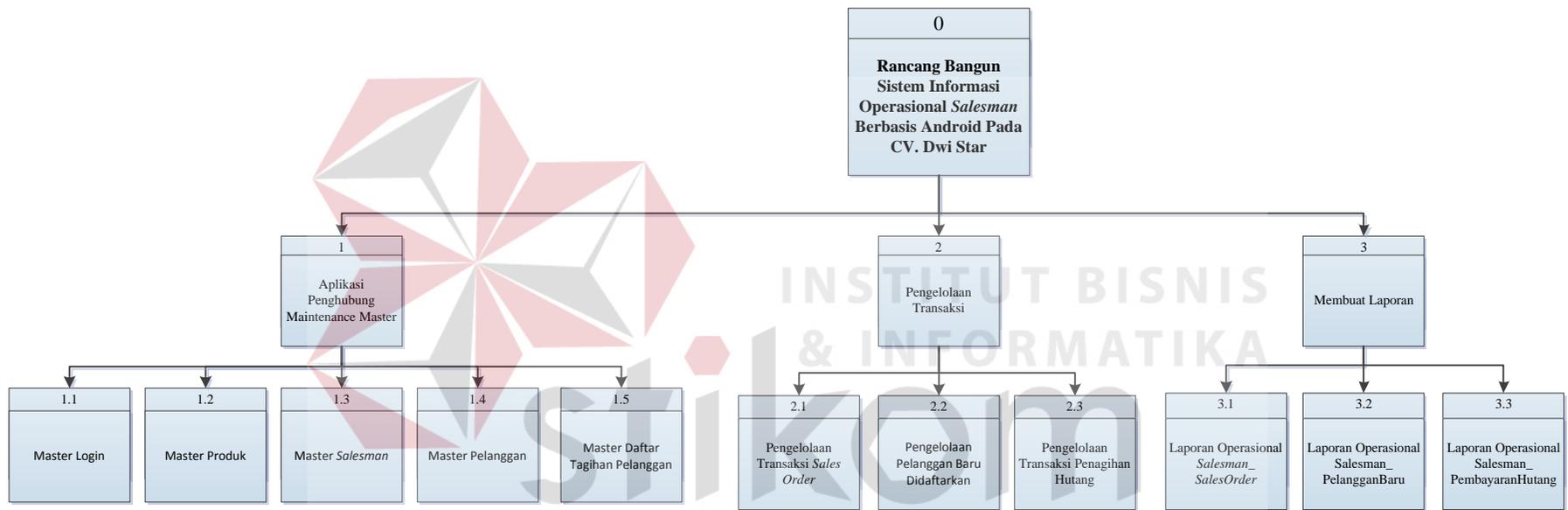
3.2.4 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan alur dari data sistem. DFD berguna untuk membantu memahami alur data sistem secara tersruktur dan jelas.

Berikut adalah alur data dari DFD:

A. Diagram Berjenjang

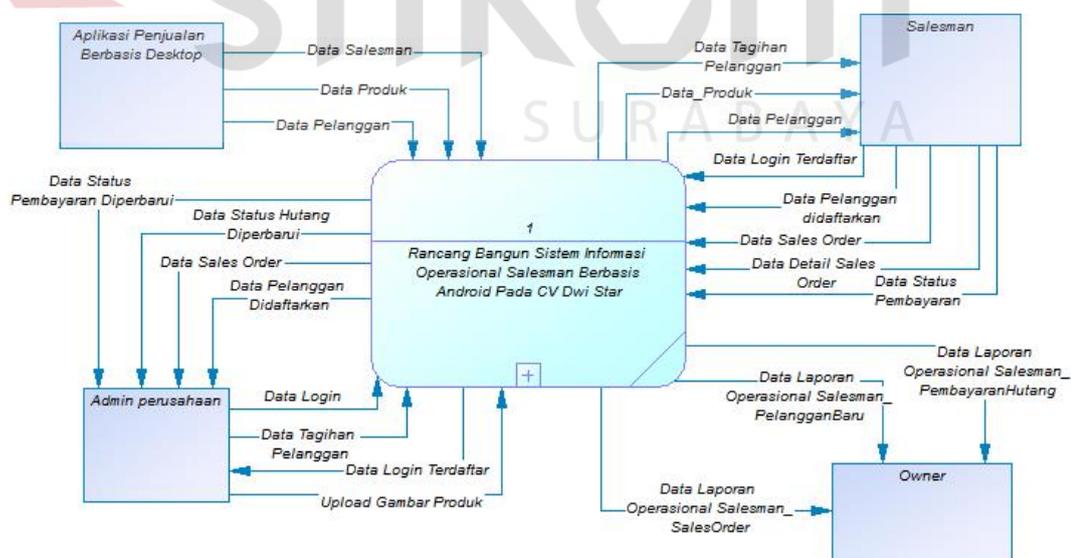
Berikut adalah diagram berjenjang dari Rancang Bangun Sistem informasi Operasional Salesman Berbasis Android:



Gambar 3.20 Diagram Berjenjang Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android

B. Context Diagram

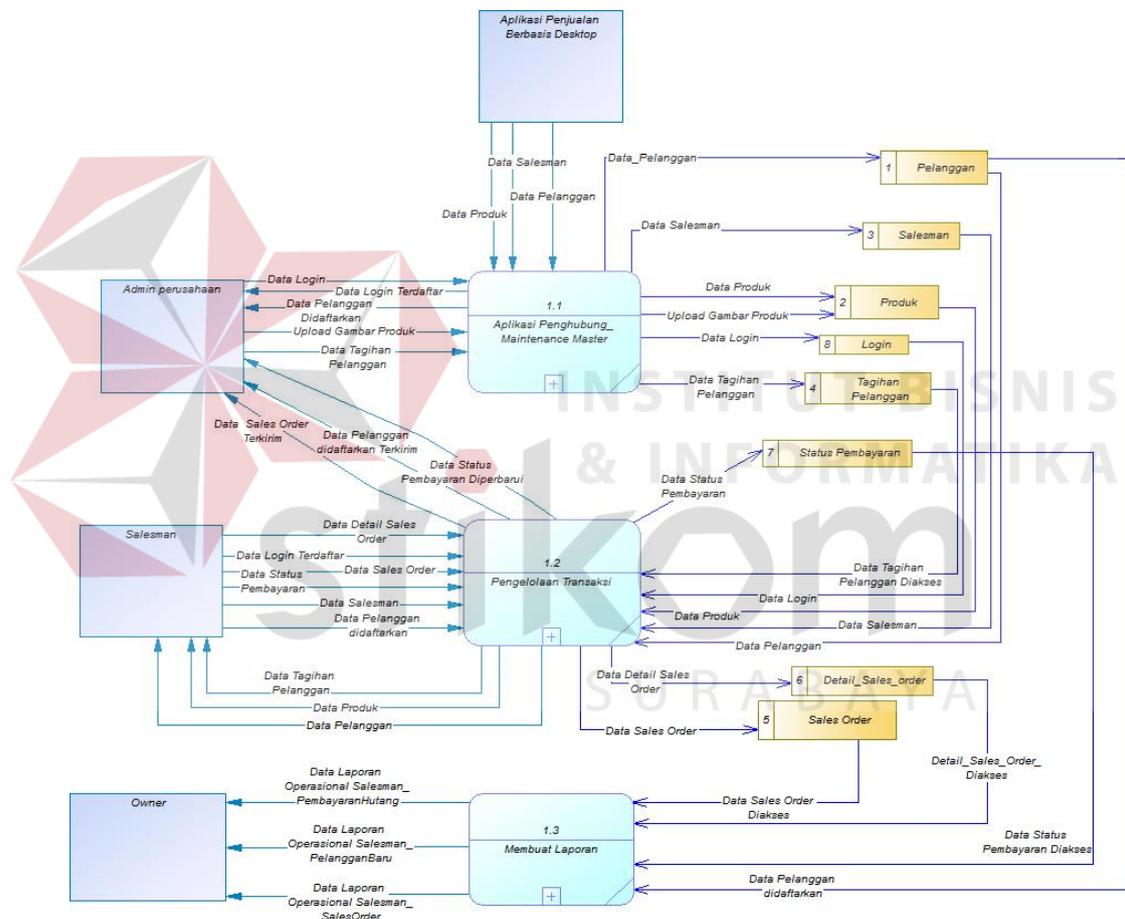
Context diagram pada sistem ini memiliki empat entitas, yaitu Aplikasi penjualan berbasis *desktop*, Admin perusahaan, *Salesman*, dan Owner. Aplikasi penjualan berbasis *desktop* merupakan aplikasi di luar area tugas akhir ini atau aplikasi yang sudah ada pada perusahaan CV. Dwi Star. Aplikasi penjualan berbasis *desktop* ini berguna untuk memberikan inputan data pelanggan, data produk, dan data pelanggan. Semua aliran data diinputkan ke dalam sistem Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android, kemudian sistem mengeluarkan respon yaitu berupa laporan operasional *salesman sales order*, laporan operasional *salesman* pelanggan baru, dan laporan operasional *salesman* pembayaran hutang. Berikut adalah gambaran dari context diagram Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android:



Gambar 3.21 Context Diagram Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android

C. DFD Level 0

DFD level 0 ini menjelaskan aliran data sistem dari tugas proses, yaitu *maintenance* master (Aplikasi Penghubung), pengelolaan transaksi, dan membuat laporan. *Maintenance* master adalah aplikasi penghubung berbasis *desktop* yang berguna untuk mengelolah data master. Berikut adalah gambaran dari DFD Level 0 Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android:

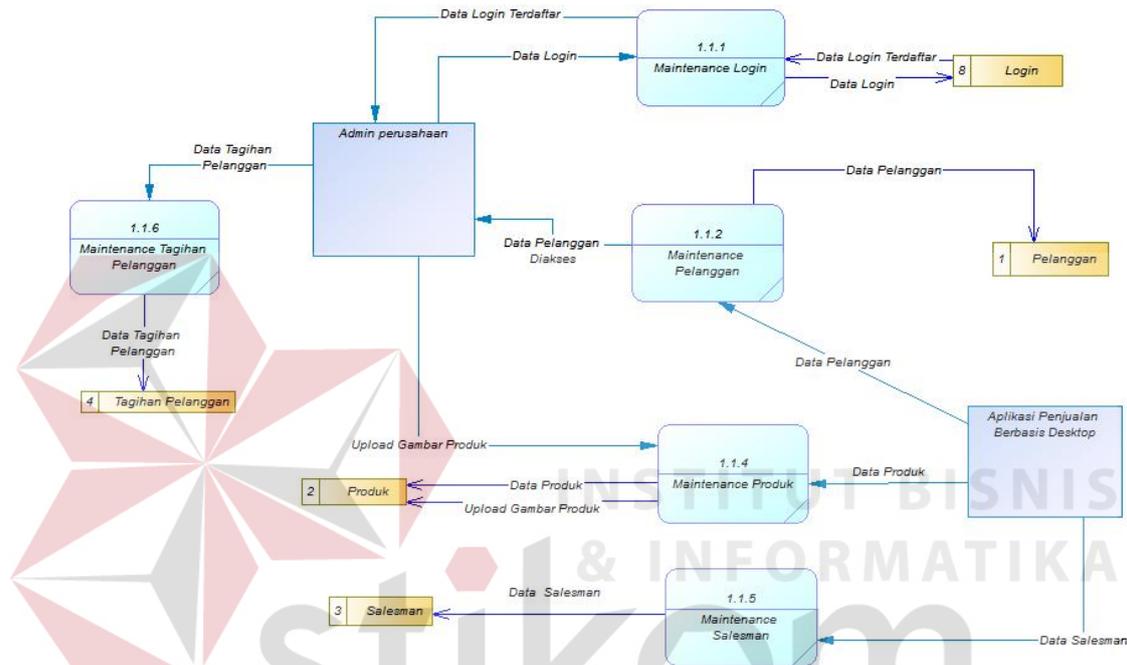


Gambar 3.22 DFD Level 0 Rancang Bangun Sistem Informasi Operasional *Salesman* Berbasis Android

D. DFD Level 1 *Maintenance* Master (Aplikasi Penghubung)

DFD level 1 *maintenance* master (Aplikasi Penghubung) ini menjelaskan aliran data sistem dari master proses, yaitu *maintenance* master login,

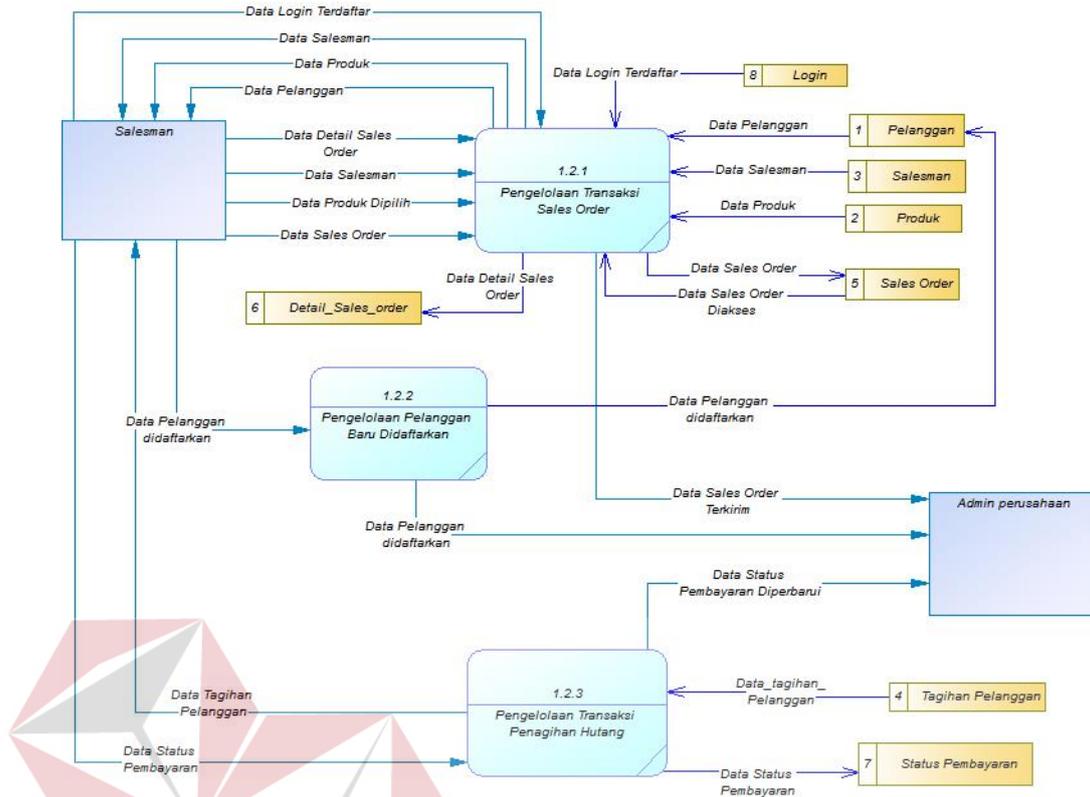
maintenance master pelanggan, *maintenance* master tagihan pelanggan, *maintenance* master produk, dan *maintenance* master salesman. *Maintenance* master adalah aplikasi penghubung berbasis *desktop* yang berguna untuk mengelola data master. Berikut adalah gambaran dari DFD Level 1 *maintenance* master:



Gambar 3.23 DFD Level 1 Maintenance Master

E. DFD Level 1 Pengelolaan Transaksi

DFD level 1 pengelolaan transaksi ini menjelaskan aliran data sistem dari proses pengelolaan transaksi, yaitu pengelolaan transaksi *sales order* dan pengelolaan transaksi penagihan hutang. Berikut adalah gambaran dari DFD Level 1 pengelolaan transaksi:



Gambar 3.24 DFD Level 1 Pengelolaan Tarnaksi

F. DFD Level 1 Membuat Laporan

DFD level 1 membuat laporan ini menjelaskan aliran data sistem dari proses pembuatan laporan, yaitu laporan operasional *salesman*. Berikut adalah gambaran dari DFD Level 1 membuat laporan:



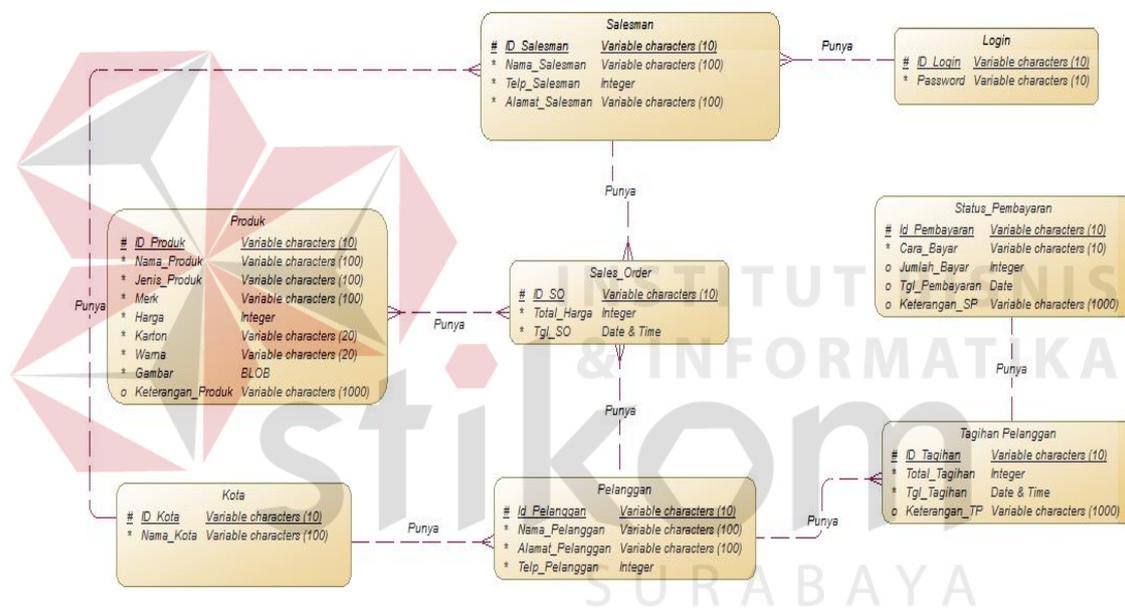
Gambar 3.25 DFD Level 1 Membuat Laporan

3.2.5 Entity Relationship Diagram (ERD)

Rancangan *database* sistem yang dibuat berupa *Entity Relational Diagram* (ERD), yaitu alat untuk merepresentasikan model data yang ada pada sistem dimana terdapat *entity* dan *relationship*.

A. Conceptual Data Model (CDM)

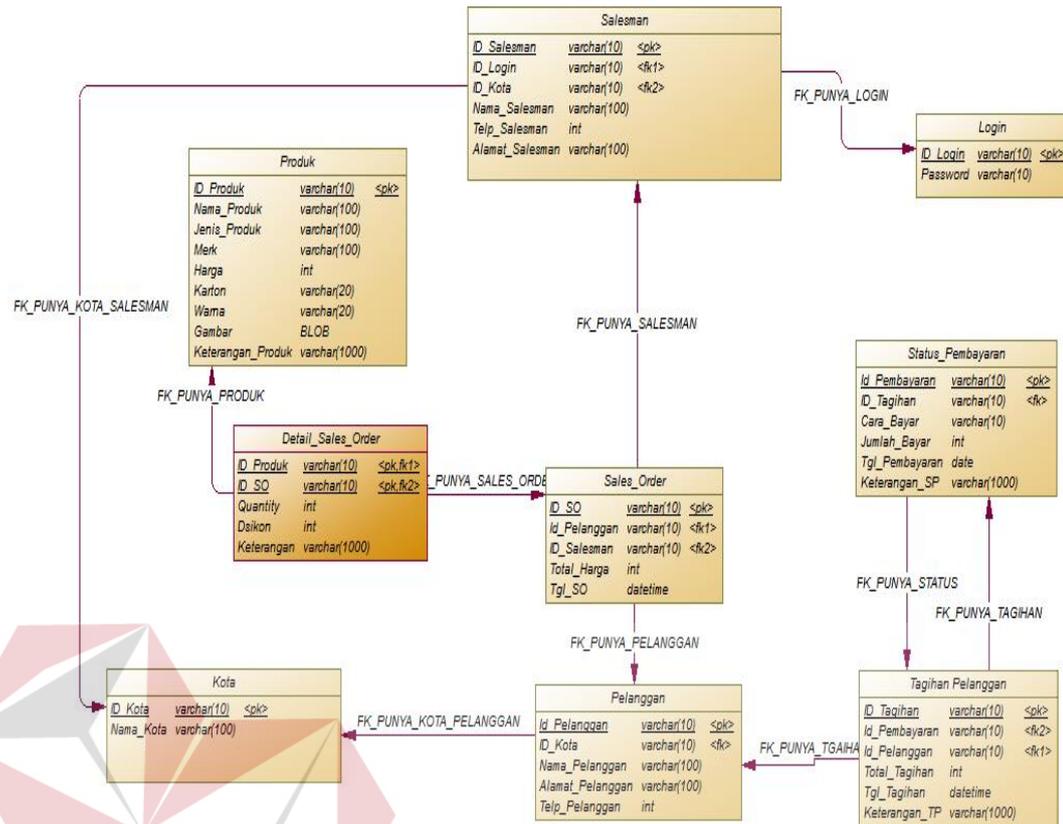
Berikut adalah gambaran dari *Conceptual Data Model* (CDM) untuk aplikasi operasional *salesman*:



Gambar 3.26 *Conceptual Data Model* (CDM)

B. Physical Data Model (PDM)

Berikut adalah gambaran dari *Physical Data Model* (PDM) untuk aplikasi operasional *salesman*:



Gambar 3.27 Physical Data Model (PDM)

C. Struktur Tabel

Dalam hal perancangan struktur tabel yang diperlukan, maka perlu dibuat atribut meliputi nama tabel, nama atribut, tipe data, serta data pelengkap seperti *primary key*, *foreign key*, dan sebagainya. Berikut adalah rancangan basis data aplikasi yang terdiri dari tabel dibawah ini:

- Nama tabel : *Salesman*

Fungsi : Menyimpan data *salesman*

Primary key : ID_Salesman

Foreign key : ID_Login
ID_Kota

Tabel 3.3 Salesman

Nama Field	Type Data	Lebar Field	Keterangan
ID_Salesman	<i>Varchar</i>	10	Id Salesman
ID_Login	<i>Varchar</i>	10	Id Login
ID_Kota	<i>Varchar</i>	10	Id Kota
Nama_Salesman	<i>Varchar</i>	100	Nama salesman
Telp_Salesman	<i>Integer</i>		No Telepon salesman
Alamat_Salesman	<i>Varchar</i>	100	Alamat Salesman

2. Nama Tabel :Login

Fungsi : Menyimpan data login

Primary key : ID_Login

Foreign key : -

Tabel 3.4 Login

Nama Field	Type Data	Lebar Field	Keterangan
ID_Login	<i>Varchar</i>	10	Id Login
Password	<i>Varchar</i>	10	Password

3. Nama Tabel : Produk

Fungsi : Menyimpan data produk

Primary key : ID_Produk

Foreign key : -

Tabel 3.5 Produk

Nama Field	Type Data	Lebar Field	Keterangan
ID_Produk	<i>Varchar</i>	10	Id Produk
Nama_Produk	<i>Varchar</i>	100	Nama produk
Jenis_Produk	<i>Varchar</i>	100	Jenis produk
Merk	<i>Varchar</i>	100	Merk produk
Harga	<i>Integer</i>		Harga Produk
Karton	<i>Varchar</i>	20	Karton Produk
Warna	<i>Varchar</i>	20	Karton Produk
Gambar	<i>BLOB</i>		Gambar Produk
Keterangan_Produk	<i>Varchar</i>	1000	Keterangan Produk

4. Nama Tabel : Detail SO

Fungsi : Menyimpan detail *Sales Order*

Primary key : -

Foreign key : ID_Produk

ID_SO

Tabel 3.6 Detail SO

Nama Field	Type Data	Lebar Field	Keterangan
ID_Produk	<i>Varchar</i>	10	Id Produk
ID_SO	<i>Varchar</i>	10	Id Sales order
Quantity	<i>Integer</i>		Kuantitas produk
Diskon	<i>Integer</i>		Diskon produk
Keterangan	<i>Varchar</i>	1000	Keterangan <i>sales order</i>

5. Nama Tabel : Sales Order

Fungsi : Menyimpan *Sales Order*

Primary key : ID_SO

Foreign key : ID_Pelanggan

ID_Salesman

Tabel 3.7 Sales Order

Nama Field	Type Data	Lebar Field	Keterangan
ID_SO	<i>Varchar</i>	10	Id Sales order
ID_Pelanggan	<i>Varchar</i>	10	Id pelanggan
ID_Salesman	<i>Varchar</i>	10	Id <i>salesman</i>
Total_Harga	<i>Integer</i>		Harga produk
Tgl_SO	<i>Datetime</i>		Tanggal <i>sales order</i>

6. Nama Tabel : Pelanggan

Fungsi : Menyimpan data pelanggan

Primary key : Id_Pelanggan

Foreign key : ID_Kota

Tabel 3.8 Pelanggan

Nama Field	Type Data	Lebar Field	Keterangan
ID_Pelanggan	<i>Varchar</i>	10	Id pelanggan
ID_Kota	<i>Varchar</i>	10	Id kota
Nama_Pelanggan	<i>Varchar</i>	100	Nama pelanggan
Alamat_Pelanggan	<i>Varchar</i>	100	Alamat pelanggan
Telp_Pelanggan	<i>Integer</i>		No Telepon pelanggan

7. Nama Tabel : Tagihan Pelanggan

Fungsi : Menyimpan data tagihan pelanggan.

Primary key : ID_Tagihan

Foreign key : ID_Pelanggan

ID_Pembayaran

Tabel 3.9 Tagihan Pelanggan

Nama Field	Type Data	Lebar Field	Keterangan
ID_Tagihan	<i>Varchar</i>	10	Id tagihan pelanggan
ID_Pembayaran	<i>Varchar</i>	10	Id pembayaran
ID_Pelanggan	<i>Varchar</i>	10	Id pelanggan
Total_Tagihan	<i>Integer</i>		Total tagihan
Ketereangan_TP	<i>Varchar</i>	1000	Keterangan tagihan pelanggan

8. Nama Tabel : Status Pembayaran

Fungsi : Menyimpan status pembayaran

Primary key : Id_Pembayaran

Foreign key : ID_Tagihan

Tabel 3.10 Status Pembayaran

Nama Field	Type Data	Lebar Field	Keterangan
Id_Pembayaran	<i>Varchar</i>	10	Id pembayaran
ID_Tagihan	<i>Varchar</i>	10	Id tagihan pelanggan
Cara_Bayar	<i>Varchar</i>	10	Cara pembayaran

Nama Field	Tipe Data	Lebar Field	Keterangan
Jumlah_Bayar	<i>Integer</i>		Uang yang harus dibayar
Tgl_Pembayaran	<i>Date</i>		Tanggal pembayaran
Keterangan_SP	<i>Varchar</i>	1000	Keterangan Pembayaran

9. Nama Tabel : Kota

Fungsi : Menyimpan kota

Primary key : ID_Kota

Foreign key :-

Tabel 3.11 Kota

Nama Field	Tipe Data	Lebar Field	Keterangan
ID_Kota	<i>Varchar</i>	10	Id kota
Nama_Kota	<i>Varchar</i>	100	Nama kota

3.2.6 Desain Input-Output

Desain *input-output* merupakan sebuah rancangan berupa *form* untuk memasukkan data dan laporan sebagai informasi yang dihasilkan dari pengolahan data. Desain *input-output* juga merupakan acuan pembuat aplikasi dalam merancang dan membangun sistem. Desain *input* ini dibedakan menjadi dua bagian, yaitu pada desain *input* pengelolaan master didesain dengan basis *desktop* yang ditujukan untuk admin sedangkan desain *input* transaksi didesain dengan basis *smartphone* (android) ditujukan kepada *salesman*.

1. Desain Input

Desain *input* merupakan perancangan desain masukan dari pengguna kepada sebuah sistem yang kemudian akan disesuaikan dengan hak akses yang

sudah ditentukan sebelumnya, sehingga data yang diakses bisa dikontrol dengan baik.

A. Desain Login



The screenshot shows a mobile application interface titled "Aplikasi Operasional Salesman". Inside the app window, there is a "LOGIN" section. It contains two text input fields: "ID:" and "Password:", both with "Enter Text" placeholder text. Below the input fields is a "Login" button.

Gambar 3.28 Desain Login

Desain Login disini berfungsi sebagai pengaksesan melalui android *mobile* dengan memasukkan ID dan *password* yang sudah ditentukan sebelumnya guna mendapatkan hak akses.

B. Desain Sinkronisasi Database



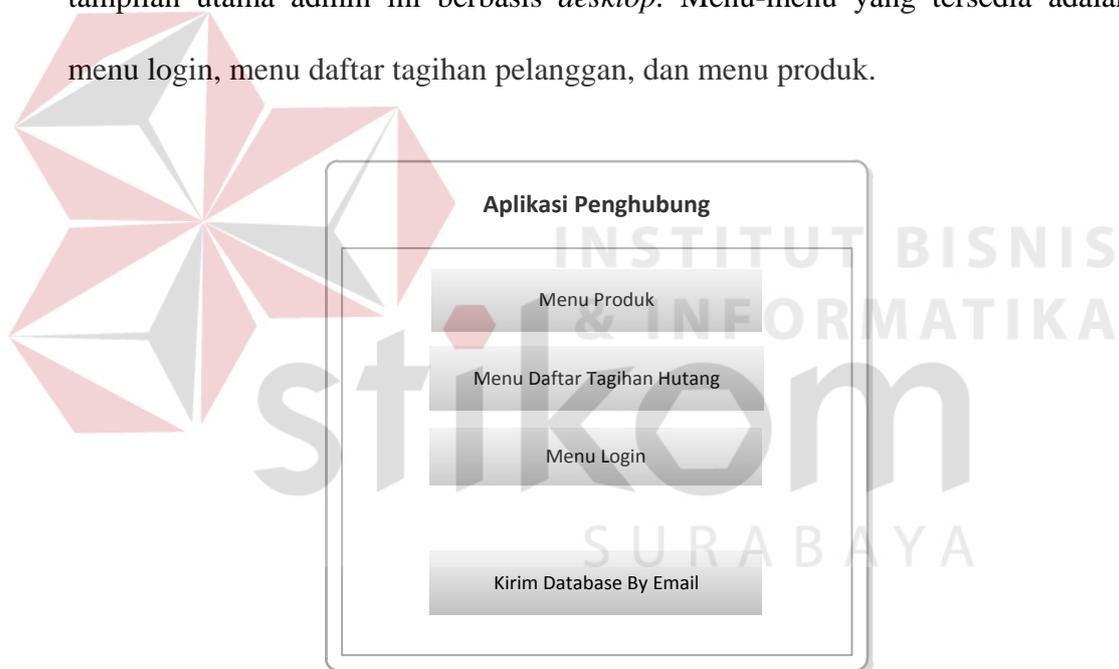
The screenshot shows a mobile application interface titled "Aplikasi Penghubung". It displays a message: "Silakan Sinkronisasi Database Terlebih Dahulu !". Below the message is a "Sinkronisasi Database" button.

Gambar 3.29 Desain Sinkronisasi Database

Desain sinkronisasi *database* ini berfungsi sebagai pensinkronan *database* Sql Server menjadi database SQLite. Desain sinkronisasi *database* digunakan oleh admin perusahaan. Sebelum admin perusahaan mengakses aplikasi penghubung, admin harus memilih *button* sinkronisasi *database* terlebih dahulu.

C. Desain Tampilan Utama Admin

Desain tampilan utama admin disini berfungsi sebagai penyedia menu khusus admin yang jika dipilih maka menu tersebut akan diproses. Desain tampilan utama admin ini berbasis *desktop*. Menu-menu yang tersedia adalah menu login, menu daftar tagihan pelanggan, dan menu produk.



Gambar 3.30 Desain Tampilan Utama Admin

D. Desain Menu Login

Desain menu login disini berfungsi sebagai untuk menampilkan data login yang telah diinputkan. Pada desain menu login ini juga terdapat fungsi simpan, ubah, hapus, dan pencarian. Menu ini dapat dilakukan oleh admin perusahaan.

Aplikasi Penghubung

Menu Login

ID:

Password:

Nama:

ID	Password	Nama
ID001	xxxxx	Joko
.....

Gambar 3.31 Desain Menu Login

E. Desain Mater *Salesman*

Aplikasi Penghubung

Menu *Salesman*

ID	Nama	Telp	Alamat
ID001	Joko	081123	Manyar
.....

Gambar 3.32 Desain Mater *Salesman*

Desain master *salesman* ini berfungsi sebagai untuk menampilkan data master *salesman* yang telah diinputkan pada aplikasi perusahaan. Pada desain

master *salesman* hanya terdapat fungsi pencarian. Menu ini dapat dilakukan oleh admin perusahaan.

F. Desain Master Pelanggan

Desain master pelanggan ini berfungsi sebagai untuk menampilkan data master pelanggan yang telah diinputkan pada aplikasi perusahaan Pada desain master *salesman* hanya terdapat fungsi pencarian. Menu ini dapat dilakukan oleh admin perusahaan.



Gambar 3.33 Desain Menu Pelanggan

G. Desain Menu Produk

Desain menu produk ini berfungsi sebagai untuk menampilkan data master produk yang telah diinputkan diinputkan pada aplikasi perusahaan Pada desain master *salesman* hanya terdapat fungsi pencarian dan penambahan gambar. Menu ini dapat dilakukan oleh admin perusahaan. Berikut adalah tampilannya:

Aplikasi Penghubung

Menu Produk

Gambar

Gambar Produk

Browse

ID:

Nama:

Jenis Produk:

Merk:

Harga:

Keterangan:

ID	Nama	Jenis	Merk	Harga
ID001	Cash box	Kotak	Eagle	RP..
.....

Gambar 3.34 Desain Menu Produk

H. Desain Menu Daftar Tagihan Pelanggan

Desain menu daftar tagihan pelanggan ini berfungsi sebagai untuk menampilkan data master menu daftar tagihan pelanggan yang telah diinputkan. Pada desain master daftar tagihan ini juga terdapat fungsi simpan, ubah, hapus, dan pencarian. Menu ini dapat dilakukan oleh admin perusahaan. Admin perusahaan dapat menambahkan data id, nama, total tagihan, tanggal tagihan, dan keterangan daftar tagihan pelanggan. Berikut adalah tampilannya:

Aplikasi Penghubung

Menu Daftar Tagihan Pelanggan

ID:

Nama Pelanggan:

Total Tagihan:

Tgl Tagihan:

Keterangan:

ID	Nama	Harga	Tgl	Harga
ID001	Budi	Rp..	15/7/15	RP..
.....

Gambar 3.35 Desain Menu Daftar Tagihan Pelanggan

I. Desain Tampilan Utama *Salesman*

Aplikasi Operasional *Salesman*

Menu Utama *Salesman*

Selamat Datang *Salesman* : Nama_Salesman

Katalog Produk

Tagih Hutang

Tambah Pelanggan Baru

Informasi Operasional *Salesman*

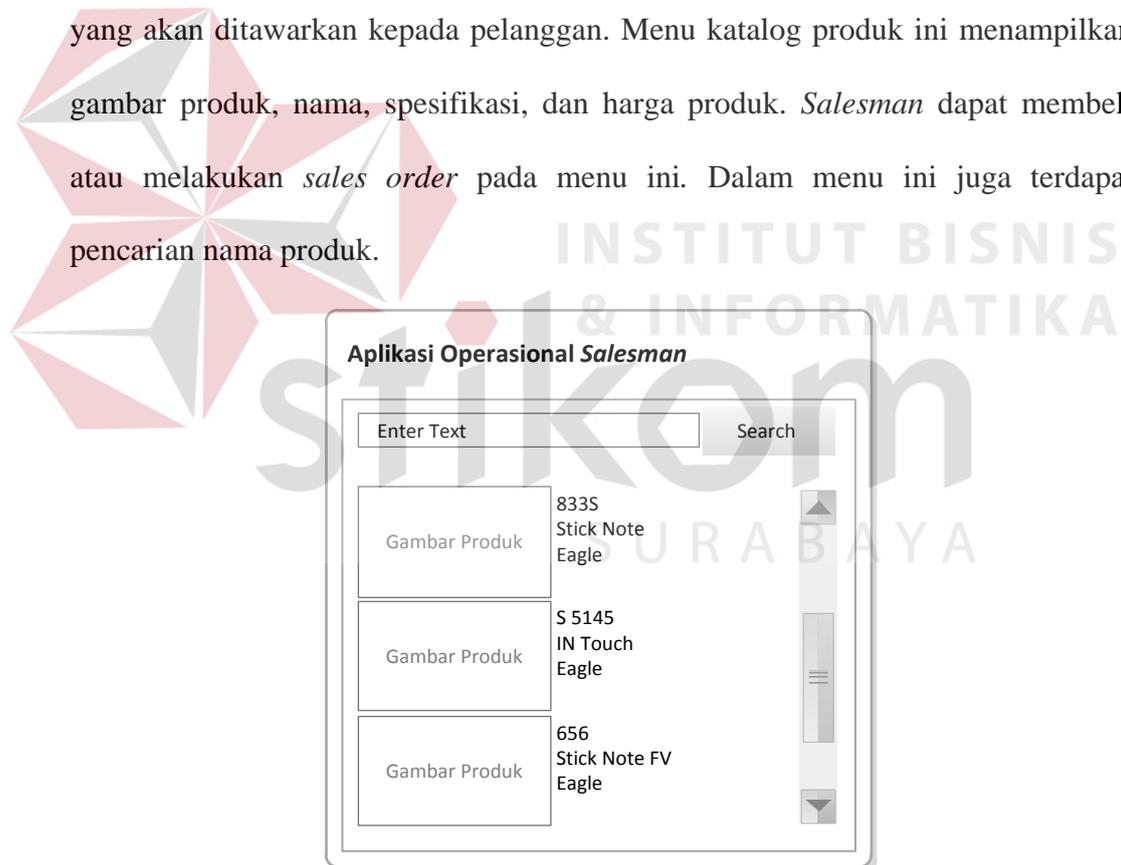
Keluar

Gambar 3.36 Desain Tampilan Utama *Salesman*

Desain Tampilan Utama *Salesman* disini berfungsi untuk menampilkan menu katalog produk, penagihan hutang, tambah pelanggan baru, dan informasi operasional *salesman*. Menu informasi operasional *salesman* menjelaskan mengenai kegiatan yang telah dilakukan oleh *salesman*. Menu Tampilan Utama *Salesman* ini diperuntukan bagi *salesman*. *Salesman* akan memilih menu-menu tersebut.

J. Desain Katalog Produk

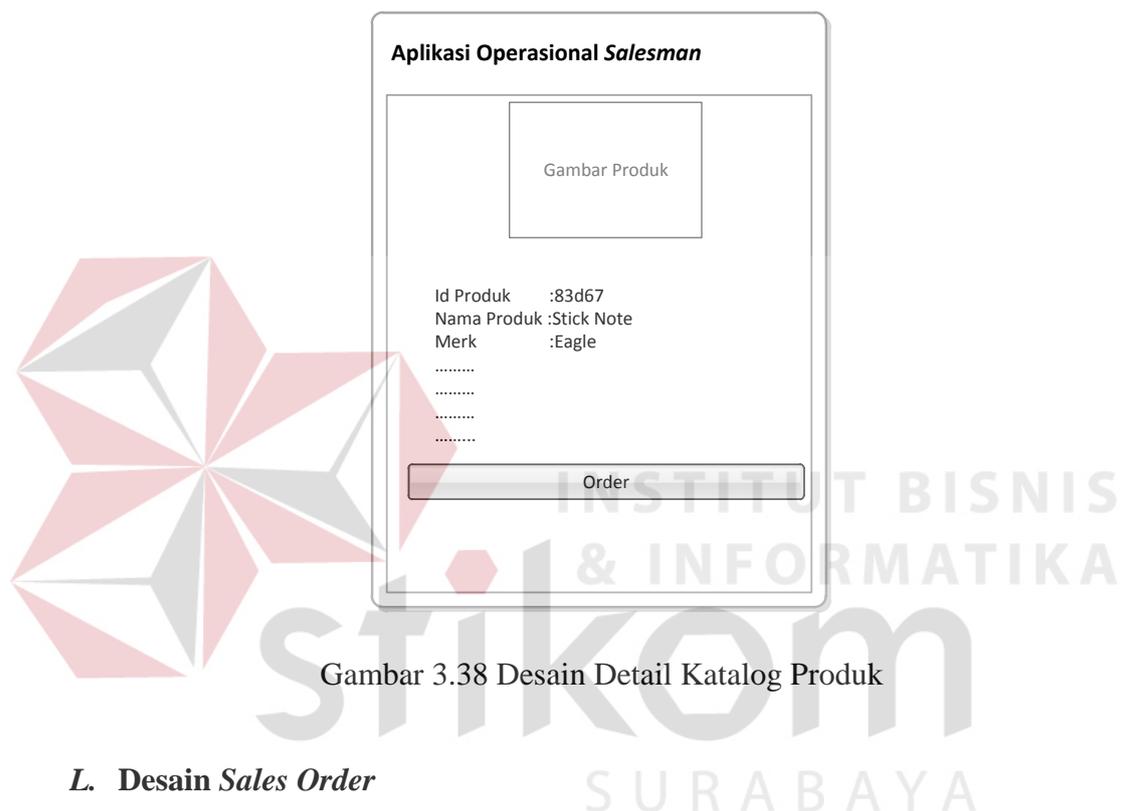
Desain katalog produk ini berfungsi untuk menampilkan produk-produk yang akan ditawarkan kepada pelanggan. Menu katalog produk ini menampilkan gambar produk, nama, spesifikasi, dan harga produk. *Salesman* dapat membeli atau melakukan *sales order* pada menu ini. Dalam menu ini juga terdapat pencarian nama produk.



Gambar 3.37 Desain Katalog Produk

K. Desain Detail Katalog Produk

Desain detail katalog produk ini berfungsi untuk menampilkan detail dari produk. Jika *salesman* memilih salah satu gambar produk, maka akan menampilkan deskripsi atau detail dari produk tersebut. Berikut adalah tampilannya:



Gambar 3.38 Desain Detail Katalog Produk

L. Desain Sales Order

Desain *sales order* disini berfungsi untuk melakukan *sales order*. *Sales order* dilakukan oleh *salesman*. *Salesman* dapat menginputkan nama pelanggan, nama produk yang telah dipilih sebelumnya, kuantitas produk, diskon, dan keterangan *sales order*. *Salesman* dapat memilih melakukan orderan kembali atau memilih mengirimkan *sales order by email* kepada admin perusahaan. Berikut adalah tampilannya:

Aplikasi Operasional *Salesman*

Sales Order

Salesman :

ID Sales Order :

Tanggal Order :

Nama Pelanggan:

Jumlah Orderan : xx

ID Produk :

Nama Produk :

Harga :

Kuantitas :

Diskon :

Sub Total Harga :

Keterangan :



Gambar 3.39 Desain *Sales Order*

M. Desain Detail Sales Order

Desain detail *sales order* ini berfungsi untuk menampilkan orderan-orderan yang telah dilakukan oleh *salesman* sebelumnya. Menu ini lanjutan dari menu *sales order* sebelumnya, jika *salesman* melakukan lebih dari satu orderan. Detail *Sales order* dilakukan oleh *salesman*. *Salesman* dapat memilih melakukan orderan kembali atau memilih mengirimkan *sales order by email* kepada admin perusahaan.

Aplikasi Operasional Salesman

Detail Sales Order

Sales Order

ID:

Nama Produk:

Nama Pelanggan:

Kuantitas:

Tgl SO:

Harga:

Keterangan:

Nama Produk:

Nama Pelanggan:

Kuantitas:

Tgl SO:

Harga:

Keterangan:

Kirim Oderan By Email

Lanjut Oderan

Gambar 3.40 Desain Detail *Sales Order*

N. Desain Menu Penagihan Hutang

Desain menu penagihan hutang ini berfungsi untuk menampilkan daftar-daftar pelanggan yang mempunyai hutang. Menu penagihan hutang ini dijalankan oleh *salesman*.

Aplikasi Operasional *Salesman*

Penagihan Hutang

Enter Text Search

TP001 Toko Buku 1	<input type="button" value="▲"/> <input type="button" value="≡"/> <input type="button" value="▼"/>
TP002 Toko Buku 2	
TP003 Toko Buku 3	

Gambar 3.41 Desain Menu Penagihan Hutang

O. Desain Detail Tagihan Hutang

Aplikasi Operasional *Salesman*

Detail Tagihan Hutang

ID Pemabayaran : TP001
Nama Pelanggan : Toko Buku 1
Total tagihan : Rp 4.500.000
Tanggal Tagihan : 24 Januari 2016
Keterangan : Jatuh Tempo

Gambar 3.42 Desain Detail Penagihan Hutang

Desain detail penagihan hutang ini berfungsi untuk menampilkan detail penagihan hutang. Detail penagihan hutang ini menampilkan id tagihan, nama pelanggan, total penagihan, tanggal tagihan, dan keterangan. Menu ini lanjutan dari menu penagihan hutang sebelumnya.

P. Desain Proses Pembayaran

Aplikasi Operasional Salesman

Proses Pembayaran

ID Pemabayaran :

ID Tagihan :

Nama Pelanggan :

Total tagihan :

Tgl Pembayaran :

Jumlah Pembayaran:

Keterangan :

Cara Pembayaran:

Tunai

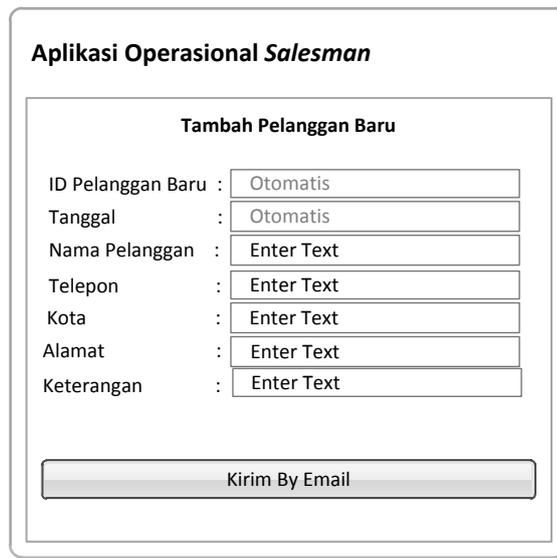
Giro

Transfer

Gambar 3.43 Desain Detail Penagihan Hutang

Desain menu detail penagihan hutang ini berfungsi untuk menginputkan pembayaran yang akan dilakukan oleh pelanggan. Menu ini digunakan oleh *salesman*. *Salesman* akan memasukkan data tanggal pembayaran, status hutang, cara pembayaran yang dilakukan oleh pelanggan, jumlah pembayaran, dan keterangan. Semua data yang telah dimasukkan akan dikirimkan kepada admin perusahaan melalui *email*.

Q. Desain Tambah Pelanggan Baru



Aplikasi Operasional Salesman

Tambah Pelanggan Baru

ID Pelanggan Baru :

Tanggal :

Nama Pelanggan :

Telepon :

Kota :

Alamat :

Keterangan :

Gambar 3.44 Desain Tambah Pelanggan Baru

Desain tambah pelanggan baru ini berfungsi untuk menginputkan data pelanggan baru. Menu tambah pelanggan baru ini digunakan oleh *salesman*. *Salesman* akan memasukkan nama pelanggan, telepon, kota, alamat, dan keterangan. Semua data yang telah dimasukkan akan dikirimkan kepada admin perusahaan melalui *email*.

R. Desain Detail Tambah Pelanggan Baru

Desain detail tambah pelanggan baru ini berfungsi untuk menampilkan data tambah pelanggan baru yang telah dilakukan sebelumnya. Hal ini dilakukan untuk mengkonfirmasi data pelanggan baru yang telah dilakukan. Berikut adalah tampilannya:

Aplikasi Operasional *Salesman*

Kirim Data Pelanggan Baru By Email

ID Pelanggan Baru : PB001
 Tanggal : 01-02-2016
 Nama Pelanggan :
 Telepon :
 Kota :
 Alamat :
 Keterangan :

Kirim By Email

Gambar 3.45 Desain Detail Tambah Pelanggan Baru

2. Desain Output

Desain *output* merupakan perancangan desain laporan(informasi) yang merupakan hasil dari pengolahan data dari proses yang ada, data yang tersimpan pada *database*, kemudian akan diolah menjadi informasi yang berguna bagi pengguna.

A. Desain Informasi Operasional *Salesman*

Desain *output* informasi operasional *salesman* ini menampilkan menu informasi *sales order*, pembayaran hutang, dan pelanggan baru. *Salesman* hanya memilih menu sesuai dengan kebutuhannya. Pada menu *sales order* dibagi menjadi tiga bagian dalam pencarian informasinya yaitu semua *sales order*, nama pelanggan, dan nama produk. Sedangkan untuk mencari informasi pembayaran hutang dibagi menjadi dua bagian yaitu menurut semua pembayaran hutang atau secara nama pelanggan. Berikut adalah tampilannya:

Aplikasi Operasional Salesman

Informasi Operasional Salesman

Sales Order:

- * Semua Sales Order
- * Nama Pelanggan
- * Produk

Pembayaran Hutang:

- * Semua Pembayaran Hutang
- * Nama Pelanggan

Pelanggan Baru:

- * Pelanggan Baru

Gambar 3.46 Desain Informasi Operasional Salesman

B. Desain Informasi Semua Sales Order

Desain informasi semua *sales order* ini menampilkan semua informasi *sales order* yang telah dilakukan oleh *salesman*. Desain informasi semua *sales order* ini diinputkan berdasarkan periode tanggal, yaitu tanggal awal dan tanggal akhir. Berikut adalah tampilannya:

Aplikasi Operasional Salesman

**Informasi
Sales Order Semua**

Periode Tanggal:

S/D

No	ID Sales Order	Nama Pelanggan	...
1	SO001/1407-001	Toko Atk Mulia	...
2	SO002/1407-001	Toko Jaya Abadi	...
3	SO003/1407-001	Toko Buku Kwan	...

Gambar 3.47 Desain Informasi Semua *Sales Order*

C. Desain Informasi Sales Order Pelanggan

Desain informasi sales order pelanggan ini menampilkan informasi *sales order* berdasarkan nama pelanggan. *Salesman* menginputkan nama pelanggan dan periode tanggal yaitu tanggal awal dan tanggal akhir. Berikut adalah tampilannya:

Aplikasi Operasional *Salesman*

**Informasi Sales Order
By Pelanggan**

Nama Pelanggan:

Periode Tanggal:
 S/D

No	ID Sales Order	Nama Pelanggan	...
1	SO001/1407-001	Toko Atk Mulia	...
2
3

Gambar 3.48 Desain Informasi *Sales Order* Pelanggan

D. Desain Informasi *Sales Order* Produk

Desain informasi *sales order* produk ini menampilkan informasi *sales order* berdasarkan nama produk. *Salesman* menginputkan nama produk dan periode tanggal yaitu tanggal awal dan tanggal akhir. Berikut tampilan desainnya:

Aplikasi Operasional *Salesman*

**Informasi Sales Order
By Produk**

Nama Produk:

Periode Tanggal:
 S/D

No	ID Sales Order	Nama Produk	...
1	SO001/1407-001	Cash Box	...
2
3

Gambar 3.49 Desain Informasi *Sales Order* Produk

E. Desain Informasi Pembayaran Hutang Semua

Aplikasi Operasional *Salesman*

**Informasi Pembayaran Hutang
Semua**

Periode Tanggal:
 S/D

No	ID Pembayaran	ID Tagihan	...
1	PEM_TP001	TP001	...
2	PEM_TP002	TP002	...
3	PEM_TP003	TP003	...

Gambar 3.50 Desain Informasi Pembayaran Hutang Semua

Desain informasi pembayaran hutang semua ini menampilkan semua pembayaran hutang yang telah dilakukan sebelumnya. Desain informasi semua pembayaran hutang ini diinputkan berdasarkan periode tanggal, yaitu tanggal awal dan tanggal akhir.

F. Desain Informasi Pembayaran Hutang Pelanggan

Aplikasi Operasional *Salesman*

Informasi Pembayaran Hutang By Pelanggan

Nama Pelanggan:

Periode Tanggal:
 S/D

No	ID Pembayaran	Nama Pelanggan	...
1	PEM_TP001	Toko Atk Mulia	...
2
3

Gambar 3.51 Desain Informasi Pembayaran Hutang Pelanggan

Desain informasi pembayaran hutang pelanggan ini menampilkan informasi pembayaran hutang berdasarkan nama pelanggan. *Salesman* menginputkan nama pelanggan dan periode tanggal yaitu tanggal awal dan tanggal akhir.

G. Desain Informasi Pelanggan Baru

Aplikasi Operasional *Salesman*

**Informasi Pelanggan Baru
Semua**

Periode Tanggal:

S/D

No	ID Pelanggan	Nama Pelanggan	...
1	PB001	Toko Abadi Jaya	...
...
...

Gambar 3.52 Desain Informasi Pelanggan Baru

Desain informasi pelanggan baru ini menampilkan semua pelanggan baru yang telah dilakukan sebelumnya. Desain informasi semua pelanggan baru ini diinputkan berdasarkan periode tanggal, yaitu tanggal awal dan tanggal akhir.

3.2.7 Desain Uji Coba (Perancangan Evaluasi)

Perancangan evaluasi merupakan kegiatan untuk merancang apa saja yang akan dilakukan, tahapan-tahapan apa saja yang dilalui dan apa saja yang akan dihasilkan menjadi jelas. Pada tahapan evaluasi akan dijelaskan mengenai pengukuran dan testing (uji coba) pada aplikasi. Pengukuran dan uji coba yang akan dilakukan akan berguna untuk menentukan apakah tujuan dari tugas akhir ini tercapai atau tidak. Uji coba aplikasi yang dilakukan akan menguji setiap *input* yang dilakukan dari setiap kejadian pada aplikasi dapat menghasilkan *output* yang

diharapkan. Desain uji coba ini menggunakan metode *black box testing*. Desain uji coba yang dilakukan adalah sebagai berikut:

A. Desain Uji Coba Katalog Produk

Desain uji coba katalog produk ini digunakan untuk merencanakan uji coba katalog produk pada aplikasi operaional *salesman* berbasis android. Perencanaan uji coba ini terdiri dari id uji coba, tujuan, input, dan hasil yang diharapkan. Berikut penjelasan mengenai perencanaan uji coba katalog produk:

Tabel 3.12 Desain Uji Coba Katalog Produk

ID	Tujuan	Input	Hasil Yang Diharapkan
A1	Menampilkan <i>listview</i> data produk yaitu id produk, nama produk, dan merk produk.	Memilih <i>button</i> katalog produk.	Aplikasi akan menampilkan <i>listview</i> data produk yaitu id produk, nama produk, dan merk produk.
A2	Menampilkan nama produk yang dicari.	Menginputkan nama produk yang dicari dan memilih <i>button search</i> .	Aplikasi akan menampilkan nama produk yang dicari.
A3	Menampilkan detail produk yaitu id produk, nama produk, merk produk, jenis produk, harga, karton, warna, dan keterangan.	Memilih produk yang diinginkan.	Aplikasi akan menampilkan detail produk yaitu id produk, nama produk, merk produk, jenis produk, harga, karton, warna, dan keterangan.

B. Desain Uji Coba Sales Order

Desain uji coba *sales order* ini digunakan untuk merencanakan uji coba *sales order* pada aplikasi operaional *salesman* berbasis android. Perencanaan uji

coba ini terdiri dari id uji coba, tujuan, input, dan hasil yang diharapkan. Berikut penjelasan mengenai perencanaan uji coba *sales order*:

Tabel 3.13 Desain Uji Coba *Sales Order*

ID	Tujuan	Input	Hasil Yang Diharapkan
B1	Menampilkan data <i>sales order</i> yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, dan orderan produk yang dipilih (id produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan).	Memilih <i>button</i> order.	Aplikasi akan menampilkan data <i>sales order</i> yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, dan orderan produk yang dipilih (id produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan).
B2	Menampilkan sub total harga secara otomatis.	Menginputkan kuantitas produk dan diskon yang diberikan.	Aplikasi akan menampilkan sub total harga secara otomatis.
B3	Menghilangkan atau menghapus orderan yang dipilih.	Memilih <i>button</i> hapus pada orderan yang dipilih.	Aplikasi akan menghilangkan atau menghapus tampilan orderan yang telah dipilih.
B4	Menampilkan jumlah orderan.	Memilih <i>button</i> order.	Aplikasi akan menampilkan jumlah orderan.
B5	Menampilkan nama pelanggan yang dipilih.	Memilih nama pelanggan.	Aplikasi akan menampilkan nama pelanggan yang dipilih.
B6	Menampilkan <i>listview</i> katalog produk.	Memilih <i>button</i> lanjutkan order.	Aplikasi akan menampilkan <i>listview</i> katalog produk.
B7	Menampilkan data orderan yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, dan orderan produk yang sudah diinputkan (id	Menginputkan kuantitas produk, diskon dan harga pada tiap produk yang akan diorder. Memilih <i>button</i> selesai order.	Aplikasi akan menampilkan data orderan yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, dan orderan produk yang

ID	Tujuan	Input	Hasil Yang Diharapkan
	produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan).		sudah diinputkan (id produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan).
B8	Menampilkan total harga <i>sales order</i> secara otomatis.	Memilih <i>button</i> selesai order.	Aplikasi akan menampilkan total harga <i>sales order</i> .
B9	Menampilkan data orderan yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, orderan produk yang sudah diinputkan (id produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan), dan total harga <i>sales order</i> pada aplikasi Gmail.	Memilih <i>button</i> kirim <i>by email</i> .	Aplikasi akan menampilkan data orderan yaitu nama <i>salesman</i> , id <i>sales order</i> , tanggal order, nama pelanggan, orderan produk yang sudah diinputkan (id produk, nama produk, harga, kuantitas, diskon, sub total harga, dan keterangan), dan total harga <i>sales order</i> pada aplikasi Gmail.

C. Desain Uji Coba Penagihan Hutang

Desain uji coba penagihan hutang ini digunakan untuk merencanakan uji coba penagihan hutang pada aplikasi operaional *salesman* berbasis android. Perencanaan uji coba ini terdiri dari id uji coba, tujuan, input, dan hasil yang diharapkan. Berikut penjelasan mengenai perencanaan uji coba penagihan hutang:

Tabel 3.14 Desain Uji Coba Penagihan Hutang

ID	Tujuan	Input	Hasil Yang Diharapkan
C1	Menampilkan <i>listview</i> data tagihan hutang yaitu id tagihan dan nama pelanggan.	Memilih <i>button</i> tagihan hutang.	Aplikasi akan menampilkan <i>listview</i> data tagihan hutang yaitu id tagihan dan nama pelanggan.

ID	Tujuan	Input	Hasil Yang Diharapkan
C2	Menampilkan nama pelanggan yang dicari.	Menginputkan nama pelanggan yang dicari dan memilih <i>button search</i> .	Aplikasi akan menampilkan nama pelanggan yang dicari.
C3	Menampilkan detail tagihan hutang yaitu id tagihan, nama pelanggan, total tagihan, tanggal tagihan, dan keterangan.	Memilih tagihan hutang atau nama pelanggan yang diinginkan.	Aplikasi akan menampilkan detail tagihan hutang yaitu id tagihan, nama pelanggan, total tagihan, tanggal tagihan, dan keterangan.
C4	Memberikan notifikasi bahwa proses pembayaran sudah dilakukan dan <i>button</i> proses pembayaran tidak ditampilkan.	Memilih tagihan hutang atau nama pelanggan yang diinginkan.	Aplikasi akan memberikan notifikasi bahwa proses pembayaran sudah dilakukan dan <i>button</i> proses pembayaran tidak ditampilkan

D. Desain Uji Coba Pembayaran Hutang

Desain uji coba pembayaran hutang ini digunakan untuk merencanakan uji coba pembayaran hutang pada aplikasi operasional *salesman* berbasis android. Perencanaan uji coba ini terdiri dari id uji coba, tujuan, input, dan hasil yang diharapkan. Berikut penjelasan mengenai perencanaan uji coba pembayaran hutang:

Tabel 3.15 Desain Uji Coba Pembayaran Hutang

ID	Tujuan	Input	Hasil Yang Diharapkan
D1	Menampilkan data proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total tagihan, tanggal	Memilih <i>button</i> proses pembayaran.	Aplikasi akan menampilkan data proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total

ID	Tujuan	Input	Hasil Yang Diharapkan
	pembayaran, jumlah pembayaran, keterangan, dan cara pembayaran (tunai/giro/transfer).		tagihan, tanggal pembayaran, jumlah pembayaran, keterangan, dan cara pembayaran (tunai/giro/transfer).
D2	Menampilkan format tanggal pada tanggal pembayaran.	Memilih dan menginputkan <i>edittext</i> tanggal pembayaran.	Aplikasi akan menampilkan format tanggal pada tanggal pembayaran.
D3	Menampilkan format rupiah pada jumlah pembayaran.	Memilih dan menginputkan jumlah pembayaran.	Aplikasi akan menampilkan format rupiah pada jumlah pembayaran.
D4	Dapat memilih satu <i>radio button</i> (tunai/giro/transfer) dan mendapatkan notifikasi cara pembayaran.	Memilih satu <i>radio button</i> (tunai/giro/transfer).	Aplikasi akan menampilkan satu <i>radio button</i> (tunai/giro/transfer) dan mendapatkan notifikasi cara pembayaran.
D5	Menampilkan notifikasi pada saat menginputkan jumlah pembayaran. Salah satu notifikasi yang akan ditampilkan adalah sebagai berikut: - Jumlah tagihan kurang Rp xxxx. - Jumlah pembayaran harus lebih kecil dari jumlah tagihan. - Tagihan lunas.	Menginputkan jumlah pembayaran dan memilih <i>button</i> kirim <i>by email</i> .	Aplikasi akan menampilkan notifikasi pada saat menginputkan jumlah pembayaran. Salah satu notifikasi yang akan ditampilkan adalah sebagai berikut: - Jumlah tagihan kurang Rp xxxx. - Jumlah pembayaran harus lebih kecil dari jumlah tagihan. - Tagihan lunas.
D6	Menampilkan data proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total tagihan, tanggal pembayaran, jumlah	Memilih <i>button</i> selesai proses pembayaran.	Aplikasi akan menampilkan data proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total tagihan, tanggal

ID	Tujuan	Input	Hasil Yang Diharapkan
	pembayaran, keterangan, dan cara pembayaran (tunai/giro/transfer) yang sudah terisi.		pembayaran, jumlah pembayaran, keterangan, dan cara pembayaran (tunai/giro/transfer) yang sudah terisi.
D7	Menampilkan informasi sisa tagihan hutang (lunas/masih ada sisa tagihan).	Memilih <i>button</i> selesai proses pembayaran.	Aplikasi akan menampilkan informasi sisa tagihan hutang (lunas/masih ada sisa tagihan).
D8	Menampilkan data proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total tagihan, tanggal pembayaran, jumlah pembayaran, keterangan, cara pembayaran (tunai/giro/transfer), dan sisa tagihan (lunas/ masih ada sisa tagihan) yang sudah terisi pada aplikasi Gmail.	Memilih <i>button</i> kirim <i>by email</i> .	Aplikasi akan menampilkan proses pembayaran yaitu id pembayaran, id tagihan, nama pelanggan, total tagihan, tanggal pembayaran, jumlah pembayaran, keterangan, cara pembayaran (tunai/giro/transfer), dan sisa tagihan (lunas/ masih ada sisa tagihan) yang sudah terisi pada aplikasi Gmail.

E. Desain Uji Coba Pelanggan Baru

Desain uji coba pelanggan baru ini digunakan untuk merencanakan uji coba pelanggan baru pada aplikasi operaional *salesman* berbasis android. Perencanaan uji coba ini terdiri dari id uji coba, tujuan, input, dan hasil yang diharapkan. Berikut penjelasan mengenai perencanaan uji coba pelanggan baru:

Tabel 3.16 Desain Uji Coba Pelanggan Baru

ID	Tujuan	Input	Hasil Yang Diharapkan
E1	Menampilkan data pelanggan baru yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan.	Memilih <i>button</i> pelanggan baru.	Aplikasi akan menampilkan data pelanggan baru yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan.
E2	Menampilkan daftar kota.	Memilih <i>edittext</i> kota.	Aplikasi akan menampilkan data kota.
E3	Menampilkan data pelanggan baru yang sudah diisi yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan.	Memilih <i>button</i> selesai pelanggan baru.	Aplikasi akan menampilkan data pelanggan baru yang sudah diisi yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan.
E4	Menampilkan data pelanggan baru yang sudah diisi yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan pada aplikasi Gmail.	Memilih <i>button</i> kirim <i>by email</i> .	Aplikasi akan menampilkan data pelanggan baru yang sudah diisi yaitu id pelanggan baru, tanggal, nama pelanggan, telepon, kota, alamat, dan keterangan pada aplikasi Gmail.