

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Pada tahap ini menjelaskan tentang kondisi suatu perusahaan saat ini berdasarkan hasil survei lapangan yang bertujuan untuk mengevaluasi permasalahan yang terjadi saat ini, sehingga dapat diajukan suatu usulan perbaikan. Pengumpulan data dilakukan untuk mengetahui kondisi atau informasi tentang perusahaan. Adapun kegiatan pengumpulan data meliputi

1. Observasi

Langkah ini dilakukan untuk mengamati dan mengidentifikasi tentang informasi dan kondisi pada Mitra 10 Wiyung secara langsung. Data-data yang telah diperoleh akan dijadikan acuan untuk memperoleh gambaran umum tentang proses penjualan di Mitra 10 Wiyung. Observasi dilaksanakan pada bulan Februari 2016 – April 2016. Observasi dilakukan di bagian kasir dan data yang diperoleh selama observasi adalah data jenis barang, data diskon barang, dan data barang.

2. Wawancara

Wawancara dalam penelitian ini dilakukan dengan cara tanya jawab langsung dengan bagian HRD Mitra 10 Wiyung. Wawancara yang dilakukan pada penelitian ini dilaksanakan secara terencana dengan berpedoman daftar pertanyaan yang telah disiapkan, dalam istilah lain wawancara seperti ini disebut wawancara terstruktur. Untuk membuat pertanyaan berpedoman dengan topik penelitian yang akan dibahas. Wawancara dilaksanakan pada

bulan pertama sampai bulan kelima. Tujuan wawancara dalam penelitian ini yaitu memudahkan peneliti untuk menjawab permasalahan yang ada.

3. Studi Pustaka

Langkah ini mengumpulkan informasi dan literatur yang berkaitan dengan penelitian yang dilakukan. Sumber informasi dan literatur ini berupa jurnal, karya ilmiah dan buku pendukung. Adapun karya ilmiah yang pernah di baca adalah karya ilmiah yang berupa tugas akhir yang mempunyai permasalahan yang hampir sama dengan yang akan dibahas. Studi pustaka ini dilaksanakan selama penelitian ini berlangsung.

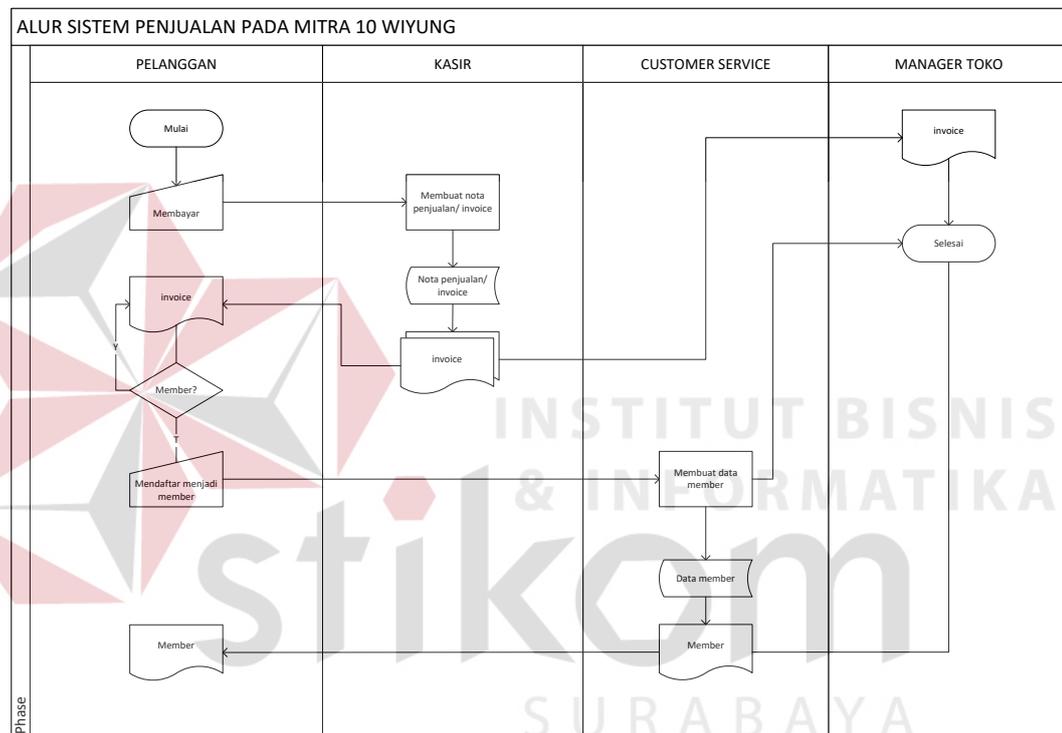
3.1.1. Identifikasi Masalah

PT Catur Mitra Sejati Sentosa adalah perusahaan yang bergerak sebagai distributor bahan bangunan melahirkan konsep belanja bahan bangunan pertama di Indonesia dengan nama Mitra10, salah satunya adalah Mitra 10 Wiyung. Berdasarkan hasil wawancara dan observasi yang dilakukan, proses penjualan yang terjadi di perusahaan dimulai saat pelanggan masuk ke toko dan disambut oleh pegawai. Kemudian pelanggan memilih barang atau jika kesulitan, pegawai akan siap untuk membantu pelanggan dalam memilih barang. Setelah itu pelanggan membayar barang yang dibeli di kasir.

Tetapi pihak kasir sendiri tidak mengetahui secara pasti stok barang sebenarnya yang tersedia karena hanya mengetahui dari sistem saja dan terkadang stok barang ada tidak sesuai dengan sistem. Perusahaan tidak memiliki laporan stok barang sehingga saat manager toko melakukan pemeriksaan hasil akhir penjualan, manager toko memeriksa satu per satu data barang yang ada sehingga dibutuhkan waktu yang lama. Perusahaan juga tidak dapat mengetahui pendapatan

mereka setiap bulan dikarenakan hasil penjualan setiap harinya sistem akan langsung mengirimkan data penjualan ke pusat.

Berdasarkan identifikasi masalah tersebut, maka perusahaan membutuhkan aplikasi untuk memudahkan perusahaan dalam melakukan kegiatan penjualan dan dalam melakukan proses pemeriksaan hasil penjualan, stok barang dan pendapatan yang didapat.



Gambar 3.1. Alur sistem penjualan pada Mitra 10 Wiyung

Proses penjualan yang terjadi pada Mitra 10 Wiyung, yaitu dimulai dari pelanggan membayar bayar yang telah dipilih. Kemudian kasir membuat invoice dan menyimpannya ke *database* yang ada lalu mencetak 2 rangkap invoice dan memberikan 1 rangkap invoice tersebut ke pelanggan dan 1 diberikan ke manager toko. Jika pelanggan ingin menjadi member, maka pelanggan mendaftar ke bagian *customer service* dan kemudian *customer service* mendata dan menyimpannya, kemudian mencetak dan diberikan kartu member ke pelanggan.

3.1.2. Analisis Kebutuhan

Analisa kebutuhan merupakan tahap sebelum melakukan perancangan sistem. Kebutuhan hasil analisa ini harus dapat dilaksanakan, diukur, diuji sesuai dengan kebutuhan bisnis yang telah diidentifikasi dan didefinisikan hingga detail yang memadai untuk desain sistem. Kebutuhan adalah sebagai berikut.

1. Informasi mengenai stok barang.
2. Informasi mengenai rekapitulasi penjualan.
3. Informasi mengenai laporan pendapatan.

3.1.3. Spesifikasi Kebutuhan Sistem

Spesifikasi kebutuhan sistem adalah proses mendeskripsikan kebutuhan dari informasi yang ada pada sistem. Spesifikasi kebutuhan sistem sebagai berikut:

1. Informasi mengenai stok barang.

Informasi ini didapatkan dari jenis barang, nama barang, jumlah dan harga barang. Untuk proses ini hasil akan keluar secara otomatis saat ingin mencari stok barang.

2. Informasi mengenai rekapitulasi penjualan.

Informasi ini didapatkan dari proses penjualan yang terjadi. Agar dapat menghasilkan rekapitulasi penjualan dibutuhkan data nota penjualan dan proses ini akan langsung tampil saat perhitungan penjualan telah selesai dilakukan.

3. Informasi mengenai laporan pendapatan.

Informasi ini didapatkan dari proses penjualan yang dilakukan dan pembelian ke distributor. Proses ini akan muncul saat menghitung total penjualan yang ada dan total pembelanjaan ke distributor.

3.2. Perancangan Sistem

Tahap ini adalah gambaran atau rancangan dari penelitian tugas akhir ini. Hal ini bertujuan untuk program dapat berfungsi dan dapat berjalan sesuai dengan yang diharapkan, yaitu dapat membantu proses penjualan menjadi lebih baik. Beberapa tahapan yang dilakukan dalam perancangan sistem penelitian ini adalah pembuatan *Sysflow*, *Input-Process-Output*, *Context Diagram*, *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, dan struktur *database* yang digunakan dalam program aplikasi ini.

3.2.1. *Input-Process-Output*

Pada gambar 3.2. menjelaskan tentang alur *input-process-output* untuk aplikasi penjualan. Adapun penjelasan dari gambar 3.2. adalah sebagai berikut

1. Input

a. Data barang

Data barang berisi data barang awal yang harus dimasukkan ke dalam sistem seperti nama barang, jenis barang, harga barang, jumlah barang.

b. Data promo

Data promo berisi data promo awal yang harus dimasukkan ke dalam sisem seperti jumlah promo dan nama promo.

c. Data jenis barang

Data jenis barang berisi data jenis barang yang harus dimasukkan ke dalam sistem seperti jenis barang dan keterangan jenis barang.

d. Data COA

Data pelanggan berisi data awal pelanggan yang harus dimasukkan ke dalam sistem seperti no akun dan nama akun.

e. Data pelanggan

Data pelanggan berisi data awal pelanggan yang harus dimasukkan ke dalam sistem seperti id pelanggan, nama pelanggan, alamat, jenis kelamin, dan nomor telepon.

2. Proses

a. Pengelolaan data awal

Proses pengelolaan data awal adalah proses yang dilakukan untuk mengelola data awal menjadi data yang sesuai untuk sistem. Misalnya data awal adalah data barang melakukan proses pengelolaan data awal sehingga menghasilkan daftar barang.

b. Pendaftaran member

Proses pendaftaran member merupakan proses yang dilakukan pada saat pelanggan ingin menjadi member di Mitra 10 Wiyung Surabaya. Persyaratan jika menjadi member yaitu jika pelanggan membeli barang hingga 1 juta.

c. Penjualan

Proses penjualan merupakan proses barang dagang dijual ke pelanggan atau member agar perusahaan mendapatkan keuntungan. Pada proses

penjualan dibutuhkan daftar barang, daftar jenis barang dan daftar diskon, data pelanggan atau data member sehingga dapat menghasilkan nota penjualan.

d. Pemesanan Barang *Job Order*

Proses pemesanan barang *Job Order* merupakan memesan barang yang bersifat *job order*. Pada proses penjualan dibutuhkan daftar barang, daftar jenis barang dan, sehingga dapat menghasilkan nota penjualan dan data pelanggan.

e. Pembayaran Barang *Job Order*

Proses pembayaran barang *Job Order* merupakan pembayaran barang yang bersifat *job order*. Pada proses penjualan dibutuhkan tanggal transaksi, nomor transaksi dan jumlah transaksi, sehingga dapat menghasilkan invoice.

f. Pelaporan

Pelaporan merupakan proses membuat laporan dari semua proses yang telah dilakukan. Pada proses pelaporan dapat berasal dari proses penjualan atau dapat juga diambil dari nota penjualan. Pelaporan menghasilkan rekapitulasi stok barang, rekapitulasi penjualan dan laporan pendapatan.

3. Output

a. Daftar barang

Daftar barang merupakan hasil yang didapat dari proses pengelolaan data awal. Pada daftar barang berisi id barang, nama barang, jenis barang, jumlah barang, harga barang.

b. Daftar jenis barang

Daftar jenis barang merupakan hasil dari proses pengelolaan data awal. Pada daftar jenis barang berisi id jenis barang, nama jenis barang dan keterangan jenis barang.

c. Daftar promo

Daftar diskon merupakan hasil dari proses pengelolaan data awal. Pada daftar promo terdapat id promo, jumlah promo, nama diskon, dan tanggal berlaku promo.

d. Daftar COA

Daftar COA merupakan hasil dari proses pengelolaan data awal. Pada daftar COA terdapat id diskon, diskon dan keterangan diskon.

e. Nota penjualan

Nota penjualan merupakan hasil dari proses penjualan. Pada nota penjualan terdapat nomor penjualan, tanggal transaksi, nama barang, jenis barang, jumlah barang, harga barang, total harga barang.

f. Data *member*

Daftar *member* merupakan hasil dari proses pengelolaan data awal. Pada daftar *member* terdapat id *member*, nama, alamat, jenis kelamin, nomor telepon, dan diskon *member*.

g. Data pelanggan

Daftar pelanggan merupakan hasil dari proses pengelolaan data awal. Pada daftar pelanggan terdapat id pelanggan, nama, alamat, jenis kelamin, nomor telepon.

h. Nota pemesanan

Nota pemesanan merupakan hasil dari proses pemesanan barang *job order*.

Pada nota pemesanan terdapat nomor pemesanan, tanggal pesan, nama barang, jenis barang, jumlah barang, harga barang, total harga barang.

i. Invoice barang *job order*

Invoice barang *job order* merupakan hasil dari proses pembayaran barang *job order*. Pada invoice barang *job order* terdapat nomor invoice, tanggal transaksi, nama barang, jenis barang, jumlah barang, harga barang, total harga barang.

j. Laporan stok barang

Laporan stok barang merupakan hasil dari proses pelaporan. Pada laporan stok barang terdapat nama barang, jumlah barang, harga barang, total harga barang.

k. Rekapitulasi penjualan tunai

Rekapitulasi penjualan tunai merupakan hasil dari proses pelaporan. Rekapitulasi penjualan tunai berisi tentang semua hasil yang didapat dari hasil penjualan yang terjadi. Pada laporan penjualan tunai terdapat nomor laporan penjualan, tanggal pembuatan laporan, nomor nota penjualan, nama barang, jumlah barang, total harga barang

l. Rekapitulasi pelunasan pemesanan *job order*

Rekapitulasi pelunasan pemesanan *job order* merupakan hasil dari proses pelaporan. Rekapitulasi pelunasan pemesanan *job order* berisi tentang semua hasil yang didapat dari hasil penjualan yang terjadi. Pada rekapitulasi pelunasan pemesanan *job order* terdapat nomor laporan

pelunasan pemesanan *job order*, tanggal pemesanan, nomor nota pelunasan pemesanan *job order*, nama barang, jumlah barang, total harga barang

m. Laporan pendapatan

Laporan pendapatan merupakan hasil dari proses pelaporan. Pada laporan berisi jumlah yang telah dihasilkan perusahaan dari proses penjualan. Pada laporan pendapatan terdapat nomor laporan pendapatan, tanggal laporan pendapatan, total harga barang, total seluruhnya penjualan.

n. Jurnal penjualan

Jurnal penjualan merupakan hasil dari proses pelaporan. Pada jurnal penjualan berisi jumlah yang telah dihasilkan perusahaan dari proses pelaporan. Pada jurnal penjualan terdapat nomor jurnal penjualan, tanggal jurnal penjualan, debit, kredit, no akun, dan nama akun.

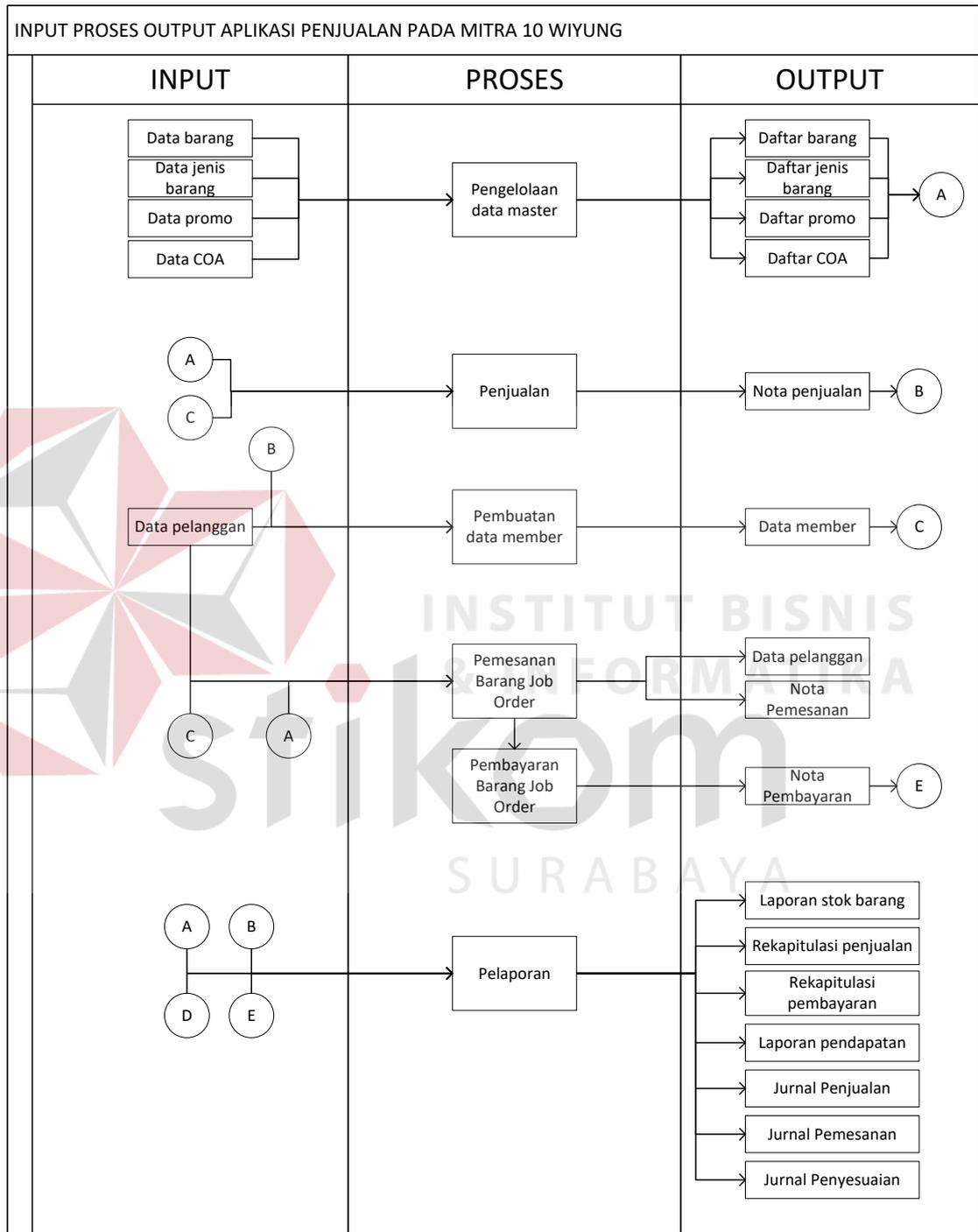
o. Jurnal pemesanan

Jurnal pemesanan merupakan hasil dari proses pelaporan. Pada jurnal pemesanan berisi jumlah yang telah dihasilkan perusahaan dari proses pelaporan. Pada jurnal pemesanan terdapat nomor jurnal pemesanan, tanggal jurnal pemesanan, debit, kredit, no akun, dan nama akun.

p. Jurnal penyesuaian

Jurnal penjualan merupakan hasil dari proses pelaporan. Pada jurnal penyesuaian berisi jumlah yang telah dihasilkan perusahaan dari proses pelaporan. Pada jurnal penyesuaian terdapat nomor jurnal penyesuaian, tanggal jurnal penyesuaian, debit, kredit, no akun, dan nama akun.

Kemudian setelah melakukan desain aplikasi, membuat program dengan menggunakan Visual Studio dan SQL Server.

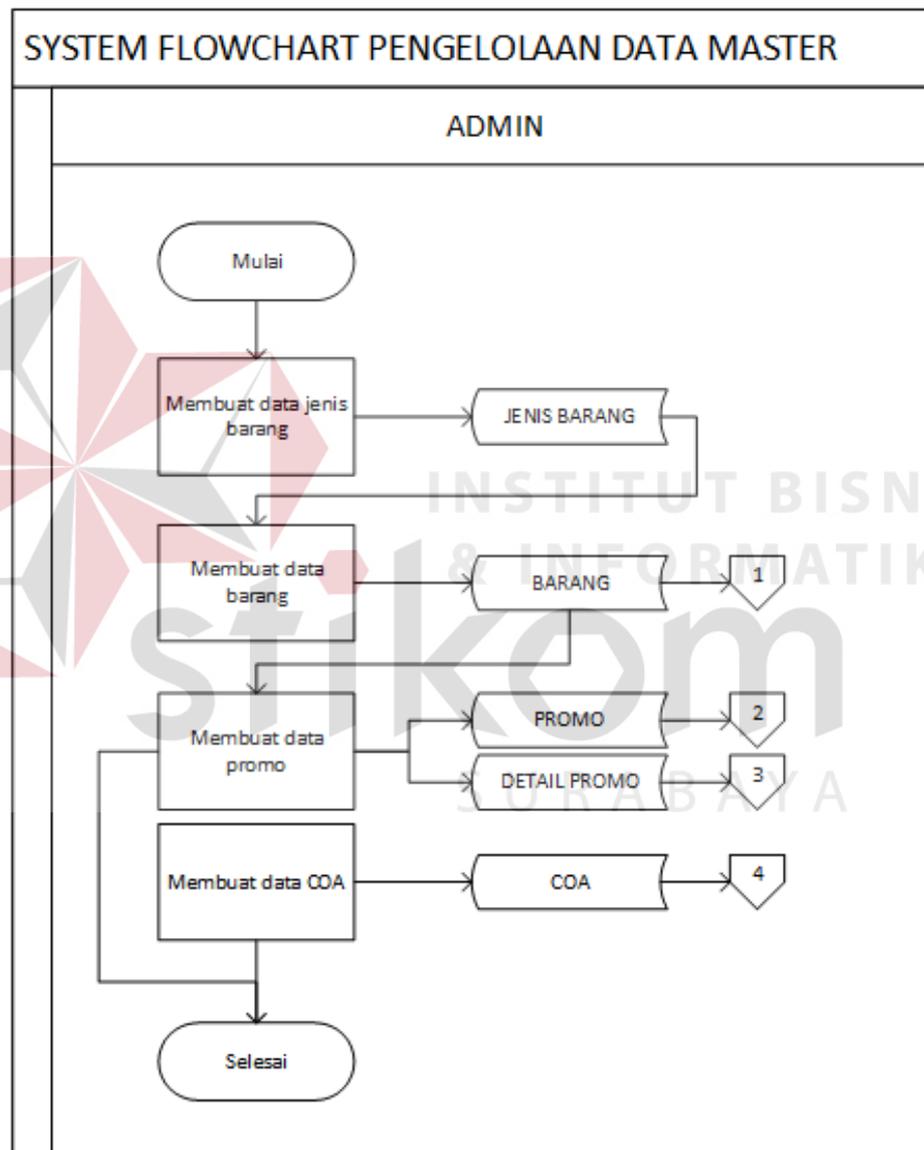


Gambar 3.2 Input-Proses-Output (IPO) Aplikasi Penjualan pada Mitra 10 Wiyung

3.2.2. System Flowchart

System Flow ini menggambar tentang alur sistem proses penjualan yang dibuat sesuai dengan kebutuhan di Mitra 10 Wiyung. Berikut ini merupakan *system flow* penjualan pada Mitra 10 Wiyung:

a. System Flowchart Pengelolaan Data Master

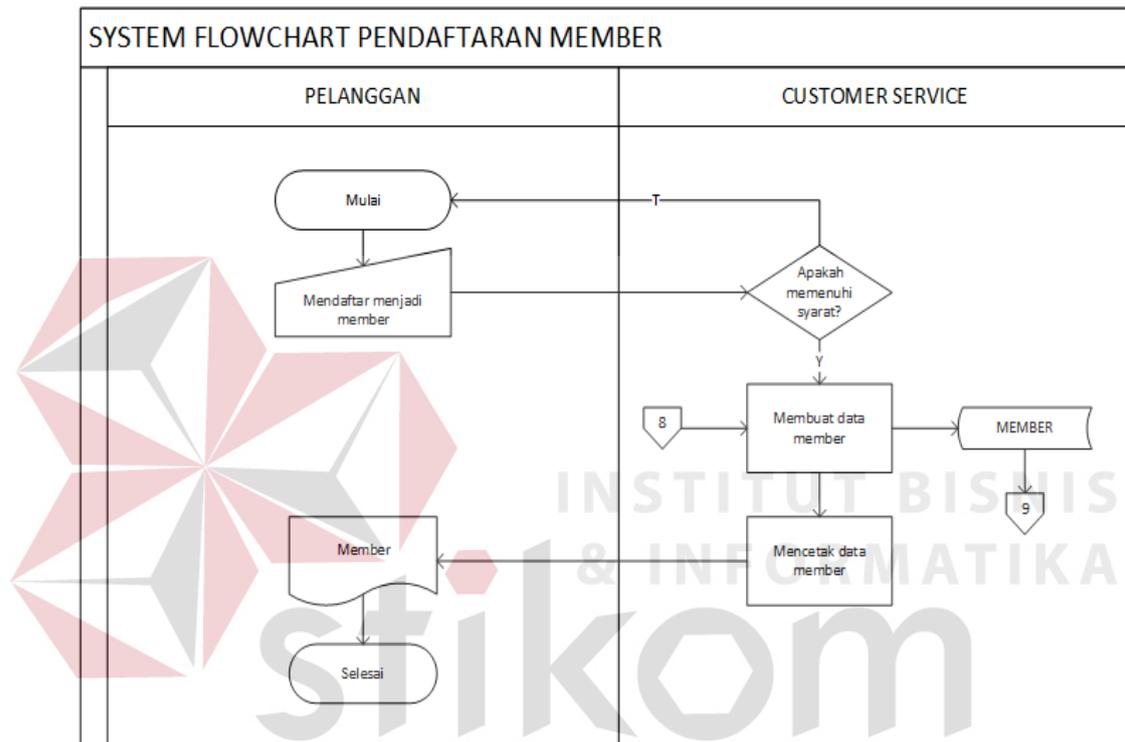


Gambar 3.3. System Flowchart Pengelolaan Data Master

Pada alur pengelolaan data master, dimulai dari admin membuat data jenis barang dan kemudian disimpan ke *database*. Setelah itu, admin membuat data

diskon dan disimpan ke *database*. Terakhir, admin membuat data barang dengan mengambil data dari data jenis barang dan data diskon pada *database* jenis barang dan diskon. Setelah membuat data barang, maka akan disimpan di *database* barang dan selesailah proses pada gambar 3.3.

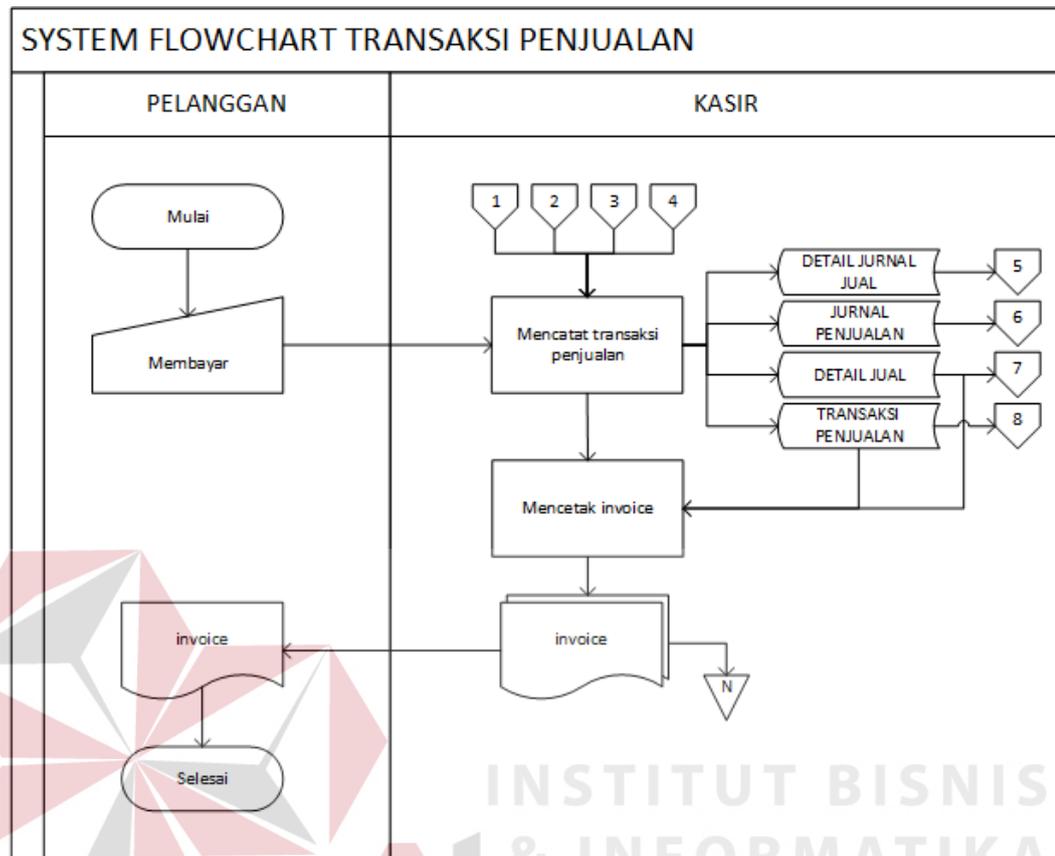
b. System Flowchart Pembuatan data member



Gambar 3.4. System Flowchart Pendaftaran Member

Pada gambar 3.4 menjelaskan alur *system flowchart* yang dibuat. Dimulai dari pengelolaan data master. Pada saat pendaftaran *member*, jika pelanggan ingin menjadi *member*, maka pelanggan akan mendaftar ke bagian *customer service*. Setelah itu *customer service* akan membuat data *member* dan disimpan ke dalam *database*. Kemudian dicetak dan diberikan pada pelanggan yang telah mendaftar tersebut.

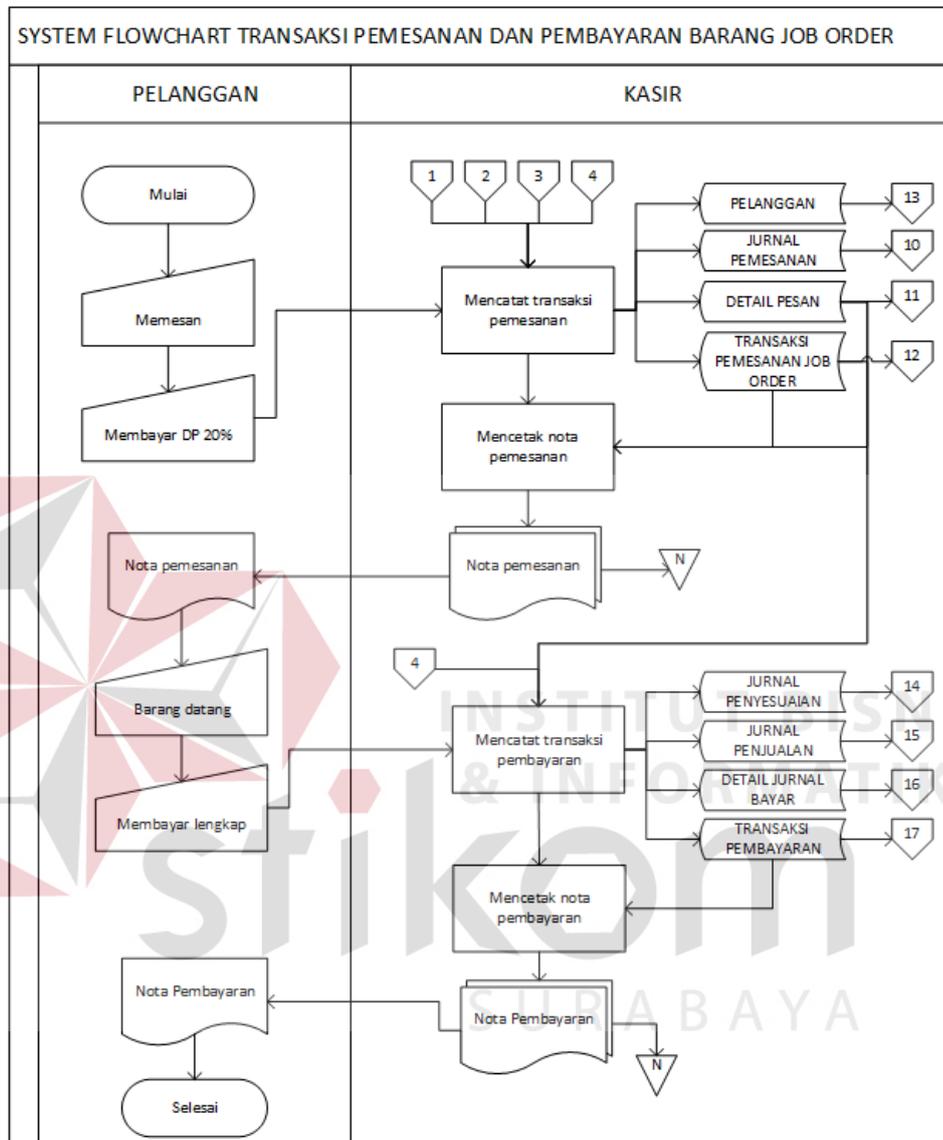
c. *System Flowchart* Transaksi Penjualan



Gambar 3.5. *System Flowchart* Transaksi Penjualan

Pada gambar 3.5 menjelaskan proses alur *system flowchart* dibagian transaksi penjualan. Dimulai dari pelanggan membayar barang yang telah dipilih ke kasir. Kemudian kasir membuat nota penjualan/ *invoice* melalui data barang dan data member yang ada pada gambar 3.4. setelah membuat nota penjualan/ *invoice*, maka disimpan di jurnal penjualan dan transaksi penjualan tunai dan dicetak rangkap 2 yang 1 diberikan ke pelanggan dan 1 untuk kasir sendiri. Demikian alur proses penjualan pada gambar 3.5.

d. *System Flowchart* Transaksi Pemesanan dan Pelunasan Barang *Job Order*



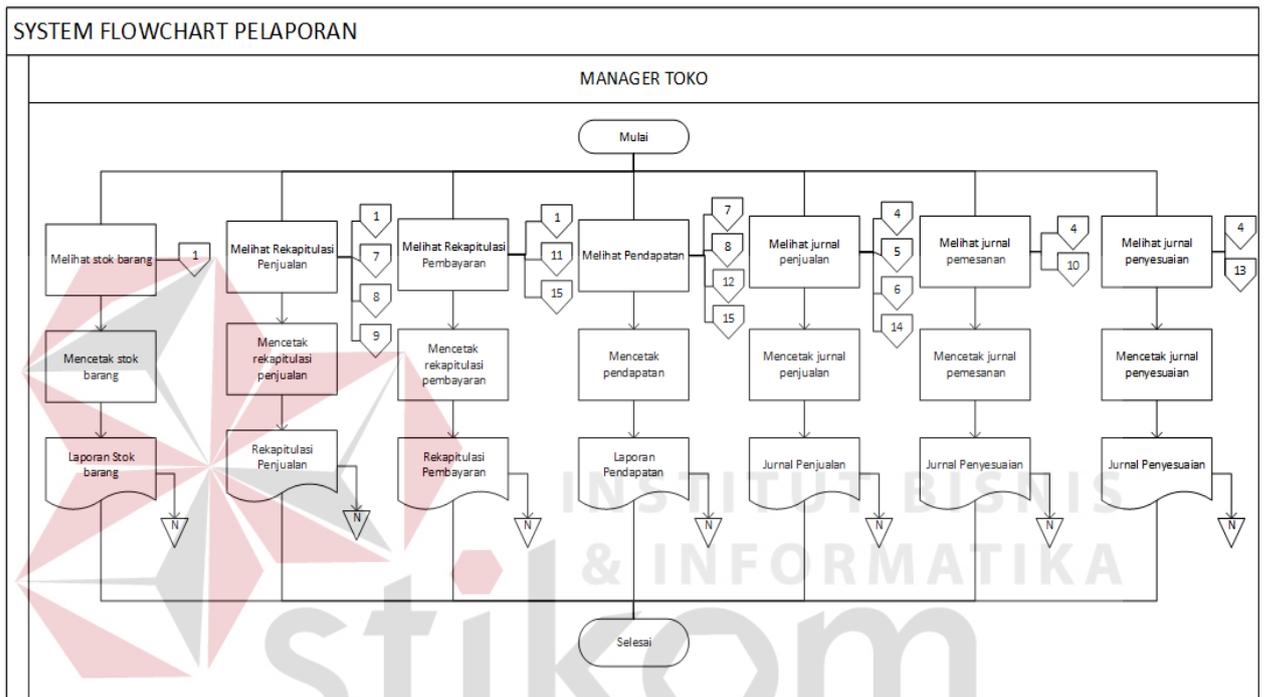
Gambar 3.6. *System Flowchart* Transaksi Pemesanan dan Pelunasan Barang *Job*

Order

Pada gambar 3.6 menjelaskan proses alur *system flowchart* dibagian transaksi pemesanan dan pelunasan barang *job order*. Dimulai dari pelanggan memesan barang yang telah dipilih ke kasir, dan membayar DP sebesar 20%. Kemudian kasir membuat nota pemesanan/ *invoice* melalui data barang dan data

member yang ada pada gambar 3.5. setelah membuat nota pemesanan/ *invoice*, maka disimpan di jurnal pemesanan dan transaksi pemesanan. Kemudian pelanggan menerima nota pemesanan dan membayar lengkap kekurangan pembayaran. Demikian alur proses penjualan pada gambar 3.6.

e. *System Flowchart Pelaporan*

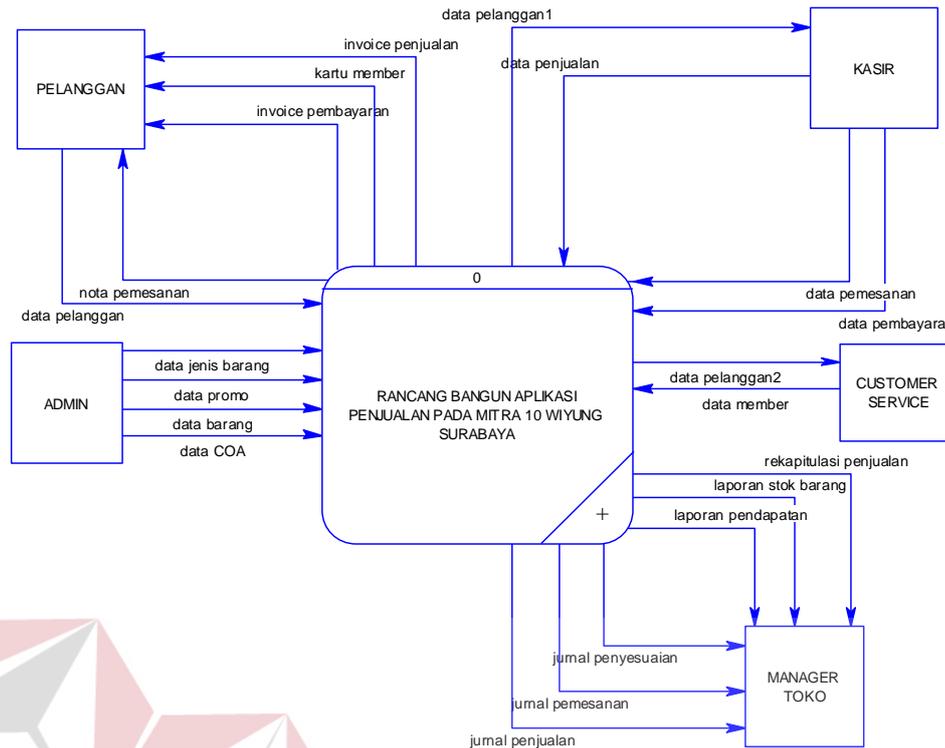


Gambar 3.7. *System Flowchart Pelaporan*

Pada gambar 3.7. merupakan alur *system flowchart* pelaporan. Pada pelaporan ini, memiliki laporan stok barang yang dihasilkan.

3.2.3. *Context Diagram*

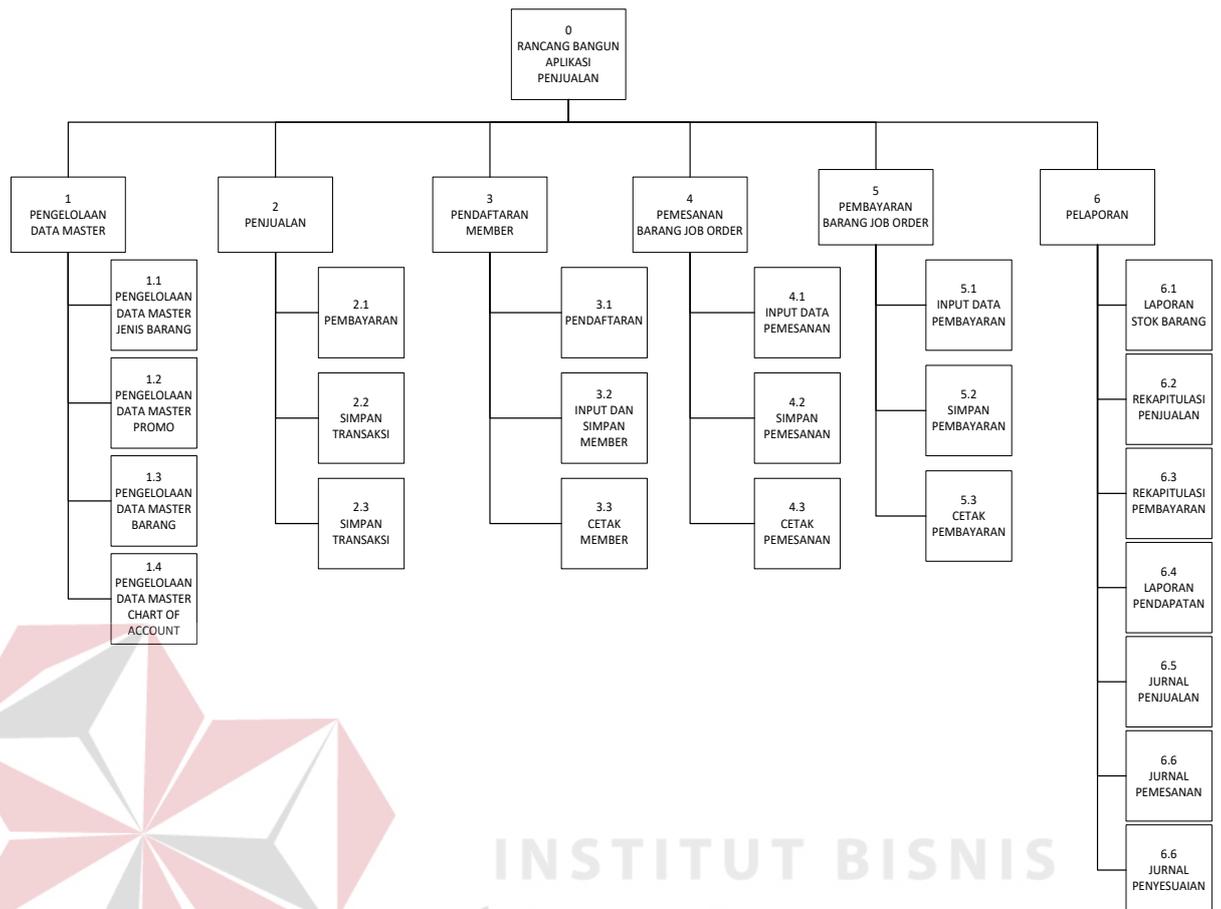
Context Diagram ini terdiri dari suatu proses dan menggambarkan ruang lingkup pada Mitra 10 Wiyung. Context diagram dari sistem penjualan yang terdiri dari 5 entitas, yaitu pelanggan, *customer service*, admin, manager toko, dan kasir. Pelanggan, *customer service*, admin, manajer toko, dan kasir merupakan entitas yang ada diluar ruang lingkup sistem seperti ditunjukkan pada gambar 3.6.



Gambar 3.8. Context Diagram

3.2.4. Hierarchy Input-Process-Output

HIPO dalam penelitian ini ditunjukkan pada gambar 3.9., dimana pada rancang bangun aplikasi penjualan mempunyai 6 proses yaitu pengelolaan data *master*, pembuatan data *member*, pemesanan barang *job order*, pembayaran barang *job order*, penjualan dan pelaporan. Untuk pengelolaan data memiliki 4 proses yaitu penginputan *master* jenis barang, penginputan *master* promo, penginputan data *master chart of account* dan penginputan *master* barang. Untuk pembuatan data *member* memiliki 3 proses yaitu pendaftaran *member*, dan penginputan *member*. Untuk penjualan memiliki 4 proses yaitu pembayaran, pembuatan *invoice*, simpan *invoice*, dan cetak *invoice*. Sedangkan untuk pelaporan memiliki 7 proses yaitu laporan stok barang, rekapitulasi penjualan, rekapitulasi pembayaran, laporan pendapatan, jurnal penyesuaian, jurnal pemesanan, dan jurnal penjualan.



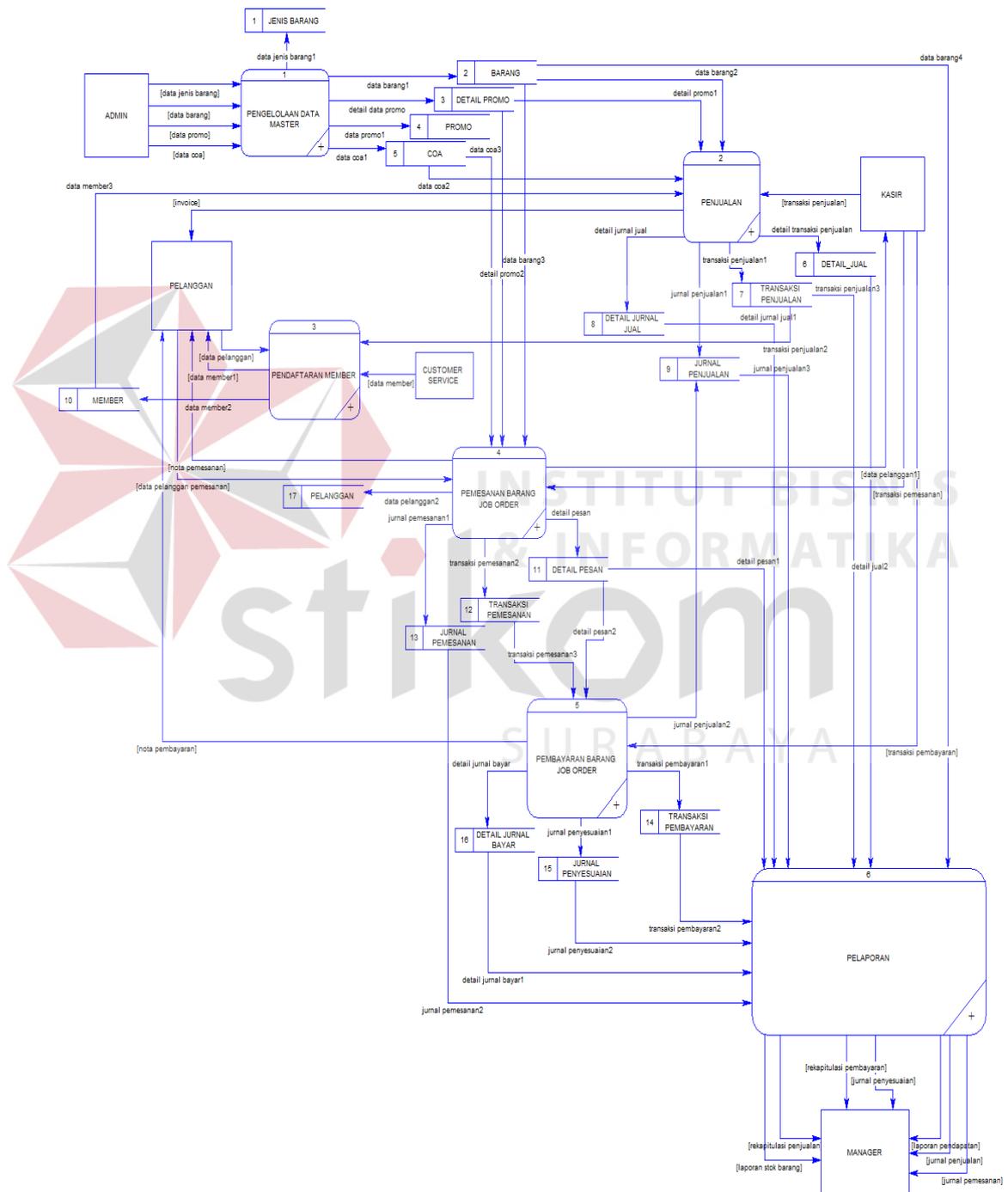
Gambar 3.9. *Hierarchy Input-Process-Output*

3.2.5. *Data Flow Diagram (DFD)*

Data Flow Diagram adalah gambaran aliran data pada suatu sistem, yang mempunyai tingkatan (memiliki beberapa level). Sistem didesain dari tingkatan konseptual tinggi, kemudian didesain sistem yang lebih detail. Pada tingkatan pertama ini mempunyai 4 proses yang hasil *decompose* dari *context diagram* yaitu proses pengelolaan data master, pembuatan data master, penjualan, dan pelaporan. Setelah itu setiap proses akan di *decompose* dan akan menghasilkan beberapa proses. *Data Flow Diagram* dari aplikasi penjualan dapat dilihat pada gambar 3.8. *Data Flow Diagram* dari hasil *decompose* dapat dilihat pada gambar 3.9. sampai gambar 3.12.

a. *Data Flow Diagram Level 0.*

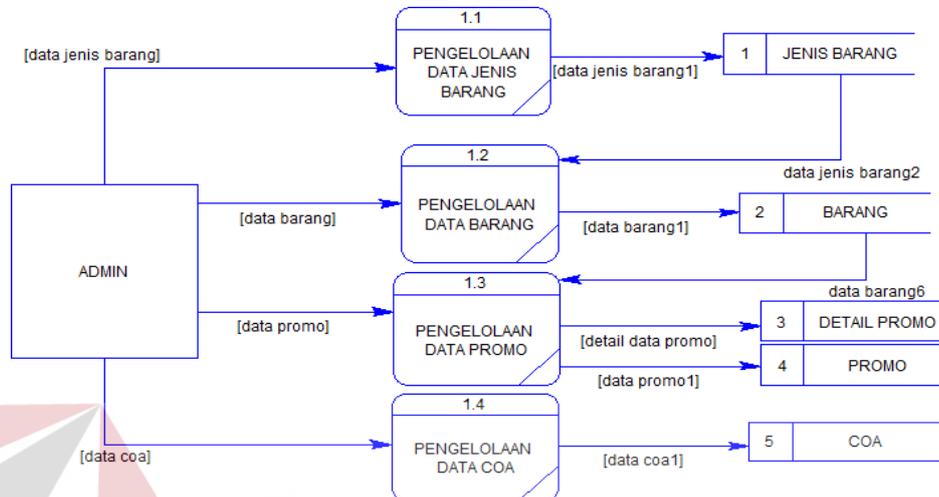
Setelah membuat HIPO, selanjutnya membuat *Data Flow Diagram (DFD)* Level 0 yaitu diagram yang lebih rinci dari *Context Diagram* yang ada pada gambar 3.10.



Gambar 3.10. *Data Flow Diagram Level 0.*

b. Data Flow Diagram Level 1 – Pengelolaan Data Master

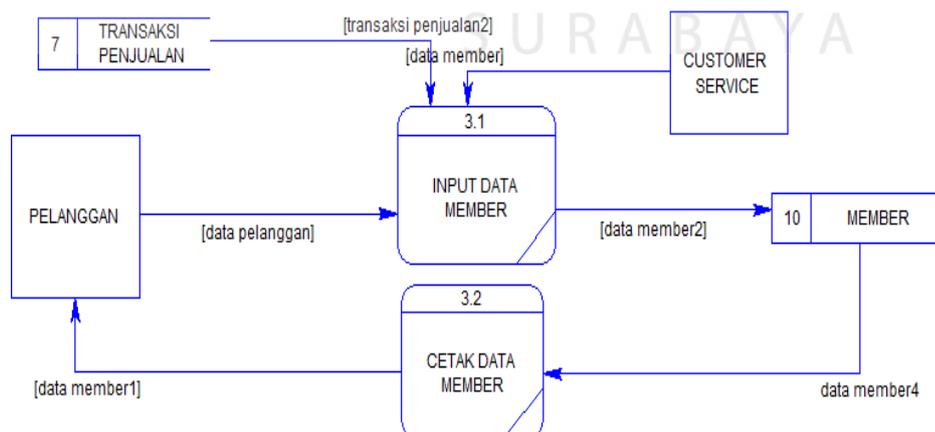
Gambar 3.11. adalah sub-proses pengelolaan data *master* yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung.



Gambar 3.11. Data Flow Diagram Level 1 – Pengelolaan Data Master

c. Data Flow Diagram Level 1 – Pembuatan data member

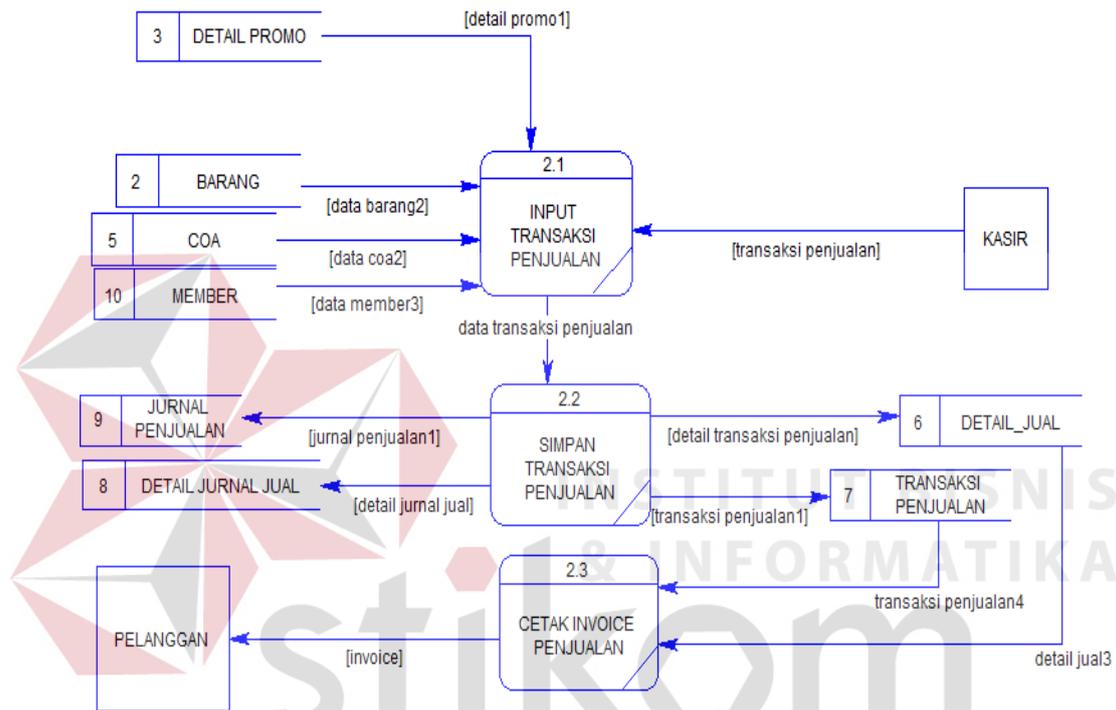
Gambar 3.12 adalah sub-proses pembuatan data *member* yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung. DFD *Level 1* ini bersiri dari proses input data *member* dan cetak data *member*.



Gambar 3.12. Data Flow Diagram Level 1 – Pembuatan data member

d. Data Flow Diagram Level 1 – Penjualan

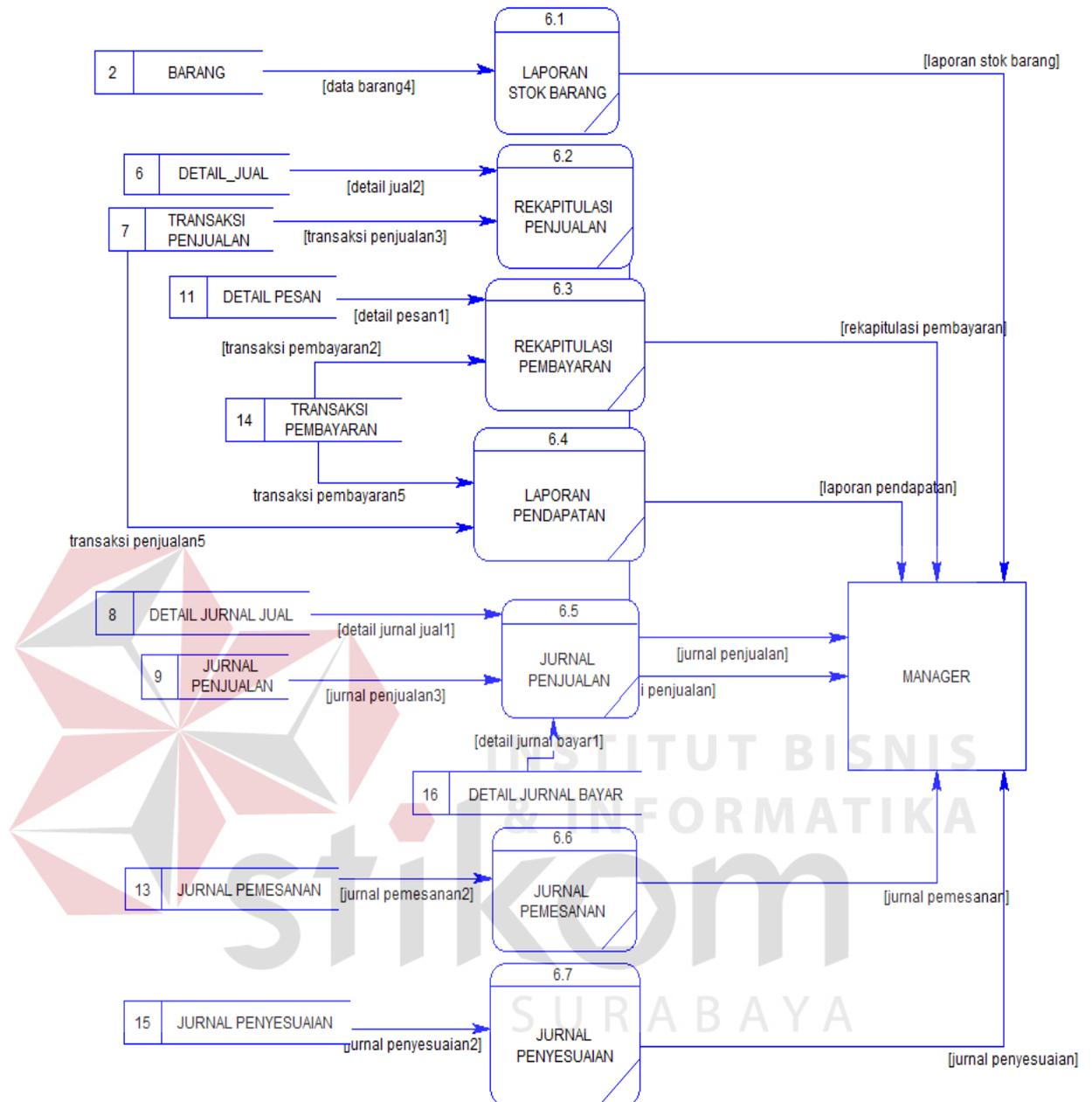
Gambar 3.13. adalah sub-proses penjualan yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung. DFD *Level 1* ini bersiri dari proses input transaksi penjualan, simpan transaksi penjualan, dan cetak invoice penjualan.



Gambar 3.13. Data Flow Diagram Level 1 – Penjualan

e. Data Flow Diagram Level 1 - Pelaporan

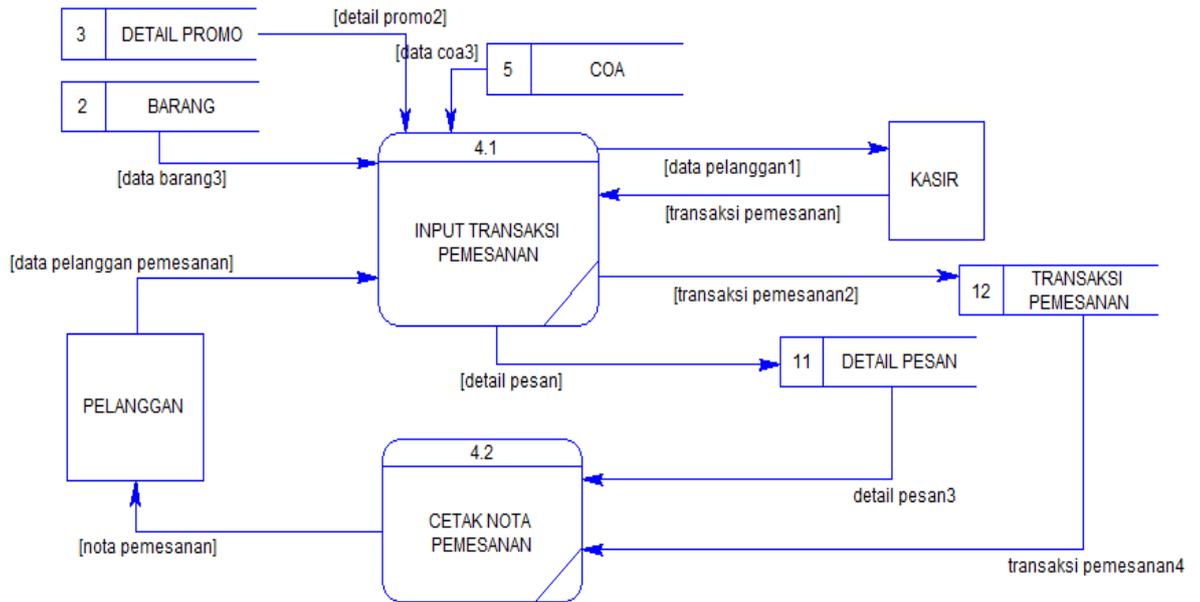
Gambar 3.14. adalah sub-proses pelaporan yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung. DFD *Level 1* ini bersiri dari proses laporan stok barang, rekapitulasi penjualan, rekapitulasi pembayaran, laporan pendapatan, jurnal penjualan, jurnal pemesanan, dan jurnal penyesuaian.



Gambar 3.14.. *Data Flow Diagram Level 1 – Pelaporan*

f. *Data Flow Diagram Level 1 – Pemesanan Barang Job Order*

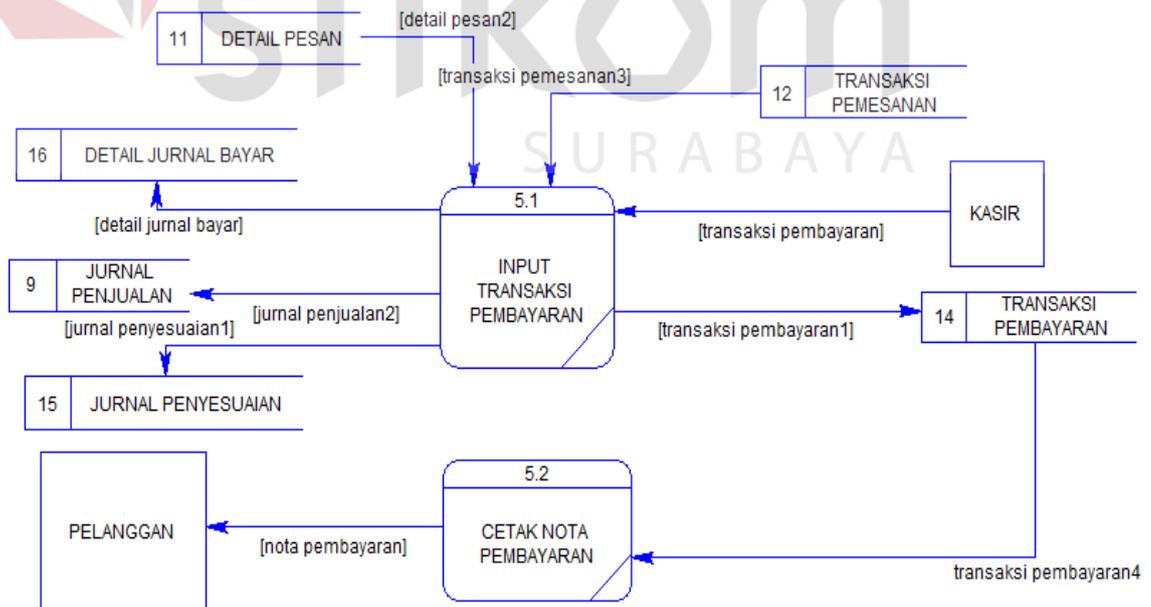
Gambar 3.15. adalah sub-proses pelaporan yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung. DFD *Level 1* ini bersiri dari proses input transaksi pemesanan, dan cetak nota pemesanan.



Gambar 3.15. Data Flow Diagram Level 1 – Pemesanan Barang Job Order

g. Data Flow Diagram Level 1 – Pembayaran Barang Job Order

Gambar 3.16. adalah sub-proses pelaporan yang ada pada DFD *Level 0* aplikasi penjualan pada Mitra 10 Wiyung. DFD *Level 1* ini bersiri dari proses input transaksi pembayaran, dan cetak nota pembayaran.



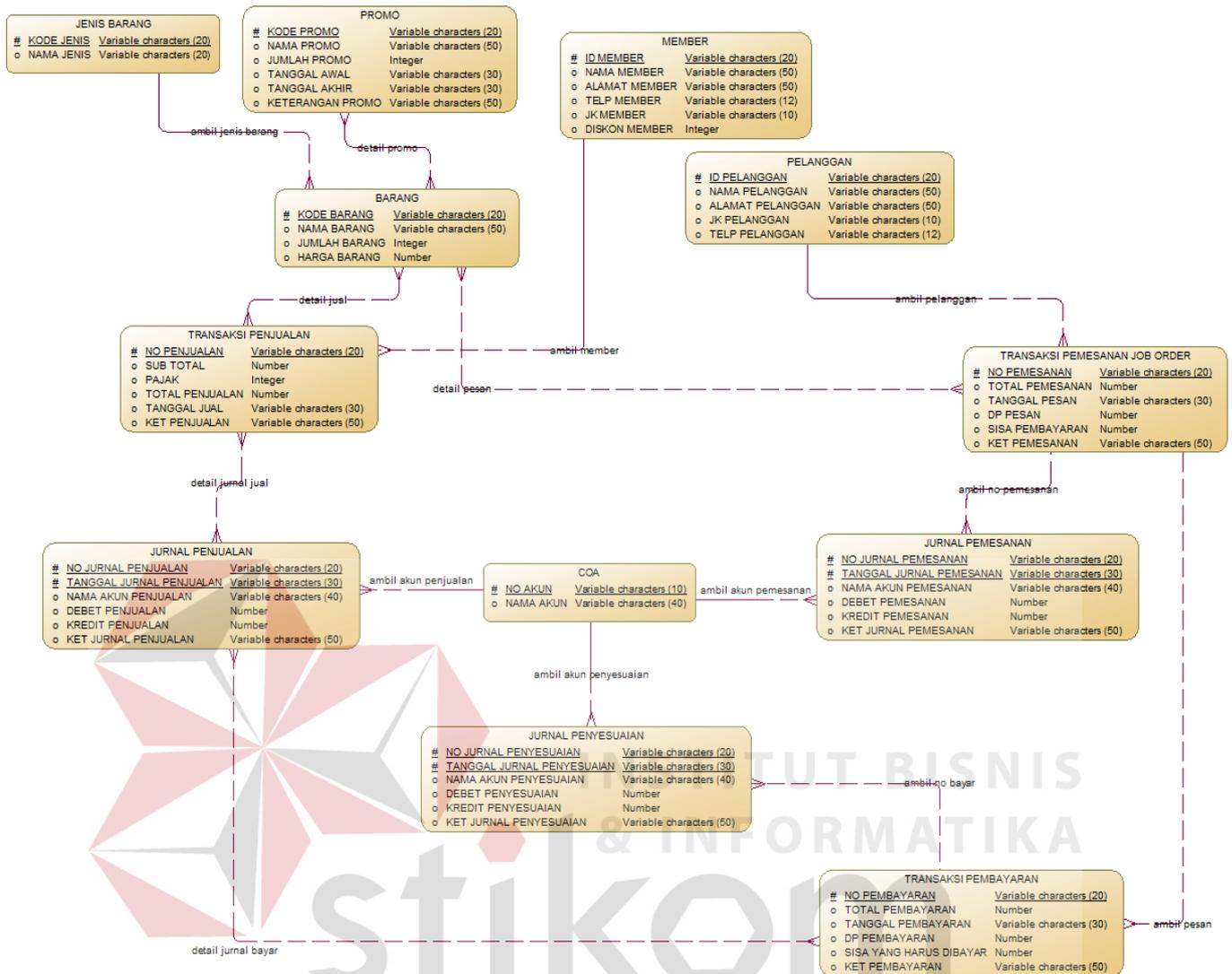
Gambar 3.16. Data Flow Diagram Level 1 – Pembayaran Barang Job Order

3.2.6. *Entity Relationship Diagram (ERD)*

ERD adalah suatu desain sistem yang digunakan untuk menggambarkan/mendokumentasikan kebutuhan untuk sistem pemrosesan *database*. ERD juga menunjukkan struktur keseluruhan dari data user dan hubungan (relasi) antar tabel. Data-data ERD tersebut digambarkan dengan menggunakan simbol *entity*. Dalam perancangan sistem ini terdapat beberapa *entity* yang saling terkait untuk menyediakan data-data yang dibutuhkan oleh sistem yaitu:

a. *Conceptual Data Model (CDM)*

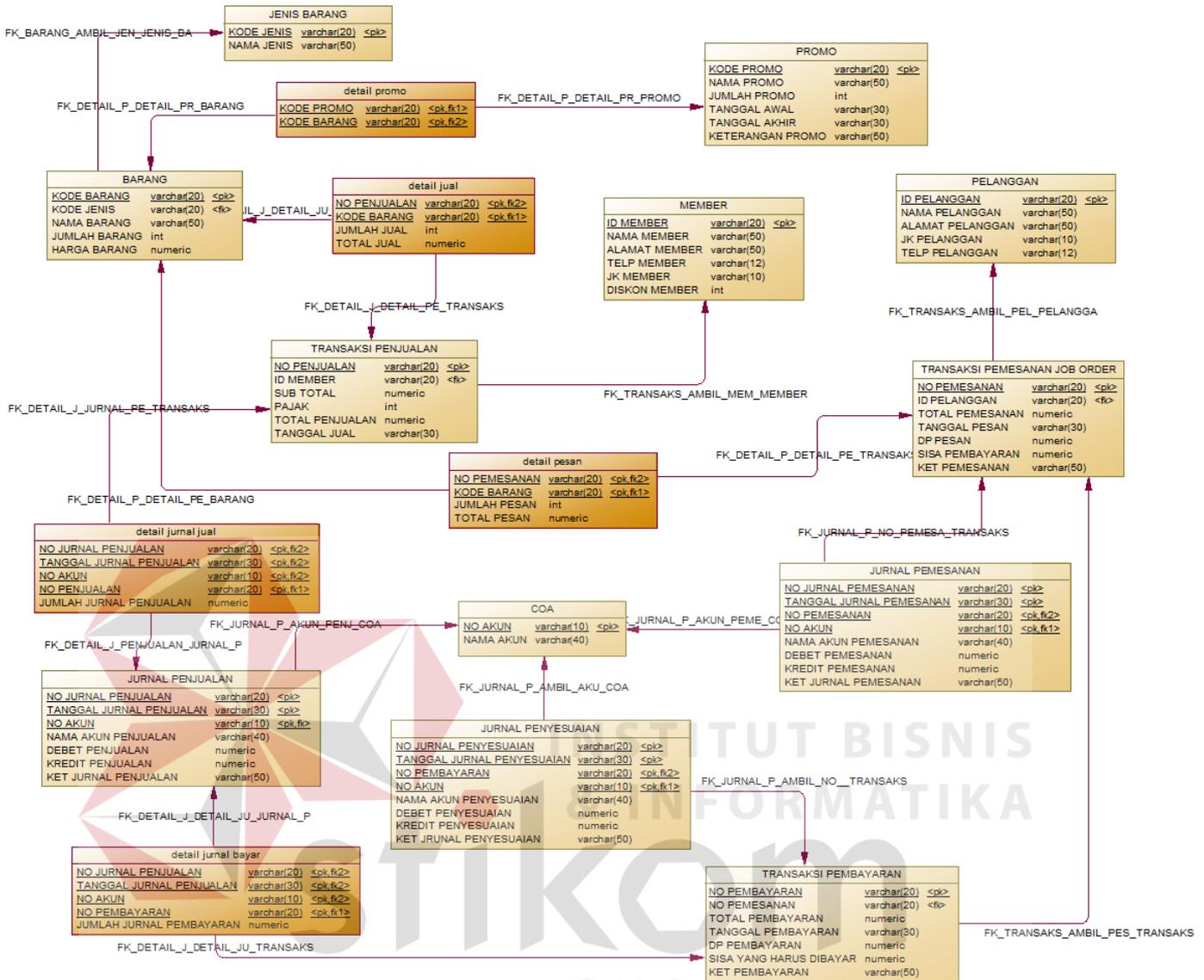
Conceptual Data Model menggambarkan konsep keseluruhan struktur basis data yang dirancang aplikasi. Dalam *CDM* belum tergambar jelas bentuk tabel penyusun basis data dan juga *field* yang terdapat pada setiap tabel. Tabel-tabel tersebut mempunyai *relationship* atau hubungan tetapi tidak terlihat pada kolom yang mana hubungan antar tabel tersebut. Dalam *CDM* juga didefinisikan kolom mana yang menjadi *primary key* dan kolom mana yang menjadi *foreign key*. *CDM* yang dirancang untuk aplikasi pada penelitian ini terdiri dari 12 tabel, yaitu tabel jenis barang, promo, member, barang, pelanggan, coa, transaksi penjualan, transaksi pemesanan job order, jurnal penjualan, jurnal pemesanan, jurnal penyesuaian, dan transaksi pembayaran. Adapun desain *CDM* dan hubungan antar tabel pada penelitian ini dapat dilihat pada gambar 3.17.



Gambar 3.17. Conceptual Data Model Aplikasi Penjualan

b. Physical Data Model (PDM)

Physical Data Model menggambarkan secara detail konsep rancangan struktur basis data yang dirancang untuk suatu program. *PDM* merupakan hasil *generate* dari *Conceptual Data Model* (CDM). Pada *PDM* tergambar jelas tabel-tabel penyusun basis data beserta *field-field* terdapat pada setiap tabel. Adapun *PDM* untuk aplikasi ini dapat dilihat pada gambar 3.18.



Gambar 3.18. *Physical Data Model* Aplikasi Penjualan

3.2.7. Struktur Basis Data

Dari *Physical Data Model* yang telah terbentuk, dapat disusun struktur basis data yang nantinya akan digunakan untuk menyimpan data yang diperlukan. Berikut ini akan diuraikan struktur tabel yang nantinya digunakan dalam pembuatan sistem seperti di bawah ini:

a. Tabel Jenis Barang

Nama Tabel : Jenis Barang

Primary Key : kode_jenis

Foreign Key : -

Fungsi :Tabel untuk memasukkan data jenis barang yang akan dijual

Tabel 3.1. Jenis Barang

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	kode_jenis	<i>Varchar</i>	20	<i>Primary Key</i>
2.	nama_jenis	<i>Varchar</i>	20	-

b. Tabel Promo

Nama Tabel : Promo

Primary Key : kode_promo

Foreign Key : -

Fungsi :Tabel untuk memasukkan data Promo yang ada

Tabel 3.2 Promo

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	kode_promo	<i>Varchar</i>	20	<i>Primary Key</i>
2.	nama_promo	<i>Varchar</i>	50	
3.	jumlah_promo	<i>Integer</i>		
4.	tanggal_awal	<i>Varchar</i>	30	
5.	tanggal_akhir	<i>Varchar</i>	30	
6.	keterangan_promo	<i>Varchar</i>	50	

c. Tabel Detail Promo

Nama Tabel : Detail_Promo

Primary Key : kode_promo, kode_barang

Foreign Key : kode_promo, kode_barang

Fungsi : Tabel untuk memasukkan data Detail Promo yang ada

Tabel 3.3 Detail Promo

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	kode_promo	Varchar	20	<i>Primary Key, Foreign Key 1</i>
2.	kode_barang	Varchar	50	<i>Primary Key, Foreign Key 2</i>

d. Tabel Barang

Nama Tabel : Barang

Primary Key : kode_barang

Foreign Key : kode_jenis

Fungsi : Tabel untuk memasukkan data barang yang akan dijual

Tabel 3.4. Barang

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	kode_barang	Varchar	20	<i>Primary Key</i>
2.	kode_jenis	Varchar	20	<i>Foreign Key</i>
3.	nama_barang	Varchar	50	
4.	jumlah_barang	Integer		
5.	harga_barang	Numeric		

e. Tabel Member

Nama Tabel : Member

Primary Key : id_member

Foreign Key : -

Fungsi :Tabel untuk memasukkan data member

Tabel 3.5. Member

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	id_member	Varchar	15	Primary Key
2.	nama_member	Varchar	50	
3.	alamat_member	Varchar	50	
4.	telp_member	Varchar	12	
5.	jk_member	Varchar	10	
6.	diskon_member	Integer		

f. Tabel COA

Nama Tabel : COA

Primary Key : no_akun

Foreign Key : nama_akun

Fungsi :Tabel untuk memasukkan data COA

Tabel 3.6. COA

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_akun	Varchar	10	Primary Key
2.	Nama_akun	Varchar	40	

g. Tabel Pelanggan

Nama Tabel : Pelanggan

Primary Key : id_pelanggan

Foreign Key : -

Fungsi :Tabel untuk memasukkan data pelanggan

Tabel 3.7. Pelanggan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	ID_pelanggan	Varchar	20	Primary Key
2.	Nama_pelanggan	Varchar	50	
3.	Alamat_pelanggan	Varchar	50	
4.	JK_pelanggan	Varchar	10	
5.	Telp_pelanggan	Varchar	12	

h. Tabel Transaksi Penjualan

Nama Tabel : Transaksi_Penjualan

Primary Key : no_penjualan

Foreign Key : id_member

Fungsi :Tabel untuk memasukkan transaksi penjualan yang dilakukan

Tabel 3.8. Transaksi Penjualan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_penjualan	Varchar	20	Primary Key
2.	ID_member	Varchar	20	Foreign Key
3.	Sub_total	Numeric		

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
4.	Pajak	<i>Integer</i>		
5.	Total_penjualan	<i>Numeric</i>		
6.	Tgl_jual	<i>Varchar</i>	30	

i. Tabel Detail Jual

Nama Tabel : Detail_Jual

Primary Key : kode_barang, no_penjualan

Foreign Key : kode_barang, no_penjualan

Fungsi :Tabel untuk memasukkan detail penjualan yang dilakukan

Tabel 3.9. Detail Jual

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	Kode_barang	<i>Varchar</i>	20	<i>Primary Key, Foreign Key 1</i>
2.	No_penjualan	<i>Varchar</i>	20	<i>Primary Key, Foreign Key 2</i>
3.	Jumlah_jual	<i>Integer</i>		
4.	Total_jual	<i>Numeric</i>		

j. Tabel Transaksi Pemesanan Job Order

Nama Tabel : Transaksi_Pemesanan_Job_Order

Primary Key : no_pemesanan

Foreign Key : id_pelanggan

Fungsi :Tabel untuk memasukkan data transaksi pemesanan job order

Tabel 3.10. Transaksi Pemesanan Job Order

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_pemesanan	Varchar	10	Primary Key
2.	ID_pelanggan	Varchar	20	Foreign Key
3.	Total_pemesanan	Numeric		
4.	Tanggal_pesanan	Varchar	30	
5.	DP_pesanan	Numeric		
6.	Sisa_pembayaran	Numeric		
7.	Ket_pemesanan	Varchar	50	

k. Tabel Detail Pesan

Nama Tabel : Detail_pesanan

Primary Key : no_pemesanan, kode_barang

Foreign Key : no_pemesanan, kode_barang

Fungsi : Tabel untuk memasukkan detail pemesanan job order

Tabel 3.11. Detail Pesan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_pemesanan	Varchar	20	Primary Key, Foreign Key 1
2.	Kode_barang	Varchar	20	Primary Key, Foreign Key 2
3.	Jumlah_pesanan	Integer		
4.	Total_pesanan	Numeric		

l. Tabel Transaksi Pembayaran

Nama Tabel : Transaksi_pembayaran

Primary Key : no_pembayaran

Foreign Key : no_pemesanan

Fungsi :Tabel untuk memasukkan data transaksi pembayaran

Tabel 3.12. Transaksi Pembayaran

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_pembayaran	Varchar	20	Primary Key
2.	No_pemesanan	Varchar	20	Foreign Key
3.	Total_pembayaran	Numeric		
4.	Tanggal_pembayaran	Varchar	30	
5.	DP_pembayaran	Numeric		
6.	Sisa_yang_harus_dibayar	Numeric		
7.	Ket_pembayaran	Varchar	50	

m. Tabel Detail Jurnal Bayar

Nama Tabel : Detail_jurnal_bayar

Primary Key : no_jurnal_penjualan, no_pembayaran,
Tanggal_pembayaran, no_akun

Foreign Key : no_jurnal_penjualan, no_pembayaran,
Tanggal_pembayaran, no_akun

Fungsi :Tabel untuk memasukkan detail jurnal pembayaran

Tabel 3.13. Detail Jurnal Bayar

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_jurnal_penjualan	<i>Varchar</i>	20	<i>Primary Key, Foreign Key 1</i>
2.	No_pembayaran	<i>Varchar</i>	20	<i>Primary Key, Foreign Key 2</i>
3.	Tanggal_pembayaran	<i>Varchar</i>	30	<i>Primary Key, Foreign Key 3</i>
4.	No_akun	<i>Varchar</i>	10	<i>Primary Key, Foreign Key 4</i>
5.	Jumlah_jurnal_pembayaran	<i>Numeric</i>		

n. Tabel Jurnal Penjualan

Nama Tabel : Jurnal_penjualan

Primary Key : no_jurnal_penjualan, Tanggal_jurnal_Penjualan, no_akun

Foreign Key : no_akun

Fungsi : Tabel untuk memasukkan data jurnal penjualan

Tabel 3.14. Jurnal Penjualan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_jurnal_penjualan	<i>Varchar</i>	20	<i>Primary Key</i>
2.	Tanggal_jurnal_Penjualan	<i>Varchar</i>	30	<i>Primary Key</i>
3.	No_akun	<i>Varchar</i>	10	<i>Primary Key, Foreign Key 1</i>
4.	Nama_akun_penjualan	<i>Varchar</i>	40	
5.	Debet_penjualan	<i>Numeric</i>		
6.	Kredit_penjualan	<i>Numeric</i>		
7.	Ket_jurnal_penjualan	<i>Varchar</i>	50	

o. Tabel Detail Jurnal Jual

Nama Tabel : Detail_jurnal_jual

Primary Key : no_jurnal_penjualan, no_penjualan,
Tanggal_jurnal_penjualan, no_akun

Foreign Key : no_jurnal_penjualan, no_penjualan,
Tanggal_jurnal_penjualan, no_akun

Fungsi :Tabel untuk memasukkan detail jurnal penjualan

Tabel 3.15. Detail Jurnal Penjualan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_jurnal_penjualan	Varchar	20	Primary Key, Foreign Key 1
2.	No_penjualan	Varchar	20	Primary Key, Foreign Key 2
3.	Tanggal_jurnal_penjualan	Varchar	30	Primary Key, Foreign Key 3
4.	No_akun	Varchar	10	Primary Key, Foreign Key 4
5.	Jumlah_jurnal_penjualan	Numeric	10	

p. Tabel Jurnal Penyesuaian

Nama Tabel : Jurnal_penyesuaian

Primary Key : no_jurnal_penyesuaian, Tanggal_jurnal_Penyesuaian,
no_akun

Foreign Key : no_pembayaran, no_akun

Fungsi :Tabel untuk memasukkan data jurnal penyesuaian

Tabel 3.16. Jurnal Penyesuaian

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_jurnal_penyesuaian	Varchar	20	Primary Key
2.	Tanggal_jurnal_Penyesuaian	Varchar	30	Primary Key
4.	No_pembayaran	Varchar	20	Foreign Key 1
3.	No_akun	Varchar	10	Primary Key, Foreign Key 2
4.	Nama_akun_penyesuaian	Varchar	40	
5.	Debet_penyesuaian	Numeric		
6.	Kredit_penyesuaian	Numeric		
7.	Ket_jurnal_penyesuaian	Varchar	50	

q. Tabel Jurnal Pemesanan

Nama Tabel : Jurnal_pemesanan

Primary Key : no_jurnal_pemesanan, Tanggal_jurnal_Pemesanan,
no_akun

Foreign Key : no_pemesanan, no_akun,

Fungsi :Tabel untuk memasukkan data jurnal pemesanan

Tabel 3.17. Jurnal Pemesanan

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
1.	No_jurnal_pemesanan	Varchar	20	Primary Key
2.	Tanggal_jurnal_Pemesanan	Varchar	30	Primary Key
4.	No_pemesanan	Varchar	20	Foreign Key 1
3.	No_akun	Varchar	10	Primary Key, Foreign Key 2
4.	Nama_akun_pemesanan	Varchar	40	
5.	Debet_pemesanan	Numeric		

<i>No.</i>	<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Key</i>
6.	Kredit_pemesanan	Numeric		
7.	Ket_jurnal_pemesanan	Varchar	50	

3.2.8. Desain Interface Input – Output

Desain *input* dan *output* dibuat untuk menggambarkan jalannya sistem. Desain *input* dan *output* adalah suatu rancangan dari *form-form* yang mengimplementasikan masukan dan rancangan keluaran yang berupa laporan-laporan yang akan digunakan sebagai dokumentasi.

Berikut adalah desain *input* dan *output*:

a. Desain Input

1. Desain Form Login

Form login terdiri dari dua *textbox*, yaitu nama pengguna (*username*) dan kata sandi (*password*), dan tombol untuk *login* dan juga tombol untuk kembali (*back*) seperti pada gambar 3.19.



Gambar 3.19. Form Login

2. Desain *Form* Menu Utama

Form ini adalah menu utama setelah user melakukan *login*. *Form* menu utama terdiri dari sub menu untuk *form* lainnya, seperti *form master* yang berisi pelanggan, member, jenis barang, promo, coa, dan barang. Kemudian *form* transaksi yang terdiri dari penjualan, pemesanan, dan pembayaran. Kemudian *form* laporan yang berisi stok barang, rekap penjualan, pendapatan, dan jurnal. Seperti terlihat pada gambar 3.20.



Gambar 3.20. *Form* Menu utama

3. Desain *form Master*

Form Master adalah *form* yang berfungsi untuk *input* data, yaitu data pegawai, data member, data jenis barang, data diskon, dan data barang.

Contohnya pada *form master member*, data yang dimasukkan adalah *id member*, nama, alamat, no telp, dan jenis kelamin. Untuk *form* yang lain terlihat pada gambar 3.21 – gambar 3.26.

Gambar 3.21. *Form Master Member*

Gambar 3.21. merupakan desain dari *form master member*. *Form master member* digunakan untuk menyimpan data member di Mitra 10 Wiyung. Pada form ini terdapat tombol simpan untuk meyimpan data member. Selain itu, ada juga tombol ubah untuk megubah data member dan tombol hapus untuk menghapus data member yang ada.

Gambar 3.22. *Form Master Jenis Barang*

Form master jenis barang yang ada pada gambar 3.22. merupakan *form* yang digunakan untuk menyimpan data jenis barang. Pada *form* ini terdapat tombol simpan untuk menyimpan data jenis barang. Terdapat juga

tombol ubah untuk mengubah data jenis barang yang ada dan hapus untuk menghapus data jenis barang dari *database*.

Gambar 3.23. *Form Master Promo*

Form master promo digunakan untuk menyimpan data promo yang ada. Pada *form* ini terdapat tombol simpan untuk menyimpan data ke *database*, tombol ubah untuk mengubah data dan tombol hapus untuk menghapus data dari *database* seperti pada gambar 3.23.

Gambar 3.24. *Form Master Barang*

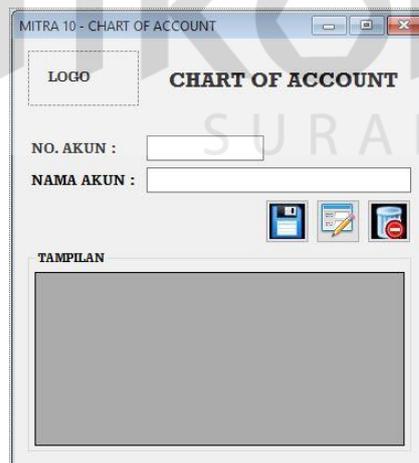
Pada gambar 3.24. merupakan *form master* barang yang digunakan untuk menyimpan data barang. Pada *form* ini terdapat tombol simpan data

barang ke *database*. Ada juga tombol ubah untuk mengubah data barang dan tombol hapus untuk menghapus data barang dari *database*.



Gambar 3.25. *Form Master Pelanggan*

Pada gambar 3.25. merupakan *form master* pelanggan yang digunakan untuk menyimpan data pelanggan. Pada *form* ini terdapat tombol simpan data pelanggan ke *database*. Ada juga tombol ubah untuk mengubah data pelanggan dan tombol hapus untuk menghapus data pelanggan dari *database*.



Gambar 3.26. *Form Master COA*

Pada gambar 3.26. merupakan *form master* COA yang digunakan untuk menyimpan data COA. Pada *form* ini terdapat tombol simpan data

COA ke *database*. Ada juga tombol ubah untuk mengubah data COA dan tombol hapus untuk menghapus data COA dari *database*.

4. Desain *form* Transaksi

Form Transaksi adalah *form* yang digunakan saat proses transaksi penjualan. Terdapat 3 jenis transaksi yaitu transaksi penjualan, transaksi pemesanan, dan transaksi pembayaran. Di dalam *form* penjualan terdapat kolom untuk memasukkan data transaksi yaitu tanggal transaksi, nomor transaksi, kode barang, nama barang, harga barang, diskon, harga promo, jumlah barang, dan total harga. Terdapat juga kolom untuk sub total, pajak total penjualan, tunai, kembali, dan id member. Form juga terdiri dari bagian tampilan untuk melihat data transaksi, dan tombol untuk menyimpan, menghapus, dan mencetak, dan mencari *member* yang sudah ada seperti terlihat pada gambar 3.27.

MITRA 10 - TRANSAKSI PENJUALAN

LOGO

PENJUALAN

TANGGAL : 28/08/2016

NO. TRANSAKSI :

KODE BARANG : Cari

NAMA BARANG :

HARGA :

DISKON :

HARGA PROMO :

JUMLAH :

TOTAL HARGA : +

TUNAI :

KEMBALI :

ID MEMBER : 🔍

TAMPILAN

SUB TOTAL :

PAJAK :

TOTAL JUAL :

KETERANGAN :

💾 🗑️

Gambar 3.27. *Form* Transaksi Penjualan

Di dalam *form* pemesanan terdapat kolom untuk memasukkan data pemesanan yaitu tanggal transaksi, nomor transaksi, kode barang, nama barang, harga barang, diskon, harga promo, jumlah barang, dan total harga. Terdapat juga kolom untuk sub total, pajak total penjualan, keterangan, uang muka, sisa pembayaran, dan id pelanggan. Form juga terdiri dari bagian tampilan untuk melihat data transaksi, dan tombol untuk menyimpan, menghapus, mencetak, membuat data pelanggan baru, dan mencari data pelanggan yang sudah ada, seperti terlihat pada gambar 3.28.

The image shows a screenshot of a software application window titled "MITRA 10 - TRANSAKSI PEMESANAN". The main area is a form for creating a transaction. It has several sections:

- PEMESANAN**: Contains fields for "TANGGAL" (29/08/2016), "NO. TRANSAKSI" (TS0003), "KODE BARANG", "NAMA BARANG", "HARGA", "DISKON", "HARGA PROMO", "JUMLAH", "TOTAL HARGA", "SUB TOTAL", "PAJAK" (10), "TOTAL JUAL", and "KETERANGAN".
- Payment Information**: Fields for "UANG MUKA", "SISA PEMBAYARAN", and "ID PELANGGAN".
- TAMPILAN**: A separate window with a search icon and a magnifying glass icon.

At the bottom of the form, there are icons for saving (floppy disk), deleting (trash), and printing (printer).

Gambar 3.28. *Form* Transaksi Pemesanan

Di dalam *form* pembayaran terdapat kolom untuk memasukkan data pembayaran yaitu tanggal, nomor pemesanan, tanggal pemesanan, pembayaran dp, pembayaran sisa, dan total. Terdapat juga kolom untuk tunai, kembali, dan id member / id pelanggan. Form juga terdiri dari tombol untuk menyimpan, dan mencari data pelanggan atau data *member* yang sudah ada, seperti terlihat pada gambar 3.29.

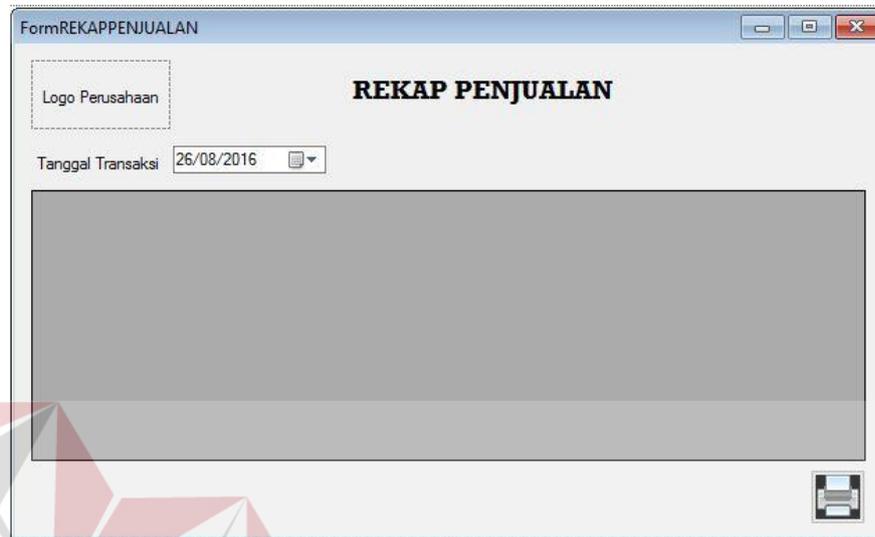
Gambar 3.29. *Form* Transaksi Pembayaran

5. Desain *form* Laporan

Form laporan berfungsi untuk menampilkan laporan hasil penyimpanan data untuk memberikan informasi terkait yang dibutuhkan. Dalam Laporan terdapat tiga *form* yaitu *form* stok barang untuk melaporkan persediaan stok barang tertentu, *form* rekapitulasi penjualan untuk melaporkan dan menampilkan hasil penjualan pada tanggal tertentu, dan *form* jurnal untuk menampilkan dan melaporkan jurnal yang diterima seperti terlihat pada gambar 3.30 – gambar 3.34.

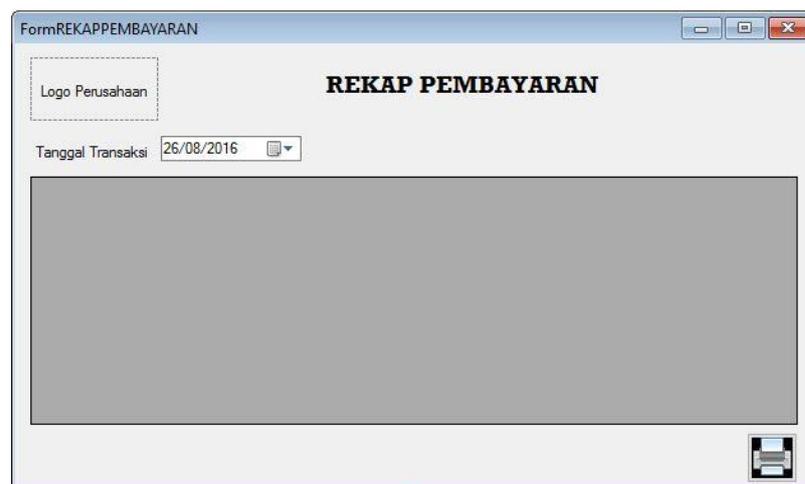
Gambar 3.30. *Form* Laporan Stok Barang

Pada gambar 3.30. merupakan *form* laporan stok barang. Pada *form* ini, tampilan *datagridview* akan data barang muncul secara otomatis. Dan ada tombol *print* yang akan langsung tersambung ke *crystal report* untuk di cetak.



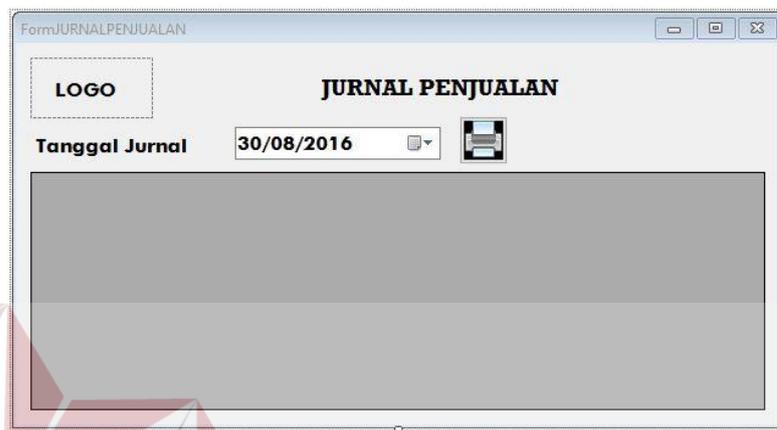
Gambar 3.31. *Form* Rekapitulasi Penjualan

Pada gambar 3.31. merupakan *form* rekapitulasi penjualan yang digunakan untuk membuat laporan rekapitulasi penjualan. Pada *form* ini terdapat tanggal rekap, *datagridview*, dan tombol cetak untuk mencetak data penjualan yang ada.



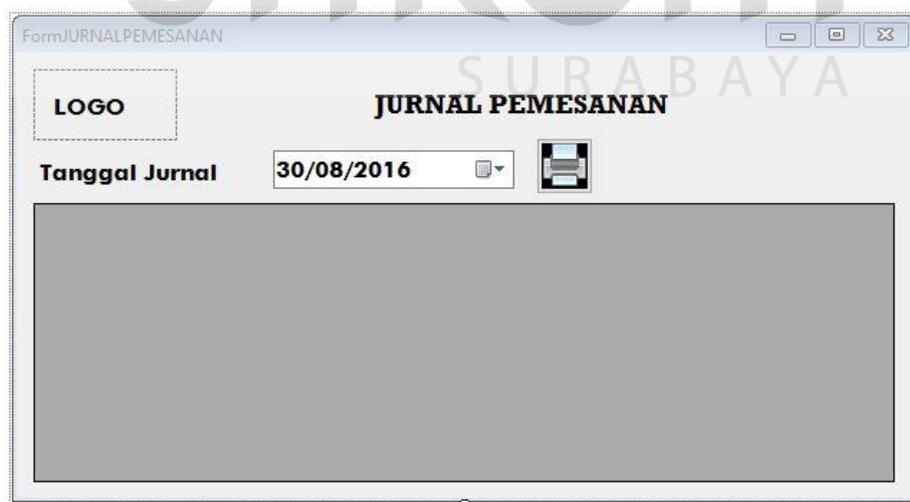
Gambar 3.32. *Form* Rekapitulasi Pembayaran

Pada gambar 3.32. merupakan *form* rekapitulasi penjualan yang digunakan untuk membuat laporan rekapitulasi pembayaran. Pada *form* ini terdapat tanggal rekap, *datagridview*, dan tombol cetak untuk mencetak data penjualan yang ada.



Gambar 3.33. *Form* Jurnal Penjualan

Pada gambar 3.33. merupakan *form* jurnal penjualan yang digunakan untuk membuat jurnal penjualan. Pada *form* ini terdapat tanggal jurnal, *datagridview*, dan tombol cetak untuk mencetak jurnal penjualan.



Gambar 3.34. *Form* Jurnal Pemesanan

Pada gambar 3.34. merupakan *form* jurnal penjualan yang digunakan untuk membuat jurnal pemesanan. Pada *form* ini terdapat tanggal jurnal, *datagridview*, dan tombol cetak untuk mencetak jurnal pemesanan.

Gambar 3.35. *Form* Jurnal Penyesuaian

Pada gambar 3.35. merupakan *form* jurnal penyesuaian yang digunakan untuk membuat jurnal penyesuaian. Pada *form* ini terdapat tanggal jurnal, *datagridview*, dan tombol cetak untuk mencetak jurnal penyesuaian.

3.3. Desain Uji Coba

Desain uji coba digunakan untuk memberikan gambaran fungsi dari sistem ini. Adapun hal hal yang akan diujikan adalah sebagai berikut.

3.3.1. Desain uji coba *Form Master*

1. Desain Uji Coba *Form Master Member*

User akan memasukkan data member baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem dan member yang telah tersimpan akan mendapatkan keuntungan berupa diskon potongan harga sebesar 20% bila melakukan transaksi. Program akan *error* bila salah satu

kolom pada data dari *member* yang tidak diisi. Kemudian pengguna dapat menyimpan data *member* baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.18.

Tabel 3.18. *Requirement Form Master Member*

Tujuan	Requirement yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang <i>member</i> .	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus

Untuk desain uji coba ini dapat dilihat pada tabel 3.19.

Tabel 3.19. Desain Uji Coba *Form Master Member*

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master Member</i>	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master Member</i>	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master Member</i>	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah

2. Desain Uji Coba *Form Master Jenis Barang*

User akan memasukkan data jenis barang baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Kode jenis barang akan langsung terbaharui dengan bertambahnya jenis barang baru pada *database*. Program akan *error* bila salah satu kolom pada data dari jenis barang yang tidak diisi. Kemudian pengguna dapat menyimpan data jenis barang baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada table 3.20.

Tabel 3.20. *Requirement Form Master Jenis barang*

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang jenis barang.	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data d. Kode Jenis Barang otomatis diperbaharui	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus c. Kode barang akan otomatis berubah dengan adanya penambahan data baru.

Untuk desain uji coba ini dapat dilihat pada tabel 3.21.

Tabel 3.21. Desain Uji Coba *Form Master Jenis Barang*

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master Jenis Barang</i>	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master Jenis Barang</i>	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master Jenis Barang</i>	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah
3	Menampilkan kode jenis barang	Aplikasi sudah terinstal	Menambah data jenis barang baru di <i>database</i>	Kode jenis barang yang ada di <i>textbox</i> , <i>datagridview</i> , dan <i>database</i> otomatis berubah dengan bertambahnya data	Kode jenis barang yang ada di <i>textbox</i> , <i>datagridview</i> , dan <i>database</i> otomatis berubah dengan bertambahnya data

3. Desain Uji Coba *Form Master* Barang

User akan memasukkan data barang baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Program akan *error* bila salah satu kolom pada data dari barang yang tidak diisi. Kemudian pengguna dapat menyimpan data barang baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel

3.22.

Tabel 3.22. *Requirement Form Master* barang

Tujuan	Requirement yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang barang.	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus

Untuk desain uji coba ini dapat dilihat pada tabel 3.23.

Tabel 3.23. Desain Uji Coba *Form Master* Barang

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master</i> Barang	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master</i> Barang	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master</i> Barang	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah

4. Desain Uji Coba *Form Master Promo*

User akan memasukkan data promo baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Program akan *error* bila salah satu kolom pada data dari promo yang tidak diisi. Kemudian pengguna dapat menyimpan data promo baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Kemudian terdapat tombol tambah untuk menambah daftar barang yang termasuk dalam promo. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.24.

Tabel 3.24. *Requirement Form Master Promo*

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang promo.	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data d. Tombol tambah data	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus c. Ketika tombol tambah ditekan, kode barang yang di inputkan akan tergabung dalam promo

Untuk desain uji coba ini dapat dilihat pada tabel 3.25.

Tabel 3.25. Desain Uji Coba *Form Master* Promo

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master</i> Promo	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master</i> Promo	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master</i> Promo	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah
4	Menguji fungsi tombol tambah data pada <i>form Master</i> Promo	Aplikasi sudah terinstal	Menekan tombol tambah data	Data yang terdapat pada kolom akan muncul di <i>datagridview</i>	Data yang terdapat pada kolom akan muncul di <i>datagridview</i>

5. Desain Uji Coba *Form Master* Pelanggan

User akan memasukkan data pelanggan baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Program akan *error* bila salah satu kolom pada data dari pelanggan yang tidak diisi. Kemudian pengguna dapat menyimpan data pelanggan baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.26.

Tabel 3.26. *Requirement Form Master* Pelanggan

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang pelanggan.	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus

Untuk desain uji coba ini dapat dilihat pada tabel 3.27.

Tabel 3.27. Desain Uji Coba *Form Master* Pelanggan

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master</i> Pelanggan	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master</i> Pelanggan	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master</i> Pelanggan	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah

6. Desain Uji Coba *Form Master* COA

User akan memasukkan data COA baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Program akan *error* bila salah satu kolom pada data dari COA yang tidak diisi. Kemudian pengguna dapat menyimpan data COA baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mengubah dan menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus atau ubah. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel

3.28.

Tabel 3.28. *Requirement Form Master* COA

Tujuan	Requirement yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang COA.	a. Tombol simpan data b. Tombol hapus data c. Tombol ubah data	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus atau ubah ditekan, data yang sudah tersimpan dapat diubah atau dihapus

Untuk desain uji coba ini dapat dilihat pada tabel 3.29.

Tabel 3.29. Desain Uji Coba *Form Master COA*

Test Case ID	Test Case	Pre Condition	Test Step	Test Data	Output yang diharapkan
1	Menguji fungsi tombol simpan data pada <i>form Master COA</i>	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master COA</i>	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol ubah data pada <i>form Master COA</i>	Aplikasi sudah terinstal	Menekan tombol ubah data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat diubah

3.3.2. Desain uji coba *Form* Transaksi Penjualan

User akan memasukkan data transaksi baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Total harga didapat dari rumus: $total = (jumlah \times harga) - (jumlah \times harga \times diskon)$(3.1)

Sub total adalah total sementara dari seluruh barang yang masuk proses transaksi. Dan total harga merupakan total keseluruhan dari harga semua barang dalam satu transaksi. Kemudian terdapat kolom pajak yang berisi pajak penjualan. Total penjualan didapat dari rumus:

$$total\ penjualan = subtotal + \left(\frac{subtotal \times pajak}{100} \right) \dots\dots\dots(3.2)$$

Kemudian tunai adalah jumlah uang yang diserahkan oleh pelanggan kepada kasir, dan kembali adalah jumlah uang kembali. Uang kembali didapatkan dengan rumus:

$$kembali = tunai - total\ penjualan \dots\dots\dots(3.3)$$

Tampilan akan menampilkan daftar barang yang dibeli dalam satu kali transaksi. Program akan *error* bila salah satu kolom pada data dari transaksi yang tidak diisi. Kolom ID *member* harus diisi terlebih dahulu untuk dapat menampilkan total jual. Jika pelanggan merupakan *member*, maka akan mendapat diskon member sebesar 30% di kolom total jual. Total penjualan jika sudah terdaftar menjadi member didapat dengan rumus:

$$total\ penjualan1 = subtotal + \left(\frac{subtotal \times pajak}{100} \right) \dots\dots\dots(3.4)$$

$$total\ penjualan = total\ penjualan1 - (total\ penjualan1 \times 0.3) \dots\dots\dots(3.5)$$

Jika pelanggan bukan *member*, namun berbelanja hingga melebihi atau sama dengan Rp 1.000.000, -, maka akan muncul suatu pertanyaan apakah pelanggan ingin mendaftar menjadi member. Jika memilih ya, maka akan tampil *form master member*. Kemudian pengguna dapat menambah data barang baru pada transaksi tersebut dengan menekan tombol tambah dan akan muncul di kolom tampilan. Untuk menyimpan data transaksi, pengguna dapat menekan tombol simpan, kemudian untuk menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus. Untuk mencetak bukti transaksi, pengguna dapat menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.30.

Tabel 3.30. *Requirement Form* Transaksi Penjualan

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang proses transaksi.	a. Tombol simpan data b. Tombol hapus data c. Tombol cetak data d. Hasil perkalian antara diskon, harga barang, dan jumlah barang e. Hasil perkalian antara total harga dan pajak	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus ditekan, data yang sudah tersimpan dapat dihapus c. Total harga didapat setelah jumlah barang dikalikan dengan harga barang dan diskon. d. Total penjualan didapat setelah total harga dikalikan dengan pajak

Untuk desain uji coba ini dapat dilihat pada tabel 3.31.

Tabel 3.31. Desain Uji Coba *Form* Transaksi Penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji fungsi tombol simpan data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol cetak data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol cetak data	Data transaksi yang telah dilakukan dapat langsung dicetak	Data transaksi yang telah dilakukan dapat langsung dicetak
4	Menampilkan hasil perhitungan total harga	Aplikasi sudah terinstal	Menekan tombol tambah atau menambah barang baru	Data akan muncul di kolom total harga dan <i>datagridview</i>	Data akan muncul di kolom total harga dan <i>datagridview</i>

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
5	Menampilkan hasil perhitungan total penjualan	Aplikasi sudah terinstal	Jika total harga sudah didapatkan, kemudian total harga dikalikan dengan pajak	Data akan muncul di kolom total penjualan dan <i>datagridview</i>	Data akan muncul di kolom total penjualan dan <i>datagridview</i>



3.3.3. Desain uji coba *Form* Transaksi Pemesanan

User akan memasukkan data pemesanan baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Total harga didapat dari rumus: $total = (jumlah \times harga) - (jumlah \times harga \times diskon)$(3.1)

Sub total adalah total sementara dari seluruh barang yang masuk proses transaksi. Dan total harga merupakan total keseluruhan dari harga semua barang dalam satu transaksi. Kemudian terdapat kolom pajak yang berisi pajak penjualan. Total penjualan didapat dari rumus:

$$total\ penjualan = subtotal + \left(\frac{subtotal \times pajak}{100} \right) \dots\dots\dots(3.2)$$

Kemudian uang muka adalah jumlah uang yang dibayarkan di muka oleh pelanggan kepada kasir sebesar 20% dari total jual, dan sisa pembayaran adalah jumlah sisa yang harus dibayarkan oleh pelanggan. Uang muka dan sisa pembayaran didapatkan dengan rumus:

$$uang\ muka = total\ penjualan \times 0.2 \dots\dots\dots(3.3)$$

$$sisa\ pembayaran = total\ penjualan - uang\ muka \dots\dots\dots(3.4)$$

Tampilan akan menampilkan daftar barang yang dibeli dalam satu kali transaksi. Program akan *error* bila salah satu kolom pada data dari transaksi yang tidak diisi.

Kemudian pengguna dapat menambah data barang baru pada transaksi tersebut di dengan menekan tombol tambah dan akan muncul di kolom tampilan. Untuk menyimpan data pemesanan, pengguna dapat menekan tombol simpan, kemudian untuk menghapus data, pengguna dapat memilih pada kolom data yang akan dihapus, kemudian tekan tombol hapus. Untuk mencetak bukti pemesanan,

pengguna dapat menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.32.

Tabel 3.32. *Requirement Form* Transaksi Pemesanan

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang proses transaksi.	a. Tombol simpan data b. Tombol hapus data c. Tombol cetak data d. Hasil perkalian antara diskon, harga barang, dan jumlah barang e. Hasil perkalian antara total harga dan pajak f. Hasil perhitungan uang muka dan sisa pembayaran.	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol hapus ditekan, data yang sudah tersimpan dapat dihapus c. Total harga didapat secara otomatis setelah jumlah barang dikalikan dengan harga barang dan diskon. d. Total penjualan didapat secara otomatis setelah total harga dikalikan dengan pajak e. Total penjualan jika terdaftar <i>member</i> didapat secara otomatis setelah total harga dikalikan dengan pajak dan diskon member

Untuk desain uji coba ini dapat dilihat pada tabel 3.33.

Tabel 3.33. Desain Uji Coba *Form* Transaksi Pemesanan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji fungsi tombol simpan data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol hapus data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol hapus data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat dihapus
3	Menguji fungsi tombol cetak data pada <i>form Master Transaksi</i>	Aplikasi sudah terinstal	Menekan tombol cetak data	Data transaksi yang telah dilakukan dapat langsung dicetak	Data transaksi yang telah dilakukan dapat langsung dicetak
4	Menampilkan hasil perhitungan total harga	Aplikasi sudah terinstal	Menekan tombol tambah atau menambah barang baru	Data akan muncul di kolom total harga dan <i>datagridview</i>	Data akan muncul di kolom total harga dan <i>datagridview</i>

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
5	Menampilkan hasil perhitungan total penjualan	Aplikasi sudah terinstal	Jika total harga sudah didapatkan, kemudian total harga dikalikan dengan pajak	Data akan muncul di kolom total penjualan dan <i>datagridview</i>	Data akan muncul di kolom total penjualan dan <i>datagridview</i>
6	Menampilkan hasil perhitungan total penjualan <i>member</i>	Aplikasi sudah terinstal	Jika total harga sudah didapatkan, kemudian total harga dikalikan dengan pajak dan diskon <i>member</i>	Data akan muncul di kolom total penjualan dan <i>datagridview</i>	Data akan muncul di kolom total penjualan dan <i>datagridview</i>



3.3.4. Desain uji coba *Form* Transaksi Pembayaran

Pengguna akan memasukkan data pembayaran baru ke dalam database dengan tujuan agar data tersebut dapat disimpan kedalam sistem. Program akan *error* bila salah satu kolom pada data dari pembayaran yang tidak diisi. Kolom no transaksi akan otomatis terisi sesuai dengan banyaknya transaksi yang tersimpan di *database*. Pengguna dapat menginputkan no pemesanan, setelah itu kolom tanggal pesan, pembayaran dp, pembayaran sisa, total, dan id *member*. pelanggan kemudian dapat mengisi kolom tunai, dan kolom kembali akan otomatis terisi. Kemudian pengguna dapat menyimpan data pembayaran baru di *database* dengan menekan tombol simpan dan akan muncul di kolom tampilan. Untuk mencari id *member* pengguna dapat menekan tombol cari. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.34.

Tabel 3.34. *Requirement Form* Transaksi Pembayaran

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang proses transaksi pembayaran.	a. Tombol simpan data b. Tombol cari <i>member</i> c. Hasil pengurangan antara tunai dan pembayaran sisa.	a. Ketika tombol simpan ditekan, data akan tersimpan pada database b. Ketika tombol cari ditekan, program mencari data yang sudah tersimpan di <i>database</i> c. Kembali didapat secara otomatis setelah tunai dikalikan dengan pembayaran sisa.

Untuk desain uji coba ini dapat dilihat pada tabel 3.35.

Tabel 3.35. Desain Uji Coba *Form* Transaksi Pembayaran

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji fungsi tombol simpan data pada <i>form</i> transaksi pembayaran	Aplikasi sudah terinstal	Menekan tombol simpan data	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>	Data yang telah diinputkan akan disimpan ke dalam <i>database</i> dan akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cari data pada <i>form</i> transaksi pembayaran	Aplikasi sudah terinstal	Menekan tombol cari data	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat ditampilkan	Data yang telah tersimpan di <i>database</i> dan yang terdapat pada <i>datagridview</i> dapat ditampilkan

3.3.5. Desain uji coba *Form* Laporan Stok Barang

Program akan menampilkan data barang di *datagridview* untuk melihat jumlah stok atau ketersediaan barang yang telah disimpan. Kemudian mencetak laporan dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.36.

Tabel 3.36. *Requirement Form* Laporan Stok Barang

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang stok barang.	<ul style="list-style-type: none"> a. <i>Datagridview</i> b. Tombol cetak laporan 	<ul style="list-style-type: none"> a. Ketika program dijalankan akan tampil data barang di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka laporan dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.37.

Tabel 3.37. Desain Uji Coba *Form* Laporan Stok

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data barang telah tersimpan di database akan muncul di <i>datagridview</i>	Data barang telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> stok barang	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak



3.3.6. Desain uji coba *Form Rekapitulasi Penjualan*

Program akan menampilkan data barang di *datagridview* untuk melihat data transaksi penjualan yang sudah dilakukan di perusahaan berdasarkan tanggal yang dipilih. Kemudian mencetak laporan dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.38.

Tabel 3.38. *Requirement Form Rekapitulasi Penjualan*

Tujuan	<i>Requirement yang diuji</i>	Hasil yang diharapkan
Menghasilkan informasi tentang transaksi penjualan	a. <i>Datagridview</i> b. Tombol cetak laporan	a. Ketika program dijalankan akan tampil data transaksi penjualan di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka laporan dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.39.

Tabel 3.39. Desain Uji Coba *Form* Rekapitulasi Penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data transaksi penjualan telah tersimpan di database akan muncul di <i>datagridview</i>	Data transaksi penjualan telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> rekapitulasi penjualan	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak

3.3.7. Desain uji coba *Form Rekapitulasi Pembayaran*

Program akan menampilkan data barang di *datagridview* untuk melihat data transaksi pembayaran yang sudah dilakukan di perusahaan berdasarkan tanggal yang dipilih. Kemudian mencetak laporan dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.40.

Tabel 3.40. *Requirement Form Rekapitulasi Pembayaran*

Tujuan	Requirement yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang transaksi pembayaran	a. <i>Datagridview</i> b. Tombol cetak laporan	a. Ketika program dijalankan akan tampil data transaksi pembayaran di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka laporan dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.41.

Tabel 3.41. Desain Uji Coba *Form* Rekapitulasi Pembayaran

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data transaksi pembayaran telah tersimpan di database akan muncul di <i>datagridview</i>	Data transaksi pembayaran telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> rekapitulasi pembayaran	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak

3.3.8. Desain uji coba *Form* Jurnal Penjualan

Program akan menampilkan data barang di *datagridview* untuk melihat data jurnal penjualan berdasarkan tanggal yang dipilih. Kemudian mencetak jurnal dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.42.

Tabel 3.42. *Requirement Form* Jurnal Penjualan

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang jurnal penjualan	<ul style="list-style-type: none"> a. <i>Datagridview</i> b. Tombol cetak jurnal 	<ul style="list-style-type: none"> a. Ketika program dijalankan akan tampil data jurnal penjualan di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka jurnal dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.43.

Tabel 3.43. Desain Uji Coba *Form* Jurnal Penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data jurnal penjualan telah tersimpan di database akan muncul di <i>datagridview</i>	Data jurnal penjualan telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> Jurnal Penjualan	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak

3.3.9. Desain uji coba *Form* Jurnal Pemesanan

Program akan menampilkan data barang di *datagridview* untuk melihat data jurnal pemesanan berdasarkan tanggal yang dipilih. Kemudian mencetak jurnal dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.44.

Tabel 3.44. *Requirement Form* Jurnal Pemesanan

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang jurnal Pemesanan	<ul style="list-style-type: none"> a. <i>Datagridview</i> b. Tombol cetak jurnal 	<ul style="list-style-type: none"> a. Ketika program dijalankan akan tampil data jurnal penjualan di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka jurnal dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.45.

Tabel 3.45. Desain Uji Coba *Form* Jurnal Pemesanan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data jurnal pemesanan telah tersimpan di database akan muncul di <i>datagridview</i>	Data jurnal pemesanan telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> Jurnal Pemesanan	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak

3.3.10. Desain uji coba *Form* Jurnal Penyesuaian

Program akan menampilkan data barang di *datagridview* untuk melihat data jurnal Penyesuaian berdasarkan tanggal yang dipilih. Kemudian mencetak jurnal dengan menekan tombol cetak. Sebelum dilakukan pengujian, maka dibutuhkan suatu *requirement* seperti dijelaskan pada tabel 3.46.

Tabel 3.46. *Requirement Form* Jurnal Penyesuaian

Tujuan	<i>Requirement</i> yang diuji	Hasil yang diharapkan
Menghasilkan informasi tentang jurnal penyesuaian	<ul style="list-style-type: none"> a. <i>Datagridview</i> b. Tombol cetak jurnal 	<ul style="list-style-type: none"> a. Ketika program dijalankan akan tampil data jurnal penjualan di <i>datagridview</i> b. Ketika tombol cetak ditekan, maka jurnal dapat dicetak melalui <i>printer</i>

Untuk desain uji coba ini dapat dilihat pada tabel 3.47.

Tabel 3.47. Desain Uji Coba *Form* Jurnal Penyesuaian

<i>Test Case ID</i>	<i>Test Case</i>	<i>Pre Condition</i>	<i>Test Step</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>
1	Menguji tampilan <i>datagridview</i>	Aplikasi sudah terinstal	Menjalankan program	Data jurnal penyesuaian telah tersimpan di database akan muncul di <i>datagridview</i>	Data jurnal penyesuaian telah tersimpan di database akan muncul di <i>datagridview</i>
2	Menguji fungsi tombol cetak data pada <i>form</i> Jurnal Penyesuaian	Aplikasi sudah terinstal	Menekan tombol cetak data	Data yang terdapat pada <i>datagridview</i> dapat dicetak	Data yang terdapat pada <i>datagridview</i> dapat dicetak



INSTITUT BISNIS
& INFORMATIKA

stikom

SURABAYA