BAB IV

PERANCANGAN *E-BUSINESS* PENJUALAN PULSA ELEKTRIK

4.1 Sistem Informasi Yang Diusulkan

4.1.1 Kebijakan Sistem Informasi Yang Diusulkan

Kebijakan dan pengendalian intern yang penulis usulkan pada Procell adalah sebagai berikut :

1. Daftar menjadi *reseller* pulsa elektrik melalui distributor sebesar Rp. 50.000
2. Daftar menjadi *client* menggunakan aplikasi *E-Business* Penjualan Pulsa Elektrik.
3. Setelah terdaftar menjadi *client* lalu mengisi *form* harga jual provider
4. Lakukan pengisian no pin dan no server sebelum melakukan transaksi jual pulsa
5. Aktivitas penjualan pulsa menggunakan aplikasi *E-Business* Penjualan Pulsa Elektrik
6. Aktivitas konfirmasi deposit pulsa menggunakan aplikasi *E-Business* Penjualan Pulsa Elektrik
7. Pendapatan diterima hasil dari penjualan pulsa elektrik
8. Transaksi pembelian deposit pulsa dapat di simpan ke dalam database dengan memasukkan *report* pembelian deposit dari distributor melalui telepon seluler
9. Transaksi penjualan pulsa elektrik dapat di simpan ke dalam database dengan memasukkan *report* penjualan pulsa dari Distributor melalui telepon seluler
10. Cek laporan keuangan jurnal, buku besar, neraca, dan laba rugi menggunakan aplikasi *E-Business* Penjualan Pulsa Elektrik

11. *Assistant Owner* mengelola perusahaan saat *Owner* tidak berada di perusahaan

4.1.2 Fungsi Yang Terkait Yang Diusulkan

Fungsi yang terkait di dalam *e-business* penjualan pulsa elektrik pada Procell adalah sebagai berikut:

1. Distributor
2. *Owner/Assistant Owner*
3. Costumer
4. *Partner*
5. *Supplier*
6. Rekanan

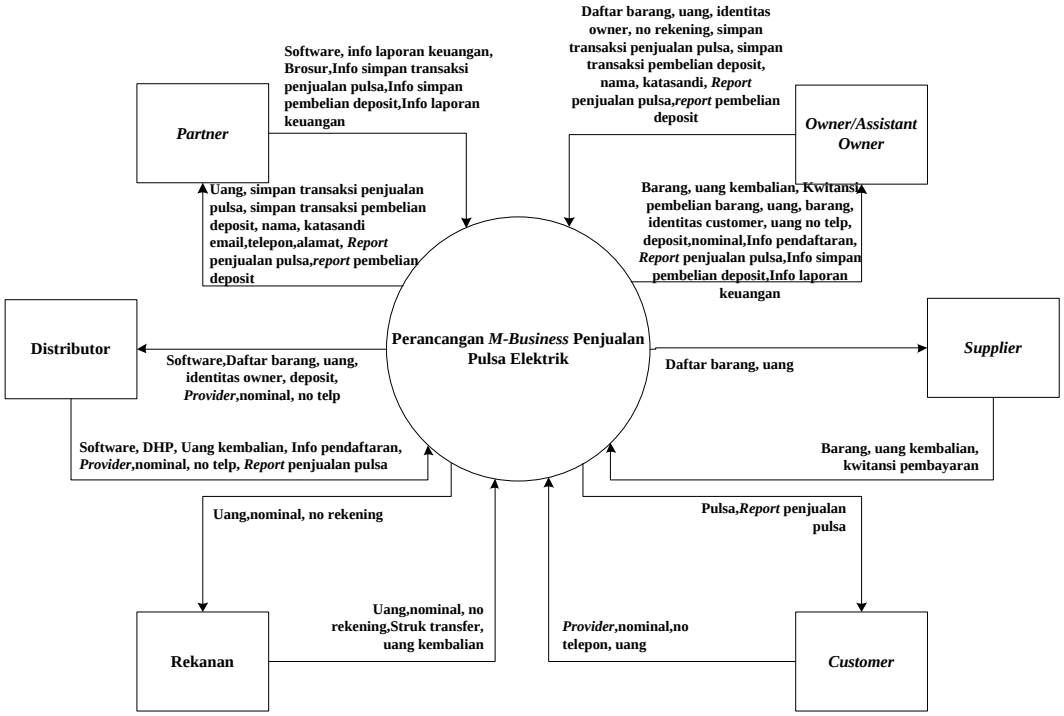
4.2 Perancangan Model Sistem Yang Diusulkan

Pada bagian ini penulis akan memaparkan usulan yang penulis usulkan dalam Perancangan *E-Business* Penjualan Pulsa Elektrik. Sistem yang penulis usulkan sistem yang telah terkomputerisasi dan berbasis telepon seluler Android. Sistem ini diharapkan dapat menjadi solusi dari permasalahan-permasalahan yang telah penulis paparkan pada Bab I.

4.2.1 Diagram Arus Data Yang Diusulkan

4.2.1.1 Diagram Konteks

Diagram konteks yang diusulkan pada *E-Business* Penjualan Pulsa Elektrik pada Procell adalah sebagai berikut :



Gambar 4.1 Diagram Konteks yang Diusulkan

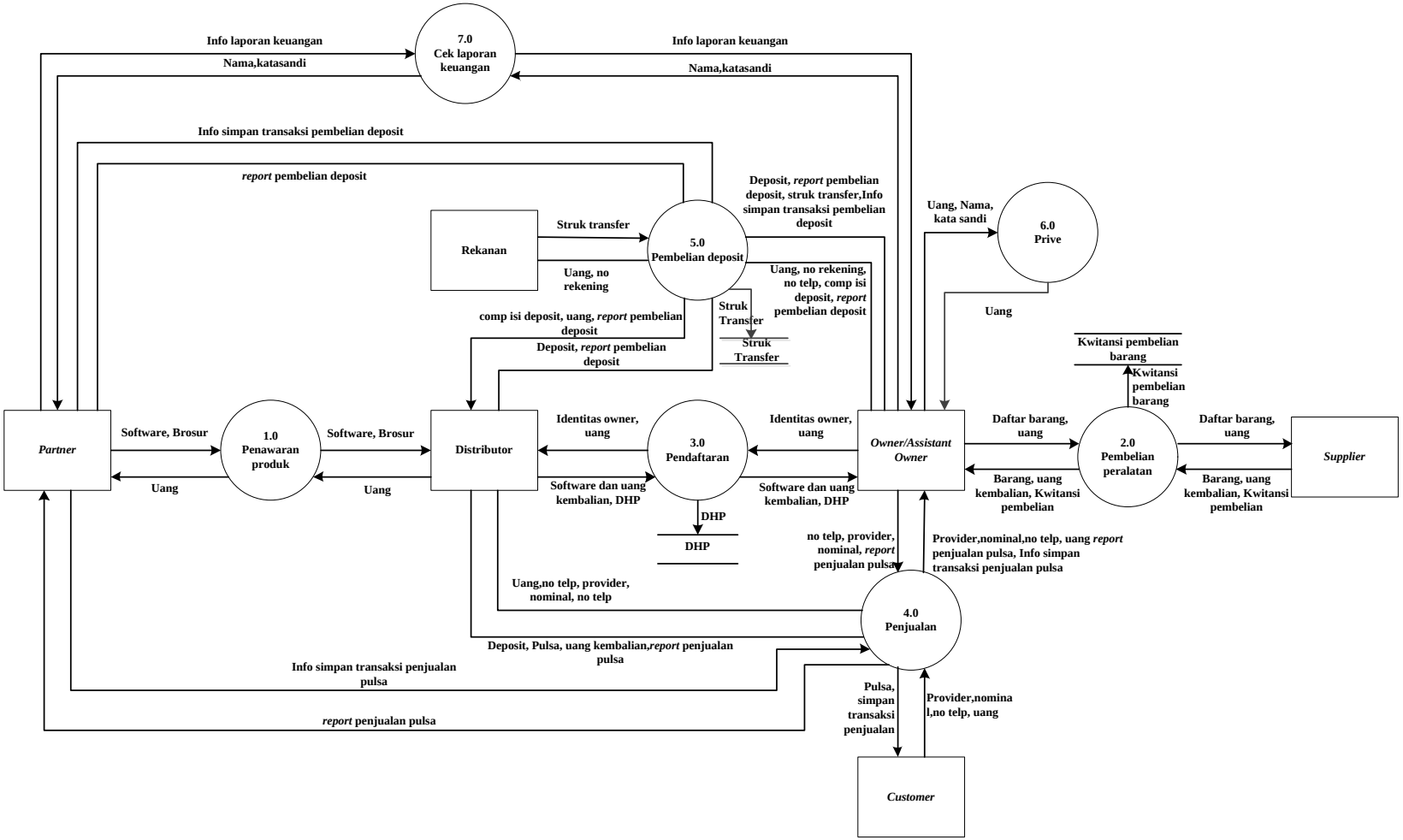
Keterangan :

DHP : Daftar Harga Provider

Pada diagram konteks *E-Business* Penjualan Pulsa Elektrik entitas-entitas yang terlibat adalah *Partner*, Distributor, *Owner/Assisten Owner*, *Supplier*, Customer, Rekanan. Awal mula transaksi dilakukan oleh Partner yang mempromosikan produknya ke Distributor.

4.2.1.2 Diagram Arus Data Level Nol Yang Diusulkan

Diagram arus data level 0 pada *E-Business* Penjualan Pulsa Elektrik terdiri dari 7 proses. Berikut ini gambaran dari diagram arus data level 0 tampak pada gambar di bawah ini:



4.2 Diagram Arus Data Level 0 Yang Diusulkan

Diagram arus data level 0 ini terdiri dari 7 proses, yang terdiri dari proses penawaran produk, pembelian peralatan, pendaftaran, penjualan, pembelian deposit, prive, cek laporan keuangan. Berikut adalah rincian dari proses-proses yang terdapat pada diagram arus data level 0 :

1. Proses Penawaran Produk

   Proses dimana Partner menawarkan sebuah produk kepada Distributor.

2. Proses Pembelian Peralatan

   Proses dimana *Owner/Assistan Owner* mengawali bisnis dengan membeli peralatan ke Supllier untuk kelancaran proses bisnisnya.

3. Proses Pendaftaran

   Proses pendaftaran yang dilakukan *Owner/Assistan Owner* ke Distributor untuk menjadi *reseller* pulsa elektrik.

4. Proses Penjualan

   Proses dimana *Owner/Assistan Owner* melakukan bisnis penjualan pulsa elektrik kepada Customer.

5. Proses Pembelian Deposit

   Proses dimana *Owner/Assistan Owner* membeli deposit kepada Distributor. Pembelian bisa dilakukan secara langsung atau melalui Rekanan.

6. Proses Prive

   Proses dimana *Owner/Assistan Owner* melakukan pengambilan uang untuk kebutuhan pribadi.

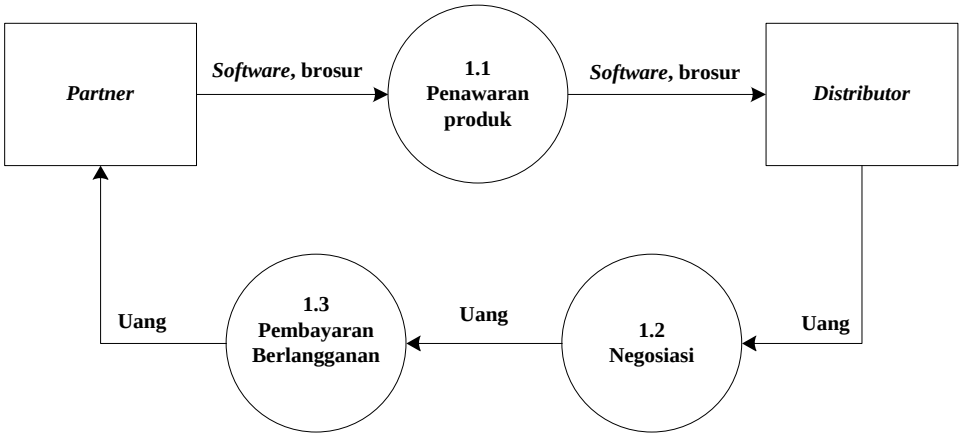
7. Proses Cek Laporan Keuangan

   Proses dimana *Owner/Assistan Owner* melakukan pengecekan laporan keuangan.

4.2.1.3 Diagram Arus Data Level 1

4.2.1.3.1 Diagram Arus Data Level 1 Proses 1

Diagram arus data level 1 proses 1 pada *E-Business* Penjualan Pulsa Elektrik terdapat 7 proses. Gambaran diagram arus data level 1 proses 1 tampak pada gambar di bawah ini :



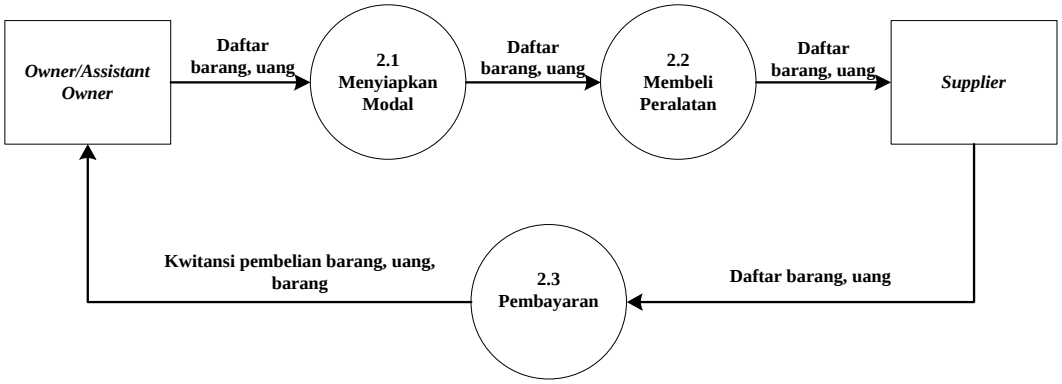
Gambar 4.3 Diagram Arus Data Yang Diusulkan Level 1 Proses 1

Diagram arus data level 1 proses 1 ini terdiri dari 3 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Penawaran Produk yang diberikan oleh Partner untuk mengajak berkerjasama dengan membawa berkas brosur dan *software.*
2. Proses Negosiasi yang dilakukan oleh Partner dan Distributor bagaimana hasil keputusan yang didapat, apabila ingin bekerjasama.
3. Proses Pembayaran Berlangganan ini dilakukan apabila Distributor ingin menggunakan produk yang diberikan oleh Partner.

4.2.1.3.2 Diagram Arus Data Level 1 Proses 2

Diagram arus data level 1 proses 2 pada *E-Business* Penjualan Pulsa Elektrik terdapat 3 proses. Gambaran diagram arus data level 1 proses 2 tampak pada gambar di bawah ini :



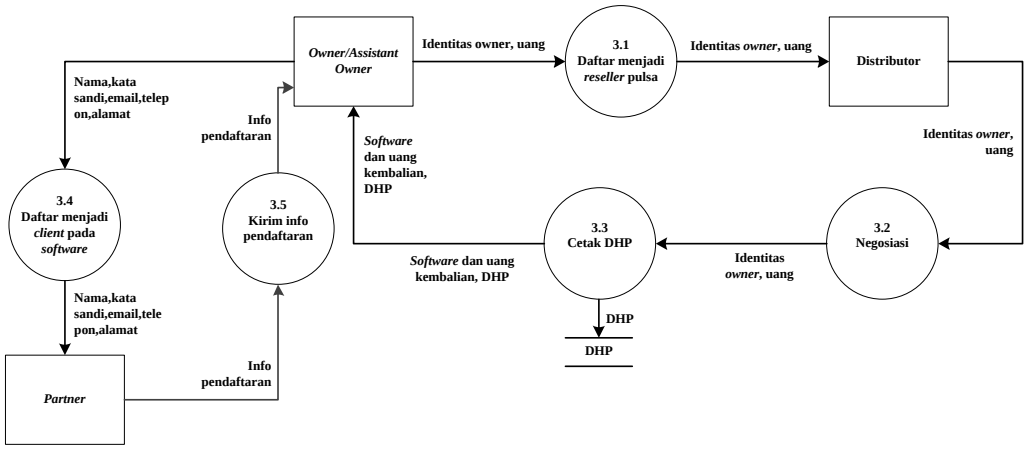
Gambar 4.4 Diagram Arus Data Level 1 Proses 2 Yang Diusulkan

Diagram arus data level 1 proses 2 ini terdiri dari 3 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Menyiapkan Modal adalah proses dimana *Owner/Assistant Owner* menyiapkan modal untuk melakukan bisnis penjualan pulsa elektrik.
2. Proses Membeli Peralatan adalah proses membeli peralatan yang diperlukan untuk memulai bisnis.
3. Proses Pembayaran adalah proses *Owner/Assisntant Owner* membayar pembelian yang dilakuan kepada *Supplier*

4.2.1.3.3 Diagram Arus Data Level 1 Proses 3

Diagram arus data level 1 proses 3 pada *E-Business* Penjualan Pulsa Elektrik terdapat 5 proses. Gambaran diagram arus data level 1 proses 3 tampak pada gambar di bawah ini :



Gambar 4.5 Diagram Arus Data Level 1 Proses 3 Yang Diusulkan

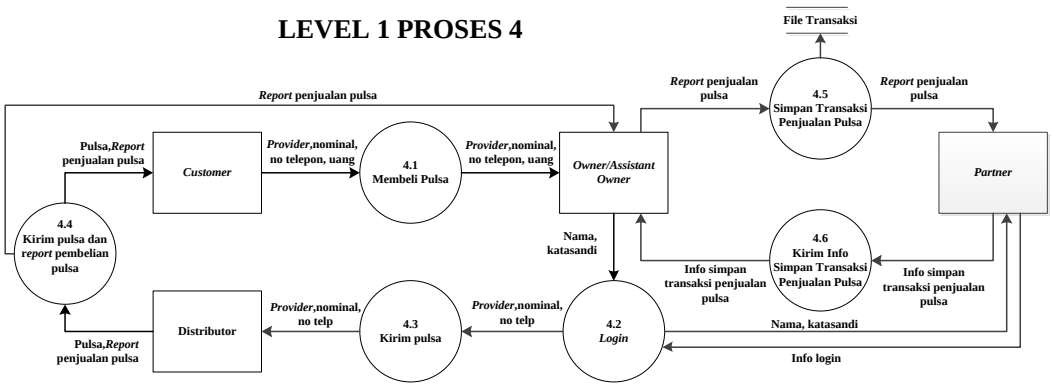
Diagram arus data level 1 proses 3 ini terdiri dari 5 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Daftar menjadi *reseller* pulsa adalah proses pendaftaran menjadi *reseller* pula elektrik yang dilakukan oleh *Owner/Assisntant Owner* ke Distributor.
2. Proses Negosiasi adalah proses dimana distributor dan *Owner/Assisntant Owner* melakukan perbincangan untuk bisa mendaftar menjadi *reseller.*
3. Proses Cetak DHP adalah proses pencetakan daftar harga pulsa setelah melakukan pendaftaran menjadi *reseller.*

4. Proses Daftar menjadi *client* pada *software* adalah proses dimana setelah menjadi *reseller* kemudian mendaftarkan pada aplikasi yang telah diberikan oleh distributor.
5. Proses Kirim info pendaftaran adalah proses pemberian info jika pendaftar sudah mendaftar.

4.2.1.3.4 Diagram Arus Data Level 1 Proses 4

Diagram arus data level 1 proses 4 pada *E-Business* Penjualan Pulsa Elektrik terdapat 4 proses. Gambaran diagram arus data level 1 proses 4 tampak pada gambar di bawah ini :



Gambar 4.6 Diagram Arus Data Level 1 Proses 4 Yang Diusulkan

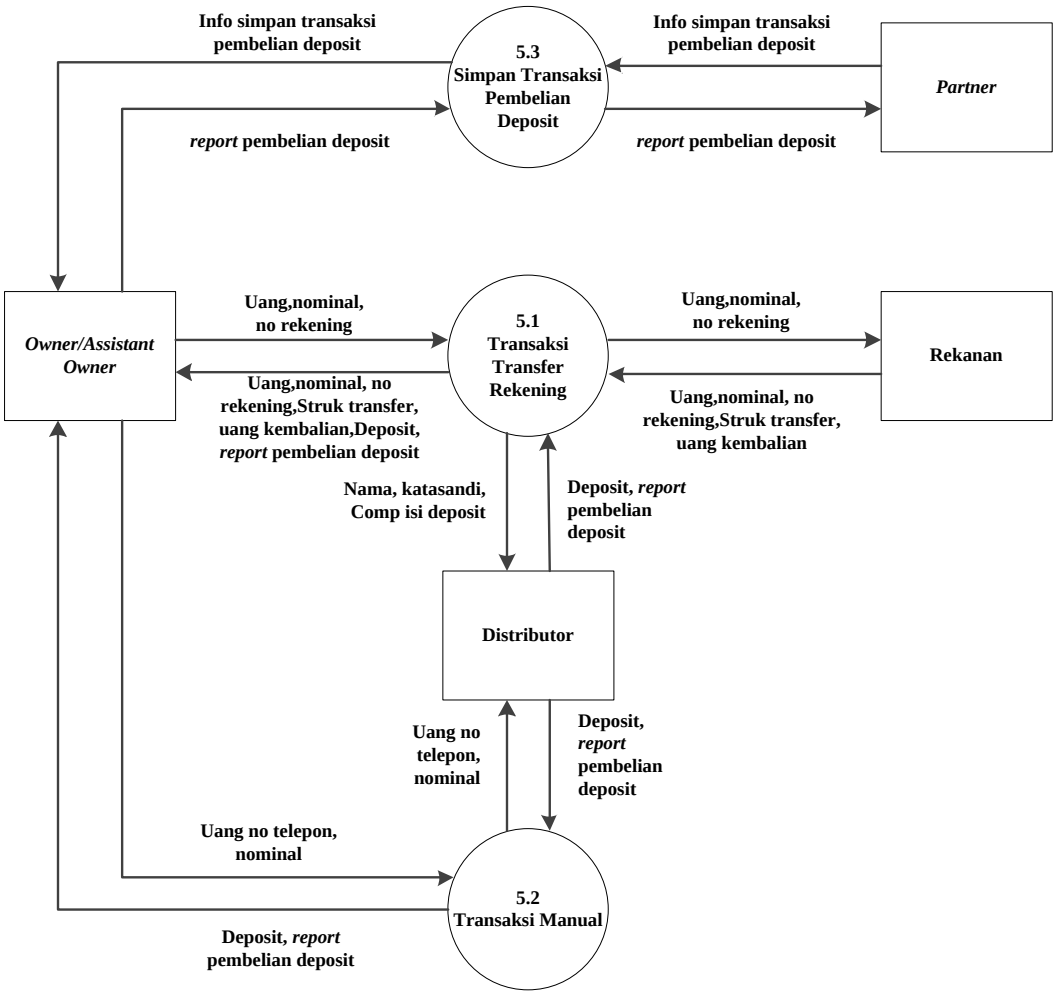
Diagram arus data level 1 proses 4 ini terdiri dari 4 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Membeli Pulsa dimana *Customer* membeli pusa dengan memberikan nama *provider*, nominal, no telepon, uang kepada *Owner/Assistant Owner*
2. Proses *Login* adalah proses membuka aplikasi dengan *login* terlebih dahulu.

3. Proses Kirim Pulsa adalah proses pengiriman pulsa kepada *Customer* yang ditujukan ke *server* terlebih dahulu
4. Proses Kirim pulsa dan *report* pembelian pulsa adalah proses pengiriman pulsa melalui sistem yang digunakan Distributor dan memberi *report* kepada *Customer* dan *Owner/Assistant Owner*
5. Proses Simpan Transaksi Penjualan Pulsa adalah proses *Owner/Assistant Owner* menyimpan data transaksi penjualaan pulsa elektrik ke *Partner*
6. Proses Kirim Info Simpan Transaksi Penjualan Pulsa adalah proses pengiriman info simpan transaksi penjualan pulsa ke *Owner/Assitant Owner*

4.2.1.3.5 Diagram Arus Data Level 1 Proses 5

Diagram arus data level 1 proses 5 pada *E-Business* Penjualan Pulsa Elektrik terdapat 3 proses. Gambaran diagram arus data level 1 proses 5 tampak pada gambar di bawah ini :



Gambar 4.7 Diagram Arus Data Level 1 Proses 5 Yang Diusulkan

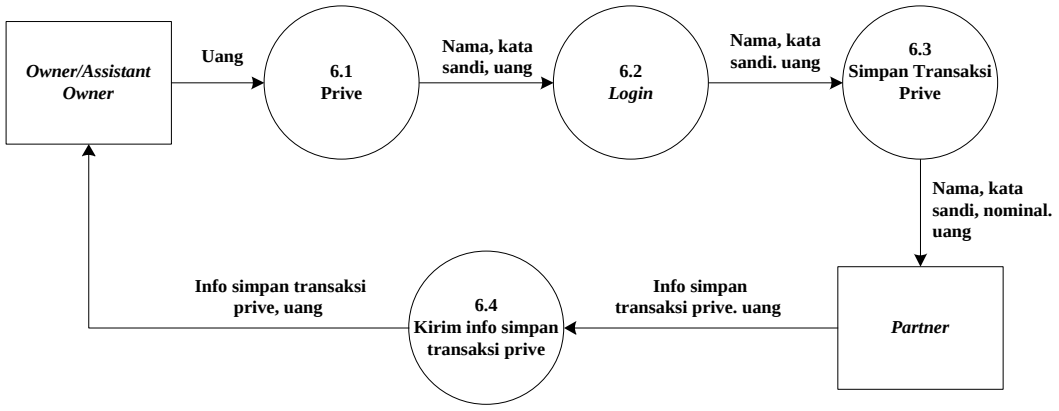
Diagram arus data level 1 proses 5 ini terdiri dari 3 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Transaksi Transfer Rekening adalah proses pembelian deposit yang dilakukan *Owner/Assistant Owner* melalui rekening *bank* yang ditujukan kepada Distributor.
2. Proses Transaksi Manual adalah proses pembelian deposit yang dilakukan *Owner/Assistant Owner* secara langsung kepada Distributor

3. Proses Simpan Transaksi Pembelian Deposit adalah proses penyimpan bukti transaksi pembelian deposit.

4.2.1.3.6 Diagram Arus Data Level 1 Proses 6

Diagram arus data level 1 proses 6 pada *E-Business* Penjualan Pulsa Elektrik terdapat 4 proses. Gambaran diagram arus data level 1 proses 6 tampak pada gambar di bawah ini :



Gambar 4.8 Diagram Arus Data Level 1 Proses 6 Yang Diusulkan

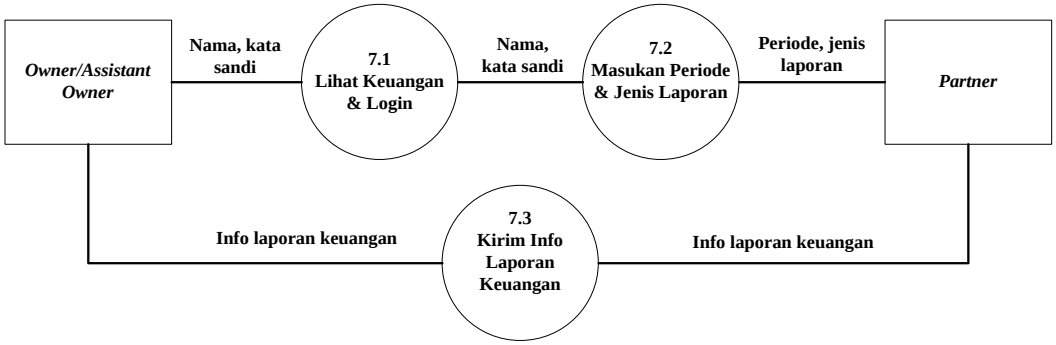
Diagram arus data level 1 proses 6 ini terdiri dari 4 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Prive adalah proses pengambilan uang untuk kebutuhan pribadi
2. Proses *Login* adalah proses untuk masuk ke dalam aplikasi yang membutuhkan nama dan kata sandi.
3. Proses Simpan Transaksi Prive adalah proses untuk menyimpan data transaksi prive ke *Partner*.

4. Proses Kirim Pesan Simpan Transaksi adalah proses info simpan transaksi kepada *Owner/Assistant Owner*

4.2.1.3.7 Diagram Arus Data Level 1 Proses 7

Diagram arus data level 1 proses 7 pada *E-Business* Penjualan Pulsa Elektrik terdapat 3 proses. Gambaran diagram arus data level 1 proses 7 tampak pada gambar di bawah ini :



Gambar 4.9 Diagram Arus Data Level 1 Proses 7 Yang Diusulkan

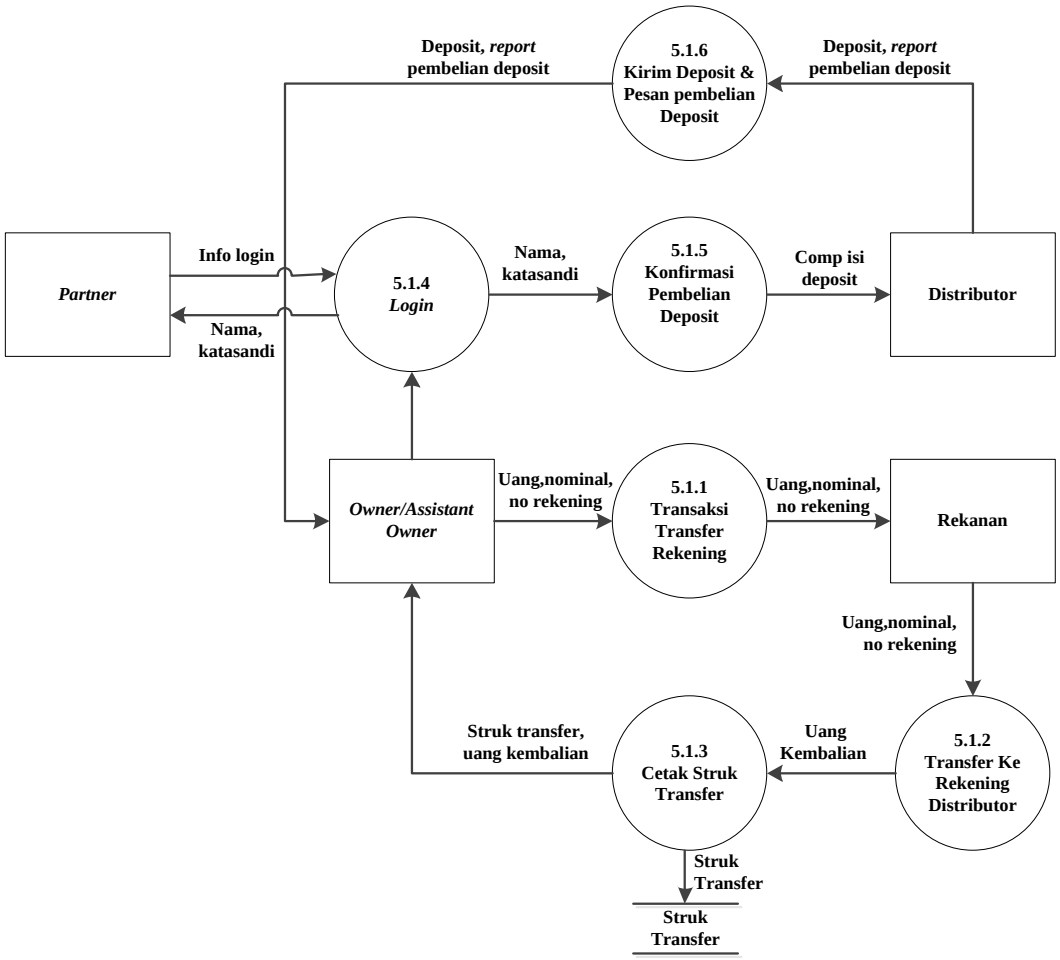
Diagram arus data level 1 proses 7 ini terdiri dari 3 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses *Login* & Lihat Keuangan adalah proses untuk masuk ke dalam aplikasi dan melihat laporan keuangan.
2. Proses Masukan Periode & Jenis Laporan adalah proses untuk menampilkan data dari periode yang dipilih dan jenis laporan yaitu jurnal, buku besar, neraca, laba rugi.
3. Proses Kirim Info Laporan Keuangan adalah proses pengiriman info laporan keuangan dan periode yang dipilih.

4.2.1.4 Diagram Alur Data Level 2

4.2.1.4.1 Diagram Arus Data Level 2 Proses 5.1

Diagram arus data level 2 proses 5.1 pada *E-Business* Penjualan Pulsa Elektrik terdapat 6 proses. Gambaran diagram arus data level 2 proses 5.1 tampak pada gambar di bawah ini :



Gambar 4.10 Diagram Arus Data Level 2 Proses 5.1 Yang Diusulkan

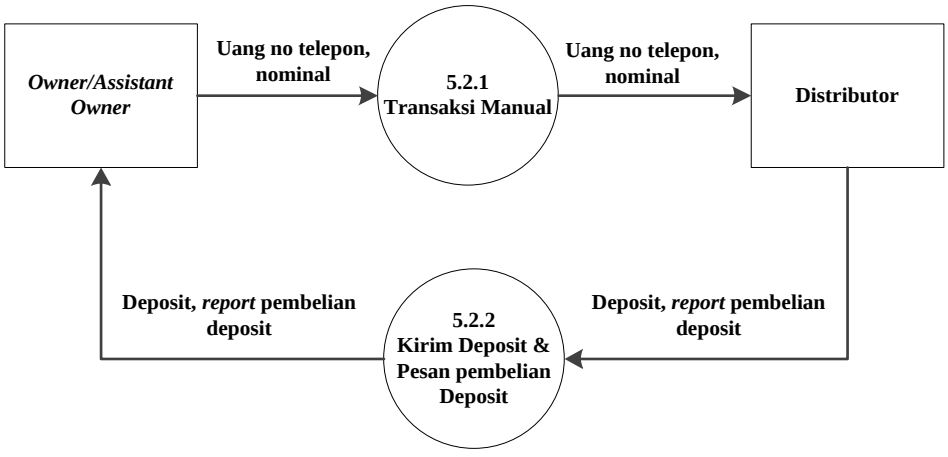
Diagram arus data level 2 proses 5.1 ini terdiri dari 6 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Transaksi Transfer Rekening adalah proses pembelian deposit dengan cara transfer rekening ke rekening Distributor

2. Proses Transfer Ke Rekening Distributor adalah proses pentransferan dana dari rekening Owner/Assitant Owner ke rekening Distributor.
3. Proses Cetak Struk Transfer adalah proses cetak sebagai bukti transaksi transfer.
4. Proses *Login* adalah proses masuk ke aplikasi dengan memasukkan nama dan katasandi.
5. Proses Konfirmasi Pembelian Deposit adalah proses dimana *Owner/Assitant Owner* mengkonfirmasikan atas pembelian deposit melalui sms.
6. Proses Kirim Deposit & Pesan Pembelian Deposit adalah proses pengiriman deposit & pesan pembelian depsosit oleh Distributor.

4.2.1.4.2 Diagram Arus Data Level 2 Proses 5.2

Diagram arus data level 2 proses 5.2 pada *E-Business* Penjualan Pulsa Elektrik terdapat 3 proses. Gambaran diagram arus data level 2 proses 5.2 tampak pada gambar di bawah ini :



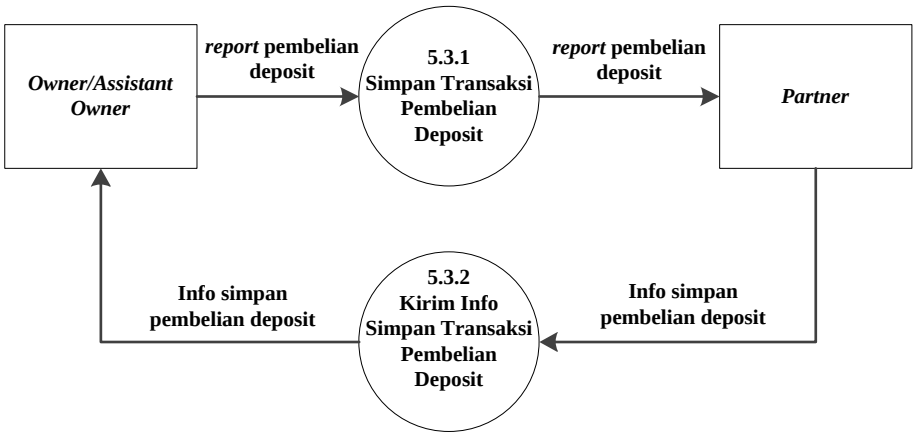
Gambar 4.11 Diagram Arus Data Level 2 Proses 5.2 Yang Diusulkan

Diagram arus data level 2 proses 5.2 ini terdiri dari 3 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Transaksi Manual adalah proses pembelian deposit yang dilakukan secara langsung ke Distributor.
2. Proses Isi Deposit adalah proses pengisian deposit yang dilakukan oleh Distributor kepada *Owner/Assistant Owner*s.
3. Proses Kirim Info Simpan Transaksi Pembelian Deposit adalah pengiriman info kepada *Owner/Assistant Owner* atas pembelian deposit

4.2.1.4.3 Diagram Arus Data Level 2 Proses 5.3

Diagram arus data level 2 proses 5.3 pada *E-Business* Penjualan Pulsa Elektrik terdapat 2 proses. Gambaran diagram arus data level 2 proses 5.3 tampak pada gambar di bawah ini :



Gambar 4.12 Diagram Arus Data Level 2 Proses 5.3 Yang Diusulkan

Diagram arus data level 2 proses 5.3 ini terdiri dari 2 proses. Berikut ini adalah rincian dari proses-proses tersebut :

1. Proses Simpan Transaksi Pembelian Deposit adalah proses penyimpanan data transaksi pembelian deposit
2. Proses Kirim Info Simpan Pembelian Deposit adalah proses pengiriman info simpan transaksi pembelian deposit.

4.2.2 Kamus Data Yang Diusulkan

Kamus data yang diusulkan pada *E-Business* Penjualan Pulsa Elektrik adalah sebagai berikut:

Tabel 4.1 Kamus Data Untuk DHP

Nama Arus Data	DHP
Alias	Daftar Harga Provider
Bentuk Data	Dokumen/Formulir
Arus Data	Proses 3.0
Penjelasan	Daftar harga pulsa dari distributor
Periode	1
Volume	1
Struktur Data	Provider, kode produk, harga

Tabel 4.2 Kamus Data Untuk Kwitansi Pembelian Barang

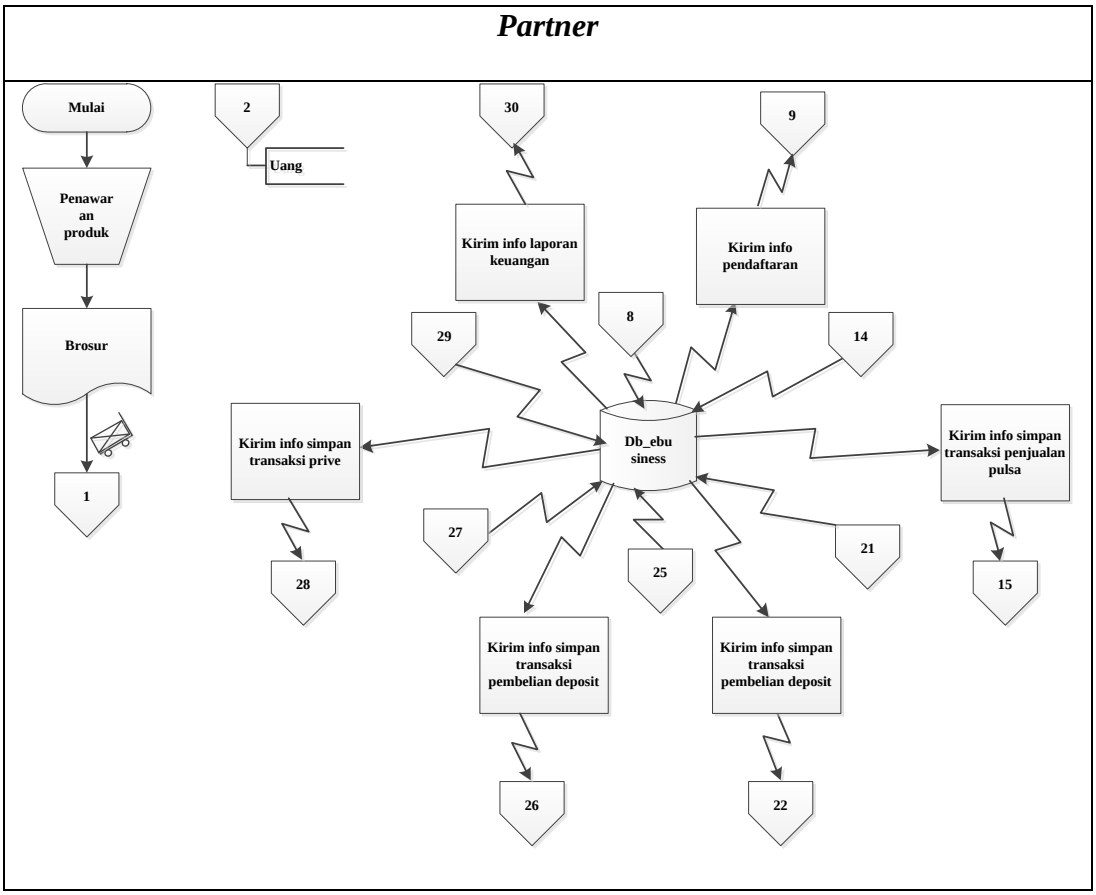
Nama Arus Data	Kwitansi Pembelian Barang
Alias	Kwitansi Pembelian Barang
Bentuk Data	Dokumen/Formulir
Arus Data	Proses 2.0
Penjelasan	Kwitansi bukti pembelian peralatan
Periode	1
Volume	1
Struktur Data	Tanggal, nama barang, harga

Tabel 4.3 Kamus Data Untuk Struk Transfer

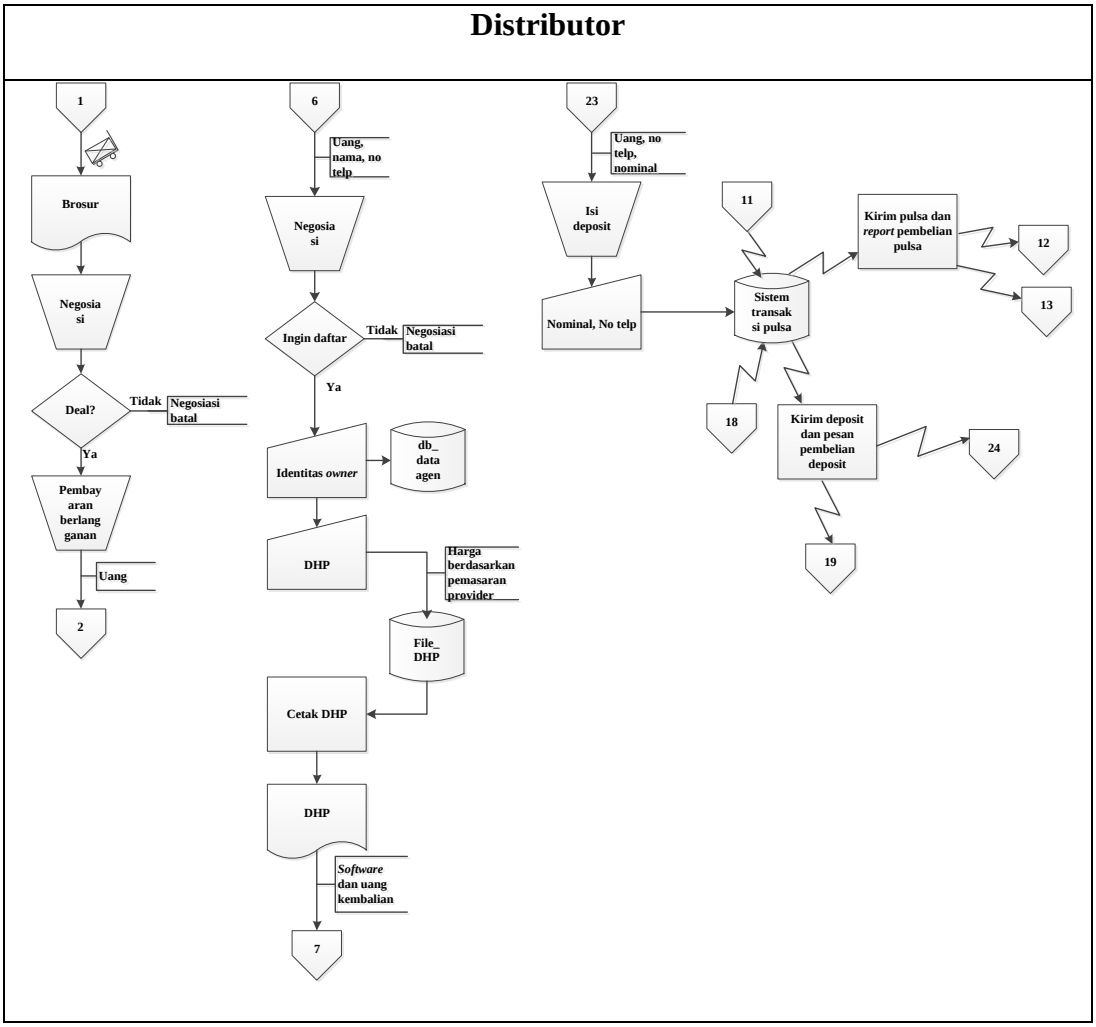
Nama Arus Data	Struk Transfer
Alias	Struk Transfer
Bentuk Data	Dokumen/Formulir
Arus Data	Proses 5.1
Penjelasan	Bukti pembelian deposit
Periode	1
Volume	1
Struktur Data	Tanggal, nominal, no rekening

4.2.3 Bagan Alir (Flowchart) Sistem Yang Diusulkan

Bagan alir sistem yang diusulkan *E-Business* Penjualan Pulsa Elektrik pada Procell adalah sebagai berikut:

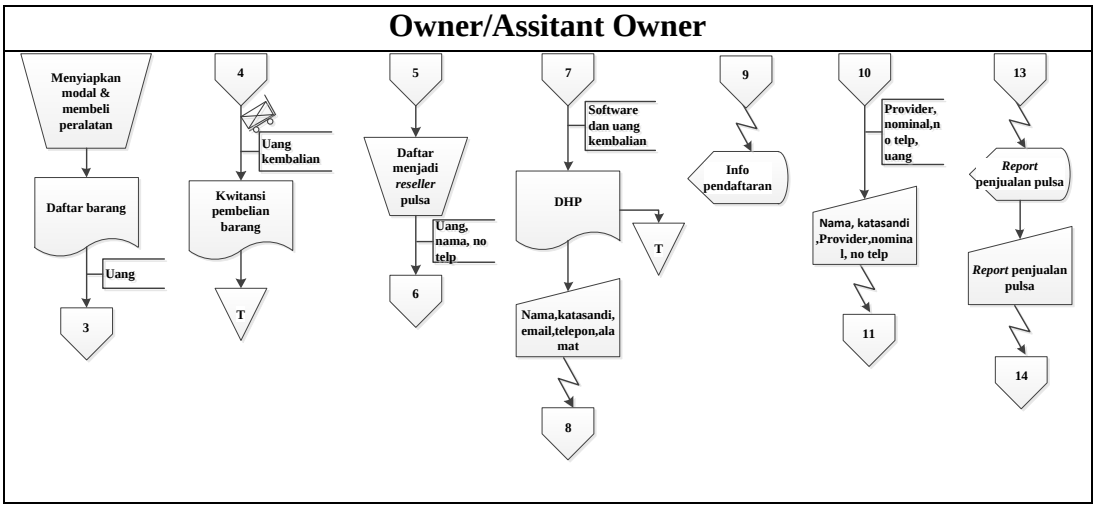


Gambar 4.13 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang Diusulkan

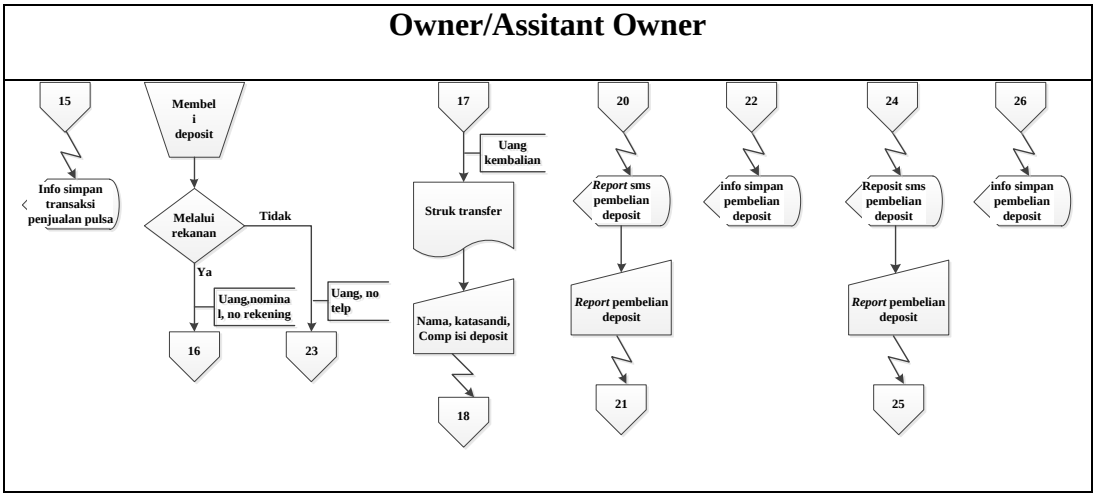


Gambar 4.14 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang

Diusulkan Lanjutan 1

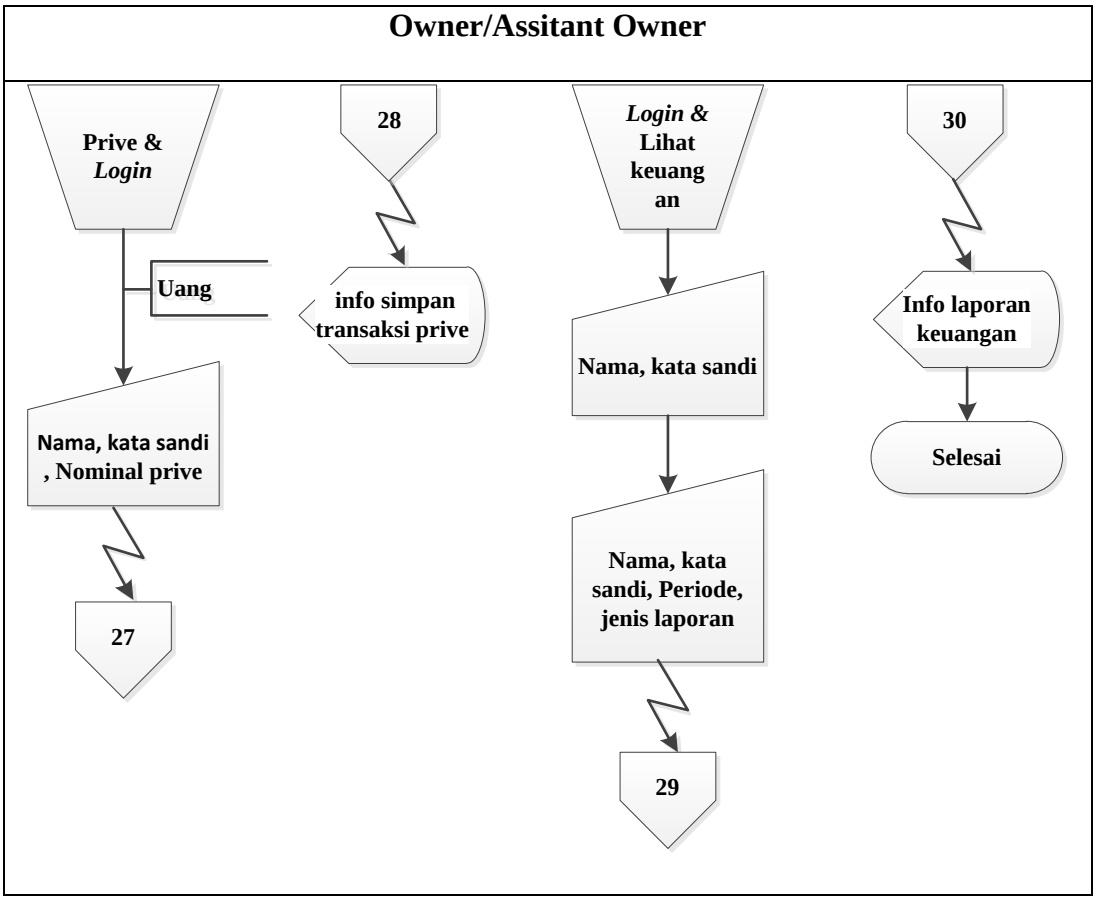


Gambar 4.15 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang Diusulkan Lanjutan 2



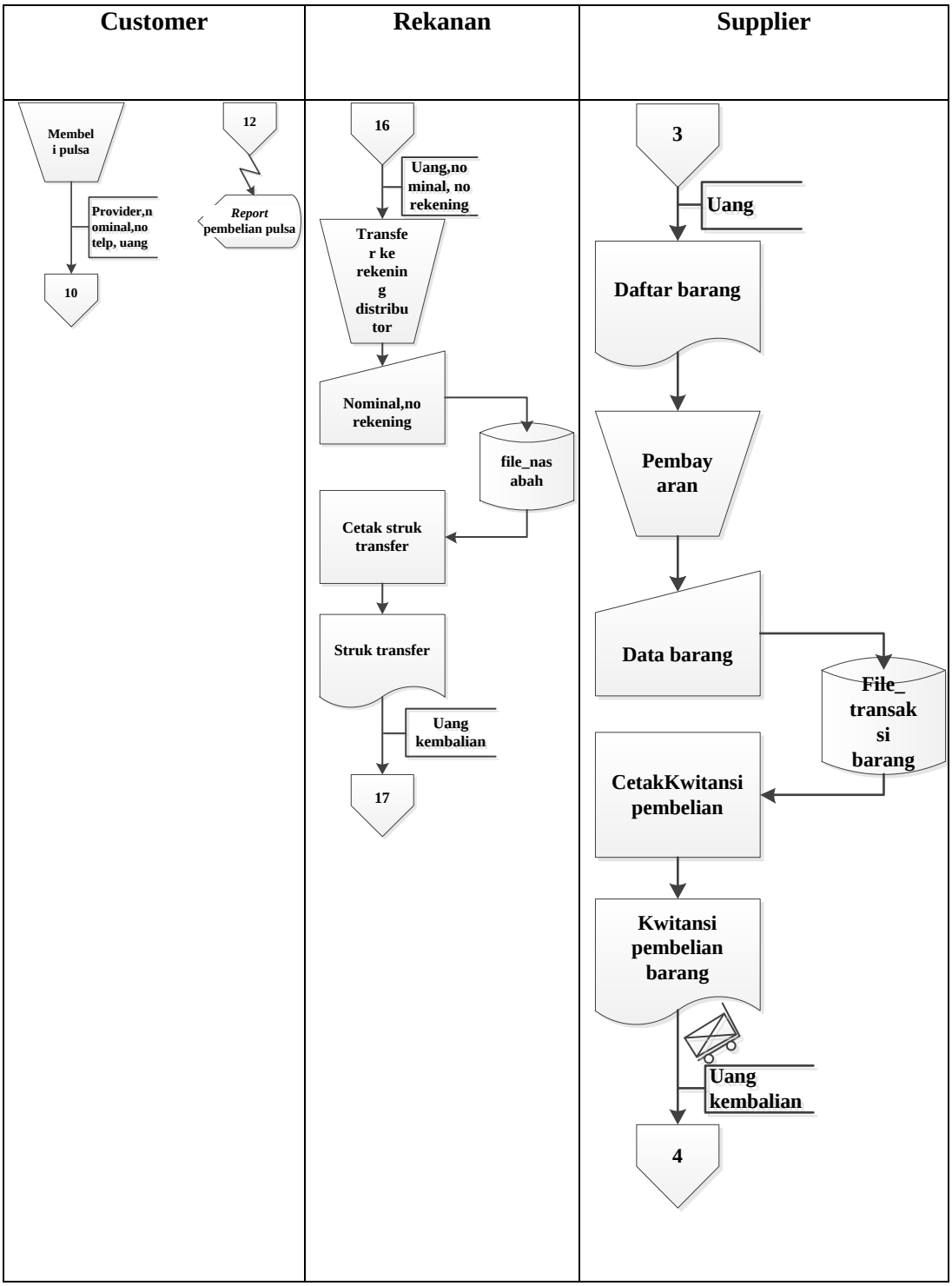
Gambar 4.16 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang

Diusulkan Lanjutan 3



Gambar 4.17 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang

Diusulkan Lanjutan 4



Gambar 4.18 Bagan Alir Sistem *E-Business* Penjualan Pulsa Elektrik Yang

Diusulkan Lanjutan 5

Keterangan**:**

DHP : Daftar Harga Provider

4.2.4 Perancangan Basis Data

4.2.4.1 Normalisasi Yang Diusulkan

Berikut ini adalah normalisasi yang penulis usulkan dalam Peracangan *E-Business* Penjualan Pulsa Elektrik :

Tabel 4.4 Bukti Kwitansi Pembelian Deposit Bentuk Unnormal

No	Tanggal	Banyaknya	Nama Barang	Jumlah	Total
10943	20/3/2012	50000	Deposit	50000	50000

Tabel 4.5 Bukti Kwitansi Pembelian Deposit Bentuk Normal ke Satu (1-NF)

No	Tanggal	Banyaknya	Nama Barang	Jumlah	Total
10943	20/3/2012	50000	Deposit	50000	50000

Tabel 4.6 Bukti Kwitansi Pembelian Deposit Bentuk Normal ke Dua (2-NF)

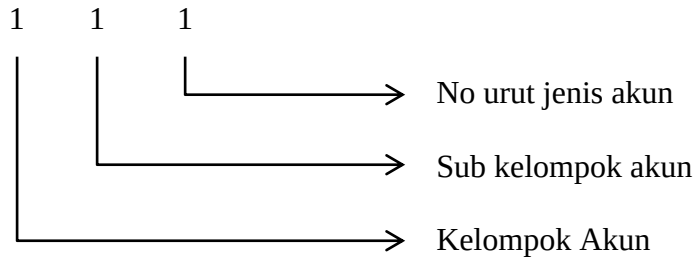
Relasi Tabel Transaksi Pembelian Deposit

No	Jumlah	Nama Barang	Total
10943	50000	Deposit	50000

4.2.4.2 Pengkodean Pada Kunci Utama

Adapun perancangan pengkodean yang digunakan oleh penulis adalah sebagai berikut :

A. Kode Akun



4.2.4.3 Perancangan Struktur Tabel

Perancangan struktur tabel adalah salah satu hal yang paling utama dalam merancang sebuah program. Hal ini dikarenakan tabel-tabel tersebut yang akan menyimpan data-data yang diolah di dalam program. Sehingga dalam pembuatannya diperlukan perancangan struktur tabel yang tepat agar tidak terjadi kesalahan yang berdampak kepada jalannya program. Berikut ini adalah hasil perancangan tabel yang penulis hasilkan untuk Perancangan *E-Business* Penjualan Pulsa Elektrik pada Procell:

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	ID	varchar(20)	latin1_swedish_ci		No			
	PASSWORD	varchar(20)	latin1_swedish_ci		No			
	EMAIL	varchar(20)	latin1_swedish_ci		No			
	ALAMAT	varchar(60)	latin1_swedish_ci		No			
	NOTELP	varchar(20)	latin1_swedish_ci		No			
	KODEPOS	varchar(10)	latin1_swedish_ci		No			
	AWALAN	varchar(20)	latin1_swedish_ci		No			
	TGL_MASUK	datetime			No			

Gambar 4.19 Struktur Tabel User

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	MESSAGE	varchar(200)	latin1_swedish_ci		Yes	NULL		
	KODE_PRODUK	varchar(9)	latin1_swedish_ci		Yes	NULL		
	NOMINAL	int(9)			Yes	NULL		
	NO_TUJUAN	varchar(14)	latin1_swedish_ci		Yes	NULL		
	SN	varchar(20)	latin1_swedish_ci		Yes	NULL		
	TGL	varchar(9)	latin1_swedish_ci		Yes	NULL		
	JAM	varchar(9)	latin1_swedish_ci		Yes	NULL		
	SALDO_AWAL	int(9)			Yes	NULL		
	HARGA_DISTRIBUTOR	int(9)			Yes	NULL		
	SALDO_AKHIR	int(9)			Yes	NULL		
	WAKTU_SIMPAN	datetime			Yes	NULL		

Gambar 4.20 Struktur Tabel Transaksi

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	TANGGAL	varchar(10)	latin1_swedish_ci		Yes	NULL		
	NOMOR_BUKTI	varchar(20)	latin1_swedish_ci		Yes	NULL		
	KETERANGAN	varchar(50)	latin1_swedish_ci		Yes	NULL		
	REF	varchar(9)	latin1_swedish_ci		Yes	NULL		
	DEBIT	int(9)			Yes	NULL		
	KREDIT	int(9)			Yes	NULL		
	TANGGALX	datetime			Yes	NULL		

Gambar 4.21 Struktur Tabel Jurnal

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	TANGGAL	varchar(10)	latin1_swedish_ci		Yes	NULL		
	KETERANGAN	varchar(25)	latin1_swedish_ci		Yes	NULL		
	REF	varchar(9)	latin1_swedish_ci		Yes	NULL		
	DEBIT	int(9)			Yes	NULL		
	KREDIT	int(9)			Yes	NULL		
	SDEBIT	int(9)			Yes	NULL		
	SKREDIT	int(9)			Yes	NULL		
	TANGGALX	datetime			Yes	NULL		

Gambar 4.22 Struktur Tabel Buku Besar

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	OPERATOR	varchar(9)	latin1_swedish_ci		Yes	NULL		
	KODE_PRODUK	varchar(9)	latin1_swedish_ci		Yes	NULL		
	HARGA_JUAL	int(9)			Yes	NULL		
	TGL_UPDATE	datetime			Yes	NULL		

Gambar 4.23 Struktur Tabel Harga Jual

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	TANGGAL	varchar(10)	latin1_swedish_ci		Yes	NULL		
	NOMOR_BUKTI	varchar(50)	latin1_swedish_ci		Yes	NULL		
	HARGA	int(9)			Yes	NULL		
	TANGGALX	datetime			Yes	NULL		

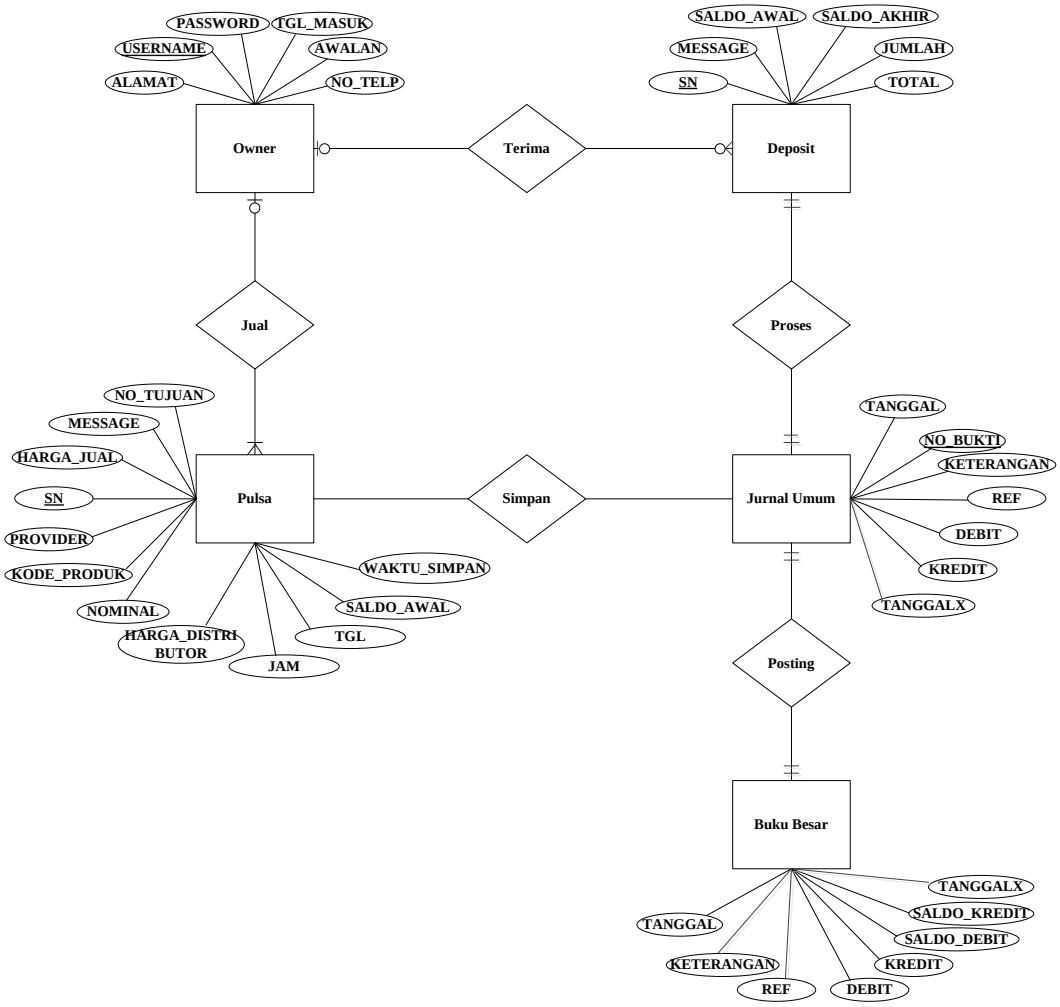
Gambar 4.24 Struktur Tabel Deposit Keluar

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	TANGGAL	varchar(10)	latin1_swedish_ci		Yes	*NULL*		
	NOMOR_BUKTI	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
	OPERATOR	varchar(20)	latin1_swedish_ci		Yes	*NULL*		
	HPP	int(9)			Yes	*NULL*		
	HARGA_JUAL	int(9)			Yes	*NULL*		
	LABA	int(9)			Yes	*NULL*		
	TANGGALX	datetime			Yes	*NULL*		

Gambar 4.25 Struktur Tabel Deposit Masuk

4.2.4.4 Diagram Relasi Entitas Yang Diusulkan

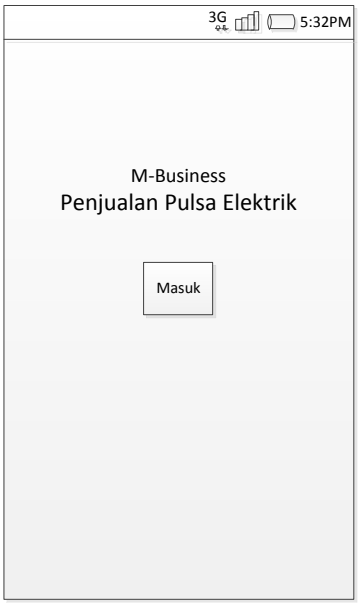
Gambaran relasi antar entitas atau *entiiy relationship* diagram dalam perancangan *E-Business* Penjualan Pulsa Elektrik adalah sebagai berikut:



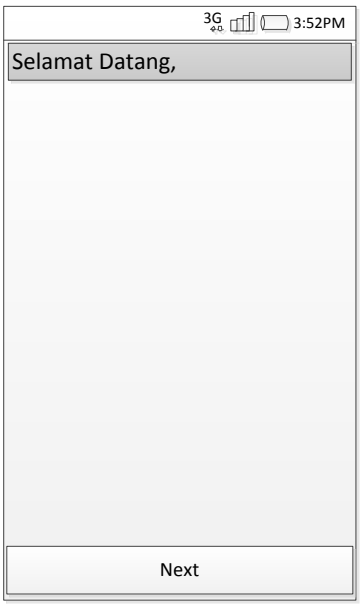
Gambar 4.26 *Entity Relationship Diagram* Usulan

4.3 Perancangan Antar Muka Dan Laporan

4.3.1 Perancangan Antarmuka



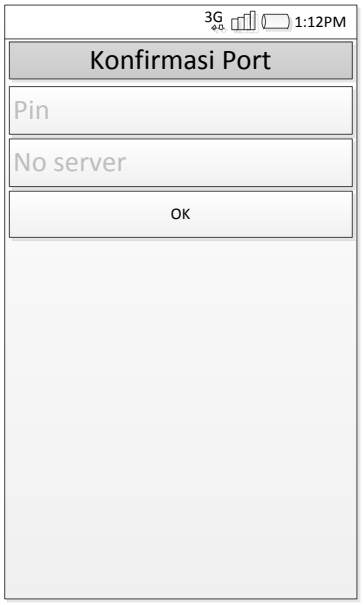
Gambar 4.27 Tampilan Perancangan Antarmuka pada *Activity Splash*



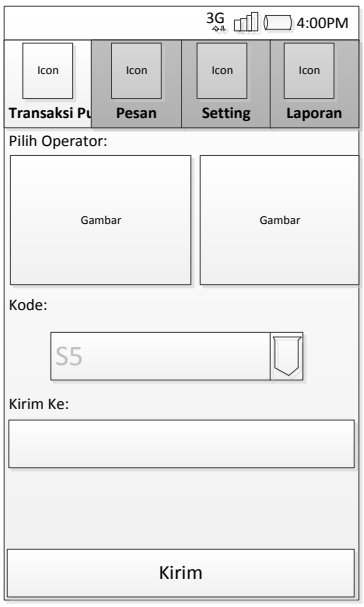
Gambar 4.28 Tampilan Perancangan Antarmuka pada *Activity Panduan*



Gambar 4.29 Tampilan Perancangan Antarmuka pada *Activity Login*



Gambar 4.30 Tampilan Perancangan Antarmuka pada *Activity Konfirmasi Port*



Gambar 4.31 Tampilan Perancangan Antarmuka pada *Activity Menu*



Gambar 4.32 Tampilan Perancangan Antarmuka pada *Activity Daftar*

4.3.2 Perancangan Laporan

Tabel 4.7 Tampilan Perancangan Antarmuka Pada Proses Jurnal Umum

PT. X
Jurnal Umum
Periode xx/xx/xxxx

Tanggal	Nomor Bukti	Keterangan	Ref	Debit	Kredit
xx/xx/xxxx	MDL1	Kas	111	xxx	-
		Modal	411	-	xxx
xx/xx/xxxx	PRL1	Peralatan	120	xxx	-
		Kas	111	-	xxx
xx/xx/xxxx	PLK1	Perlengkapan	113	xxx	-
		Kas	111	-	xxx
xx/xx/xxxx	SN	Deposit Pulsa	511	xxx	-
		Kas	111	-	xxx
xx/xx/xxxx	SN	Kas	111	xxx	-
		Penjualan	411	-	xxx
xx/xx/xxxx	SN	Kas	111	xxx	-
		Penjualan	411	-	xxx
xx/xx/xxxx	SN	Beban Kerugian Deposit Pulsa	512	xxx	-
		Deposit Pulsa	511	-	xxx
xx/xx/xxxx	PRV1	Prive	312	xxx	-
		Kas	111	-	xxx
TOTAL				**xxx**	**xxx**

Tabel 4.8 Tampilan Perancangan Antarmuka Pada Proses Buku Besar Umum

PT. X
Buku Besar Umum
Periode xx/xx/xxxx

Nama Akun: Kas Nomor Akun: 111

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
xx/xx/xxxx	Modal	311	xxx	-	xxx	-
xx/xx/xxxx	Peralatan	120	-	xxx	xxx	-
xx/xx/xxxx	Perlengkapan	113	-	xxx	xxx	-
xx/xx/xxxx	Deposit Pulsa	511	-	xxx	xxx	-
xx/xx/xxxx	Penjualan	411	xxx	-	xxx	-
xx/xx/xxxx	Penjualan	411	xxx	-	xxx	-
xx/xx/xxxx	Prive	312	xxx	-	xxx	-

Tabel 4.9 Tampilan Perancangan Antarmuka Pada Proses Neraca Saldo

PT.X
Laporan Neraca Saldo
Periode xx/xx/xxxx

Akun	Debit	Kredit
Kas	xxx	-
Perlengkapan	xxx	-
Peralatan	xxx	-
Modal	-	xxx
Prive	xxx	-
Penjualan	-	xxx
Beban Kerugian Deposit Pulsa	xxx	-
Deposit Pulsa	xxx	-

Tabel 4.10 Tampilan Perancangan Antarmuka Pada Proses Laba/Rugi

PT.X
Laporan Laba Rugi
Periode xx/xx/xxxx

Pendapatan		xxx
Harga Pokok Penjualan:		
Pembelian Deposit Pulsa	xxx	xxx
Laba Kotor		xxx
Beban usaha:		
Beban Perlengkapan	xxx	
Beban Kerugian Deposit Pulsa	xxx	xxx
Laba Bersih		xxx

Tabel 4.11 Tampilan Perancangan Antarmuka Pada Proses Neraca

PT.X
Neraca
Periode xx/xx/xxxx

AKTIVA		Kewajiban dan Modal	
Aktiva Lancar		Kewajiban	
Kas	xxx	Kewajiban Lancar	
Perlengkapan	xxx	Utang dagang	xxx
Persediaan	xxx	Utang bunga	xxx
Jumlah Aktiva	xxx	Utang gaji	xxx
Aktiva Tetap		Jumlah Kewajiban Lancar	xxx
Peralatan	xxx	Kewajiban Jangka Panjang	
		Utang bank	xxx
Nilai Buku	xxx		
		Jumlah Kewajiban	xxx
		Modal	
		Modal	xxx
Jumlah Aktiva	xxx	Jumlah Kewajiban dan Modal	xxx

Tabel 4.12 Tampilan Perancangan Antarmuka Pada Proses Perubahan Modal

PT.X
Laporan Perubahan Modal
Periode xx/xx/xxxx

Modal Awal		xxx
Laba Bersih	xxx	
Pengambilan Prive	xxx	xxx
		xxx
Modal Akhir		xxx

Tabel 4.13 Tampilan Perancangan Antarmuka Pada Proses Laporan Pendukung

PT.X
Laporan Pendukung
Periode xx/xx/xxxx

Deposit Masuk

Tanggal	Nomor Bukti	Harga
xxx	xxx	xxx
Total	xxx	xxx

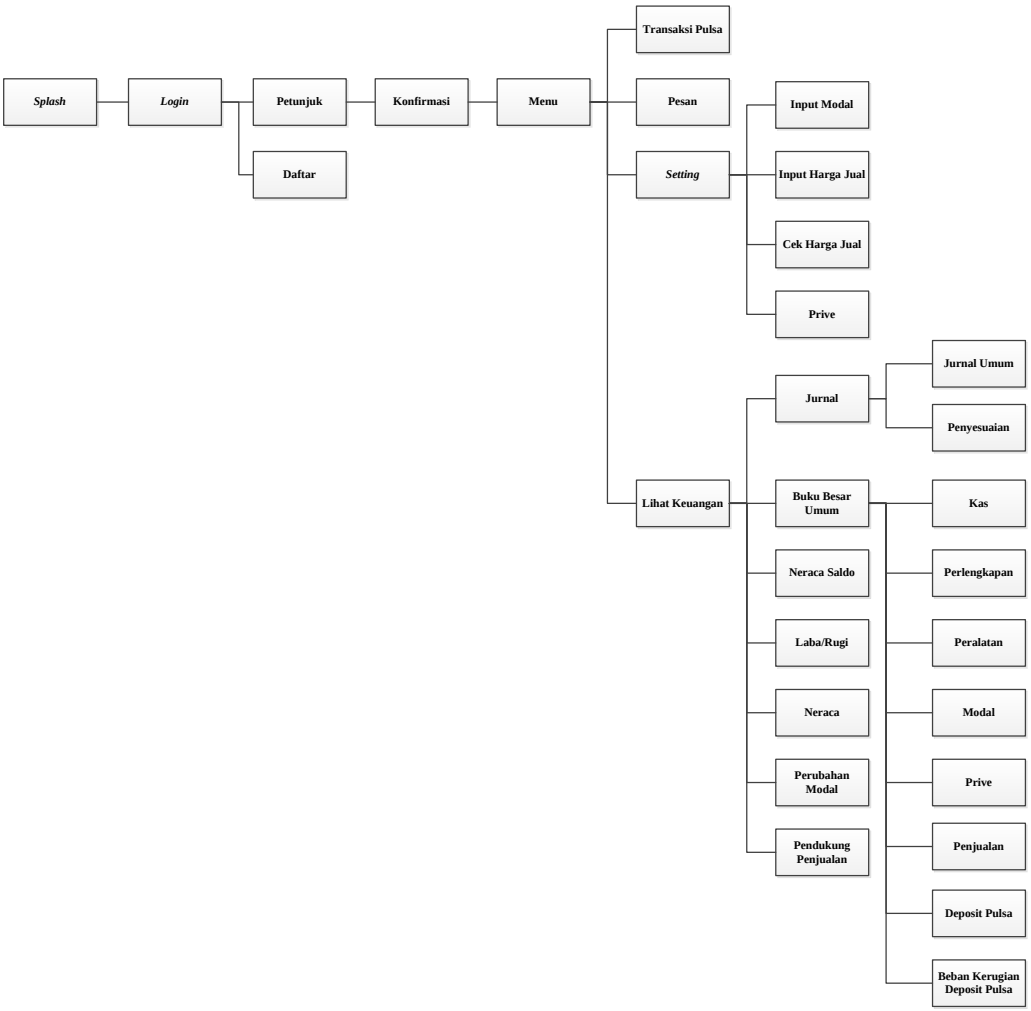
Deposit Keluar

Tanggal	No Bukti	Operator	HPP	Harga Jual	Laba
xxx	xxx	xxx	xxx	xxx	xxx
Total			xxx	xxx	xxx

Total Pembelian Deposit	xxx
Total HPP	xxx
Sisa Deposit	xxx
Total Laba	xxx

4.4 Perancangan Struktur Menu

Struktur menu program yang penulis rancang terdiri dari menu Transaksi Pulsa, Pesan, *Setting*, dan Lihat Keuangan.



Gambar 4.33 Struktur Menu Utama Program Keseluruhan

4.5 Tampilan Dan Kode Program

Berikut adalah tampilan dan kode program yang penulis rancang:



Gambar 4.34 Tampilan *Activity Splash*

Tabel 4.14 Tabel *Listing* Program *Activity Splash*

***Listing* Program**	**Keterangan**

package com.angga.blackproject;
 	
import android.os.Bundle;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.app.Activity;
import android.content.Intent;
 	
public class Splash extends Activity {
 	
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_splash);
 	
final TextView tv = (TextView) findViewById(R.id.textView1);
final ProgressBar pb = (ProgressBar) findViewById(R.id.progressBar1);
final Button btnMasuk = (Button) findViewById(R.id.btnMasuk);
btnMasuk.setOnClickListener(new View.OnClickListener() {
	Splash

Tabel 4.15 Tabel *Listing* Program *Activity Splash* (Lanjutan 1)

***Listing* Program**	**Keterangan**

public void onClick(View v) {
// TODO Auto-generated method stub
 	
tv.setVisibility(View.VISIBLE);
 	
pb.setVisibility(View.VISIBLE);
 	
btnMasuk.setVisibility(View.INVISIBLE);
Thread timer = new Thread() {
public void run() {
try {
//
berapalama splashscreen akan ditampilkan dalam
//
milisecond
 	
sleep(1000);
 	
} catch (Exception e) {
// TODO: handle exception
 	
e.printStackTrace();
} finally {
//
activity yang akan dijalankan setelah
//
splashscreen
//
selesai
finish();
Intent i = new Intent(Splash.this, Login.class);
 	
startActivity(i);
}
}
};
timer.start();
}
});
 	
}
 	
}	



Gambar 4.35 Tampilan *Activity Login*

Tabel 4.16 Tabel *Listing* Program *Activity Login*

***Listing* Program**	**Keterangan**
‘FRONT-END package com.angga.blackproject; import java.util.ArrayList; import org.apache.http.NameValuePair; import org.apache.http.message.BasicNameValuePair; import android.app.Activity; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.view.Window; import android.widget.Button; import android.widget.EditText; import android.widget.Toast; public class Login extends Activity implements OnClickListener { Button btnMasuk, btnDaftar; EditText nama, katasandi; private String url = "http://10.0.2.2/android/login.php"; @Override public void onCreate(Bundle	*Login*

Tabel 4.17 Tabel *Listing* Program *Activity Login* (Lanjutan 1)

***Listing* Program**	**Keterangan**

savedInstanceState) {
super.onCreate(savedInstanceState);
 	
requestWindowFeature(Window.FEATURE_NO_TITLE);
setContentView(R.layout.activity_login);
 	
nama = (EditText) findViewById(R.id.textNama);
katasandi = (EditText) findViewById(R.id.textKataSandi);
 	
btnMasuk = (Button) findViewById(R.id.btnMasuk);
btnDaftar = (Button) findViewById(R.id.btnDaftar);
btnMasuk.setOnClickListener(this);
btnDaftar.setOnClickListener(this);
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
if (v == btnMasuk) {
kirimData();
} else {
Intent i = new Intent(this, Daftar.class);
startActivity(i);
}
}
 	
private void kirimData() {
ArrayList<NameValuePair> kirimkephp = new ArrayList<NameValuePair>();
kirimkephp.add(new BasicNameValuePair("auser", nama.getText()
.toString()));
kirimkephp.add(new BasicNameValuePair("apass", katasandi.getText()
.toString()));
String respon = null;
if (nama.length() == 0 && katasandi.length()==0) {
nama.requestFocus();
Toast.makeText(this, "Masukkan nama dan password Anda", Toast.LENGTH_SHORT).show();
} else {
try {
respon = ClientToServer.eksekusiHttpPost(url, kirimkephp);
String res = respon.toString();
res = res.trim();	

Tabel 4.18 Tabel *Listing* Program *Activity Login* (Lanjutan 2)

***Listing* Program**	**Keterangan**

``
            if (res.equals("Login Sukses")) {
                        Toast.makeText(this,
                                    "Selamat
Datang, " + nama.getText().toString(),
 	
     Toast.LENGTH_SHORT).show();
                        Intent   i   =   new
Intent(this, Panduan.class);
                        i.putExtra("username",
nama.getText().toString());
                        i.putExtra("password",
katasandi.getText().toString());
                        startActivity(i);
                  } else {
 	
     Toast.makeText(this,res.toString(),Toast.
LENGTH_SHORT).show();
                  }
 	
            } catch (Exception e) {
                  e.printStackTrace();
                  Toast.makeText(this,
"Jaringan   tidak   terhubung",
Toast.LENGTH_SHORT)
                                    .show();
            }
      }
     }
 	
}
`BACK-END
<?php
        $user=$_POST['auser'];
        $pass=$_POST['apass'];
        $user=str_replace(" ", "",$user);
 	
$con =mysql_connect("localhost","","")
                        or die("koneksi
gagal");
mysql_select_db("anggagan_ebusiness",$con);
$sql = "SELECT * FROM user WHERE ID='$user'
AND PASSWORD='$pass'";
$hasil = mysql_query($sql,$con);
$num = mysql_num_rows($hasil);
 	
if ($num>0){
        echo "Login Sukses";
} else {
        echo "Gagal Login";
}
?>
``	



Gambar 4.36 Tampilan *Activity Panduan*

Tabel 4.19 Tabel *Listing* Program *Activity Panduan*

***Listing* Program**	**Keterangan**

`package com.angga.blackproject;`
 	
`import android.app.Activity;`
`import android.content.Intent;`
`import android.os.Bundle;`
`import android.view.View;`
`import android.view.View.OnClickListener;`
`import android.widget.Button;`
`import android.widget.TextView;`
 	
`public class Panduan extends Activity implements OnClickListener {`
`String hasilnama, hasilpass;`
`TextView status;`
 	
`@Override`
`public void onCreate(Bundle savedInstanceState) {`
`super.onCreate(savedInstanceState);`
 	
`setContentView(R.layout.activity_panduan);`
 	
`Bundle extra = getIntent().getExtras();`
`hasilnama = extra.getString("username");`
`hasilpass = extra.getString("password");`
	Panduan pemakai

Tabel 4.20 Tabel *Listing* Program *Activity Panduan* (Lanjutan 1)

***Listing* Program**	**Keterangan**

status = (TextView)findViewById(R.id.txStatus);
status.setText("Selamat Datang, " + hasilnama);
 	
Button btNext = (Button) findViewById(R.id.btNext);
btNext.setOnClickListener(this);
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
Intent i = new Intent(this, Port.class);
i.putExtra("username", hasilnama);
i.putExtra("password", hasilpass);
startActivity(i);
}
}	



Gambar 4.37 Tampilan *Activity Konfirmasi*

Tabel 4.21 Tabel *Listing* Program *Activity Konfirmasi*

***Listing* Program**	**Keterangan**

package com.angga.blackproject;
 	
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
 	
public class Port extends Activity implements OnClickListener {
EditText pin, noserver;
Button btnSet;
String hasilnama, hasilpass;
 	
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_port);
 	
Bundle extra = getIntent().getExtras();
hasilnama = extra.getString("username");
hasilpass = extra.getString("password");
 	
pin = (EditText) findViewById(R.id.textPin);
noserver = (EditText) findViewById(R.id.textNoServer);
 	
btnSet = (Button) findViewById(R.id.btnSet);
btnSet.setOnClickListener(this);
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
if(pin.length()!=0 && noserver.length() != 0){
AlertDialog.Builder ab = new AlertDialog.Builder(this)
.setMessage("Port: " + pin.getText().toString()
+ " No Server: "
+ noserver.getText().toString())
.setTitle("Konfirmasi")
.setPositiveButton("Ya",
new
	Konfirmasi Port

Tabel 4.22 Tabel *Listing* Program *Activity Konfirmasi* (Lanjutan 1)

***Listing* Program**	**Keterangan**

DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int which) {
// TODO Auto-generated method stub
 	
Intent i = new Intent(Port.this, TabMenu.class);
 	
i.putExtra("username", hasilnama);
 	
i.putExtra("password", hasilpass);
 	
i.putExtra("pin", pin.getText().toString());
 	
i.putExtra("noserver", noserver.getText().toString());
 	
startActivity(i);
}
})
.setNegativeButton("Tidak",
new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog,
int which) {
// TODO Auto-generated method stub
}
});
ab.show();
}else{
Toast.makeText(getBaseContext(), "Isi Pin dan No server", Toast.LENGTH_SHORT).show();
}
 	
}
}	



Gambar 4.38 Tampilan *Activity Transaksi Pulsa*

Tabel 4.23 Tabel *Listing* Program *Activity Pulsa*

***Listing* Program**	**Keterangan**
package com.angga.blackproject; import java.io.NotActiveException; import android.app.Activity; import android.app.AlertDialog; import android.app.PendingIntent; import android.content.BroadcastReceiver; import android.content.Context; import android.content.DialogInterface; import android.content.Intent; import android.content.IntentFilter; import android.os.Bundle; import android.telephony.SmsManager; import android.view.View; import android.view.View.OnClickListener; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.ImageView; import android.widget.Spinner; import android.widget.Toast; public class TransaksiPulsa extends Activity implements OnClickListener { Spinner spinKode; Button btKirim; ImageView simpati, as, mentari, im3, xl, smart, esia, fren, flexi, three,	Transaksi Pulsa

Tabel 4.24 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 2)

***Listing* Program**	**Keterangan**

```
            axis, starone;
    String msg, hasilnama, hasilpass, hasilpin,
hasilnoserver;
    EditText NoTujuan, Pin, NoServer;

    String[] simpatiKode = { "S5", "S10", "S20",
"S25", "S50", "S100" };
    String[] asKode = { "A5", "A10", "A20", "A25",
"A50", "A100" };
    String[] mentariKode = { "M2", "M5", "M10",
"M25", "M50", "M100" };
    String[] im3Kode = { "I2", "I5", "I10", "I25",
"I50", "I100" };
    String[] xlKode = { "X1", "X5", "X10", "X25",
"X50", "X100", "XE10",
            "XE50", "XE100" };
    String[] smartKode = { "SM5", "SM10", "SM20",
"SM50", "SM100" };
    String[] esiaKode = { "E1", "E5", "E10",
"E11", "E15", "E20", "E25", "E50",
            "E100" };
    String[] frenKode = { "FR10", "FR25", "FR50",
"FR100" };
    String[] flexiKode = { "FL5", "FL10", "FL20",
"FL50", "FL100" };
    String[] threekode = { "T1", "T5", "T10",
"T20", "T30", "T50", "T100" };
    String[] axisKode = { "AX1", "AX2", "AX3",
"AX4", "AX5", "AX10", "AX15",
            "AX20", "AX25", "AX50", "AX100" };
    String[] staroneKode = { "ST5", "ST10",
"ST25", "ST50", "ST100" };

    @Override
    public          void          onCreate(Bundle
savedInstanceState) {
      super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_transaksipuls
a);

      Bundle extra = getIntent().getExtras();
      hasilnama = extra.getString("username");
      hasilpass = extra.getString("password");
      hasilpin = extra.getString("pin");
      hasilnoserver = extra.getString("noserver");

      simpati            =            (ImageView)
findViewById(R.id.simpati);
      as = (ImageView) findViewById(R.id.as);
      mentari            =            (ImageView)
findViewById(R.id.mentari);
      im3 = (ImageView) findViewById(R.id.im3);
      xl = (ImageView) findViewById(R.id.xl);
```

Tabel 4.25 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 3)

***Listing* Program**	**Keterangan**

esia = (ImageView) findViewById(R.id.esia);
fren = (ImageView) findViewById(R.id.fren);
flexi = (ImageView) findViewById(R.id.flexi);
three = (ImageView) findViewById(R.id.three);
axis = (ImageView) findViewById(R.id.axis);
starone = (ImageView) findViewById(R.id.starone);
 	
simpati.setOnClickListener(this);
as.setOnClickListener(this);
mentari.setOnClickListener(this);
im3.setOnClickListener(this);
xl.setOnClickListener(this);
smart.setOnClickListener(this);
esia.setOnClickListener(this);
fren.setOnClickListener(this);
flexi.setOnClickListener(this);
three.setOnClickListener(this);
axis.setOnClickListener(this);
starone.setOnClickListener(this);
 	
spinKode = (Spinner) findViewById(R.id.spinKode);
NoTujuan = (EditText) findViewById(R.id.txNoTujuan);
NoTujuan.requestFocus();
 	
btKirim = (Button) findViewById(R.id.btKirim);
btKirim.setOnClickListener(this);
 	
ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, simpatiKode);
 	
aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa);
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
switch (v.getId()) {
case R.id.simpati:
Toast.makeText(getBaseContext(), "Simpati", Toast.LENGTH_SHORT)
.show();
ArrayAdapter<String> aa = new ArrayAdapter<String>(this,	

Tabel 4.26 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 4)

***Listing* Program**	**Keterangan**

android.R.layout.simple_spinner_item, simpatiKode);
 	
aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa);
break;
case R.id.as:
Toast.makeText(getBaseContext(), "AS", Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa1 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, asKode);
 	
aa1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa1);
break;
case R.id.mentari:
Toast.makeText(getBaseContext(), "Mentari", Toast.LENGTH_SHORT)
.show();
ArrayAdapter<String> aa2 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, mentariKode);
 	
aa2.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa2);
break;
case R.id.im3:
Toast.makeText(getBaseContext(), "IM3", Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa3 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, im3Kode);
 	
aa3.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa3);
break;
case R.id.xl:
Toast.makeText(getBaseContext(), "XL", Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa4 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, xlKode);	

Tabel 4.27 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 5)

***Listing* Program**	**Keterangan**

aa4.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa4);
break;
case R.id.smart:
Toast.makeText(getBaseContext(), "Smart", Toast.LENGTH_SHORT)
.show();
ArrayAdapter<String> aa5 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, smartKode);
 	
aa5.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa5);
break;
case R.id.esia:
Toast.makeText(getBaseContext(), "Esia", Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa6 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, esiaKode);
 	
aa6.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa6);
break;
case R.id.fren:
Toast.makeText(getBaseContext(), "Fren", Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa7 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, frenKode);
 	
aa7.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinKode.setAdapter(aa7);
break;
case R.id.flexi:
Toast.makeText(getBaseContext(), "Flexi", Toast.LENGTH_SHORT)
.show();
ArrayAdapter<String> aa8 = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item, flexiKode);	

Tabel 4.29 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 7)

***Listing* Program**	**Keterangan**

aa8.setDropDownViewResource(android.R.layout.simple_
spinner_dropdown_item);
spinKode.setAdapter(aa8);
break;
case R.id.three:
Toast.makeText(getBaseContext(), "3",
Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa9 = new
ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item,
threekode);
 	
aa9.setDropDownViewResource(android.R.layout.si
mple_spinner_dropdown_item);
spinKode.setAdapter(aa9);
break;
case R.id.axis:
Toast.makeText(getBaseContext(), "AXIS",
Toast.LENGTH_SHORT).show();
ArrayAdapter<String> aa10 = new
ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item,
axisKode);
 	
aa10.setDropDownViewResource(android.R.layout.s
imple_spinner_dropdown_item);
spinKode.setAdapter(aa10);
break;
case R.id.starone:
Toast.makeText(getBaseContext(),
"StarOne", Toast.LENGTH_SHORT)
.show();
ArrayAdapter<String> aa11 = new
ArrayAdapter<String>(this,
 	
android.R.layout.simple_spinner_item,
staroneKode);
 	
aa11.setDropDownViewResource(android.R.layout.s
imple_spinner_dropdown_item);
spinKode.setAdapter(aa11);
break;
 	
case R.id.btKirim:
msg =
spinKode.getSelectedItem().toString() + "."
+
NoTujuan.getText().toString() + "." + hasilpin;
if (NoTujuan.length() > 0 &&
msg.length() > 0) {
try {	

Tabel 4.30 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 8)

***Listing* Program**	**Keterangan**

```
nomor tujuan");
            }
            break;
      default:
            break;
      }

    }

    public void sendSMS(String tujuan,
String pesan) {
      String SENT = "SMS_SENT";
      String        DELEIVERED        =
"SMS_DELIVERED";

      PendingIntent         pi         =
PendingIntent.getBroadcast(this, 0,
                  new Intent(SENT), 0);
      PendingIntent    deliveri_pi     =
PendingIntent.getBroadcast(this, 0,
                  new
Intent(DELEIVERED), 0);

      BroadcastReceiver  brSend  =  new
BroadcastReceiver() {
            @Override
            public                  void
onReceive(Context arg0, Intent arg1) {
                  // TODO Auto-generated
method stub
                  switch
(getResultCode()) {
                  case
Activity.RESULT_OK:

    Toast.makeText(getBaseContext(),
"Terkirim",

    Toast.LENGTH_SHORT).show();
                        break;
                  case
SmsManager.RESULT_ERROR_GENERIC_FAILURE:

    Toast.makeText(getBaseContext(),
"Generic Failure",

    Toast.LENGTH_SHORT).show();
                        break;
                  case
SmsManager.RESULT_ERROR_NO_SERVICE:

    Toast.makeText(getBaseContext(),
"No Serveice",
```

Tabel 4.31 Tabel *Listing* Program *Activity Pulsa* (Lanjutan 9)

***Listing* Program**	**Keterangan**

case SmsManager.RESULT_ERROR_NULL_PDU:
 	
Toast.makeText(getBaseContext(), "Null PDU",
 	
Toast.LENGTH_SHORT).show();
break;
case SmsManager.RESULT_ERROR_RADIO_OFF:
 	
Toast.makeText(getBaseContext(), "Radio Off",
 	
Toast.LENGTH_SHORT).show();
break;
default:
break;
}
}
};
BroadcastReceiver brDelivered = new BroadcastReceiver() {
 	
@Override
public void onReceive(Context context, Intent intent) {
// TODO Auto-generated method stub
switch (getResultCode()) {
case Activity.RESULT_OK:
 	
Toast.makeText(getBaseContext(), "SMS Diterima",
 	
Toast.LENGTH_SHORT).show();
break;
case Activity.RESULT_CANCELED:
 	
Toast.makeText(getBaseContext(), "Sms gagal",
 	
Toast.LENGTH_SHORT).show();
break;
}
}
};
registerReceiver(brSend, new IntentFilter(SENT));
registerReceiver(brDelivered, new IntentFilter(DELEIVERED));
SmsManager smsman = SmsManager.getDefault();
smsman.sendTextMessage(tujuan, null,	



Gambar 4.39 Tampilan *Activity Pesan*

Tabel 4.32 Tabel *Listing* Program *Activity Pesan*

***Listing* Program**	**Keterangan**
package com.angga.blackproject; import java.util.ArrayList; import android.app.Activity; import android.content.ContentResolver; import android.content.Intent; import android.content.IntentFilter; import android.database.Cursor; import android.net.Uri; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.widget.AdapterView; import android.widget.AdapterView.OnItemClickListener; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ListView; import android.widget.Toast; public class SmsList extends Activity implements OnItemClickListener, OnClickListener { Button btnSimpan, btnKonfirmasiDeposit; String hasilnama, hasilpass, hasilpin, hasi	Pesan masuk

Tabel 4.33 Tabel *Listing* Program *Activity Pesan* (Lanjutan 1)

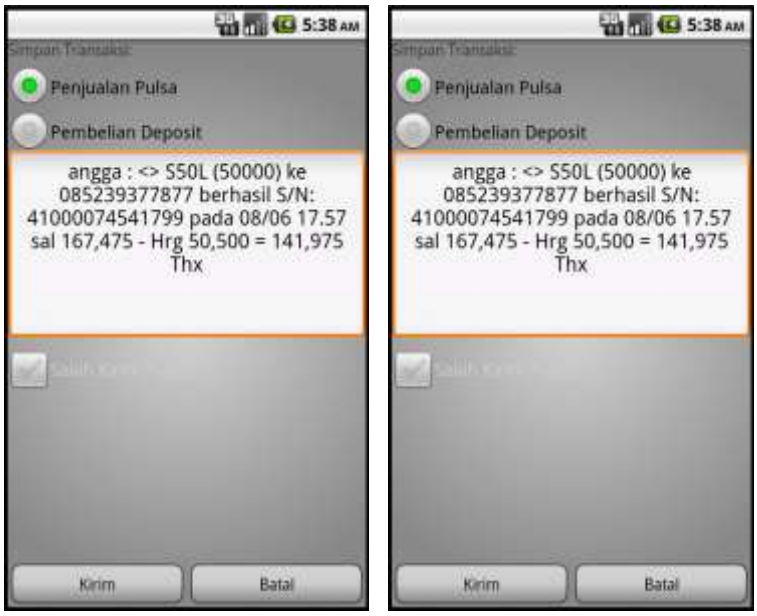
***Listing* Program**	**Keterangan**

smsList.add(str);
} while (cursor.moveToNext());
 	
ListView smsListView = (ListView) findViewById(R.id.SmsList);
smsListView.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1, smsList));
smsListView.setOnItemClickListener(this);
}
 	
ArrayList<String> smsList = new ArrayList<String>();
 	
public void onItemClick(AdapterView<?> parent, View view, int pos, long id) {
try {
String[] splitted = smsList.get(pos).split("\n");
String sender = splitted[0];
String encryptedData = "";
for (int i = 1; i < splitted.length; ++i) {
encryptedData += splitted[i];
}
String data = sender
+ "\n"
+ StringCryptor.decrypt(new String(SmsReceiver.PASSWORD),
 	
encryptedData);
Toast.makeText(this, data, Toast.LENGTH_SHORT).show();
} catch (Exception e) {
e.printStackTrace();
}
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
if (v == btnSimpan) {
Intent i = new Intent(this, SimpanTransaksi.class);
i.putExtra("username", hasilnama);
i.putExtra("password", hasilpass);
startActivity(i);
}
}
 	
}	

Tabel 4.34 Tabel *Listing* Program *Activity Pesan* (Lanjutan 2)

***Listing* Program**	**Keterangan**

`@Override`
`public void onCreate(Bundle savedInstanceState) {`
`super.onCreate(savedInstanceState);`
`setContentView(R.layout.activity_smslist);`
 	
`Bundle extra = getIntent().getExtras();`
`hasilnama = extra.getString("username");`
`hasilpass = extra.getString("password");`
`hasilpin = extra.getString("pin");`
`hasilnoserver = extra.getString("noserver");`
 	
`btnSimpan = (Button) findViewById(R.id.btnSimpan);`
`btnSimpan.setOnClickListener(this);`
 	
`/*`
`* You can also register your intent filter here. And here is example`
`* how to do this.`
`*/`
 	
`IntentFilter filter = new IntentFilter(`
 	
`"android.provider.Telephony.SMS_RECEIVED");`
 	
`filter.setPriority(IntentFilter.SYSTEM_HIGH_PRIORITY);`
`registerReceiver(new SmsReceiver(), filter);`
 	
`ContentResolver contentResolver = getContentResolver();`
`Cursor cursor = contentResolver.query(Uri.parse("content://sms/inbox"),`
`null, null, null, null);`
 	
`int indexBody = cursor.getColumnIndex(SmsReceiver.BODY);`
`int indexAddr = cursor.getColumnIndex(SmsReceiver.ADDRESS);`
 	
`if (indexBody < 0 || !cursor.moveToFirst())`
`return;`
 	
`smsList.clear();`
 	
`do {`
`String str = "Pengirim: " +`
`cursor.getString(indexAddr) + "\n"`	



Gambar 4.40 Tampilan *Activity Simpan Transaksi*

Tabel 4.35 Tabel *Listing* Program *Activity Transaksi*

***Listing* Program**	**Keterangan**
‘FRONT-END package com.angga.blackproject; import java.util.ArrayList; import org.apache.http.NameValuePair; import org.apache.http.message.BasicNameValuePair; import android.app.Activity; import android.app.AlertDialog; import android.content.DialogInterface; import android.content.res.ColorStateList; import android.graphics.Color; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.widget.Button; import android.widget.CheckBox; import android.widget.CompoundButton; import android.widget.CompoundButton.OnCheckedChangeListen er; import android.widget.EditText; import android.widget.LinearLayout; import android.widget.RadioButton; import android.widget.Toast;	Simpan Transaksi

Tabel 4.36 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 1)

***Listing* Program**	**Keterangan**

public class SimpanTransaksi extends Activity implements OnClickListener, OnCheckedChangeListener {
Button btnKirim, btnBatal;
LinearLayout ln1,ln2;
EditText pesan;
CheckBox cbSlhKirim;
RadioButton rbPenjualanPulsa, rbPembelianDeposit;
String hasilnama, hasilpass, hasilpin, hasilnoserver, jenis,rugi;
private String url = "http://10.0.2.2/android/simpantransaksi.php";
 	
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
 	
setContentView(R.layout.activity_simpantransaksi);
 	
Bundle extra = getIntent().getExtras();
hasilnama = extra.getString("username");
hasilpass = extra.getString("password");
 	
pesan = (EditText) findViewById(R.id.txPesan);
ln1=(LinearLayout)findViewById(R.id.linear1);
ln2=(LinearLayout)findViewById(R.id.linear2);
cbSlhKirim =(CheckBox)findViewById(R.id.checkSlhKirim);
cbSlhKirim.setOnCheckedChangeListener(this);
 	
rbPenjualanPulsa = (RadioButton)findViewById(R.id.radioPenjualanPulsa);
 	
rbPembelianDeposit=(RadioButton)findViewById(R.id.radioPembelianDeposit);
 	
rbPenjualanPulsa.setOnCheckedChangeListener(this);
 	
rbPembelianDeposit.setOnCheckedChangeListener(this);
 	
btnKirim = (Button) findViewById(R.id.btnKirim);
btnBatal = (Button)	

Tabel 4.37 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 2)

***Listing* Program**	**Keterangan**

findViewById(R.id.btnBatal);
btnKirim.setOnClickListener(this);
btnBatal.setOnClickListener(this);
}
 	
public void onClick(View v) {
// TODO Auto-generated method stub
if (v == btnKirim) {
if(rbPenjualanPulsa.isChecked() && pesan.length()>=15){
kirimData();
}else if(rbPembelianDeposit.isChecked() && pesan.length()>=8){
kirimData();
}else{
Toast.makeText(this, "Format sms harus sesuai!", Toast.LENGTH_LONG).show();
}
} else {
finish();
}
}
 	
private void kirimData() {
ArrayList<NameValuePair> kirimkephp = new ArrayList<NameValuePair>();
kirimkephp.add(new BasicNameValuePair("message", pesan.getText()
.toString()));
kirimkephp.add(new BasicNameValuePair("anama", hasilnama));
kirimkephp.add(new BasicNameValuePair("apass", hasilpass));
kirimkephp.add(new BasicNameValuePair("ajenis", jenis));
kirimkephp.add(new BasicNameValuePair("aya", "tidak"));
kirimkephp.add(new BasicNameValuePair("arugi", rugi));
String respon = null;
try {
respon = ClientToServer.eksekusiHttpPost(url, kirimkephp);
String res = respon.toString();
res = res.trim();
Toast.makeText(this, res.toString(),
 	
Toast.LENGTH_SHORT).show();
try {
AlertDialog.Builder ab = new AlertDialog.Builder(this)
 	
.setMessage(res.toString(	

Tabel 4.38 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 3)

***Listing* Program**	**Keterangan**

DialogInterface.OnClickListener() {
 	
public void onClick(
 	
DialogInterface dialog,
 	
int which) {
 	
// TODO Auto-generated method stub
 	
yakirimdata();
}
})
 	
.setNegativeButton("Tidak",
new
DialogInterface.OnClickListener() {
 	
public void onClick(
 	
DialogInterface dialog,
 	
int which) {
 	
// TODO Auto-generated method stub
}
});
ab.show();
} catch (Exception e) {
// TODO: handle exception
}
} catch (Exception e) {
e.printStackTrace();
}
}
 	
public void yakirimdata(){
ArrayList<NameValuePair> kirimkephp = new
ArrayList<NameValuePair>();
kirimkephp.add(new
BasicNameValuePair("message", pesan.getText()
.toString()));
kirimkephp.add(new
BasicNameValuePair("anama", hasilnama));
kirimkephp.add(new
BasicNameValuePair("apass", hasilpass));
kirimkephp.add(new
BasicNameValuePair("ajenis", jenis));
kirimkephp.add(new
BasicNameValuePair("aya", "ya"));
kirimkephp.add(new
BasicNameValuePair("arugi", rugi));
String respon = null;
try	

Tabel 4.39 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 4)

***Listing* Program**	**Keterangan**

```
res = res.trim();
            if (res.equals("Berhasil")) {
                  Toast.makeText(this,       "Data
berhasil disimpan",

    Toast.LENGTH_LONG).show();
                  finish();
            }
      } catch (Exception e) {
            // TODO: handle exception
      }
    }

    public  void  onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
      // TODO Auto-generated method stub
      if(rbPenjualanPulsa.isChecked()){
            ln1.setVisibility(View.VISIBLE);
            ln2.setVisibility(View.VISIBLE);

    cbSlhKirim.setVisibility(View.VISIBLE);
            pesan.setText("PROCELL : <TRX LANCAR>
S10L  (10000)  ke  085239377877  berhasil  S/N:
41000074541799 pada 08/06 17.57 sal 167,475 - Hrg
10,000 = 141,975 Thx");
            pesan.setHint("contoh  :  PROCELL  :
<TRX  LANCAR>  S10L  (10000)  ke  085239377877
berhasil S/N: 41000074541799 pada 08/06 17.57 sal
167,475 - Hrg 10,000 = 141,975 Thx");
            pesan.requestFocus();
            if(pesan.length()==0){
                  pesan.setError("Paste-kan
report sms penjualan pulsa di sini");
            }
            jenis="penjualanpulsa";
      }else{
            cbSlhKirim.setChecked(false);
            cbSlhKirim.setVisibility(View.GONE);
            ln1.setVisibility(View.VISIBLE);
            ln2.setVisibility(View.VISIBLE);
            pesan.setText("PROCELL   :   Deposit
(PROCELL - 1012-0024) bertambah Rp. 200,000 Saldo
Rp. 222.225 Thx");
            pesan.setHint("contoh  :  PROCELL  :
Deposit  (PROCELL  -  1012-0024)  bertambah  Rp.
200,000 Saldo Rp. 222.225 Thx");
            pesan.requestFocus();
            if(pesan.length()==0){
                  pesan.setError("Paste-kan
report sms pembelian deposit di sini");
            }
            jenis="pembeliandeposit";
      }
      if(cbSlhKirim.isChecked()){
```

Tabel 4.40 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 5)

***Listing* Program**	**Keterangan**

```
rugi="tidakrugi";
      }
    }
}
'BACK-END
<?php
$msgx =$_POST['message'];
$msg  =str_replace(",", "",$msgx);
$msg  =str_replace("(", "",$msg);
$msg  =str_replace(")", "",$msg);
$msg  =str_replace(".", "",$msg);
$msg =explode(" ", $msg);

$jenis      =$_POST['ajenis'];
$id   =$_POST['anama'];
$pass =$_POST['apass'];
$ya   =$_POST['aya'];
$rugi =$_POST['arugi'];
$tgl =gmdate("Y/m/d H:i:s", time()+60*60*7);
$date =gmdate("d/m/y", time()+60*60*7);

$con =mysql_connect("localhost","root","")
           or die("koneksi gagal");
mysql_select_db("anggagan_ebusiness",$con);

//SELEKSI USER
$sql = "SELECT * FROM user WHERE ID='$id' AND
PASSWORD='$pass'";
$hasil = mysql_query($sql,$con);
$row = mysql_fetch_row($hasil);

    if($jenis=="penjualanpulsa"  &&  $ya=="tidak"
&& $rugi=="tidakrugi"){
//SELEKSI HARGA
$sql = "SELECT * FROM $row[6]"."hargajual WHERE
KODE_PRODUK='$msg[4]'";
$hasil = mysql_query($sql,$con);
$bar = mysql_fetch_row($hasil);
$selisih=$bar[2]-$msg[18];
echo              "===PenjualanPulsa===\nProvider
".$bar[0]."\nNominal             ".$msg[5]."\nKode
".$msg[4]."\nHarga         distributor         Rp.
".$msg[18]."\nHarga jual Rp. ".$bar[2]."\nSelisih
harga Rp. ".$selisih."\nSimpan transaksi?";
}   elseif($jenis=="penjualanpulsa"            &&
$ya=="tidak" && $rugi=="rugi"){
//SELEKSI HARGA
$sql = "SELECT * FROM $row[6]"."hargajual WHERE
KODE_PRODUK='$msg[4]'";
$hasil = mysql_query($sql,$con);
$bar = mysql_fetch_row($hasil);
$selisih=$bar[2]-$msg[18];
echo "Salah Kirim\n===PenjualanPulsa===\nProvider
```

Tabel 4.41 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 6)

***Listing* Program**	**Keterangan**

```
".$msg[18]."\nHarga jual Rp. ".$bar[2]."\nSelisih
harga Rp. ".$selisih."\nSimpan transaksi?";
}   elseif($jenis=="pembeliandeposit"          &&
$ya=="tidak"){
echo       "===PembelianDeposit===\nKode       SN
".$msg[5]."\nSebesar   Rp.   ".$msg[8]."\n\nSimpan
transaksi?";
}

    if($jenis=="penjualanpulsa"  &&  $ya=="ya"  &&
$rugi=="tidakrugi"){
//INPUT TRANSAKSI
$sql    =    "INSERT    INTO    $row[6]"."transaksi
VAlUES('$msgx',  '$msg[4]',  '$msg[5]',  '$msg[7]',
'$msg[10]',  '$msg[12]',  '$msg[13]',  '$msg[15]',
'$msg[18]', '$msg[20]', '$tgl')";
$hasil = mysql_query($sql,$con);
//SELEKSI HARGA
$sql  =  "SELECT  *  FROM  $row[6]"."hargajual  WHERE
KODE_PRODUK='$msg[4]'";
$hasil = mysql_query($sql,$con);
$bar = mysql_fetch_row($hasil);
//INPUT JURNAL
$sql    =    "INSERT    INTO    $row[6]"."jurnal
VALUES('$date',    '$msg[10]',    'Kas',    '111',
$bar[2], '0', '$tgl')";
$hasil = mysql_query($sql,$con);
$sql  =  "INSERT  INTO  $row[6]"."jurnal  VALUES('',
'', 'Penjualan', '411', '0', $bar[2], '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR KAS
$sql      =      "SELECT      SUM(DEBIT)      AS
total_debit,SUM(KREDIT)   AS   total_kredit   FROM
$row[6]"."bbkas";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo       =       ($ttl['total_debit']       -
$ttl['total_kredit']) + $bar[2];
$sql    =    "INSERT    INTO    $row[6]"."bbkas
VALUES('$date',  'Penjualan',  '411',  '$bar[2]',
'0', '$saldo', '0', '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR PENJUALAN
$sql  =  "SELECT  SUM(KREDIT)  AS  total_kredit  FROM
$row[6]"."bbpenjualan";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo = $ttl['total_kredit'] + $bar[2];
$sql   =   "INSERT   INTO   $row[6]"."bbpenjualan
VALUES('$date',  'Kas',  '111',  '0',  '$bar[2]',
'0', '$saldo', '$tgl')";
$hasil = mysql_query($sql,$con);
    if($hasil){
    echo "Berhasil";
```

Tabel 4.42 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 7)

***Listing* Program**	**Keterangan**

```
}

}   elseif($jenis=="penjualanpulsa" && $ya=="ya"
&& $rugi=="rugi"){
//INPUT TRANSAKSI
$sql   =   "INSERT   INTO   $row[6]"."transaksi
VAlUES('$msgx', '$msg[4]', '$msg[5]', '$msg[7]',
'$msg[10]', '$msg[12]', '$msg[13]', '$msg[15]',
'$msg[18]', '$msg[20]', '$tgl')";
$hasil = mysql_query($sql,$con);
//SELEKSI HARGA
$sql = "SELECT * FROM $row[6]"."hargajual WHERE
KODE_PRODUK='$msg[4]'";
$hasil = mysql_query($sql,$con);
$bar = mysql_fetch_row($hasil);
//INPUT JURNAL
$sql    =    "INSERT    INTO    $row[6]"."jurnal
VALUES('$date',   '$msg[10]',   'Beban   Kerugian
Deposit Pulsa', '512', $bar[2], '0', '$tgl')";
$hasil = mysql_query($sql,$con);
$sql = "INSERT INTO $row[6]"."jurnal VALUES('',
'', 'Deposit Pulsa', '-', '0', $bar[2], '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR KERUGIAN DEPOSIT PULSA
$sql = "SELECT SUM(SDEBIT) AS total_debit FROM
$row[6]"."bbkerugiandepositpulsa";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo = $ttl['total_debit'] + $bar[2];
$sql          =          "INSERT          INTO
$row[6]"."bbkerugiandepositpulsa   VALUES('$date',
'Deposit Pulsa', '-', '$bar[2]', '0', '$saldo',
'0', '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR DEPOSIT PULSA
$sql      =      "SELECT      SUM(DEBIT)      AS
total_debit,SUM(KREDIT)   AS   total_kredit   FROM
$row[6]"."bbkas";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo       =       ($ttl['total_debit']       -
$ttl['total_kredit']) + $bar[2];
$sql   =   "INSERT   INTO   $row[6]"."bbdepositpulsa
VALUES('$date', 'Beban Kerugian Deposit Pulsa',
'512', '0', '$bar[2]', '$saldo', '0',  '$tgl')";
$hasil = mysql_query($sql,$con);
if($hasil){
echo "Berhasil";
}
}   elseif($jenis=="pembeliandeposit"          &&
$ya=="ya"){
//INPUT TRANSAKSI
$sql = "INSERT INTO $row[6]"."transaksi
```

Tabel 4.43 Tabel *Listing* Program *Activity Transaksi* (Lanjutan 8)

***Listing* Program**	**Keterangan**

'', '', '', '$msg[11]', '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT JURNAL
$sql = "INSERT INTO $row[6]"."jurnal VALUES('$date', '$msg[5]', 'Pembelian Deposit Pulsa', '511', '$msg[8]', '0', '$tgl')";
$hasil = mysql_query($sql,$con);
$sql = "INSERT INTO $row[6]"."jurnal VALUES('', '', 'Kas', '111', '0', '$msg[8]', '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR KAS
$sql = "SELECT SUM(DEBIT) AS total_debit,SUM(KREDIT) AS total_kredit FROM $row[6]"."bbkas";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo = ($ttl['total_debit'] - $ttl['total_kredit']) - $msg[8];
$sql = "INSERT INTO $row[6]"."bbkas VALUES('$date', 'Deposit Pulsa', '-', '0', '$msg[8]', '$saldo', '0', '$tgl')";
$hasil = mysql_query($sql,$con);
//INPUT BUKU BESAR DEPOSIT PULSA
$sql = "SELECT SUM(DEBIT) AS total_debit FROM $row[6]"."bbdepositpulsa";
$hasil = mysql_query($sql,$con);
$ttl = mysql_fetch_array($hasil);
$saldo = $ttl['total_debit'] + $msg[8];
$sql = "INSERT INTO $row[6]"."bbpembeliandepositpulsa VALUES('$date', 'Kas', '111', '$msg[8]', '0', '$saldo', '0', '$tgl')";
$hasil = mysql_query($sql,$con);
if($hasil){
echo "Berhasil";
}
}
mysql_close();
?>	



Gambar 4.41 Tampilan *Activity Setting*

Tabel 4.44 Tabel *Listing* Program *Activity Setting*

***Listing* Program**	**Keterangan**

package com.angga.blackproject;
 	
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import
android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
 	
public class Setting extends Activity implements OnItemClickListener {
ListView list;
String[] Menu = { "Cek Harga Jual", "Update Harga Jual", "Update Modal" };
String hasilnama, hasilpass, hasilpin, hasilnoserver;
 	
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
 	
setContentView(R.layout.activity_tabsetting);
 	
Bundle extra = getIntent().getExtras();
	Setting

Tabel 4.45 Tabel *Listing* Program *Activity Setting* (Lanjutan 1)

***Listing* Program**	**Keterangan**

hasilnama = extra.getString("username");
hasilpass = extra.getString("password");
hasilpin = extra.getString("pin");
hasilnoserver = extra.getString("noserver");
 	
list = (ListView) findViewById(R.id.listtool);
ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
 	
android.R.layout.simple_list_item_1, Menu);
list.setAdapter(aa);
list.setOnItemClickListener(this);
}
 	
public void onItemClick(AdapterView<?> arg0, View arg1, int pos, long arg3) {
// TODO Auto-generated method stub
if (Menu[pos].equals("Update Harga Jual")) {
Intent i = new Intent(this, ListHargaJual.class);
i.putExtra("username", hasilnama);
i.putExtra("password", hasilpass);
i.putExtra("pin", hasilpin);
i.putExtra("noserver", hasilnoserver);
startActivity(i);
} else if (Menu[pos].equals("Update Modal")) {
Intent i = new Intent(this, Modal.class);
i.putExtra("username", hasilnama);
i.putExtra("password", hasilpass);
i.putExtra("pin", hasilpin);
i.putExtra("noserver", hasilnoserver);
startActivity(i);
}	



Gambar 4.42 Tampilan *Activity Lihat Keuangan*

Tabel 4.22 Tabel *Listing* Program *Activity Keuangan*

***Listing* Program**	**Keterangan**
'FRONT-END package com.angga.blackproject; import android.app.Activity; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.view.View.OnClickListener; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.CompoundButton; import android.widget.CompoundButton.OnCheckedChangeListen er; import android.widget.EditText; import android.widget.RadioButton; import android.widget.Spinner; import android.widget.Toast; public class LihatKeuangan extends Activity implements OnClickListener, OnCheckedChangeListener { RadioButton rbJurnal, rbBukuBesar, rbNeraca, rbLabaRugi, rbPerBulan, rbPerTahun; String hasilnama, hasilpass, hasilpin, hasilnoserver, periode; Button btnCek; EditText txTahun; Spinner spinBula	Lihat Keuangan

Tabel 4.23 Tabel *Listing* Program *Activity Keuangan* (Lanjutan 1)

***Listing* Program**	**Keterangan**

```
<?php
if($periode=="pertahun"){
$sql = "SELECT * FROM $row[6]"."jurnal WHERE
YEAR(TANGGALX)='$tahun'";
}else{
$sql = "SELECT * FROM $row[6]"."jurnal WHERE
MONTH(TANGGALX)='$bln 'AND
YEAR(TANGGALX)='$tahun'";
}
$hasil = mysql_query($sql,$con);
$i=1;
while($bar=mysql_fetch_row($hasil)){
$i++;
        <td style='border-bottom-color:#FFF'
align='center'>".$bar[3]."</td>
        <td style='border-bottom-color:#FFF'
align='right'>".number_format($bar[4])."</td>
        <td style='border-bottom-color:#FFF'
align='right'>".number_format($bar[5])."</td>
        </tr>";
}else{
echo"
<tr style='border-bottom-color:#CCC'>
        <td  align='right'>".$bar[2]."</td>
        <td  align='center'>".$bar[3]."</td>
        <td
align='right'>".number_format($bar[4])."</td>
        <td
align='right'>".number_format($bar[5])."</td>
        </tr>";
?>
<?php
if($periode=="pertahun"){
$sql = "SELECT SUM(DEBIT) AS
total_debit,SUM(KREDIT) AS total_kredit FROM
$row[6]"."jurnal WHERE YEAR(TANGGALX)='$tahun'";
}else{
$sql = "SELECT SUM(DEBIT) AS
total_debit,SUM(KREDIT) AS total_kredit FROM
$row[6]"."jurnal WHERE MONTH(TANGGALX)='$bln' AND
YEAR(TANGGALX)='$tahun'";
        <td
align='right'><b>".number_format($ttl['total_debit'
])."</b></td>
        <td
align='right'><b>".number_format($ttl['total_kredit
'])."</b></td>
        </tr>";
mysql_close($con);
?>
</table>
</body>
</html>
```

Angga

Jurnal Umum

Periode September 2012

Tanggal	Nomor Bukti	Keterangan	Ref	Debit	Kredit
21/09/12	MDL1	Kas	111	7,000,000	0
		Modal	411	0	7,000,000
21/09/12	PRL1	Peralatan	120	1,600,000	0
		Kas	111	0	1,600,000
21/09/12	PLK1	Perlengkapan	113	500,000	0
		Kas	111	0	500,000
21/09/12	1012-0024	Deposit Pulsa	511	400,000	0
		Kas	111	0	400,000
21/09/12	41000074541799	Kas	111	52,000	0
		Penjualan	411	0	52,000
21/09/12	41000074541799	Kas	111	52,000	0
		Penjualan	411	0	52,000
21/09/12	41000074541799	Kas	111	52,000	0
		Penjualan	411	0	52,000
21/09/12	PRV1	Prive	312	200,000	0
		Kas	111	0	200,000
TOTAL				**9,856,000**	**9,856,000**

Gambar 4.43 Tampilan *Jurnal Umum*

Angga

Buku Besar Umum

Periode September 2012

Nama Akun : Kas — Nomor Akun : 111

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Modal	311	7,000,000	0	7,000,000	0
21/09/12	Peralatan	120	0	1,600,000	5,400,000	0
21/09/12	Perlengkapan	113	0	500,000	4,900,000	0
21/09/12	Deposit Pulsa	511	0	400,000	4,500,000	0
21/09/12	Penjualan	411	52,000	0	4,552,000	0
21/09/12	Penjualan	411	52,000	0	4,604,000	0
21/09/12	Penjualan	411	52,000	0	4,656,000	0
21/09/12	Prive	113	0	200,000	4,456,000	0

Nama Akun : Perlengkapan — Nomor Akun : 116

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	111	500,000	0	500,000	0

Nama Akun : Peralatan — Nomor Akun : 121

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	111	1,600,000	0	1,600,000	0

Nama Akun : Modal — Nomor Akun : 311

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	311	0	7,000,000	7,000,000	0

Nama Akun : Prive — Nomor Akun : 312

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	111	200,000	0	200,000	0

Nama Akun : Penjualan — Nomor Akun : 411

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	111	0	52,000	52,000	0
21/09/12	Kas	111	0	52,000	104,000	0
21/09/12	Kas	111	0	52,000	156,000	0

Nama Akun : Deposit Pulsa — Nomor Akun : 511

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
21/09/12	Kas	111	400,000	0	400,000	0

Nama Akun : Beban Kerugian Deposit Pulsa — Nomor Akun : 512

Tanggal	Keterangan	Ref	Debit	Kredit	Saldo	
					Debit	Kredit

Gambar 4.44 Tampilan *Buku Besar Umum*

Angga

Neraca Saldo

Periode September 2012

Akun	Debit	Kredit
Kas	4,456,000	0
Perlengkapan	500,000	0
Peralatan	1,600,000	0
Modal	0	7,000,000
Prive	200,000	0
Penjualan	0	156,000
Beban Kerugian Deposit Pulsa	0	0
Deposit Pulsa	400,000	0
TOTAL	**7,156,000**	**7,156,000**

Gambar 4.45 Tampilan *Neraca Saldo*

Angga

Laporan Laba Rugi

Periode September 2012

Penjualan			156,000
Persediaan Awal	0		
Pembelian	400,000		
Barang siap dijual		400,000	
Persediaan Akhir		248,500	
HPP			151,500
Laba Kotor			4,500
Beban usaha:			
Beban Perlengkapan	0		
Beban Kerugian Deposit Pulsa	0		0
Laba Bersih			**4,500**

Gambar 4.46 Tampilan *Laba Rugi*

Angga

Neraca

Periode September 2012

AKTIVA		KEWAJIBAN DAN MODAL	
Aktiva Lancar		**Kewajiban**	
Kas	4,456,000	Kewajiban Lancar	
Perlengkapan	500,000	Utang dagang	0
Persediaan	248,500	Utang bunga	0
Jumlah Aktiva Lancar	5,204,500	Utang gaji	0
		Jumlah Kewajiban Lancar	0
Aktiva Tetap			
Peralatan	1,600,000		
		Kewajiban Jangka Panjang	
Nilai Buku	1,600,000	Utang Bank	0
		Jumlah Kewajiban	0
		Modal	
		Modal	6,804,500
Jumlah Aktiva	**6,804,500**	**Jumlah Kewajiban dan Modal**	**6,804,500**

Gambar 4.47 Tampilan *Neraca*

Angga

Laporan Perubahan Modal

Periode September 2012

Modal angga, 1 September 2012		7,000,000
Laba bersih	4,500	
Pengambilan Prive	200,000	-195,500
Modal angga, 31 September 2012		**6,804,500**

Gambar 4.48 Tampilan *Perubahan Modal*

Angga
Laporan Pendukung Penjualan
Periode September 2012

Deposit Masuk

Tanggal	Nomor Bukti	Harga
21/09/12	1012-0024	400,000
Total		400,000

Deposit Keluar

Tanggal	Nomor Bukti	Operator	HPP	Harga Jual	Laba
21/09/12	41000074541799	telkomsel	50,500	52,000	1,500
21/09/12	41000074541799	telkomsel	50,500	52,000	1,500
21/09/12	41000074541799	telkomsel	50,500	52,000	1,500
Total			151,500	156,000	4,500

Total Pembelian Deposit	400,000
Total HPP	151,500
Sisa Deposit	248,500
Total Laba	4,500

Gambar 4.49 Tampilan *Perndukung Penjualan*



Gambar 4.50 Tampilan *Cek Harga Jual*

4.6 Konversi Dan Komponen Sistem

4.6.1 Pengguna *(Brainware)*

Sumber daya manusia yang diperlukan untuk menjalankan program *e-business* penjualan pulsa elektrik perlengkapan ini minimal mengerti cara menggunakan telepon seluler berbasis android dan sedikitnya mengerti akuntansi.

4.6.2 Perangkat Keras *(Hardware)*

Hardware yang diperlukan untuk menjalankan program aplikasi *e-business* penjualan pulsa elektrik perlengkapan ini adalah: *Handphone Platform Android* minimal versi *Froyo, Memory RAM* minimal *512mb, Proccecor* minimal 512Mhz.

4.6.3 Perangkat Lunak *(Software)*

4.6.3.1 Perangkat Lunak Sistem

Perangkat lunak sistem yang digunakan untuk mengembangkan program yaitu Windows 7. Karena sistem operasi ini dapat mendukung *e-business* penjualan pulsa elektrik yang penulis rancang.

4.6.3.2 Perangkat Lunak Pemograman

Perangkat lunak pemrograman yang penulis digunakan untuk membuat program *e-business* penjualan pulsa elektrik ini adalah Eclipse. Karena aplikasi ini memudahkan dalam merancang program dan mendukung *e-business* penjualan pulsa elektrik.

4.6.3.3 Perangkat Lunak Aplikasi

Perangkat lunak aplikasi yang digunakan untuk membuat program aplikasi *e-business* penjualan pulsa elektrik ini adalah MySQL karena MySQL mudah untuk dikonfirgurasi dengan program yang penulis rancang.

4.7 Koneksi Jaringan *Internet* (*Interconection Network)*

Berdasarkan perancangan *e-business* penjualan pulsa elektrik yang berbasis *client server* dengan menggunakan Android dan MySQL, maka diperlukannya koneksi jaringan *internet*.

4.8 Kelebihan Dan Kelemahan Sistem Yang Diusulkan

4.8.1 Kelebihan Sistem Yang Diusulkan

Program aplikasi *e-business* penjualan pulsa elektrik yang penulis rancang dengan menggunakan *software* Eclipse dengan *database* MySQL yang penulis buat memiliki beberapa kelebihan. Berikut kelebihan dari program yang penulis buat :

A. Program ini memiliki beberapa fitur pengendalian keamanan baik untuk mengakses program dengan cara penggunaan hak akses program dengan *user* yang berbeda.
B. Program aplikasi ini sudah berbasis *client server*, sehingga memudahkan bagi pengguna untuk melakukan transaksi *mobile.*
C. Pada saat transaksi sudah terdapat pengendalian sehingga tidak ada transaksi yang ganda.

D. Program aplikasi yang penulis rancang memiliki tampilan *visual* yang memudahkan *user* untuk melakukan transaksi, dan terdapat fitur konfirmasi untuk meminimalisir kesalahan dalam bertransaksi.

E. Transaksi dan Laporan keuangan dapat dilakukan melalui telepon seluler dimana saja dan kapan saja.

4.8.2 Kelemahan Sistem Yang Diusulkan

Selain kelebihan, program aplikasi *e-business* penjualan pulsa elektrik dengan menggunakan Android dan *database* MySQL ini juga memiliki kelemahan. Berikut ini adalah beberapa kelemahan dari sistem yang penulis usulkan :

A. Program aplikasi ini memerlukan koneksi jaringan *internet* untuk melakukan penyimpanan data transaksi.

B. Program aplikasi ini dalam pelaporan keuangan tidak termasuk biaya sms.