

BAB III

ANALISIS MASALAH DAN RANCANGAN PROGRAM

III.1. Analisis Masalah

Proses analisa sistem merupakan langkah kedua pada pengembangan sistem. Analisa sistem dilakukan untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Sistem sso (*single sign on*) data belum begitu banyak diketahui oleh seorang operator, seorang operator ingin melakukan pengamanan data terhadap dokumennya menggunakan *password* yang ada pada aplikasi atau program yang digunakan. Berkembangnya teknologi informasi secara otomatis akan menambah jumlah data pribadi. Hal ini secara otomatis dapat lupa terhadap pemberian *password* tersebut karena terlalu banyak data yang sudah dibuat.

Untuk itu, sistem yang penulis rancang adalah sistem yang melakukan sekali login tidak dua kali login dengan menggunakan *single sign on*, agar data penting tersebut tidak dapat dibuka sembarangan orang. Dalam tahap pengembangan sistem *single sign on* ini, analisa sistem merupakan hal yang harus dilakukan sebelum proses perancangan sistem.

III.2. Strategi Pemecahan Masalah

Adapun strategi pemecahan masalah dari sistem *single sign on* data yang dirancang adalah sebagai berikut :

1. Untuk melakukan autentikasi *login* agar semua *user* dapat melakukan dengan sekali *login*.
2. Agar semua *user* tidak perlu mendaftarkan ulang untuk menjalankan sistem aplikasi yang dibangun karena semua *user* terfokus pada *active directory*.
3. Pada bagian sisi Admin, Admin dapat melihat aktifitas yang dilakukan oleh *user*.

III.3. Evaluasi Sistem Yang Berjalan

Sistem pengiriman dan penerimaan data terhadap aplikasi yang dibangun saat ini hanya bersifat *single user*. Kelemahan dari sistem ini adalah banyak terbuang waktu dalam melakukan setiap kali masuk kedalam program tersebut dan hanya satu orang yang dapat mengakses aplikasi.. Karena setiap harinya pekerjaan itu semakin banyak dan harus dapat diselesaikan dengan cepat.

Maka solusi yang penulis buat untuk mengatasi masalah tersebut adalah membuat suatu sistem hanya sekali *login* dan terpusat dalam satu *active directory* artinya semua data dan aplikasi terpusat dalam satu *active directory* dalam setiap sekali melakukan akses data.

III.4. Analisa Kebutuhan *Hardware* Dan *Software*

Kebutuhan non fungsional menjabarkan apa-apa saja yang harus dimiliki oleh sistem agar dapat berjalan. Analisis kebutuhan non fungsional bertujuan untuk mengetahui sistem seperti apa yang cocok diterapkan, perangkat keras dan

perangkat lunak apa saja yang dibutuhkan serta siapa saja pengguna yang akan menggunakan sistem ini.

1. Aspek Perangkat Keras

Perangkat keras adalah semua bagian fisik komputer dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

Adapun kebutuhan aspek perangkat keras bertujuan untuk menciptakan aplikasi atau perangkat lunak keamanan data didalam folder tersebut terdiri dari *Prosesor Intel Core I3, harddisk, Memory RAM 2 GB*.

2. Aspek Perangkat Lunak (*software*)

Perangkat lunak adalah program yang dibuat oleh seseorang atau beberapa programmer untuk menciptakan sistem keamanan data didalam *folder* biasanya programmer ini posisinya dibagian IT. Tanpa adanya perangkat lunak ini komponen perangkat keras tidak dapat berfungsi, adapun aplikasi dan *software* yang digunakan dalam pembuatan keamanan data tersebut terdiri dari sistem operasi windows 7, aplikasi xampp.

III.5. Desain Sistem

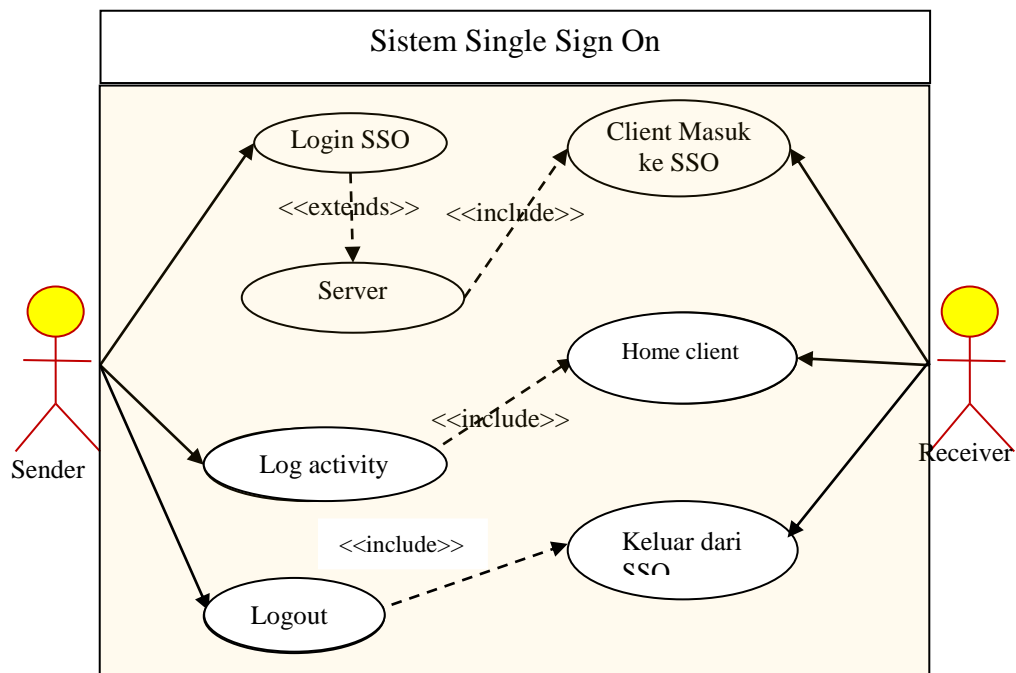
Setelah tahapan analisis sistem, maka selanjutnya dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki

efisiensi kerja suatu sistem yang telah ada. Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

1. Perancangan *Use Case Diagram*
2. Perancangan *Sequence Diagram*
3. Perancangan *Activity Diagram*
4. Perancangan *Output dan Input*

III.5.1. *Use Case Diagram*

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Diagram *Use Case* tersebut dapat dilihat pada gambar III.1.



Gambar III.1. *Use Case Diagram*

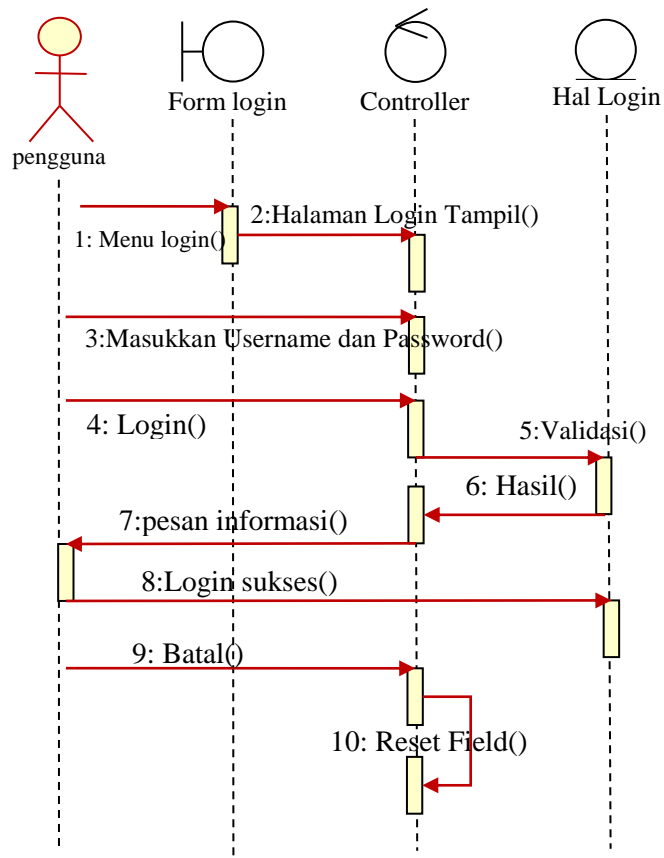
Pada gambar III.1 memberikan informasi tentang perancangan pada aplikasi jaringan single sign on, pertama sekali pengguna sender melakukan *login sso*, jika *username* dan *passwordnya* sesuai maka mengirimkan ke *server* untuk mengecek *protection* yang diciptakan oleh *server* tentang algoritma *encrypt* dan *decrypt*, jika algoritma ini sinkron dengan aplikasi xampp maka pengguna *receiver* dapat mempergunakan aplikasi *client* tetapi disini jika sender melakukan *logout* maka aplikasi *client* tidak dapat mempergunakan program karena *single sign on* diciptakan hanya dapat melakukan sekali *login*.

III.5.2. Sequence Diagram

Sequence diagram menunjukkan bagaimana operasi yang dilakukan secara detail. *Sequence* diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* dalam menjalankan sistem, adapun *sequence* yang dilakukan terdiri dari.

1. Sequence Login

Login digunakan untuk masuk ke *form login* yang berisi nama *username* dan *password*, untuk lebih jelasnya dapat dilihat pada gambar III.2.

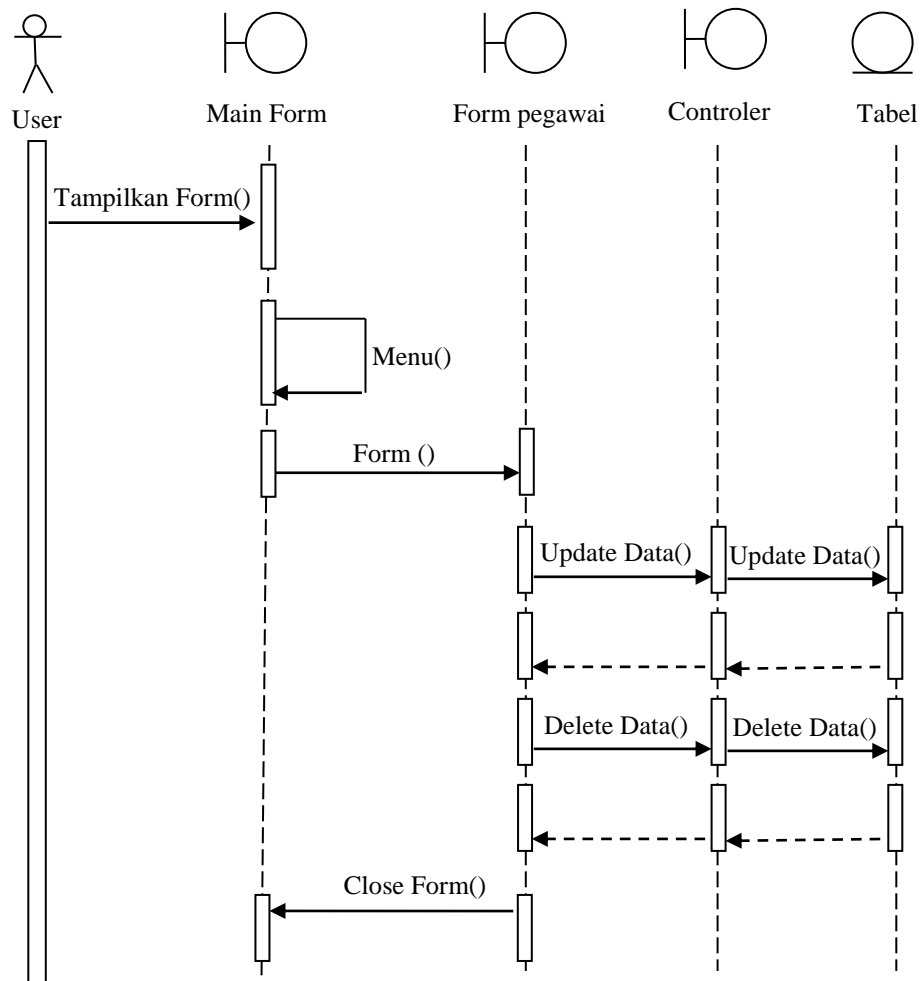


Gambar III.2. Sequence Diagram Login

Dari gambar III.2 menunjukkan bahwa seorang pengguna jika ingin masuk ke halaman utama harus terlebih dahulu memasukkan nama dan kata sandi dengan benar, selanjutnya masuk kedalam menu utama.

2. Sequence Halaman Data

Halaman data digunakan sebagai tempat memasukan data, untuk lebih jelasnya dapat dilihat pada gambar III.3.

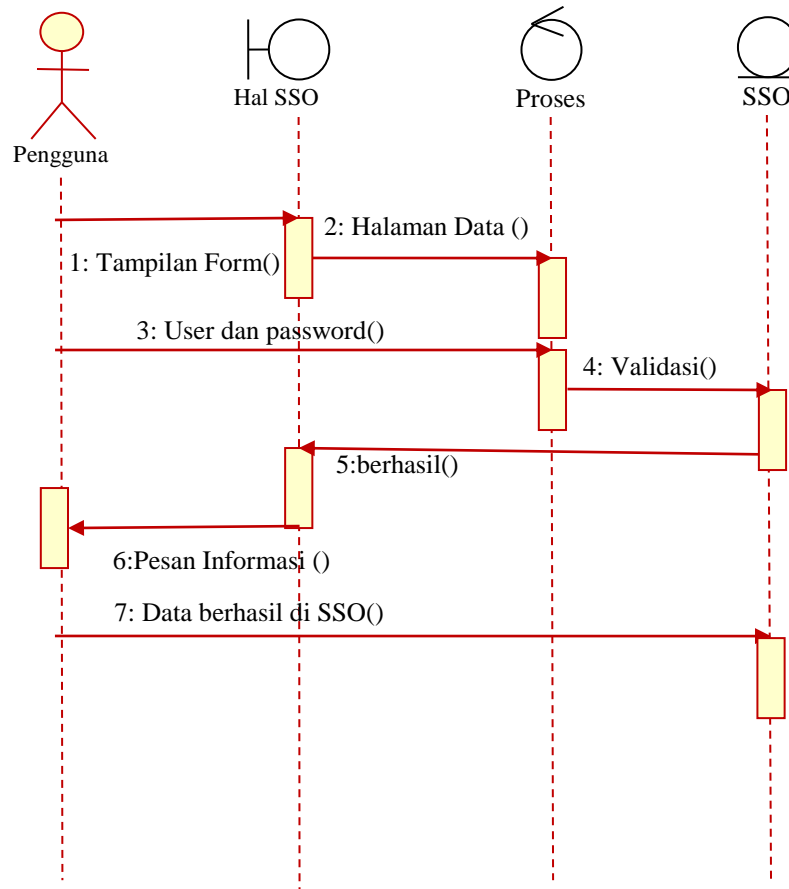


Gambar III.3. Sequence Diagram Halaman Data

Dari gambar III.3 menunjukkan bahwa seorang pengguna dapat melakukan eksekusi dari masing-masing menu yang sudah ditentukan untuk memasuki kehalaman data harus terlebih dahulu masukan kedalam *single sign on*.

3. Sequence SSO (*single sign on*)

SSO digunakan untuk melakukan sekali logi, untuk lebih jelasnya dapat dilihat pada gambar III.4.



Gambar III.4. Sequence SSO (single sign on)

Dari gambar III.4 menunjukkan bahwa seorang *user* jika ingin melakukan sso dalam sebuah jaringan *localhost*.

III.6. Rancangan Program

Rancangan program *single sign on* secara fisik yaitu perancangan bentuk fisik atau bagan arsitektur sistem yang diusulkan. Dalam merancang suatu sistem perlu diketahui hal yang akan menunjang sistem, agar dapat mempermudah pengolahan data nantinya. Pengolahan data ini diharapkan dapat mempermudah

dalam hal penyajian, pelayanan dan pembuatan berbagai laporan data yang dibutuhkan.

1. Rancangan *Form Login* SSO

Dalam perancangan *login* ini digunakan untuk masuk kedalam menu *desktop*, untuk lebih jelasnya dapat dilihat pada gambar III.5.

<i>Form Login</i>	
<input type="text" value="user"/>	
<input type="text" value="login"/>	
<input type="button" value="LOGIN"/>	<input type="button" value="BATAL"/>

Gambar III.5. Rancangan Form Login SSO

Rancangan login yang terdapat pada gambar III.10 terdiri dari dua *textbox*, dan dua *button*. Dari masing-masing *tool* tersebut tugasnya berbeda-beda. Pertama user harus memasukkan data nama, lalu user diperintahkan untuk memasukkan kata sandi setelah kedua input ini sudah diisi dengan benar, *user* di minta untuk melakukan eksekusi dengan cara melakukan klik pada tombol *login*, jika ingin membatalkan eksekusi *user* dapat mengklik tombol batal.

2. Rancangan Form Sistem Server SSO

	SISTEM SERVER SINGLE SIGN ON		
	Log out		
Home	Programer	Log_Activity	
<div style="display: flex; justify-content: space-around; margin-top: 20px;"> Animasi: Besar Animasi: Kecil Animasi: Normal </div>			

Gambar III.6. Rancangan Tampilan Home Server

Setelah *sender* melakukan *login server* maka akan muncul *form* baru yakni *Form Server Single Sign On* seperti pada gambar III.6. Pada gambar diatas terdapat beberapa *button* yakni *Programer Log_Activity* serta *log out*.

Apabila *sender* mengklik tombol *programmer* akan muncul *form* baru seperti gambar III.7. dibawah ini

	SISTEM SERVER SINGLE SIGN ON												
	Log out												
Home	Programer	Log_Activity											
<table border="1" style="margin: 10px auto; width: 80%;"> <tr> <td colspan="2" style="text-align: center;">Programer</td> </tr> <tr> <td colspan="2">Penulis</td> </tr> <tr> <td>Nama</td> <td>Aris Tri Putra</td> </tr> <tr> <td>Nim</td> <td>1110000113</td> </tr> <tr> <td>Jurusan</td> <td>Teknik Informatika</td> </tr> </table>				Programer		Penulis		Nama	Aris Tri Putra	Nim	1110000113	Jurusan	Teknik Informatika
Programer													
Penulis													
Nama	Aris Tri Putra												
Nim	1110000113												
Jurusan	Teknik Informatika												

Gambar III.7. Rancangan Tampilan Programmer

Pada tampilan *form programmer* akan menjelaskan data data seorang *programmer*, setelah itu seorang *sender* dapat melihat aktivitas *client*. Aktivitas clien dapat dilihat seperti pada gambar perancangan III.8. berikut

SISTEM SERVER SINGLE SIGN ON			
Home	Programmer		Log_Activity
			Kembali
No	User	Tanggal	Activity

Gambar III.8. Rancangan Tampilan *Log Activity*

Form Log Activity merupakan sebuah *form* yang berfungsi sebagai laporan kegiatan *client*, data yang ditampilkan adalah *user*, tanggal membuka sistem jaringan *client*, serta *activity* yakni aktivitas yang telah dilakukan.

3. Rancangan *Form Menu Utama Receiver/Jaringan Client*

Dalam perancangan menu utama *receiver* ini digunakan sebagai pusat dari program perancangan keamanan data dari masing-masing menu, untuk lebih jelasnya dapat dilihat pada gambar III.9.


Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON	
Server Jaringan SSO Sedang Login		
Data Buku		
Master	Transaksi	Laporan
<div style="border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;">Icon</div> Buku	<div style="border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;">Icon</div> Penjualan	<div style="border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;">Icon</div> Penjualan
<div style="border: 1px solid black; width: 40px; height: 40px; margin: 0 auto; text-align: center; line-height: 40px;">Icon</div> Kategori		
Potensi Utama, 2016 – Aris # Jaringan SSO		

Gambar III.9. Rancangan Form Menu Utama Receiver

Rancangan menu utama receiver terdiri dari beberapa menu yaitu, menu buku, menu input, menu lihat, menu edit dan about. Menu buku digunakan untuk memasukan data buku dan dapat melihat data buku berdasarkan dari masing-masing judul. Menu transaksi untuk melakukan transaksi tentang penjualan buku yang telah dilakukan dan dilengkapi dengan tanda simbol secara detail dalam menampilkan isi data tentang penjualan buku tersebut.

4. Rancangan *Form Input Data Buku*

Dalam perancangan *form input* data ini berfungsi melakukan memasukan data, untuk lebih jelasnya dapat dilihat pada gambar III.10.

Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON
Server Jaringan SSO Sedang Login	
TAMBAH DATA BUKU	
	
Kode Buku	<input type="text"/>
Judul Buku	<input type="text"/>
Pengarang	<input type="text"/>
Harga	<input type="text"/>
Deskripsi	<input type="text"/>
Kategori	<input type="text" value="▼"/>
Cover	<input type="button" value="Telusuri"/> Tidak ada berkas dipilih
<input type="button" value="simpan"/>	

Gambar III.10. Rancangan *Form Input Data*

Rancangan *form* input yang terdapat pada gambar III.7 terdiri dari beberapa *field*. Setelah semuanya data sudah dimasukkan dengan benar selanjutnya melakukan penyimpanan data tersebut dengan mengklik tombol *simpan* untuk memproses datanya kedalam *database*.

5. Rancangan *Form Input* Katagori

Dalam perancangan *form* kategori ini berfungsi melakukan memasukan data buku berdasarkan kategori, untuk lebih jelasnya dilihat pada gambar III.11.

Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON
Server Jaringan SSO Sedang Login	
TAMBAH DATA KATEGORI	
Kode Kategori	<input type="text"/>
Nama Kategori	<input type="text"/>
<input type="button" value="Simpan"/>	

Gambar III.11. Rancangan *Form Input* Kateegori

Rancangan *form input* kategori yang terdapat pada gambar III.8 terdiri dari *field-field* seperti kode kategori dan nama kategori. Masukan data sesuai dengan *field* tersebut setelah pengguna sudah selesai memasukan data tersebut selanjutnya pengguna diminta untuk melakukan penyimpanan data ke database dengan melakukan klik tombol simpan. Jadi tombol simpan digunakan untuk melakukan proses penyimpanan ke dalam *database* MySQL, dengan catatan kode kategori tidak boleh sama dengan kode yang sudah ada, jika kode kategori sama maka data yang dimasukan tidak dapat disimpan dan itu dianggap kode yang kembar.

6. Rancangan *Form Input Laporan*

Dalam perancangan *form* input laporan ini berfungsi melakukan proses laporan berdasarkan tanggal dalam melakukan pencetakan ke layar maupun ke *printer*, untuk lebih jelasnya dapat dilihat pada gambar III.12.



Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON
Server Jaringan SSO Sedang Login	
Mulai Tanggal	<input type="text" value="dd/mm/yyyy"/>
Sampai Tanggal	<input type="text" value="dd/mm/yyyy"/>
<input type="button" value="Proses"/>	

Gambar III.12. Rancangan *Form Input Laporan*

Pada gambar III.9 memberitahukan dalam melakukan penginputan data tanggal, memasukan data tanggal didahuu dengan tanggal, lalu bulan dan tahun tidak boleh terbalik, jika sudah benar selanjutnya klik tombol proses.

7. Rancangan *Form Output Data Buku*

Form Output menampilkan semua data buku berdasarkan kategori dan dilengkapi dengan jumlah buku berdasarkan kategori tersebut. Bentuk daripada *form* halaman lihat yang dirancang dapat dilihat pada gambar III.13.

Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON					
Server Jaringan SSO Sedang Login						
Data Buku						
Kategori : xxxxxxxxxxxx						 
No	Cover	Kode	Judul Buku	Pengarang	Harga	Aksi
999	xxxxxxx	xxxxxx	xxxxxxxxx	xxxxxxx	9999	del
Jumlah : 999						

Gambar III.13. Rancangan *Form Output Data Buku*

Rancangan *form output* data buku yang terdapat pada gambar III.10 dapat menampilkan gambar covernya, berdasarkan jenis buku disebut dengan kategori dari masing-masing buku tersebut, dan dapat menambahkan data buku dengan mengklik tombol tambah yang bersimbol tanda tambah, dan untuk kembali ke menu utama pengguna dapat mengklik tombol panah kiri.

8. Rancangan *Form Output* Transaksi Penjualan

Form Output transaksi penjualan. Bentuk daripada *form* halaman lihat yang dirancang dapat dilihat pada gambar III.14.

Gambar	<p>SISTEM JARINGAN CLIENT SINGLE SIGN ON</p>																		
<p>Server Jaringan SSO Sedang Login</p> <p>No Faktur : xxxxx Tanggal : dd/mm/yyyy</p> <p style="text-align: center;"><input type="button" value="Kembali"/></p>																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">No</th> <th style="width: 15%;">Kode</th> <th style="width: 25%;">Judul Buku</th> <th style="width: 15%;">Harga</th> <th style="width: 15%;">Jumlah</th> <th style="width: 20%;">Sub Total</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">999</td> <td style="text-align: center;">xxxxxx</td> <td style="text-align: center;">xxxxx</td> <td style="text-align: center;">xxxxxxxxx</td> <td style="text-align: center;">xxxxxxxxx</td> <td style="text-align: center;">9999</td> </tr> <tr> <td colspan="5" style="text-align: right; padding-right: 10px;">Grand Total</td> <td style="text-align: center; border: 1px solid black;">9999</td> </tr> </tbody> </table>		No	Kode	Judul Buku	Harga	Jumlah	Sub Total	999	xxxxxx	xxxxx	xxxxxxxxx	xxxxxxxxx	9999	Grand Total					9999
No	Kode	Judul Buku	Harga	Jumlah	Sub Total														
999	xxxxxx	xxxxx	xxxxxxxxx	xxxxxxxxx	9999														
Grand Total					9999														

Gambar III.14. Rancangan *Form Output* Transaksi Penjualan

Rancangan *form* output transaksi penjualan yang terdapat pada gambar III.11, rancangan ini dilengkapi dengan grand total untuk mengetahui berapa jumlah sub total.

9. Rancangan *Form Output* Transaksi Penjualan Periode

Form Output transaksi penjualan periode . Bentuk daripada *form* halaman lihat yang dirancang dapat dilihat pada gambar III.15.

Gambar	SISTEM JARINGAN CLIENT SINGLE SIGN ON		
Server Jaringan SSO Sedang Login			
Data Transaksi Penjualan Periode : dd/mm/yyyy			<input type="button" value="Print"/>
No	No Faktur	Tanggal	Total
999	xxxxxx	Dd/mm/yyyy	9999
Grand Total :			9999

Gambar III.15. Rancangan *Form Output* Transaksi Penjualan Periode

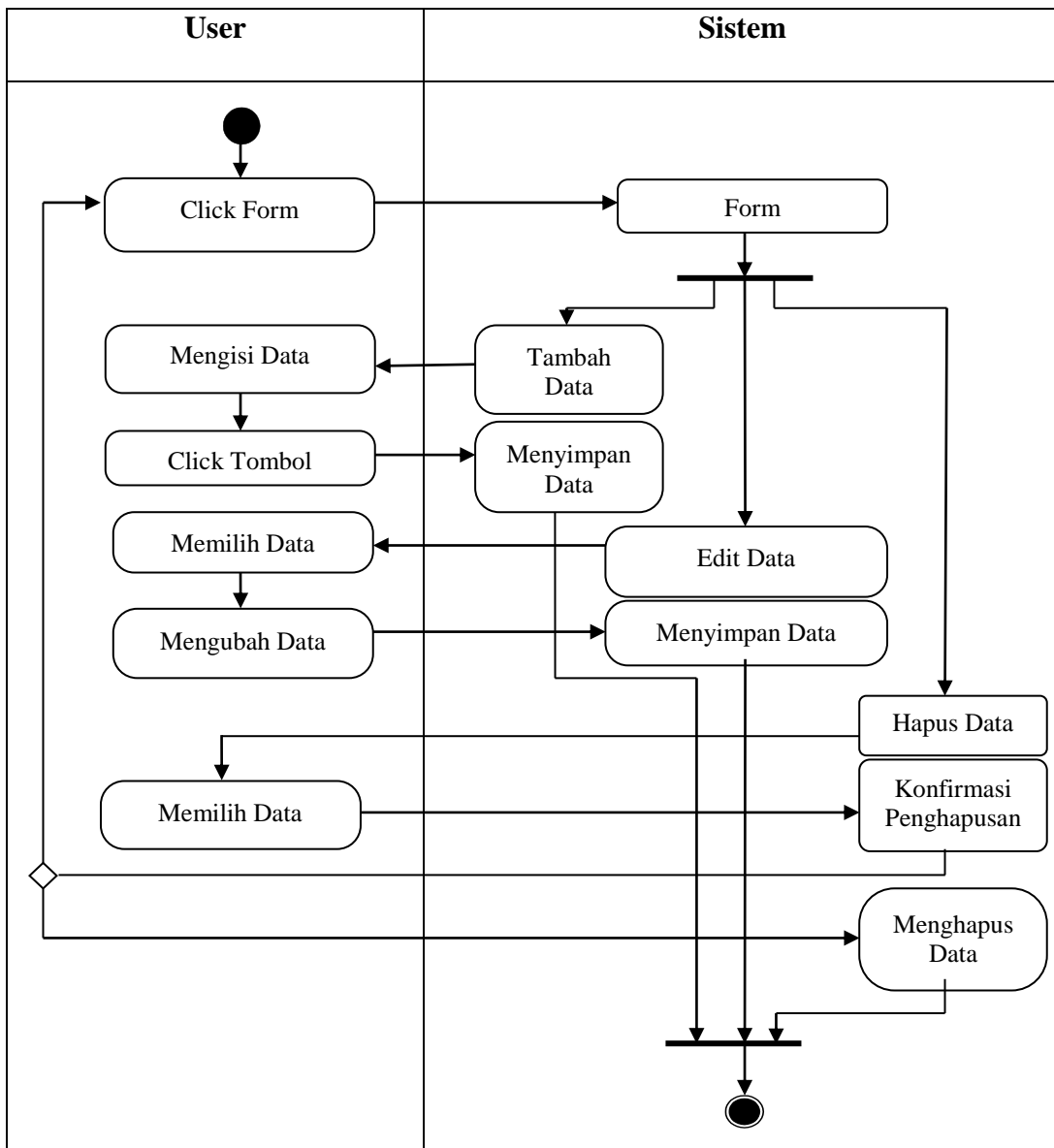
Rancangan *form* output transaksi penjualan periode yang terdapat pada gambar III.15. rancangan ini dilengkapi dengan grand total untuk mengetahui berapa jumlah total. Rancangan ini berdasarkan tanggal mulai sampai dengan tanggal akhir yang ditentukan.

III.7. Logika Program

Logika program dari sistem yang diusulkan akan digambarkan dalam sebuah *activity diagram* sebagai berikut :

1. Activity Diagram

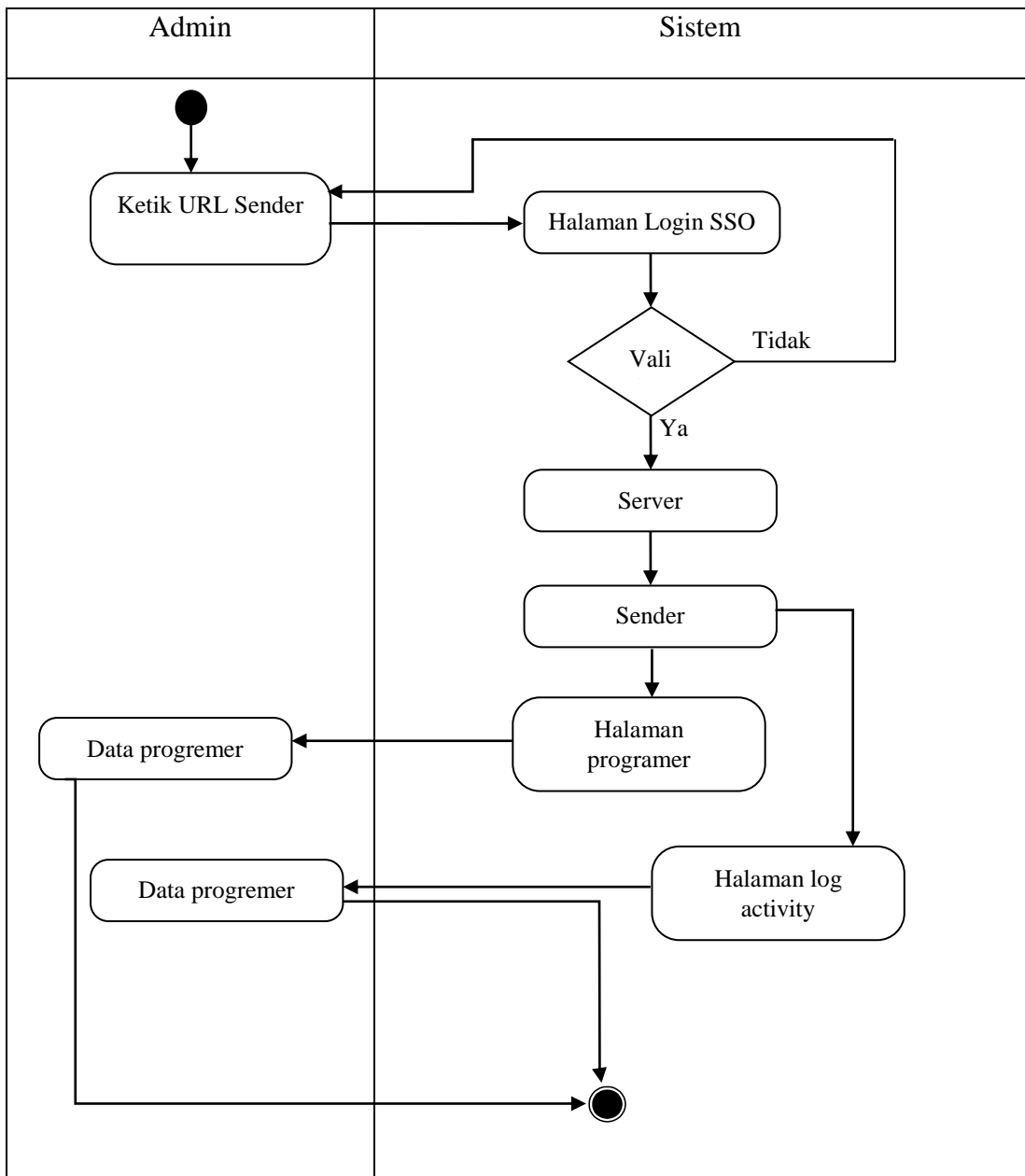
Activity diagram untuk proses pengolahan data. Activity diagram enkripsi data dapat dilihat pada gambar III.16.



Gambar III.16. Activity Diagram

2. Activity Diagram Analisa SSO

Aktivitas yang dilakukan dalam akan melakukan analisa login, kemudian akan dihadapkan ke *menu home* sisi *user* yang ditunjukkan pada gambar III.17.



Gambar III.17. Activity Analisa SSO