BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pembuatan sistem informasi *monitoring* antiran menggunakan konsep *Systems Development Life Cycle* (SDLC) yang berfungsi untuk menggambarkan tahapan di dalam proses pembuatan Rancang Bangun Sistem Informasi *Monitoring* Antrian pada Koperasi Setia Bhakti Wanita Berbasis Web, berikut adalah tahapan-tahapannya :

3.1 Analisis Sistem

Untuk pengumpulan *data-data* yang diperlukan dalam melaksanakan tugas akhir di bagian Simpan Pinjam khususnya di proses antrian pada Koperasi Setia Bhakti Wanita, ada beberapa langkah yang telah dilakukan, antara lain:

a. Wawancara/*Interview*

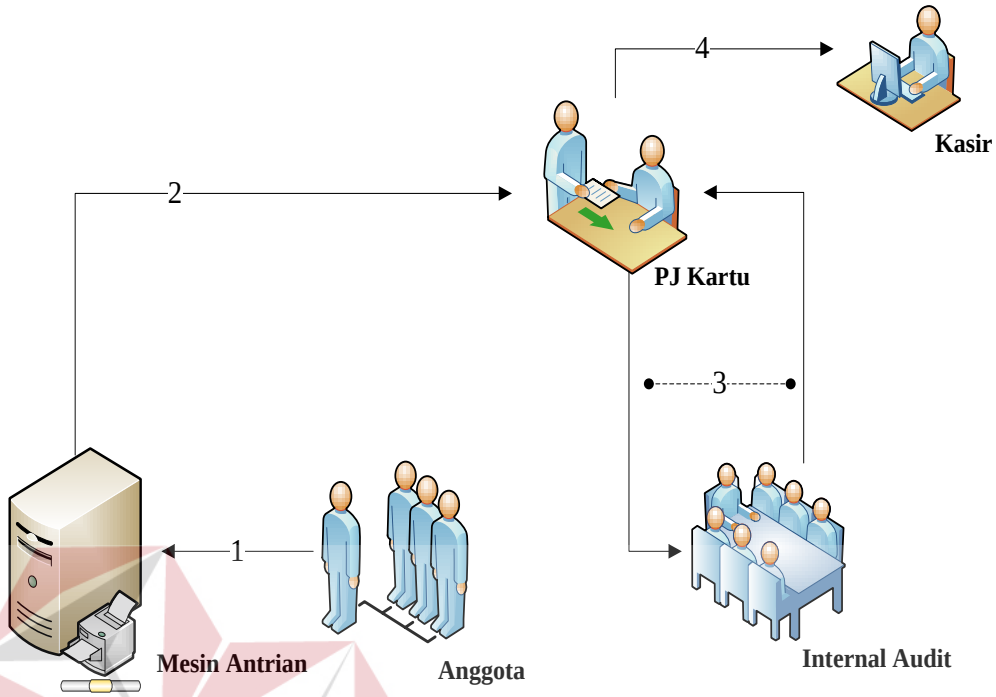
Langkah ini dilakukan untuk mengetahui permasalahan-permasalahan yang terjadi di bagian Simpan Pinjam Koperasi Setia Bhakti Wanita yang berkaitan dengan sistem antrian. Diperoleh pula kebutuhan-kebutuhan sistem yang diinginkan oleh pihak perusahaan dalam rangka pembuatan aplikasi nantinya. Narasumber untuk pengerjaan tugas akhir adalah koordinator dibagian EDP (*Entry Data Processing*). Hasil dari wawancara ini, mengetahui awal dari *point-point* permasalahan yang ada di bagian Simpan Pinjam Koperasi Setia Bhakti Wanita dengan cara mengetahui alur kerja bagian terlebih dahulu (dapat dilihat pada lampiran I) hasilnya adalah simpan pinjam mempunyai proses yang harus dipantau untuk mendapatkan laporan *monitoring* yang dibutuhkan oleh *manager* HRD karena tidak semua alur kerja

berhubungan dengan sistem antrian. Setelah mengetahui alur pada bagian simpan pinjam maka langkah selanjutnya adalah pengamatan sistem antrian.

b. Pengamatan/*Observasi*

Langkah ini dilakukan untuk melihat kondisi di bidang simpan pinjam terutama sistem antrian. *Observasi* yang dilakukan meliputi operasi mesin antrian dan sifat antrian, pada gambar 3.1 merupakan *workflow* simpanan berdasarkan anggota yang mengantri :

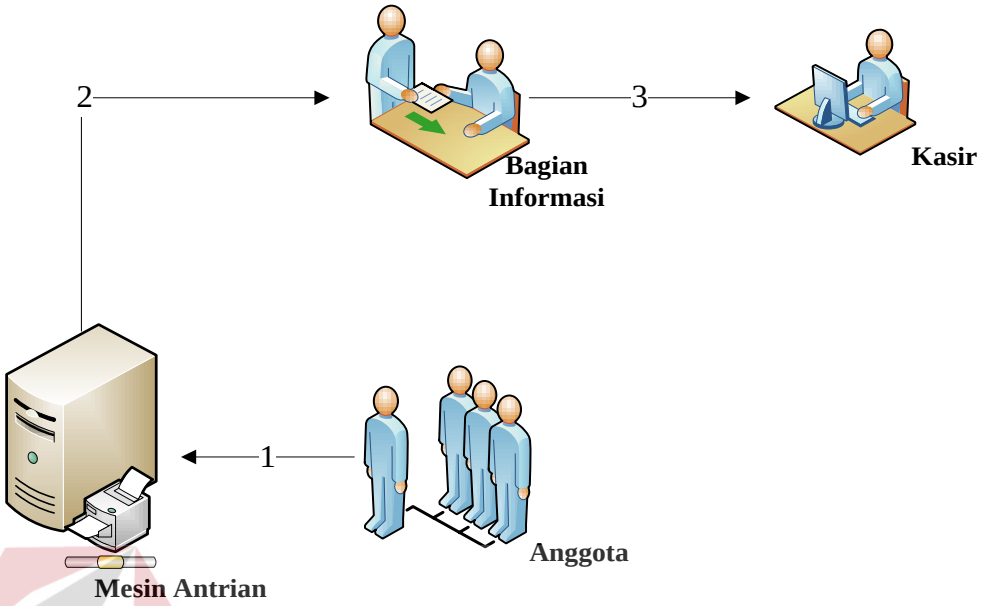
1. Langkah pertama anggota yang akan bertransaksi mengambil *struck* antrian yang berisikan informasi nomor antriannya, jumlah antrian sementara, jenis transaksi.
2. Langkah kedua anggota akan dipanggil berdasarkan nomor antriannya dan langsung menuju ke loket PJ kartu untuk melakukan angsuran ataupun simpanan, anggota menyerahkan rekapan *data* angsuran / simpanan yang nantinya akan diinput petugas loket PJ kartu.
3. Langkah ketiga rekapan dan *input*-an yang terjadi di loket PJ kartu akan diperiksa kembali oleh tim *Internal* A*udit*, sementara proses pemeriksaan berlangsung antrian di loket PJ kartu akan dilanjutkan begitu seterusnya sampai tim *internal audit* selesai memeriksa. Tim *internal audit* akan memberikan kitir setoran ke PJ kartu, setelah itu petugas PJ kartu menyerahkan ke anggota yang mempunyai *data* tersebut.
4. Langkah terakhir anggota yang sudah mendapatkan kitir setoran akan dipanggil ke loket kasir untuk menyerahkan kitir setoran dan petugas kasir akan memberikan bukti kas masuk.



Gambar 3.1 *Workflow* Simpanan

Pada gambar 3.2 merupakan *workflow* peminjaman berdasarkan anggota yang mengantri :

1. Langkah pertama anggota yang akan bertransaksi mengambil *struck* antrian yang berisikan informasi nomor antriannya, jumlah antrian sementara, jenis transaksi.
2. Langkah kedua anggota akan menuju loket bagian informasi yang menunjukan nomor antrian yang dipunyai. Petugas bagian informasi akan memberikan SPH yang berguna untuk mencairkan pinjaman di loket kasir
3. Langkah terakhir anggota akan menuju ke loket kasir, di loket kasir anggota menyerahkan SPH kepada petugas setelah itu mendapatkan dana yang telah tertera di SPH tersebut.



Gambar 3.2 *Workflow* Pinjaman

Dari hasil *observasi*, diketahui terdapat mesin antrian yang merupakan *server* pada sistem antrian, mesin antrian tersebut menghasilkan data antrian berupa nomor antrian, jenis transaksi dan waktu transaksi. Data antrian yang dihasilkan akan diberikan pada setiap loket transaksi, lalu akan diproses pada aplikasi *client* disetiap loket transaksi. Proses tersebut meliputi pemanggilan data antrian dan penyaluran data antrian ke transaksi selanjutnya.

Terdapat 4 menu pada mesin antrian yaitu PJ kartu atas, PJ kartu bawah, informasi atas, dan informasi bawah. Layanan simpan pinjam terbagi di dua lantai dan total loket transaksi ada 13 yang tersebar di dua lantai, lantai pertama terdapat 7 loket transaksi (bagian informasi 1, PJ kartu 4, dan kasir 2) dan lantai kedua terdapat 6 loket transaksi (bagian informasi 1, PJ kartu 3 dan kasir 2). Sistem antrian belum mempunyai *history* laporan antrian yang terjadi setiap hari, sehingga *manager* HRD

kesulitan dalam melakukan *monitoring* antrian. Kebutuhan *manager* HRD untuk melakukan *monitoring* antrian adalah informasi tentang petugas, layanan dan total antrian.

3.1.1 Hasil Analisis

Dari permasalahan tersebut akan dibuat sistem informasi *monitoring* yang digunakan untuk membantu *manager* HRD dalam meningkatkan layanan. *Input* data yang akan diolah didapatkan dari transfer data antrian di setiap loket transaksi. Proses *monitoring* yang akan diterapkan meliputi penangkapan data antrian, lalu akan disimpan ke *database* setelah itu akan diolah oleh sistem untuk menghasilkan *Output* dari Sistem Informasi *Monitoring* Antrian yaitu laporan *real time* antrian yang terjadi, laporan total antrian per periode, laporan petugas loket (jumlah antrian per loket), laporan pelayanan (total layanan, total waktu layanan).

Laporan yang dihasilkan dapat digunakan untuk melakukan peningkatan layanan, antara lain memberikan gambaran *manager* HRD untuk kinerja petugas pada loket transaksi, *manager* HRD melihat kinerja petugas loket dari waktu pelayanan serta total antrian tiap harinya. Dan juga memberikan gambaran pada *manager* HRD dalam setiap bulan antrian yang menumpuk agar bisa direspon dengan baik.

3.1.2 Analisis Kebutuhan Sistem

Aplikasi sistem informasi *monitoring* antrian yang akan dibangun membutuhkan data antrian sebagai data utama yang digunakan untuk menghasilkan informasi ke *manager* HRD. Data antrian sebagai acuan untuk mencatat berbagai aktifitas *server* antrian yang ada di Koperasi Setia Bhakti Wanita. Dapat dilhat pada *table* 3.1 yang merupakan kebutuhan dari *manager* HRD.

Tabel 3.1 Analisa Kebutuhan Sistem

No	Pengguna	Kebutuhan	Laporan Yang dihasilkan
1	*Manager* HRD	– Informasi antrian yang terjadi – Informasi total antrian per periode – Informasi petugas loket (jumlah antrian per loket) – Informasi pelayanan (total layanan, total waktu layanan)	– Laporan *real time* antrian yang terjadi – Laporan total antrian per periode – Laporan petugas loket (jumlah antrian per loket) – Laporan pelayanan (total layanan, total waktu layanan)

Aplikasi sistem informasi *monitoring* antrian ini akan membantu *manager* HRD dalam memonitor antrian untuk meningkatkan layanan. Pada Tabel 3.2 merupakan kebutuhan *functional* dan *non functional* dari sistem.

Tabel 3.2 *Functional* dan *Non-functional* Sistem Informasi *Monitoring* Antrian

Functional	Non Functional
– Sistem dapat menangkap data antrian pada *server*. – Sistem menyediakan halaman yang memungkinkan penggunanya mendapat laporan *monitoring* antrian.	– Sistem dapat memilah data antrian dari *server*

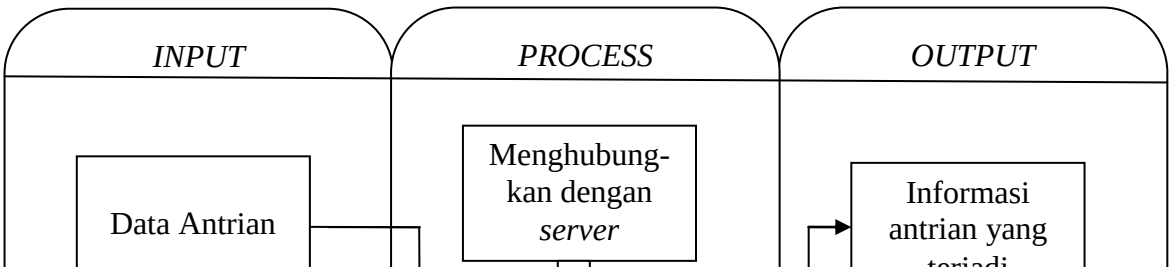
3.2 Perancangan Sistem

Perencanaan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Rancang Bangun Sistem Informasi *Monitoring* Antrian merupakan media

untuk membantu *manager* HRD dalam mengambil keputusan berdasarkan laporan yang akan dihasilkan oleh sistem.

Gambar 3.3 merupakan gambaran dari Sistem Informasi *Monitoring* Antrian yang berawal dari anggota yang akan melakukan transaksi simpan pinjam mengambil nomor antrian pada mesin antrian, lalu data dari sistem antrian akan ditangkap dan dipilah sesuai dengan informasi yang akan dibutuhkan setelah itu disimpan pada *database*. Data akan diolah oleh sistem dan menghasilkan yaitu laporan *real time* antrian yang terjadi, laporan total antrian per periode, laporan petugas loket (jumlah antrian per loket), laporan pelayanan (total layanan, total waktu layanan). Laporan yang dihasilkan oleh sistem dapat digunakan untuk melakukan peningkatan layanan, antara lain memberikan gambaran *manager* HRD untuk kinerja petugas pada loket transaksi. *Manager* HRD melihat kinerja petugas loket dari waktu pelayanan serta total antrian tiap harinya, dan juga memberikan gambaran pada *manager* HRD dalam setiap bulan antrian yang menumpuk agar bisa direspon dengan baik.

Pada Gambar 3.3 terdapat proses *monitoring* yang terdiri atas lima proses utama yaitu menghubungkan dengan *server*, menangkap data antrian, memilah data, menyimpan data, membuat laporan antrian. Titik pengamatan dalam proses *monitoring* terdapat di *server* antrian, penetapan titik pengamatan penting dalam me-*monitoring* antrian di Koperasi Setia Bhakti Wanita. Pada *server* antrian terdapat kegiatan yang berlangsung di sistem antrian maka titik pengamatan dilakukan di *server*. Proses *monitoring* menggunakan konsep *socket client-server*, karena konsep *socket client-server* tidak akan merusak sistem antrian yang sudah terbuat.

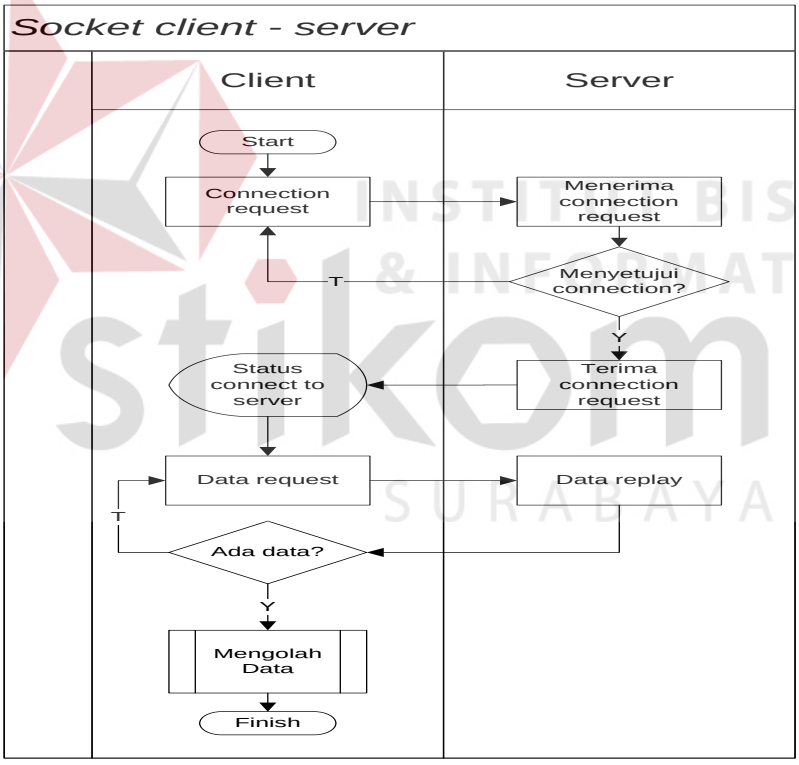


INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Gambar 3.3 Blok Diagram Sistem Informasi *Monitoring* Antrian

Pada Gambar 3.4 merupakan langkah-langkah dari *socket client-server*. Pertama-tama *client* harus terhubung dengan *server*, beranjak dari langkah pertama *client* akan melakukan proses *connection request* ke *server* untuk mengawali terhubungnya *client* dan *server*. *Server* akan menerima *connection request* dari *client* tetapi akan dipertimbangkan *connection* tersebut akan disetujui atau tidak. Apabila tidak disetujui *server* akan mengirimkan penolakan *connection* dan *client* akan mengulang proses *connection request*, apabila *server* menerima *connection request* dari *client* maka akan mengirimkan status *connect to server* untuk *client*. Setelah

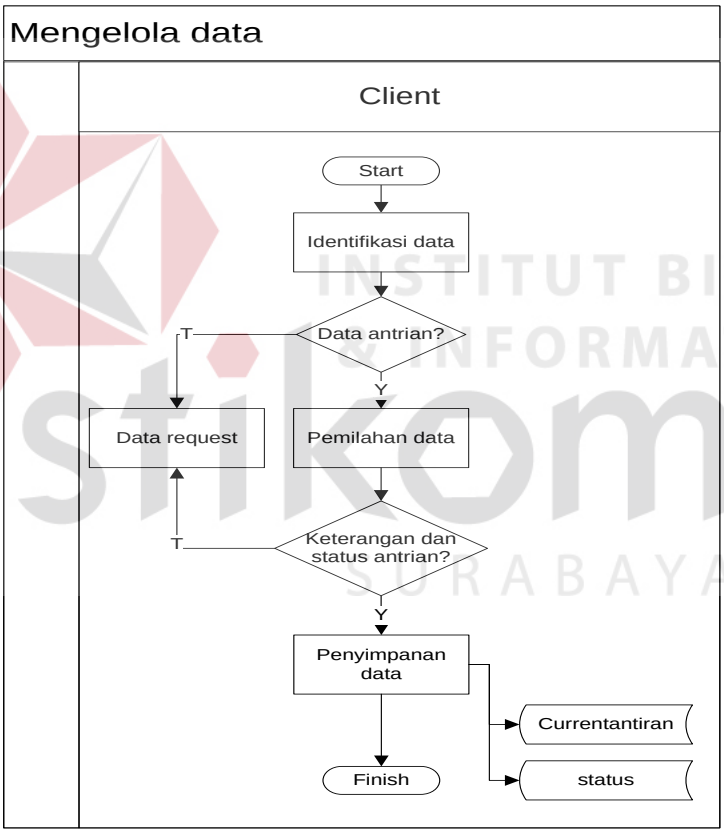
mendapatkan *connection* maka *client* akan melakukan proses *data request* untuk mendapatkan data yang terdapat pada *server*, *data request* akan direspon oleh *server* dengan proses *data replay* yang berguna memberikan data yang ada di *server*. Apabila tidak terdapat data pada saat *server* melakukan proses *data replay* maka *client* akan mengulang proses *data request*, apabila terdapat data dari proses *data replay* yang dilakukan *server* maka data tersebut akan diolah oleh *client*. Proses mengolah data dapat dilihat pada Gambar 3.5 jika data yang diolah bukan merupakan data antrian maka akan kembali pada proses *data request*.



Gambar 3.4 Penerapan *socket client-server* pada proses *monitoring*

Pada Gambar 3.5 adalah cara mengolah data yang merupakan kelanjutan dari Gambar 3.4. Langkah pertama data yang diterima dari proses sebelumnya akan diidentifikasi, apakah itu merupakan data antrian atau tidak. Apabila itu bukan

merupakan data antrian maka akan kembali ke proses data request, apabila merupakan data antrian akan dilakukan proses pemilahan data. Data antrian dipilah berdasarkan keterangan dan status antrian, apabila data antrian tidak mengandung kedua unsur tersebut maka akan ke proses data request kembali untuk meminta data yang tepat sebaliknya apabila data antrian mengandung kedua unsur tersebut maka data akan disimpan ke table yang telah disediakan. Table currentantrian untuk menyimpan keterangan antrian dan table status untuk menyimpan status antrian.

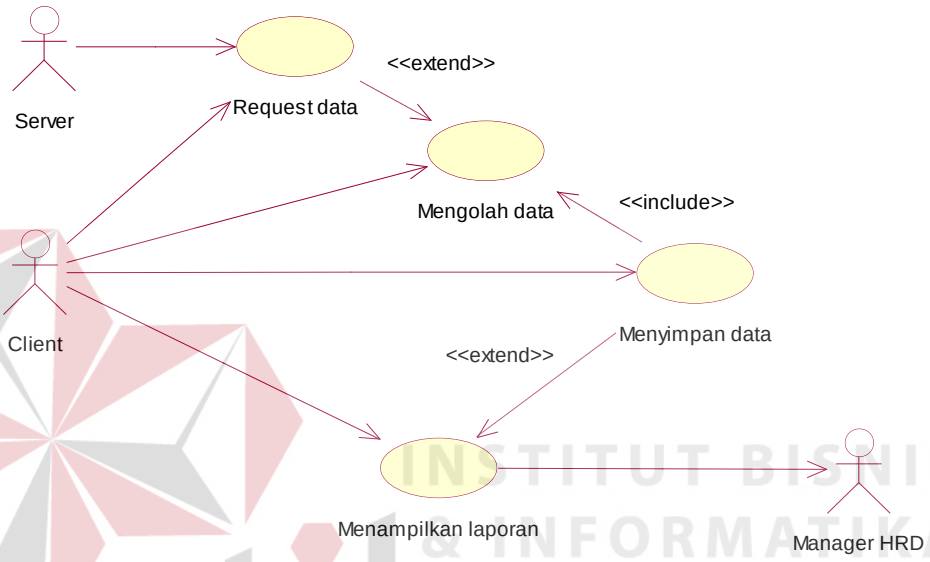


Gambar 3.5 Alur proses mengolah data

3.2.1 *Use Case Monitoring* Antrian

Pada Gambar 3.6 terdapat *use case monitoring* antrian. Pada *use case monitoring* antrian terdapat 3 aktor bisnis, aktor bisnis atau *business actor* adalah

seseorang atau sesuatu yang ada di luar organisasi dan berinteraksi dengan organisasi yang terlibat dalam kegiatan bisnis organisasi sedangkan pekerja bisnis atau *business worker* adalah suatu peranan di dalam organisasi, bukan posisi. Seseorang boleh memainkan banyak peran tetapi memengang hanya satu posisi (Sholiq, 2006).



Gambar 3.6 *Use Case Monitoring* Antrian

Pada Gambar 3.6 terdapat tiga aktor, yaitu *server* sebagai sumber data dari sistem antrian yang akan dicatat aktivitasnya oleh *client* untuk dijadikan *history* serta laporan *monitoring*. Laporan-laporan *monitoring* antrian ditujukan kepada *manager* HRD yang nantinya akan mengambil keputusan.

3.2.2 *Activity Diagram* dan *Flow of Event*

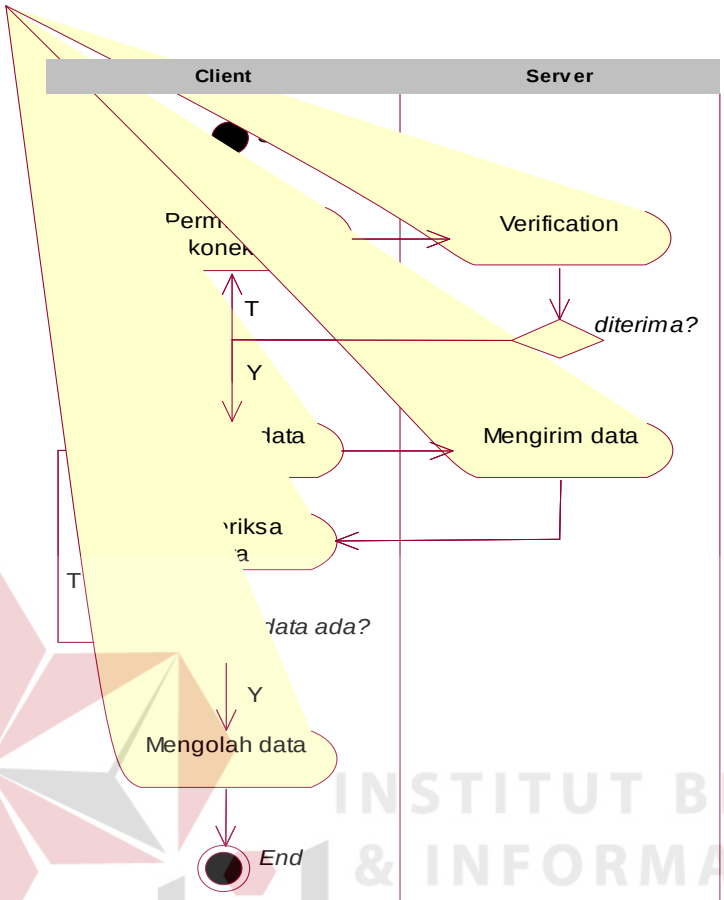
A. Proses *Request* Data

Activity diagram adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *use case* dalam bentuk grafik. Diagram ini menunjukan langkah-langkah di dalam aliran kerja, titik-titik keputusan di dalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan obyek-obyek yang digunakan dalam aliran kerja (Sholiq, 2006).

Gambar 3.7 menceritakan tentang, bagaimana *client server* berkomunikasi. Pada awalnya proses koneksi sama seperti langkah *socket client-server*, yaitu *client* akan mengirimkan koneksi yang akan diterima *server* yang artinya sudah terjadi hubungan atau ada komunikasi antara keduanya. Selanjutnya *client* akan mengecek data antrian yang sudah di data oleh *server* dengan cara melihat status antrian dan *client* akan menerima status antrian masuk jika tidak masuk maka kembali lagi pada proses mengecek data antrian, jika status antrian masuk akan menuju ke proses selanjutnya yaitu proses menangkap data.

Berikut ini disajikan *flow of events* dari proses *request data*. *Flow of events* bertujuan untuk mendokumentasikan alur logika dalam *use case* yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan. *Flow of events request data* disajikan pada Tabel 3.3 dengan kondisi akhir yang diinginkan dan kondisi akhir gagal, serta alur alternatif untuk menangani kondisi salah.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA



Gambar 3.7 *Activity Diagram Request Data*

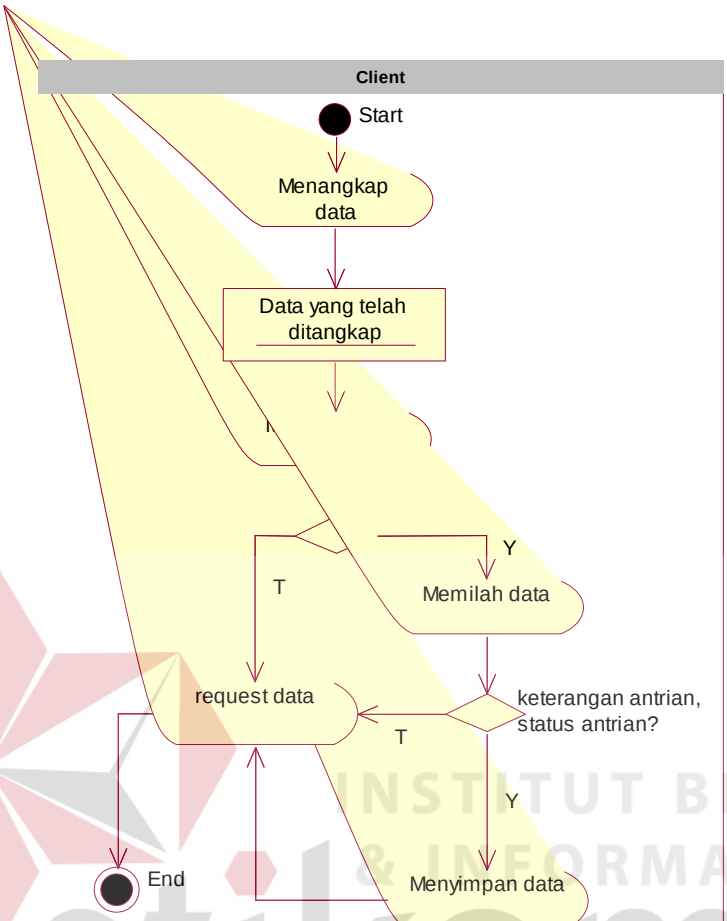
Tabel 3.3 *Flow of Event request* data

Nama Proses	***Request* data**
Kebutuhan terkait	Salah satu proses untuk memulai *monitoring* antiran yang berada di koperasi Setia Bhakti Wanita. Proses ini berhubungan langsung dengan *server*, dan disetiap kegiatan *server* akan terpantau.
Tujuan	Meminta data.
Prasyarat	Sudah terkoneksi dengan *server*.
Kondisi akhir sukses	Terdapat data atau kegiatan pada *server* antrian yang akan dilanjutkan ada proses mengolah data.
Kondisi akhir gagal	Jika tidak ada data atau dalam *server* tidak ada kegiatan, maka *client* akan meminta terus-menerus sampai terdapat data.
Aktor utama	*Server, client*

Nama Proses	***Request* data**	
Aktor sekunder	Tidak ada	
Pemicu	Terdapat kegiatan di *server* antrian	
Alur utama	**Langkah**	**Aksi**
	1.	Melakukan koneksi ke *server* antrian.
	2.	*Client* meminta data kepada *server*.
	3.	*Server* akan merespon dengan mengirimkan data atau kegiatan yang terjadi.
	4.	Melakukan validasi data.
	5.	Melakukan proses mengolah data.
Alur Perluasan	**Langkah**	**Aksi Percabangan**
		Tidak ada

B. Proses Mengolah Data

Pada Gambar 3.8 menceritakan alur proses mengolah data. Proses dimulai ketika *server* sudah mengirimkan data yang terdapat isinya atau adanya kegiatan pada *server*, pengiriman data berlangsung pada proses *request* data. Selanjutnya data akan diperiksa, karena hasil dari proses *request* data tidak seluruhnya merupakan data antrian terdapat data login petugas. Jika data bukan merupakan data antrian maka proses selanjutnya akan kembali pada proses *request* data, jika benar data antrian akan dipilah sesuai dengan yang dibutuhkan, yaitu keterangan antrian dan status antrian. Setelah mendapatkan data yang dibutuhkan *client* akan menyimpan data-data tersebut, meskipun proses terakhir adalah menyimpan data akan tetapi akan kembali ke proses *request* data dan begitu seterusnya.



Gambar 3.8 *Activity Diagram* Menangkap Data

Berikut ini disajikan *flow of events* dari menangkap data. menangkap data dilakukan setelah data tertangkap dan selanjutnya akan difilter dan diidentifikasi oleh proses ini. *Flow of events* dari proses menangkap data disajikan dalam Tabel 3.4.

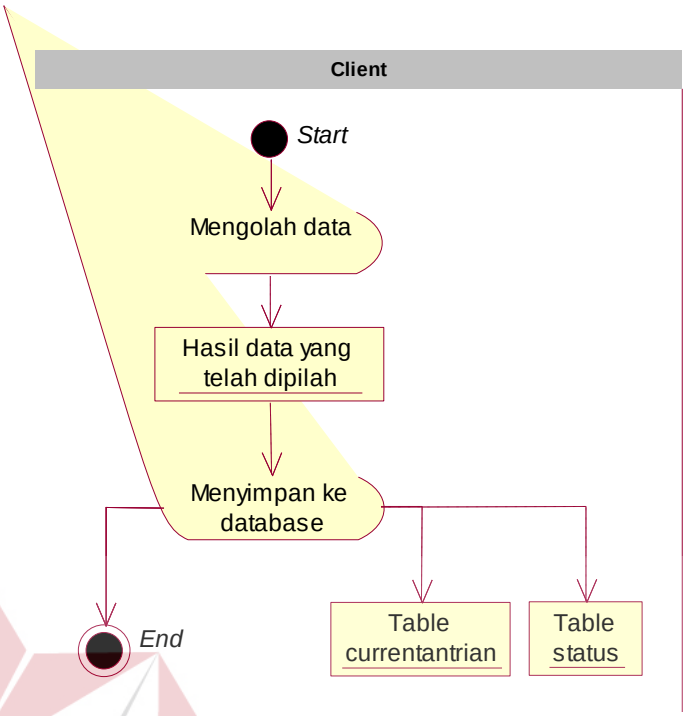
Tabel 3.4 *Flow of Events* Menangkap Data

Nama *Use Case*	**Mencatat Mutasi**
Kebutuhan terkait	Data yang dihasilkan oleh proses sebelum yaitu proses *request* data, data tersebut akan diolah pada proses ini.
Tujuan	Mendapatkan data yang dibutuhkan dengan cara memeriksa dan memilah data.
Prasyarat	Data antrian.
Kondisi akhir sukses	Hasil pemeriksaan menentukan data adalah data antrian dan akan dipilah sesuai dengan kebutuhan.

Nama *Use Case*	**Mencatat Mutasi**	
Kondisi akhir gagal	Jika bukan merupakan data antrian akan mengulang pada proses *request* data	
Aktor utama	*Client*	
Aktor sekunder	Tidak ada	
Pemicu	Proses *request* data menghasilkan data yang mempunyai isi atau terdapat kegiatan pada *server* antrian.	
Alur utama	**Langkah**	**Aksi**
	1.	Memeriksa data.
	2.	Melakukan validasi data.
	3.	Data yang merupakan data antrian akan dipilah sesuai dengan kebutuhan.
	4.	Jika data bukan merupakan data antrian akan mengulang ke proses *request* data.
	5.	Melakukan validasi data antrian.
	6.	Melakukan proses penyimpan data, data yang tersimpan merupakan data yang sudah dipilah-pilah.
	7.	Jika data antrian bukan merupakan data yang dibutuhkan maka akan mengulang ke proses *request* data.
Alur Perluasan	**Langkah**	**Aksi Percabangan**
		Tidak ada

C. Proses Penyimpanan Data

Pada Gambar 3.9 merupakan diagram tentang *client* menyimpan data antrian yang telah dipilah. Lalu menghasilkan *table* currentantrian dan *table* status yang akan digunakan pada proses pembuatan laporan antrian, data yang tersimpan di *table* currentantrian adalah waktu mengantri, nama tempat transaksi, dan jumlah antrian. Sedangkan pada *table* status terdapat waktu transaksi, nomor tempat transaksi, dan nomor antrian.



Gambar 3.9 *Activity Diagram* Menyimpan Data

Berikut ini *flow of events* dari proses menyimpan data. Menyimpan data dilakukan setelah proses menangkap data menghasilkan data yang telah dipilah dan data itu akan disimpan pada *table* antrian dan *table* statu. *Flow of events* dari proses menyimpan data dapat dilihat pada Tabel 3.5.

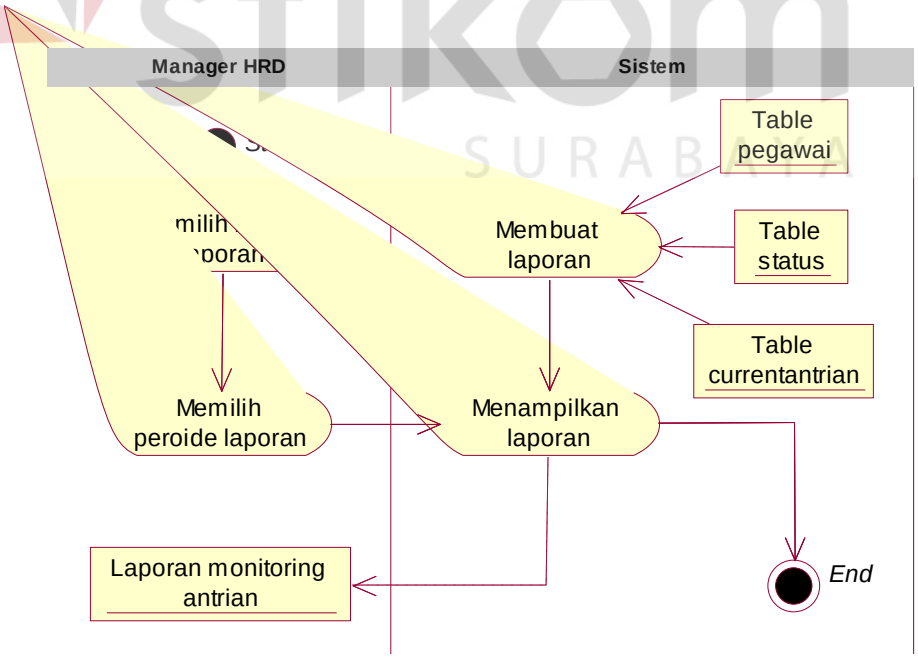
Tabel 3.5 *Flow of Events* Menyimpan Data

Nama Proses	**Menyimpan Data**
Kebutuhan terkait	Pada sisi *client* akan terus menerus menyimpan kegiatan yang terdapat di *server* dan tentu sudah dipilah datanya. Data-data itu akan dijadikan *history* untuk antrian Koperasi Setia Bhakti Wanita.
Tujuan	Menghasilkan *table* antrian dan *table* status untuk membuat laporan *monitoring* antrian.
Prasyarat	Data yang sudah dipilah.
Kondisi akhir sukses	Jika data antrian sudah dipilah maka data dapat disimpan pada *table* antrian dan *table* status.
Kondisi akhir gagal	Jika data belum dipilah maka data tidak dapat disimpan.

Nama Proses	**Menyimpan Data**	
Aktor utama	*Client*	
Aktor sekunder	Tidak ada	
Pemicu	Proses menangkap data berjalan dengan lancar.	
Alur utama	**Langkah**	**Aksi**
	1.	Melakukan proses menangkap data
	2.	Hasil dari proses tersebut merupakan masukan yang akan disimpan ke *table* antrian dan *table* status
	3.	Proses penyimpanan data
	4.	Masuknya data pada *table* currentantrian dan *table* status
Alur Perluasan	**Langkah**	**Aksi Percabangan**
		Tidak ada

D. Proses Menampilkan Laporan

Pada Gambar 3.10 merupakan diagram menampilkan laporan. Laporan yang dihasilkan telah disesuaikan dengan kebutuhan *manager* HRD, yaitu laporan antrian, laporan transaksi, laporan petugas, dan laporan pelayanan. Data dari semua laporan berasal dari *table* antrian, *table* status dan *table* petugas.



Gambar 3.10 *Activity Diagram* Menampilkan Laporan

Berikut ini *flow of events* dari proses menampilkan laporan. Proses ini berjalan ketika *manager* HRD ingin melihat laporan *monitoring* antiran. *Flow of events* dari proses menampilkan laporan dapat dilihat pada Tabel 3.6.

Tabel 3.6 *Flow of Events* Menyimpan Data

Nama Proses	**Menampilkan Laporan**	
Kebutuhan terkait	Laporan *monitoring* antrian yang disajikan kepada *manager* HRD mendapat masukan dari *table* antrian, *table* status dan *table* petugas.	
Tujuan	Memberikan informasi antrian di bagian Simpan Pinjam koperasi Setia Bhakti Wanita.	
Prasyarat	*Table* antrian dan *table* status sudah terisi dengan data yang dibutuhkan.	
Kondisi akhir sukses	Menampilkan laporan *monitoring* antrian kepada *manager* HRD	
Kondisi akhir gagal	Jika tidak ada data pada *table* antrian dan *table* status	
Aktor utama	*Manager* HRD	
Aktor sekunder	Sistem	
Pemicu	*Manager* HRD memilih menu laporan	
Alur utama	**Langkah**	**Aksi**
	1.	*Manager* HRD memilih menu laporan antrian serta memilih periode yang diinginkan.
	2.	Pembuatan laporan yang mengambil data dari *table* currentantrian, *table* status dan *table* petugas.
	3.	Sistem menampilkan laporan yang diinginkan *manager* HRD.
Alur Perluasan	**Langkah**	**Aksi Percabangan**
		Tidak ada

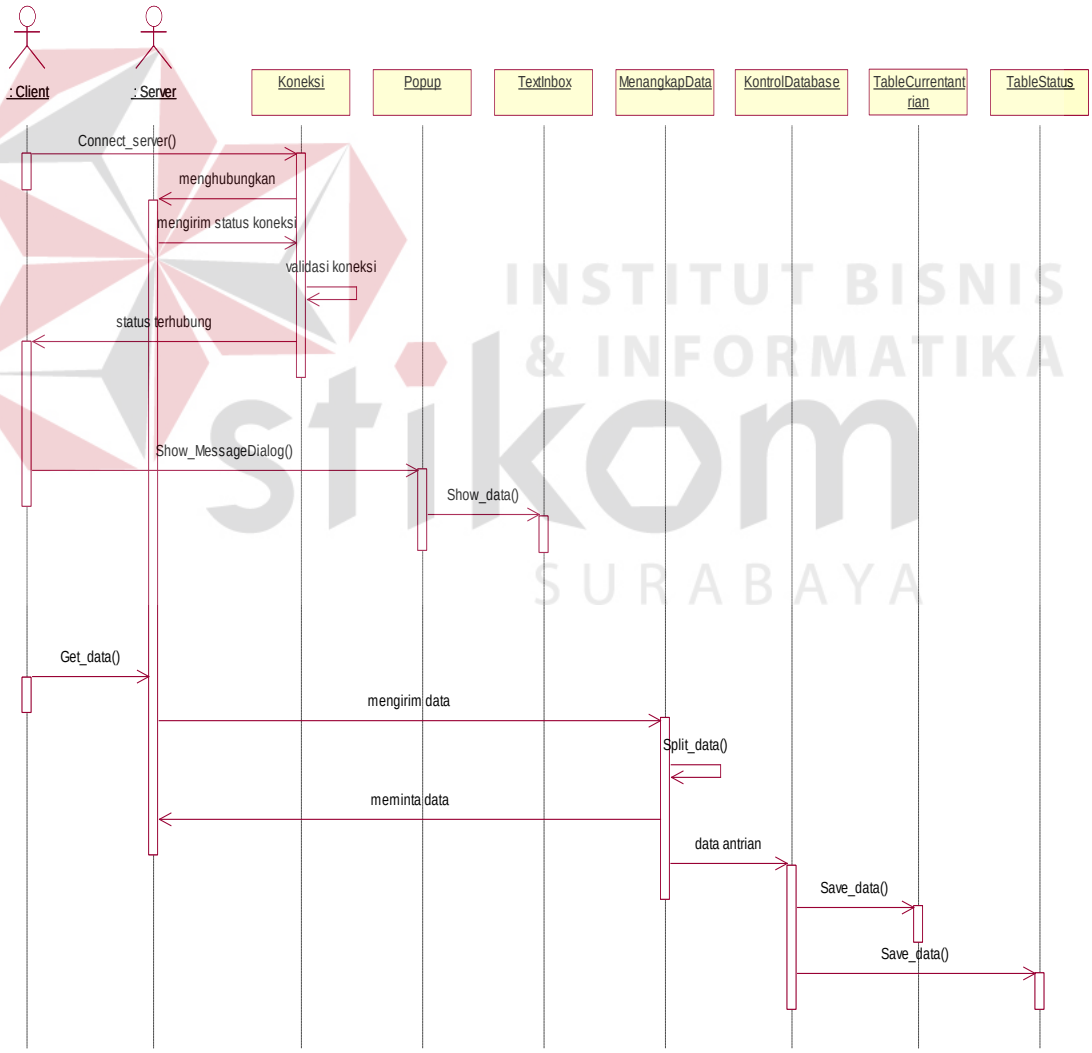
3.2.3 *Sequence Diagram*

Menurut Nugroho (2005) *sequence diagram* adalah *interaction diagram* yang memperlihatkan *event-event* yang berurutan sepanjang berjalannya waktu. *Sequence diagram* menggambarkan interaksi yang dilakukan oleh objek-objek dalam sistem. Kerja sama antar objek-objek dilaksanakan dengan saling mengirimkan pesan yang

membentuk sebuah alur kerja sama. Berikut *sequence diagram* sistem yang disarankan:

A. *Sequence Diagram* Mengolah Data

Alur yang digambarkan pada *sequence diagram* tak ubahnya adalah *flow of events* yang sudah dibuat sebelumnya. Pada *sequence diagram* ini digambarkan proses-proses yang terjadi sewajarnya. Berikut ini adalah Gambar 3.11 merupakan *sequence diagram* dari proses menangkap data.



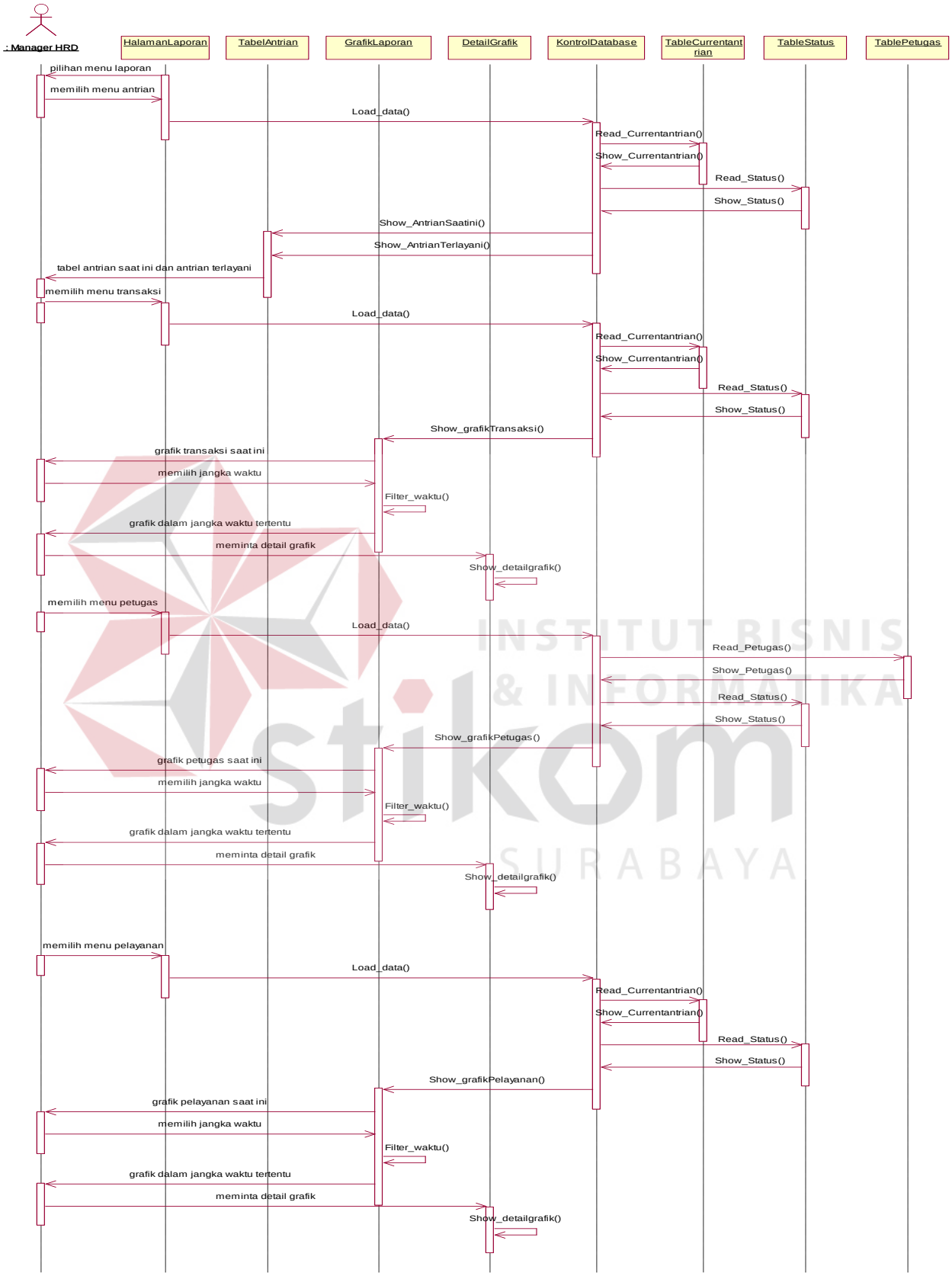
Gambar 3.11 *Sequence Diagram* Mengolah Data

Pada Gambar 3.11 terlihat proses mengolah data dimulai dari alur kegiatan *client* yang melakukan koneksi kepada *server* guna meminta data antrian, karena *server* merupakan sumber data dari sistem antrian yang akan dicatat aktivitasnya oleh *client*. Setelah terhubung maka *client* akan meminta data antrian kepada *server*, *server* akan menjalankan perintah Get_data() yang berarti *server* mengirimkan data tersebut. Pada halaman *client* terdapat *TextInbox* yang berfungsi sebagai media yang menampilkan data yang dikirim oleh *server*, pada langkah ini data akan dipilah menggunakan perintah Split_data() untuk mendapatkan data yang dibutuhkan. Apabila tidak terdapat data yang dibutuhkan maka *client* akan meminta data kembali kepada *server*, perulangan ini berfungsi apabila tidak ada data antrian yang ditangkap maka *client* tidak melanjutkan ke langkah selanjutnya sampai menemukan data antrian yang dibutuhkan. Setelah menemukan data antrian yang dibutuhkan maka data-data tersebut akan disimpan menggunakan perintah Save_data() ke *table* currentantrian dan *table* status.

B. *Sequence Diagram* Menampilkan Laporan

Sequence diagram di Gambar 3.12 merealisasikan proses menampilkan laporan. *Sequence diagram* menampilkan laporan juga menggambarkan *flow of events* dari proses menampilkan laporan yang merupakan proses akhir dari sistem informasi *monitoring* antrian.

Aliran proses menampilkan laporan dimulai dari *manager* HRD yang mengakses halaman utama laporan. Setelah itu akan menampilkan pilihan jenis laporan yang ingin dilihat dengan cara mengakses *database* maka laporan grafik akan terlihat pada *form* grafik laporan yang sudah tersedia.



INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Gambar 3.12 *Sequence Diagram* Menampilkan Laporan

3.2.4 *Class Diagram Monitoring* Antrian

Class diagram digunakan untuk menampilkan kelas-kelas atau paket-paket dalam sistem dan relasi antar mereka. Biasanya, dibuat beberapa *diagram* kelas untuk satu sistem. Satu *class diagram* menampilkan subset dari kelas-kelas dan relasinya. *Class diagram* lainnya mungkin menampilkan kelas-kelas termasuk *attribut* dan operasi dari kelas-kelas pembentuk *diagram*.

Class diagram adalah alat perancangan terbaik untuk tim pengembang perangkat lunak. *Class diagram* membantu tim pengembang mendapatkan pola kelas-kelas dalam sistem, struktur sistem sebelum menuliskan kode program, dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif rancangan (Sholiq, 2010).

Berdasarkan perencanaan sistem pada *use case diagram*, dibutuhkan kelas-kelas untuk membangun dan mendukung jalannya aplikasi. Hubungan antar kelas tersebut dapat digambarkan dalam sebuah *class diagram*. *Class diagram* dari sistem yang dibangun tidak ditampilkan secara keseluruhan, melainkan ditampilkan secara terpisah sesuai relasi-relasi yang ada untuk mempermudah pembacaan.

Kelas yang dibuat untuk aplikasi ini menggunakan pemodelan UML dengan konsep pemodelan *Model-View-Controller* (MVC). *Class model* merupakan *class* yang akan menangani segala sesuatu yang berhubungan dengan entitas. *Class view* merupakan *class* yang akan menangani segala sesuatu yang berhubungan dengan tampilan *user interface*. *Class controller* merupakan *class* yang menangani segala proses seperti proses pencarian data dan segala proses yang berhubungan dengan

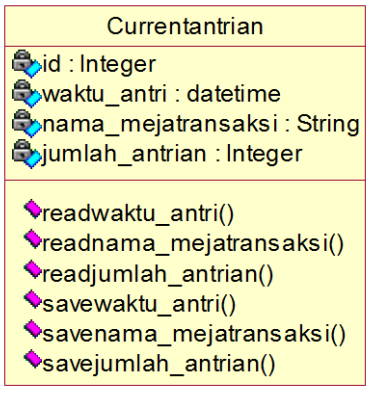
penyimpanan data pada SQL *server*. *Class diagram* yang digunakan dalam pada *monitoring* antrian adalah :

A. *Class* Model

Class Model atau *class* entitas adalah *class* yang digunakan menangani informasi yang mungkin disimpan secara permanen di dalam *database*. Berikut ini merupakan *class* model yang ada dalam sistem informasi *monitoring* antrian *Class* Model atau *class* entitas adalah *class* yang digunakan menangani informasi yang mungkin disimpan secara permanen di dalam *database*. Berikut ini merupakan *class* model yang ada dalam sistem informasi *monitoring* antrian beserta atribut dan operasi dalam *class* tersebut:

1. *Class* Currentantrian

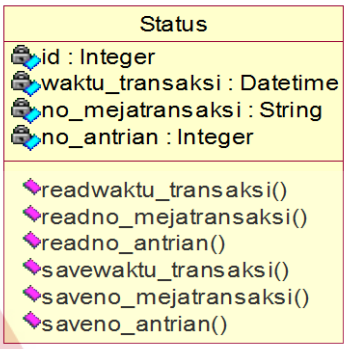
Pada Gambar 3.13 adalah notasi c*lass* Currentantrian yang berfungsi menangani data antrian yang digunakan untuk menyimpan data yang berada di sistem antrian. Atribut yang dimiliki *class* currentantrian adalah id, waktu_antri, nama_mejatransaksi, dan jumlah_antrian.



Gambar 3.13 Notasi *Class* Currentantrian

2. *Class* Status

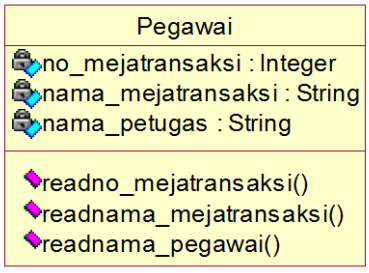
Pada Gambar 3.14 adalah notasi c*lass* Status yang berfungsi menangani data antrian yang digunakan untuk menyimpan data yang berada di sistem antrian. Atribut yang dimiliki *class* status adalah id, waktu_transaksi, no_mejatransaksi, dan no_antrian.



Gambar 3.14 Notasi *Class* Status

3. *Class* Pegawai

Pada Gambar 3.15 adalah notasi c*lass* Pegawai yang berfungsi memberi keterangan siapa dan apa nama transaksi yang dilakukan di sistem antrian tersebut. Atribut yang dimiliki *class* pegawai adalah no_mejatransaksi, nama_mejatransaksi, dan nama_pegawai.



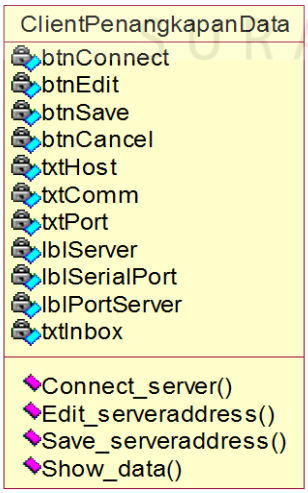
Gambar 3.15 Notasi *Class* Pegawai

B. *Class View*

Pengertian dari *class view* sebenarnya hampir sama dengan kelas pembatas (*boundary class*). Yaitu kelas yang terletak diantara sistem dengan sekelilingnya. Semua *form*, laporan, *user interface*, termasuk dalam kategori *class view*. Berikut ini *class view* pada sistem informasi *monitoring* antrian.

1. *Class PenangkapanData*

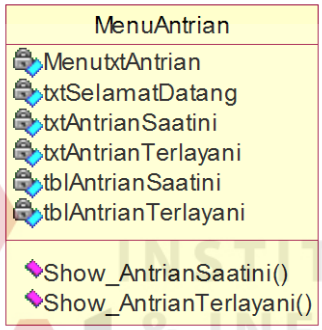
Pada Gambar 3.16 adalah notasi c*lass* ClientPenangkapanData digunakan untuk menangani fungsi-fungsi pada halaman *client* sebagai langkah awal pada aplikasi sistem informasi *monitoring* antrian. Halaman *client* ini digunakan juga untuk mendapatkan data atau kegiatan yang berada di *server* antrian. Atribut yang digunakan pada *class* PenangkapanData yaitu btnConnect, btnEdit, btnSave, btnCancel, txtHost, txtComm, txtport, lblServer , lblSerialPort, lblPortServer dan txtInbox. Operasi pada *class* ClientPenangkapanData yaitu Connect_server, Edit_serveraddress, Save_serveraddress, dan Show_data.



Gambar 3.16 Notasi *Class* ClientPenangkapanData

2. *Class* MenuAntrian

Pada Gambar 3.17 adalah notasi *class* MenuAntrian digunakan untuk menampilkan antrian yang terjadi dari sistem antrian. Atribut yang digunakan pada *class* MenuAntrian yaitu MenutxtAntrian, txtSelamatDatang, txtAntrianSaatini, txtAntrianTerlayani, tblAntrianSaatini, tblAntrianTerlayani. Operasi pada *class* MenuAntrian yaitu Show_AntrianSaatini dan Show_AntrianTerlayani.



Gambar 3.17 Notasi *Class* MenuAntrian

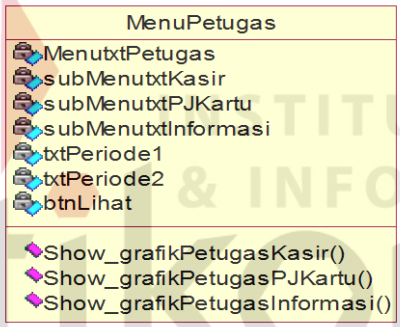
3. *Class* MenuTransaksi

Pada Gambar 3.18 adalah notasi *class* MenuTransaksi digunakan untuk menampilkan grafik laporan transaksi yang terjadi di sistem antrian. Atribut yang digunakan pada *class* MenuTransaksi yaitu MenutxtTransaksi, txtPeriode1, txtPeriode2, dan btnLihat. Operasi pada *class* MenuAntrian yaitu Show_grafikTransaksi.

Gambar 3.18 Notasi *Class* MenuTransaksi

4. *Class* MenuPetugas

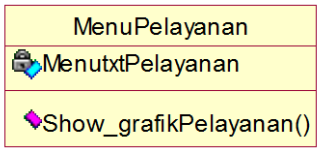
Pada Gambar 3.19 adalah notasi *class* MenuPetugas digunakan untuk menampilkan grafik laporan setiap petugas loket yang terlibat di sistem antrian. Atribut yang digunakan pada *class* MenuPetugas yaitu MenutxtPetugas, subMenutxtKasir, subMenutxtPJKartu, subMenutxtInformasi, txtPeriode1, txtPeriode2, dan btnLihat. Operasi pada *class* MenuAntrian yaitu Show_grafikPetugasKasir, Show_grafikPetugasPJKartu, dan Show_grafikPetugasInformasi.



Gambar 3.19 Notasi *Class* MenuPetugas

5. *Class* MenuPelayanan

Pada Gambar 3.20 adalah notasi *class* MenuPelayanan digunakan untuk menampilkan grafik laporan pelayanan yang terjadi di sistem antrian. Atribut yang digunakan pada *class* MenuPelayanan yaitu MenutxtPelayanan. Operasi pada *class* MenuAntrian yaitu Show_grafikPelayanan.



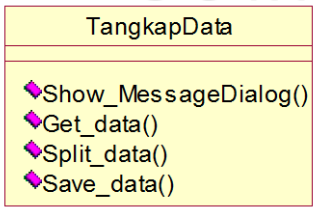
Gambar 3.20 Notasi Class MenuPelayanan

C. *Class Controller*

Class controller bertanggung jawab untuk mengkoordinasikan kegiatan-kegiatan terhadap *class* lainnya. *Class* ini bersifat optional, tetapi jika *class control* ini diputuskan untuk digunakan dalam sistem, maka lazimnya satu *class control* untuk satu *use case*. *Class control* digunakan untuk mengatur urutan kejadian dalam *use case* tersebut. *Class controller* pada sistem informasi *monitoring* antrian adalah:

1. *Class* TangkapData

Gambar 3.21 adalah notasi c*lass* TangkapData digunakan untuk mengatur jalannya menangkap data yang dilakukan oleh *client*. Show_MessageDialog, Get_data, Split_data, dan Save_data merupakan fungsi yang digunakan pada TangkapData. Fungsi Show_MessageDialog untuk menunjukan status terhubung atau tidak terhubung antara *client* dan *server* antrian, Get_data untuk meminta data ke *server*, Split_data untuk memilah data yang dibutuhkan, sedangkan Save_data untuk menyimpan data yang telah dipilah.

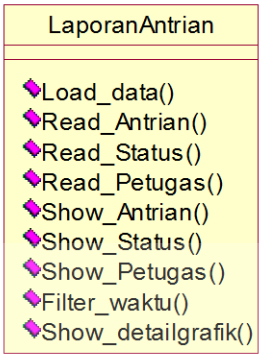


Gambar 3.21 Notasi *Class* TangkapData

2. *Class* LaporanAntrian

Gambar 3.22 adalah notasi c*lass* LaporanAntrian berfungsi sebagai pengatur tampilnya laporan. Melalui *class* ini *manager* HRD dapat melihat grafik dari antrian

yang ada serta detailnya, dan dapat memilih periode waktu dari grafik yang diinginkan. *class* ini memiliki operasi-operasi antara lain Load_data, Read_Antrian, Read_Status, Read_Petugas, Show_Antrian, Show_Status, Show_Petugas, Filter_waktu, dan Show_detailgrafik.



Gambar 3.22 Notasi Class LaporanAntrian

Berikut merupakan *class diagram* yang merupakan gabungan dari *class* yang dijelaskan di atas. *Class diagram* ini menunjukan relasi-relasi pada setiap *class* yang dapat dilihat pada Gambar 3.23.

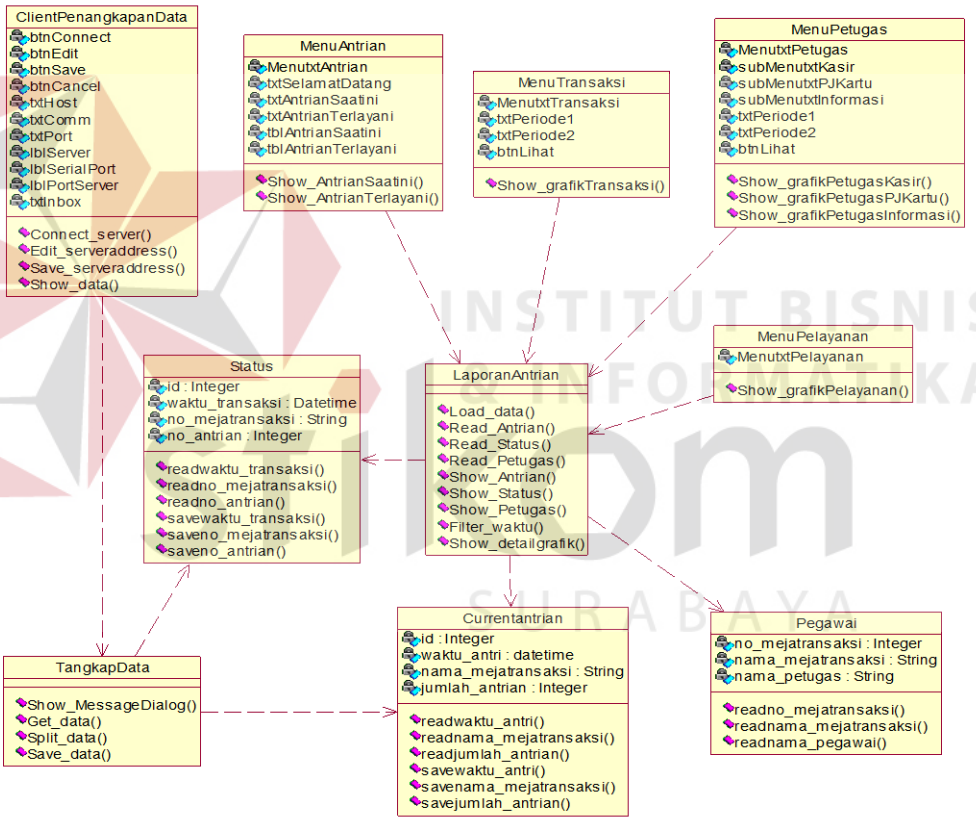
3.3 Perancangan *User Interface*

Pembuatan tampilan sangat diperlukan agar pengguna dapat berinteraksi dengan sistem, sehingga dibutuhkan perancangan secara detil mengenai tampilan aplikasi berdasarkan informasi yang ditampilkan pada layar web. Tampilan yang akan dibuat adalah tampilan *form* penangkapan data, dan *form* laporan.

3.3.1 Desain *Form* Penangkapan Data

Pada Gambar 3.24 menampilkan desain *form* menangkap data yang digunakan untuk menangkap data antrian yang berjalan pada *server*. Terdapat dua masukan yaitu *server* dan *port* yang berfungsi sebagai pengenalan alamat *server* tujuan, setelah

kedua masukan tersebut terisi maka tombol *connect* merupakan langkah selanjutnya. Tombol tersebut berfungis sebagai pengesekusi hubungan antar *client* dan *server*, inilah awal dimana data dari *server* bisa dipantau atau dilihat melalui *client* untuk diolah dengan cara memvalidasi data-data tersebut dan dipisahkan sesuai dengan prosesnya. Hasil pengolahan data-data tersebut berupa informasi yang berbentuk grafik.



Gambar 3.23 *Class diagram monitoring* antrian

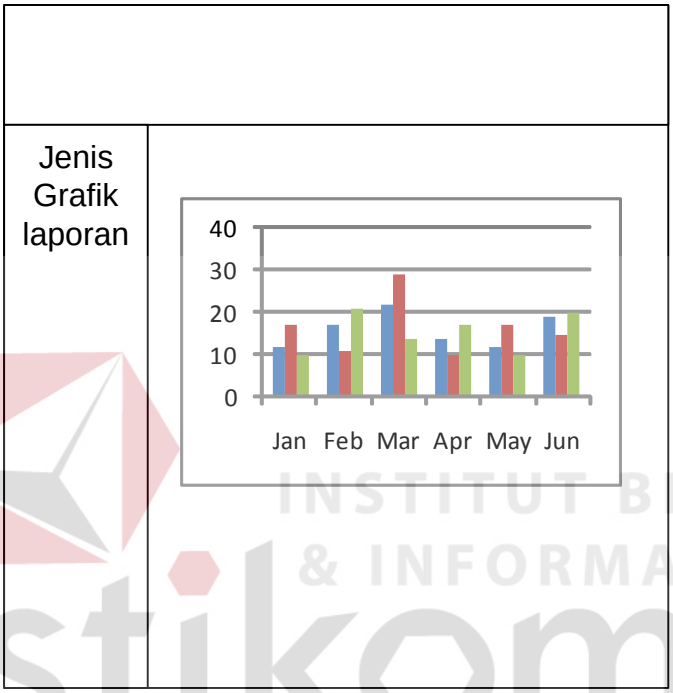
3.3.2 Desain *Form* Laporan *Monitoring* Antrian

Pada Gambar 3.25 merupakan desain *form* laporan antrian yang digunakan untuk melihat laporan akhir dari *monitoring* antrian. Didalamnya terdapat beberapa

jenis laporan yaitu laporan antrian, laporan transaksi, laporan petugas, dan laporan pelayanan. Laporan antrian berfungsi sebagai laporan yang menginformasikan bahwa total antrian yang terjadi dan terlayani ada berapa, diharapkan laporan antrian dapat membantu *manager* HRD dalam menetukan keputusan menambah loket transaksi atau tidak. Untuk laporan petugas berfungsi sebagai laporan yang menginformasikan berapa banyak petugas setiap loket transaksi melayani anggota koperasi, diharapakan dengan laporan ini *manager* HRD dapat memantau kinerja petugas dan dapat menetapkan target khusus kepada petugas untuk meningkatkan pelayanannya kepada anggota koperasi. Sedangkan laporan layanan merupakan hasil dari total waktu proses simpan maupun pinjam. Laporan ini diharapkan dapat menentukan waktu standart untuk proses simpan maupun pinjam.

Gambar 3.24 Desain *Form* Penangkapan Data

Masing-masing dari laporan pada Gambar 3.25 merupakan kebutuhan *manager* HRD untuk meningkatkan kualitas layanan. Dan terdapat detail dari tiap laporan untuk memudahkan *manager* HRD dalam memantau antrian atau mengevaluasi antrian.



Gambar 3.25 *Form* laporan *monitoring* antrian

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

3.4 Desain Uji Coba

Aplikasi yang dirancang dan diimplementasikan harus diuji untuk mengetahui tingkat keberhasilan dari pemakai aplikasi tersebut. Untuk melakukan uji coba pada aplikasi yang telah dibuat maka digunakan beberapa data transaksi yang dimasukkan sebagai *input*-an. Data transaksi yang dimasukkan bertujuan untuk menganalisa hasil *output* yang ditampilkan oleh sistem, apakah telah sesuai dengan tujuan pembuatan sistem informasi *monitoring* antrian.

Untuk aplikasi sistem informasi *monitoring* antrian, pengujian akan dilakukan dengan pendekatan metode *black box testing*. Pengujian *black box* adalah pengujian aspek *fundamental* sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan karena aplikasi sistem informasi *monitoring* antrian memiliki beberapa fungsi yang berjalan secara otomatis dibalik layar. Pengujian *black box* merupakan metode perancangan uji coba yang didasarkan pada spesifikasi perangkat lunak yang dibuat. Uji coba yang akan dilakukan yaitu uji coba fungsi aplikasi dan uji coba aplikasi kepada pengguna.

3.4.1 Uji Coba Fungsi Aplikasi

Pengujian ini dilakukan untuk mengetahui apakah fungsi-fungsi yang ada pada aplikasi berjalan dengan baik atau tidak. Adapun desain uji coba fungsi-fungsi yang akan diujikan adalah :

A. Desain Uji Coba Penangkapan Data

Pada desain uji coba proses menangkap data bertujuan untuk mengetahui apakah proses tangkap data dapat berfungsi dengan baik. Uji coba ini dilakukan dengan memasukkan antrian pada semua loket agar terciptanya data antrian di setiap loket. Desain uji coba proses menangkap data bisa dilihat pada Tabel 3.7.

Tabel 3.7 Desain Uji Coba Menangkap Data

Test Case ID	**Tujuan**	***Input***	***Output* Yang Diharapkan**	**Status**
1.	Menguji fungsi dari atribut *connection*	Mengakses lblServer, lblSerialPort, lblPortServer	Keterangan terhubung pada *server*	
2.	Menghubungkan ke *server* antrian	Alamat *server* dan *port*	Muncul pesan “terhubung”	

Test Case ID	**Tujuan**	***Input***	***Output* Yang Diharapkan**	**Status**
3.	Menguji penambahan data antrian	Mengakses *form* transaksi	Menambahnya data antrian yang tertera pada area txtInbox	
4.	Menguji pemilahan data antrian	Mengakses *form* transaksi	Tampilnya format data masukan yang dibutuhkan	

B. Desain Uji Coba Halaman Laporan

Pada desain uji coba halaman laporan bertujuan untuk mengetahui apakah halaman laporan antrian dapat berfungsi dengan baik. Uji coba ini dilakukan dengan masuk ke halaman tersebut lalu mengakses menu antrian, transaksi, petugas, dan pelayanan yang tersedia. Desain uji coba halaman laporan antrian bisa dilihat pada Tabel 3.8.

Tabel 3.8 Desain Uji Coba Halaman Laporan Antrian

Test Case ID	**Tujuan**	***Input***	***Output* yang diharapkan**	**Status**
5.	Menampilkan antrian yang terjadi	Mengakses menu antrian	Dapat menampilkan antrian yang terjadi	
6.	Menampilkan laporan transaksi	Mengakses menu transaksi	Dapat menampilkan laporan transaksi berupa grafik serta detailnya	
7.	Menampilkan laporan petugas	Mengakses menu petugas	Dapat menampilkan laporan petugas berupa grafik serta detailnya	

Test Case ID	**Tujuan**	***Input***	***Output* yang diharapkan**	**Status**
8.	Menampilkan laporan pelayanan	Mengakses menu pelayanan	Dapat menampilkan laporan pelayanan berupa grafik serta detailnya	

3.4.2 Uji Coba Aplikasi Kepada Pengguna

Pada proses uji coba ini, pengguna diharuskan melihat hasil dari *monitoring* antiran dengan catatan pada simpan pinjam terjadi transaksi dengan itu laporan *monitoring* dapat tercipta. Uji coba ini dilakukan dengan cara memberikan langsung kuesioner pada lampiran 4 kepada target pengguna yaitu *manager* HRD serta bagian EDP.

Kuesioner yang telah diisi oleh target pengguna akan dimasukan kedalam rumusan kuesioner, dimana desain table kuesioner yang telah diisi dapat dilihat pada tabel 3.9. Fungsi dari tabel tersebut untuk mengetahui hasil keseluruhan dari pendapat target pengguna terhadap hasil laporan *monitoring* antrian.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Tabel 3.9 Desain Rumusan Kuesioner

Bagian	**Pernyataan Nomor**	**Penilaian**					**Σ**	**X̄**	**Nilai Akhir**
		1	**2**	**3**	**4**	**5**			
Tampilan									
A	1.								
	2.								
	3.								
	4.								
	5.								
Navigasi									
B	1.								
	2.								
	3.								

	4.								
Materi Monitoring									
C	1.								
	2.								
	3.								
	4.								
Hasil Monitoring (Grafik laporan)									
D	1.								
	2.								
	3.								

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA