



## SISTEM INFORMASI *MONITORING* ANTRIAN PADA KOPERASI SETIA BHAKTI WANITA BERBASIS WEB

Muhamad Revo Dwi Putro<sup>1)</sup> Teguh Sutanto<sup>2)</sup> Erwin Sutomo<sup>3)</sup>

Program Studi/Jurusan Sistem Informasi

STMIK STIKOM Surabaya

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email : 1) [muhamadrevoDWIPUTRO@gmail.com](mailto:muhamadrevoDWIPUTRO@gmail.com), 2) [teguh.sutanto@gmail.com](mailto:teguh.sutanto@gmail.com), 3) [sutomo@stikom.edu](mailto:sutomo@stikom.edu)

**Abstract:** *Koperasi Wanita Setia Bhakti Wanita have loan and save activities that has been devised a system of the queue to set a queue a members will do transactions. The current system of queue couldn't give the history queue to the manager HRD, the impact is difficulty in performing manager HRD to monitoring the queue. Result of monitoring queue is information what manager HRD needs and could be useful to improve transaction services.*

*Based on the above then made monitoring system by considering the necessities manager HRD. Monitoring system proposed is assessment of concept socketserver, the first step make client that are connected by queue server. The second step, client will record history or activities in server. The third step, recorded of history is a sequence of data couldn't be used for information. In the next step, have doing sorting a list of data will be stored and used as a source of data to make report monitoring a queue.*

*The result of monitoring system is a report monitoring queue of comprising from reporting the real time a occurred queue, reporting the total queue, reporting the counter services (the amount queue per counter), reporting the services (total service, total service time) that's manager HRD used to evaluate queue system.*

**Keywords:** *monitoring, queue, saving and loan, socketserver, client server*

Koperasi wanita setia bhakti wanita adalah badan usaha milik sekelompok orang yang mengorganisasikan berbagai kegiatan ekonomi yang bertujuan untuk mensejahterakan anggotanya. Kegiatan-kegiatannya antara lain yaitu Simpan Pinjam, Swalayan, *Learning Center*, Griya Tamu, dan E-Kopwan. Dari beberapa kegiatan yang dijalankan, Simpan Pinjam merupakan proses bisnis utama yang paling mendasar karena dari dana Simpan Pinjam anggota yang diolah menjadikan koperasi ini dapat berkembang lebih besar lagi.

Dalam pelaksanaannya proses Simpan Pinjam mempunyai sistem yang dapat menangani antrian di setiap harinya. Sistem tersebut hanya mengatur nomor antrian ke loket-loket transaksi yang terdiri dari bagian informasi, PJ (Penanggung Jawab) kartu, dan kasir. 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 ke dua terdapat 6 loket transaksi (bagian informasi 1, PJ kartu 3 dan kasir 2). Proses antrian dimulai ketika anggota ingin melakukan transaksi dengan memilih salah satu dari 4 menu transaksi yang terdiri dari PJ kartu atas, PJ kartu bawah, informasi atas, dan informasi bawah. Menu PJ kartu ditujukan untuk anggota yang ingin melakukan transaksi pembayaran angsuran dan simpanan (*individu*) sedangkan menu informasi untuk anggota yang ingin melakukan transaksi pengajuan pinjaman.

*Monitoring* yang dibutuhkan *manager HRD* agar mendapatkan informasi dalam antrian Simpan Pinjam tidak bisa dilakukan secara manual. Karena deretan langkah yang panjang dan mempunyai beberapa proses yang berbeda dalam transaksinya, oleh sebab itu sebaiknya diadakan perbaikan pada sistem antrian yang sudah berjalan dengan mempertimbangkan

kebutuhan-kebutuhan *manager* HRD. *Monitoring* tersebut akan memuat *history* antrian di Simpan Pinjam, waktu yang dibutuhkan untuk proses transaksi dan karyawan-karyawan yang bertugas di Simpan Pinjam.

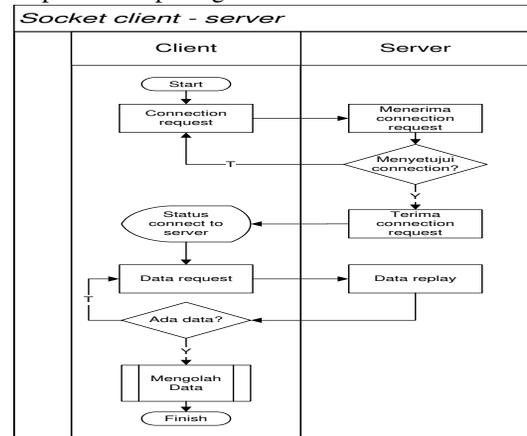
Sesuai dengan konsep sistem antrian yang dikemukakan Gross dan Harris (2001) kedatangan pelanggan untuk mendapatkan pelayanan, menunggu untuk dilayani jika fasilitas pelayanan masih sibuk, mendapatkan pelayanan dan kemudian meninggalkan sistem setelah dilayani. Dalam sistem antrian terdapat beberapa komponen dasar proses antrian yaitu kedatangan, pelayanan, dan antrian. Tiga komponen dasar dari porses antrian yang menjadi acuan untuk membuat sistem *monitoring*, oleh karena itu *Monitoring* yang akan dibuat disesuaikan dengan definisi *monitoring* yang dikemukakan Mercy (2005) yaitu sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Tujuan yang diharapkan dari sistem *monitoring* antrian dapat ditinjau dari beberapa segi. Segi objek dan subjek yang dipantau, serta hasil dari proses *monitoring* itu sendiri. Adapun beberapa tujuan dari sistem *monitoring* adalah (Amsler, dkk, 2009): 1. Memastikan suatu proses dilakukan sesuai prosedur yang berlaku, 2. Menyediakan probabilitas tinggi anakan keakuratan data bagi pelaku *monitoring*, 3. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses.

Dari permasalahan tersebut, diusulkan sistem informasi *monitoring* yang digunakan untuk membantu *manager* HRD dalam meningkatkan layanan. *Input* data yang akan diolah didapatkan dari transfer data antrian disetiap loket transaksi. Proses *monitoring* yang akan diterapkan meliputi penangkapan aliran data antrian, penangkapan data yang diterapkan berdasarkan cara kerja perkembangan *socket* yaitu menggunakan *socket clien-server*. Pada intinya terdapat dua *socket* yang berbeda fungsi, satu *socket* berfungsi sebagai *server* dan satu *socket* berfungsi sebagai *client*. *Socket* yang berfungsi sebagai *server* akan menunggu koneksi dari *socket* yang berfungsi sebagai *client*, ketika koneksi antara *server* dan *client* telah terhubung maka keduanya akan saling mengirim data (Kurniawan, 2011). Setelah mendapatkan data yang diperlukan lalu akan disimpan ke *database* setelah itu akan diolah oleh sistem untuk menghasilkan *output* dari Sistem Informasi *Monitoring* ini yaitu informasi

*real time* antrian yang terjadi, informasi total antrian per periode, informasi petugas loket (jumlah antrian per loket), informasi pelayanan (total layanan, total waktu layanan). Informasi 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. Selain itu dapat memberikan gambaran pada *manager* HRD dalam setiap bulan antrian yang menumpuk agar bisa direspon dengan baik.

**METODE**

Dalam pembuatan sistem informasi *monitoring* antrian menggunakan konsep *socket client-server*. Metode tersebut merupakan penerapan dari cara kerja *socket* itu sendiri. Dapat dilihat pada gambar 1.



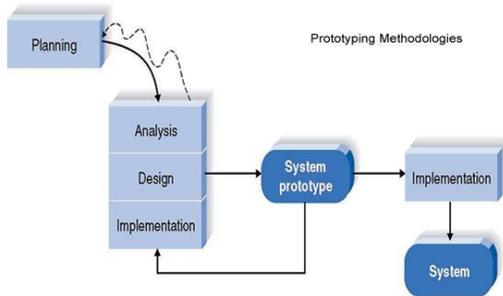
**Gambar 1** Penerapan metode *socket client-server* pada sistem *monitoring* antrian

Pada Gambar 1 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*.

**Analisis sistem**

Pembuatan sistem informasi *monitoring* antiran menggunakan konsep *Systems Development Life Cycle* (SDLC) dengan metode *prototype*(dapat dilihat pada gambar 2) yang menitik beratkan tiga proses inti dari SDLC yaitu analisa, desain, dan implementasi pada Rancang Bangun Sistem Informasi *Monitoring* Antrian pada Koperasi wanita setia bhakti wanita Berbasis Web.



**Gambar 2** Metode *Prototype* (Dennis,dkk,2013)

Sesuai dengan metode *prototype* yang digunakan untuk mendapatkan akar masalah dan langkah-langkah yang sudah direncanakan yaitu dengan melakukan 1. Analisa, 2. Desain, 3. Implementasi.

Langkah pertama merupakan pengumpulan data yang diperlukan pada antrian bagian Simpan Pinjam di Koperasi Setia Bhakti Wanita. Analisa sistem memerlukan beberapa tahapan, yaitu :

1. *Wawancara/Interview*

Tahap ini berinteraksi dengan nara sumber yang berguna untuk mengetahui masalah yang ada pada alur Simpan Pinjam. Hasil dari tahap ini 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

2. *Pengamatan/Observasi*

Tahap ini dilakukan untuk melihat kondisi Simpan Pinjam terutama sistem antriannya. Pengamatan yang dilakukan meliputi operasi mesin antrian dan sifat antrian. 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.

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).

Sistem informasi *monitoring* antrian yang akan dibangun membutuhkan data antrian sebagai data utama yang digunakan untuk menghasilkan informasi ke *manager* HRD, berikut pada table 1 merupakan kebutuhan dari *manager* HRD.

**Tabel 1** Analisa Kebutuhan Sistem

No	Pengguna	Kebutuhan	Laporan Yang dihasilkan
1	<i>Manager</i> HRD	<ul style="list-style-type: none"> <li>• Informasi antrian yang terjadi</li> <li>• Informasi total antrian per periode</li> <li>• Informasi petugas loket (jumlah antrian per loket)</li> <li>• Informasi pelayanan (total layanan, total waktu)</li> </ul>	<ul style="list-style-type: none"> <li>• Laporan <i>real time</i> antrian yang terjadi</li> <li>• Laporan total antrian per periode</li> <li>• Laporan petugas loket (jumlah antrian per loket)</li> <li>• Laporan pelayanan (total layanan, total waktu layanan)</li> </ul>

N o	Peng-guna	Kebutuhan	Laporan Yang dihasilkan
		layanan)	

Berikut merupakan kebutuhan *functional* dan *non functional* dari sistem informasi *monitoring* antrian.

**Tabel 2** *Functional* dan *Non-functional* Sistem Informasi *Monitoring* Antrian

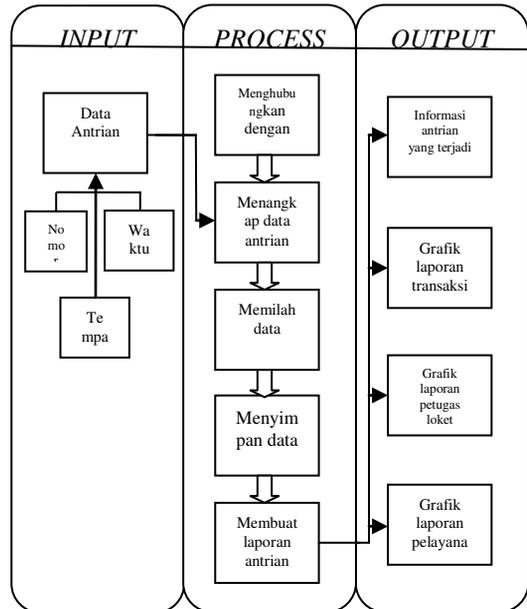
Functional	Non Functional
<ul style="list-style-type: none"> <li>Sistem dapat menangkap data antrian pada <i>server</i>.</li> <li>Sistem menyediakan halaman yang memungkinkan penggunaannya mendapat laporan <i>monitoring</i> antrian.</li> </ul>	<ul style="list-style-type: none"> <li>Sistem dapat memilah data antrian dari <i>server</i></li> </ul>

**Perancangan Desain Sistem**

Gambar 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 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 wanita

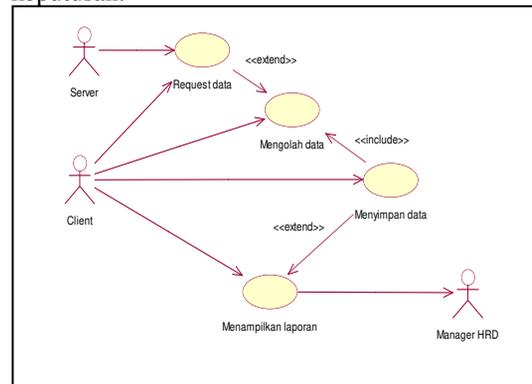
setia bhakti wanita. Pada *server* antrian terdapat kegiatan yang berlangsung di sistem antrian maka titik pengamatan dilakukan di *server*. Proses *monitoring* menggunakan konsep *socket client-server*, karena konsep *socket client-server* tidak akan merusak sistem antrian yang sudah terbuat.



**Gambar 3** Blok Diagram Sistem Informasi *Monitoring* Layanan

**Use Case Monitoring Antrian**

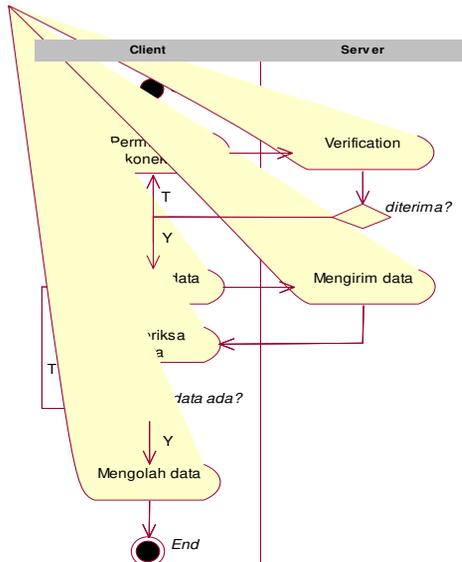
Pada Gambar 4 terdapat *use case diagram* *Monitoring* antrian. Pada *use case diagram* ini, mempunyai tiga aktor yaitu *server*, sistem, dan manajemen. Aktor *server* sebagai sumber data yang terdapat di sistem yang akan dicatat aktivitasnya oleh sistem untuk dijadikan histori serta laporan *monitoring*. Laporan-laporan *monitoring* antrian ditujukan kepada *management* yang nantinya akan mengambil keputusan.



**Gambar 4** Use Case *Monitoring* Antrian

**Activity Diagram dan Flow of Event Proses Reques Data**

Activity diagram adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *use case* dalam bentuk grafik. Diagram ini menunjukkan 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 (Sholih, 2006).



**Gambar 5** Activity Diagram Request Data

Diagram activity pada gambar 5 menceritakan tentang, bagaimana *client server* berkomunikasi. Pada awalnya proses koneksi sama seperti langkah *socketserver*, 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 didata 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 1 dengan kondisi akhir yang diinginkan dan kondisi akhir gagal, serta alur alternatif untuk menangani kondisi salah. Proses *request data* dilakukan untuk meminta data yang tertera di server, yaitu data antrian yang berjalan

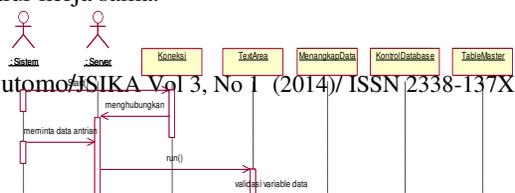
di sistem. Berikut *Flow of events* dari proses *request data* dapat dilihat pada tabel 3.

**Tabel 3** Flow of Event request data

Nama Proses	Request data	
<b>Kebutuhan terkait</b>	Salah satu proses untuk memulai <i>monitoring</i> antiran yang berada di koperasi Setia Bhakti Wanita. Proses ini berhubungan langsung dengan <i>server</i> , dan disetiap kegiatan server akan terpantau.	
<b>Tujuan</b>	Meminta status antrian yang telah terjadi.	
<b>Prasyarat</b>	Sudah terkoneksi dengan <i>server</i> .	
<b>Kondisi akhir sukses</b>	Ada antrian yang terjadi <i>client</i> akan meminta <i>server</i> untuk mengirimkan status antrian.	
<b>Kondisi akhir gagal</b>	Jika tidak ada antrian atau dalam <i>server</i> tidak ada kegiatan, maka <i>client</i> akan meminta terus-menerus sampai terdapat antrian.	
<b>Aktor utama</b>	<i>Server, client</i>	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Terdapat kegiatan di <i>server</i> antrian	
<b>Alur utama</b>	<b>Langkah</b>	<b>Aksi</b>
	1.	Melakukan koneksi ke <i>server</i> antrian.
	2.	<i>Client</i> mengirimkan proses cek data antrian kepada <i>server</i> .
	3.	<i>Server</i> akan merespon dengan mengirimkan status antrian yang telah terjadi.
	4.	Status antrian akan divalidasi.
5.	Melakukan proses menangkap data.	
<b>Alur Perluasan</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
		Tidak ada

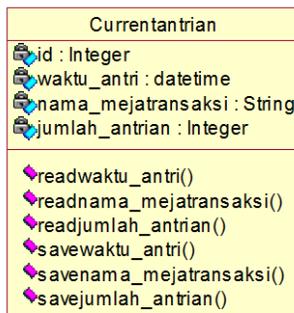
**Sequence Diagram Menangkap Data**

Diagram sekuensial (*sequence diagram*) dapat dilihat pada gambar 6 yang digunakan untuk menunjukkan alur fungsionalitas yang melalui sebuah *use case* yang disusun dalam urutan waktu. Diagram sekuensial menggambarkan interaksi yang dilakukan oleh obyek-obyek dalam sistem. Kerja sama antar obyek-obyek dilaksanakan dengan saling mengirimkan pesan yang membentuk sebuah alur kerja sama.



**Gambar 6** Sequence Diagram Menangkap Data Class Currentantrian

Class 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. Notasi class shift dapat dilihat pada gambar 7.



**Gambar 7** Notasi Class Master

**HASIL DAN PEMBAHASAN**

Tujuan dari langkah ini adalah untuk mengetahui apakah perangkat lunak yang telah dibuat sudah berjalan dengan baik dan sesuai dengan tujuan atau *output* yang diharapkan. Langkah ini memiliki beberapa tahapan, yaitu:

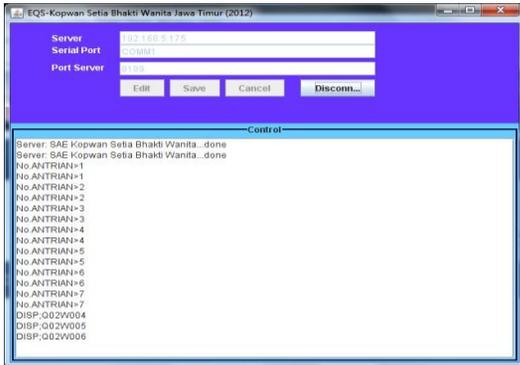
1. Uji coba fungsi perangkat lunak
  - Pengujian ini, dilakukan untuk mengetahui apakah fungsi-fungsi yang ada pada perangkat lunak berjalan dengan baik atau tidak. Adapun fungsi-fungsi yang akan diujikan adalah :
    - 1.1. Hasil uji coba penangkapan data
      - Pada uji coba proses menangkap data bertujuan untuk mengetahui apakah proses menangkap data dapat berfungsi dengan baik. Uji coba ini dilakukan dengan memasukkan antrian pada semua loket agar terciptanya data antrian disetiap loket. Pada table 4 merupakan hasil uji coba pada proses menangkap data, hasil

membuktikan bahwa *client* dapat menangkap data dari *server* sesuai dengan yang diharapkan.

**Tabel 4** Hasil uji coba menangkap data

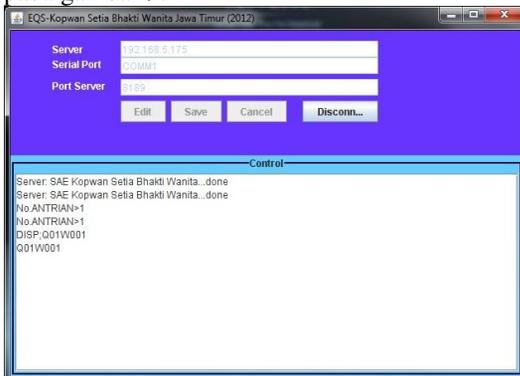
<i>Test Case ID</i>	Tujuan	Input	Output Yang Diharapkan	Status
1.	Menguji fungsi dari atribut <i>connection</i>	Mengakses <i>lblServer</i> , <i>lblSerialPort</i> , <i>lblPortServer</i>	Keterangan terhubung pada <i>server</i>	Sukses
2.	Menghubungkan ke <i>server</i> antrian	Alamat <i>server</i> dan <i>port</i>	Muncul pesan “terhubung”	Sukses
3.	Menguji penambahan data antrian	Mengakses <i>form</i> transaksi	Menambahnya data antrian yang tertera pada area <i>txtInbox</i>	Sukses
4.	Menguji pemilahan data antrian	Mengakses <i>form</i> transaksi	Tampilnya format data masukan yang dibutuhkan	Sukses

Berikut pembuktian data pada *server* antrian dapat ditangkap dapat dilihat pada gambar 8. Data yang telah tertangkap dan ditampilkan oleh *client* penangkapan data adalah data antrian yang ada, yaitu nomer antrian dan keterangan data proses itu berlangsung dimana atau nomor antrian itu dipanggil ke loket petugas mana. Pada contoh hasil tampilan penangkapan data terdapat tujuh nomor antrian yang telah diambil atau telah mengantri untuk dipanggil. Dibawah data nomor antrian terdapat format data untuk pemanggilan nomor antrian disetiap loket antrian, yang telah dipanggil yaitu nomor 4-6.



**Gambar 8** Pembuktian penangkapan data antrian

Setelah penangkapan itu terjadi maka data akan diolah dengan sesuai format data yang telah ditentukan, misalnya format “DISP:Q01W001”, terdapat arti dari format tersebut “.Q01” menunjukkan dimeja berapa transaksi dilakukan sedangkan “W001” merupakan nomor antrian yang bertransaksi. Pembuktiannya dapat dilihat pada gambar 9.



**Gambar 9** Pembuktian pengolahan data antrian 1.2.

Hasil uji coba halaman laporan 1.2. Pada uji coba halaman laporan bertujuan untuk mengetahui apakah halaman laporan dapat berfungsi dengan baik. Uji coba ini dilakukan dengan masuk kehalaman tersebut lalu mengakses menu laporan yang tersedia. Pada table 5 merupakan hasil uji coba pada proses halaman laporan, hasil membuktikan bahwa halaman laporan dapat berfungsi dengan baik sesuai dengan yang diharapkan yaitu setiap halaman laporan terjadi peningkatan data dari proses antrian pada simpan pinjam. Hasil uji coba halaman laporan dapat dilihat pada table 5.

2. Uji coba perangkat lunak 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

**Tabel 5** Hasil uji coba halaman laporan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output Yang Diharapkan</i>	<i>Status</i>
5.	Menampilkan antrian yang terjadi	Mengakses menu antrian	Dapat menampilkan antrian yang terjadi	<b>Sukses</b>
6.	Menampilkan laporan transaksi	Mengakses menu transaksi	Dapat menampilkan laporan transaksi berupa grafik serta detailnya	<b>Sukses</b>
7.	Menampilkan laporan petugas	Mengakses menu petugas	Dapat menampilkan laporan petugas berupa grafik serta detailnya	<b>Sukses</b>
4.	Menampilkan laporan pelayanan	Mengakses menu pelayanan	Dapat menampilkan laporan pelayanan berupa grafik serta detailnya	<b>Sukses</b>

memberikan langsung kuesioner kepada target pengguna yaitu *manager* HRD serta bagian EDP.

Fungsi dari tabel tersebut untuk mengetahui hasil keseluruhan dari pendapat target pengguna terhadap hasil laporan *monitoring* antrian. Table hasil kuesioner dapat dilihat pada table 6.

Tabel 6 Hasil kuesioner

Bagian	Pernyataan No	Penilaian					$\Sigma$	$\bar{X}$	Nilai Akhir
		1	2	3	4	5			
<b>Tampilan</b>									
A	1.	0	0	0	2	0	8	4	3,6
	2.	0	0	2	0	0	6	3	
	3.	0	0	0	2	0	8	4	
	4.	0	0	0	2	0	8	4	
	5.	0	0	2	0	0	6	3	
<b>Navigasi</b>									
B	1.	0	0	0	2	0	8	4	3,8
	2.	0	0	0	2	0	8	4	
	3.	0	0	1	1	0	7	3,5	
	4.	0	0	1	1	0	7	3,5	
<b>Materi Monitoring</b>									
C	1.	0	0	0	2	0	8	4	3,9
	2.	0	0	0	2	0	8	4	
	3.	0	0	0	2	0	8	4	
	4.	0	0	1	1	0	7	3,5	
<b>Hasil Monitoring (Grafik laporan)</b>									
D	1.	0	0	0	2	0	8	4	4
	2.	0	0	0	2	0	8	4	
	3.	0	0	0	2	0	8	4	

Dapat disimpulkan dari hasil keseluruhan uji coba perangkat lunak kepada pengguna untuk bagian tampilan mendapatkan 72%, untuk bagian navigasi 76%, materi *monitoring* 78%, bagian hasil *monitoring* (grafik laporan) 80%. Hal ini mengisaratkan bahwa perangkat lunak layak untuk diimplementasikan pada Koperasi Wanita Setia Bhakti Wanita.

Dari penggabungan hasil uji coba fungsi perangkat lunak dan hasil uji coba perangkat lunak kepada pengguna, dapat disimpulkan perangkat lunak akan membantu *manager* HRD untuk dapat memantau antrian pada bagian simapn pinjam pada Koperasi Setia Bhakti Wanita serta bisa diimplementasikan sesuai dengan tujuan dari penelitian ini.

### KESIMPULAN

Setelah melakukan uji coba dan evaluasi terhadap sistem informasi *monitoring* antrian

maka dapat ditarik kesimpulan, bahwa sistem informasi *monitoring* antrian ini dapat membantu *manager* HRD dalam memantau antrian yang terjadi di bagian simpan pinjam dari Koperasi Wanita Setia Bhakti Wanita. Hasil dari sistem *monitoring* antrian adalah laporan *real time* antrian yang terjadi, laporan total antrian per periode (per hari, per minggu, per bulan), laporan petugas loket (jumlah antrian per loket), laporan pelayanan (total layanan, total waktu layanan). Laporan-laporan tersebut digunakan oleh *manager* HRD untuk mengevaluasi antrian di bagian simpan pinjam.

### SARAN

Adapun saran yang dapat diberikan kepada peneliti berikutnya apabila ingin mengembangkan perangkat lunak yang telah dibuat ini agar menjadi lebih baik adalah sistem dapat dikembangkan untuk mengatasi masalah jika terjadi *server down* atau listrik padam (contoh kasus akibat masalah tersebut seperti *server* akan mengulang nomor urut antrian dan nomor urut yang tertera pada *display* setiap loket akan mengulang) dengan cara mengembalikan data antrian yang telah disimpan kedalam *database*.

### RUJUKAN

- Amsler, G. M., Findley, H. M., & Ingram, E., 2009, *Performance monitoring: guidance for the modern workplace. Supervision*, 70, 12-19.
- Dennis, Alan, Tegorden, David, & Wixom, B. H., 2013, *System Analysis and Design with UML, 4<sup>th</sup> Edition*. John Wiley & Sons, Inc.
- Gross, Donald dan Harris, Carl, 2001, *The Queueing Systems*, NewYork, McGraw-Hill, Inc.
- Kurniawan, Agus, 2011, *Pemrograman Jaringan dengan JAVA*, Yogyakarta : Penerbit ANDI.
- Mercy Corps, 2005, *Design, monitoring, and evaluation guidebook*.
- Sholih, 2006, *Pemodelan Sistem Informasi Berorientasi Obyek dengan UML*, Graha Ilmu, Yogyakarta.