

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1. Implementasi

Tahap implementasi program merupakan suatu tahap penerapan dari analisis dan desain sistem yang telah dibuat. Berikut ini adalah kebutuhan sistem yang terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang harus dipenuhi sehingga aplikasi dapat berjalan dengan baik.

4.1.1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras atau *hardware* merupakan kebutuhan komputer atau peralatan pendukung agar aplikasi dapat berjalan dengan baik. Spesifikasi *hardware* yang dibutuhkan antara lain :

Komputer *Server* :

1. *Processor dual core* atau lebih
2. Ram minimal 2GB
3. *Hardisk* minimal 80GB
4. *Keyboard*
5. *Mouse*
6. *Drive CD-ROM* atau DVD

7. Monitor
8. Monitor *display* eksternal > 21' (sesuai kebutuhan)
9. *Speaker*

Komputer *Client* :

1. *Processor* pentium 4 atau lebih
2. Ram minimal 1GB
3. *Hardisk* minimal 40GB
4. *Keyboard*
5. *Mouse*
6. *Drive* CD-ROM atau DVD
7. Monitor

Komputer Ambil Nomor :

1. *Processor* pentium 3 atau lebih
2. Ram minimal 1GB
3. *Hardisk* minimal 40GB
4. *Keyboard* (opsional)
5. *Mouse*
6. *Drive* CD-ROM atau DVD
7. Monitor dengan *Touch Screen* (opsional) atau monitor biasa

Perangkat Jaringan :

1. *Hub/switch*
 2. Kabel LAN
- Atau
3. *Wireles router*

4.1.2. Kebutuhan Perangkat Lunak

Perangkat lunak merupakan program komputer yang diperlukan untuk mengoperasikan perangkat keras untuk kebutuhan menjalankan aplikasi. Pada saat pembuatan aplikasi, perangkat lunak yang digunakan antara lain :

1. Sistem operasi Windows 7 Profesional
2. *Database* Microsoft SQL Server 2008
3. Visual Studio .Net 2008 Service Pack 2
4. .Net Framework 3.5
5. Microsoft Visio 2010
6. Power Designer 6.0
7. Crystal Report 10
8. IBM Rational Rose 7.0.0

Saat implementasi nantinya, diperlukan perangkat lunak antara lain :

Komputer Server :

1. Sistem operasi Windows XP Profesional Service Pack 2
2. *Database* Microsoft SQL Server 2008
3. .Net Framework 3.5 Runtime Environment
4. Crystal Report 10 Runtime

Komputer *Client* & Komputer Ambil Nomor

1. Sistem operasi Windows XP Profesional Service Pack 2
2. .Net Framework 3.5 Runtime Environment

4.2. Evaluasi

Pada tahap evaluasi kali ini, program yang dihasilkan akan diuji menggunakan studi kasus.

4.2.1. Tes Kasus

Diasumsikan sebuah perusahaan telekomunikasi telah membuka sebuah *customer care* untuk melayani pelanggan dengan kebutuhan :

1. Membuka 6 *counter* layanan dengan masing-masing *counter customer service* sebanyak 4 *counter* dan *counter* kasir untuk pembayaran sebanyak 2 *counter*.
2. Memiliki 4 jenis layanan yaitu, *customer service*, pelayanan keluhan, pengajuan berlangganan dan pembayaran tagihan.
3. Masing-masing *counter* melayani jenis layanan yang ditentukan sebagai berikut : *counter* 1 dan 2 merupakan *customer service* dan pelayanan keluhan. *Counter* 3 dan 4 merupakan *customer service* dan pengajuan berlangganan. *Counter* 5 dan 6 untuk pembayaran tagihan.

4. Perusahaan telah memiliki 6 orang karyawan untuk ditempatkan di *customer care* tersebut, dimana karyawan ini memiliki hak akses pada aplikasi.
5. Pada saat melayani, aplikasi berjalan. Jika *customer* yang datang pada *counter* ingin mengambil layanan lain atau dirasa perlu diarahkan ke layanan lain, *customer service*/karyawan yang bertugas di *counter* tersebut dapat membantu dengan melakukan *forwarding* ke layanan tertentu. Kemudian komputer mati karena mati listrik. Aplikasi harus berada pada nomor antrian terakhir seperti pada saat listrik belum mati
6. Pada akhir bulan, pimpinan memeinginkan data kepuasan pelanggan sebagai penilaian terhadap kinerja pegawainya.

4.2.2. Metode pengujian

Untuk menguji aplikasi, akan dilakukan uji coba *input* dan melihat *output* yang dihasilkan. Dalam pengujian akan dilakukan tes kasus/simulasi dengan memberikan *input* dengan menyiapkan target yang diharapkan dari input yang diinginkan. Pengujian akan dilakukan berulang-ulang dengan metode *Control Structure Testing* berupa *Condition Testing* atau tes kondisi yang memeriksa kondisi

logika yang diterapkan pada program untuk memastikan semua fungsi dan logika sudah berjalan dengan benar. Pengujian akan diulang minimal sepuluh kali untuk setiap *test case*.

Tabel 4.1 Tes Kasus Pengujian

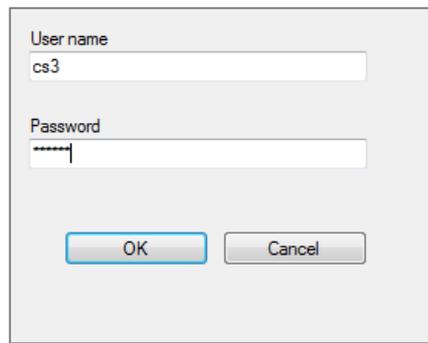
<i>Case ID</i>	<i>Case</i>	<i>Input</i>	<i>Output yang Diharapkan</i>
1	Memiliki 6 <i>counter</i> dengan 4 CS dan 2 Kasir.	<i>Input</i> data <i>counter</i> .	<i>Counter</i> tersimpan, ditampilkan pada list dan dapat diakses saat melakukan <i>mapping</i> .
2	Memiliki 4 layanan <i>customer service</i> , pelayanan keluhan, pengajuan berlangganan dan pembayaran tagihan.	<i>Input</i> data layanan.	Layanan tersimpan dan ditampilkan pada <i>list</i> dan dapat diakses saat melakukan <i>mapping</i> .
3	<i>Mapping</i> layanan ke <i>counter</i> , sebuah <i>counter</i> melayani layanan tertentu saja.	Data <i>counter</i> dan data layanan yang diinputkan sebelumnya.	Layanan tersimpan dan termapping ke <i>counter</i> .
4	User memiliki hak akses ke aplikasi.	<i>Input</i> data <i>user</i> , <i>password</i> .	<i>User</i> terdaftar dan tersimpan di <i>database</i> dan muncul pada <i>list</i> .
5	Menguji fungsi aplikasi dalam melayani antrian dan mengembalikan kondisi saat listrik padam	Nomor antrian yang dipanggil.	Tampilan nomor antrian dan panggilan suara. Nomor antrian tidak kacau saat listrik padam
6	Menghasilkan rekap data.	Pilihan tanggal rekap.	Rekap data berupa laporan yang dapat dicetak.

Tabel 4.2 Parameter Pengujian

Test ID	Tes Yang dilakukan	Tujuan
1	a. Uji <i>input</i> data <i>counter</i> b. Menghapus data <i>counter</i> c. mengedit data <i>counter</i> .	Menguji untuk menemukan <i>error</i> pada aplikasi saat <i>input</i> dan manajemen data <i>counter</i> .
2	a. Uji <i>input</i> data layanan b. Menghapus data layanan c. Mengedit data layanan.	Menguji untuk menemukan <i>error</i> pada aplikasi saat <i>input</i> dan manajemen data layanan.
3	a. Melakukan uji mapping layanan ke <i>counter</i> . b. Menghapus layanan yang telah dimapping.	Menguji fungsi pemetaan layanan ke <i>counter</i> yang disediakan
4	a. Uji <i>input</i> data <i>user</i> b. Menghapus data <i>user</i> c. Mengedit data <i>user</i>	Menguji untuk menemukan <i>error</i> pada aplikasi saat <i>input</i> dan manajemen data <i>user</i> .
5	a. Melakukan panggilan nomor antrian b. Melakukan <i>forwarding</i> antrian c. Menerima <i>input feedback</i> d. Menguji <i>handling</i> saat listrik mati (menutup dan me-restart komputer).	Menguji kemampuan aplikasi dalam menhandel gangguan listrik yang tidak mengacaukan nomor antrian
6	a. Membuat rekap b. Memilih rekap berdasarkan bulan tertentu	Menguji kemampuan aplikasi menghasilkan rekap data.

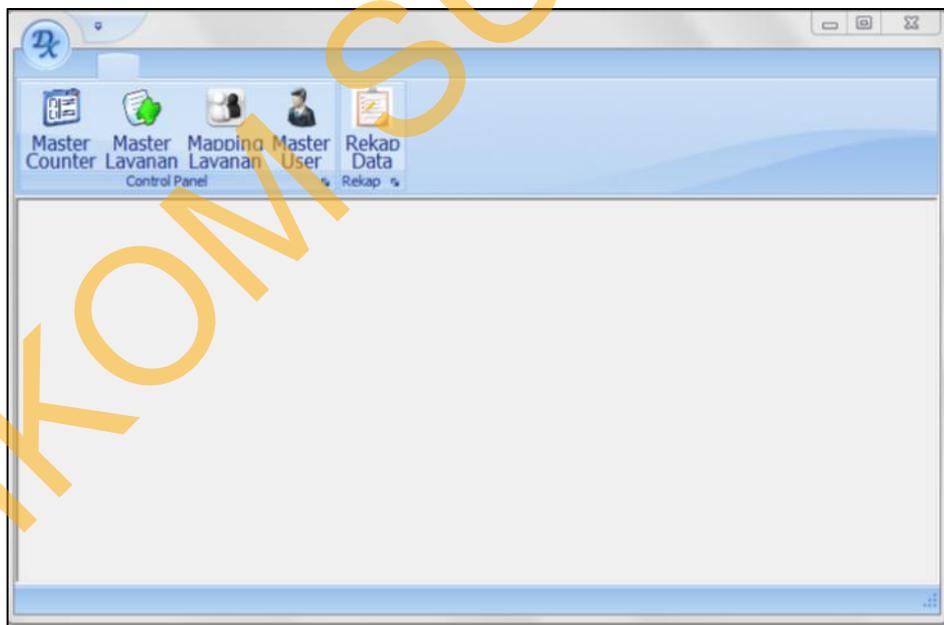
4.2.3. Penyelesaian Kasus

Setelah melakukan pemasangan perangkat dan instalasi, maka aplikasi sistem antrian ini dapat dijalankan. Yang pertama adalah *form* administrasi data yang bernama mdiFormAdmin :

A login dialog box with a light gray background. It contains two text input fields: 'User name' with the text 'cs3' and 'Password' with masked characters '*****'. Below the fields are two buttons: 'OK' and 'Cancel'.

Gambar 4.1 *Form Login* Untuk Admin

Untuk masuk ke *form* mdiFormAdmin seperti yang terlihat pada gambar 4.1, seorang *admin* harus *login* terlebih dahulu. Setelah itu akan muncul tampilan sebagai berikut mdiFormAdmin seperti pada gambar 4.2.

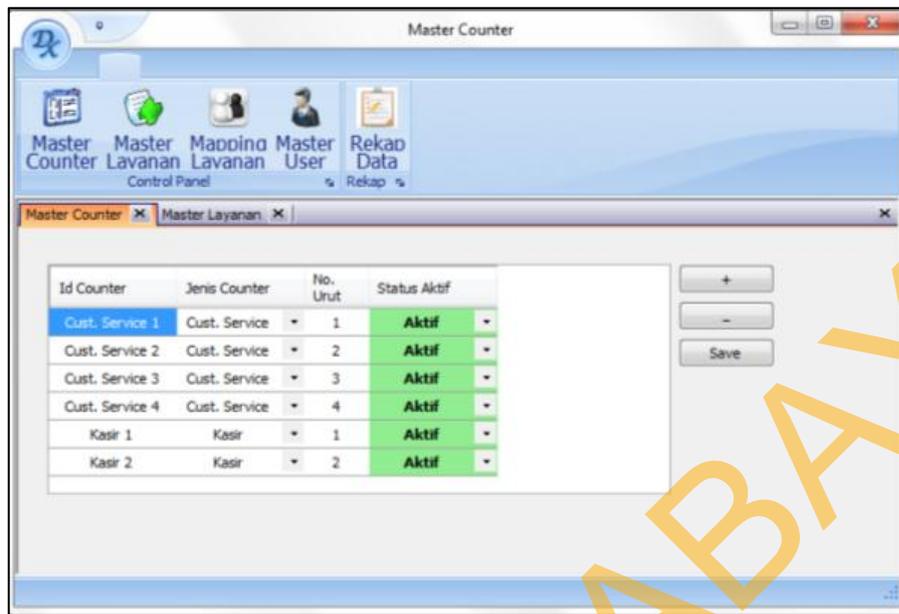


Gambar 4.2 *form* mdiFormAdmin

Terdapat 5 menu yang berupa tombol pada *form* mdiFormAdmin dengan fungsi masing-masing sebagai berikut :

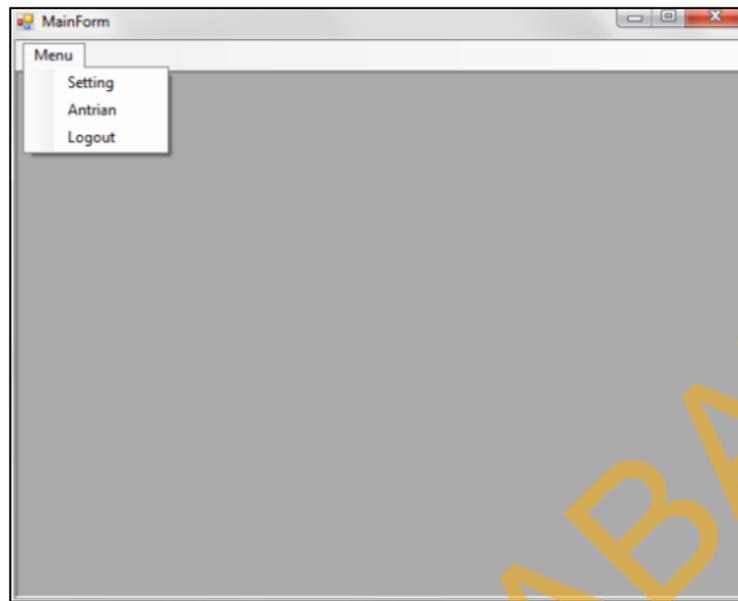
Master *Counter* untuk membuka form *controlCounter* untuk melakukan penambahan, pengurangan atau mengedit *counter*. Master Layanan untuk membuka form Master Layanan, user dapat melakukan *editing* terhadap data layanan yang tersimpan di tabel *masterLayanan*. *Mapping* Layanan untuk memunculkan *form Mapping* Layanan yang digunakan untuk melakukan mapping layanan ke *counter* yang diinginkan. Master *User* yang berfungsi untuk mengatur, menambah dan mengurangi *user*. Dan rekap Data untuk menampilkan rekap data hasil dalam proses melayani *customer*.

1. Diperlukan 6 *counter* layanan dengan 4 CS dan 2 kasir. Kebutuhan ini diselesaikan melalui *form* admin Master *Counter* dengan menginputkan data *counter* baru. Nomor urut disesuaikan oleh *admin*, kemudian disimpan dengan menekan tombol *save*. Hasil yang didapat adalah seperti pada gambar 4.3, data *counter* sudah tersimpan di tabel *master counter* dan dapat dipanggil jika dibutuhkan.



Gambar 4.3 data *counter* yang sudah tersimpan

Perusahaan kini telah memiliki 6 *counter* yang telah terdaftar untuk melayani *customer* yang datang yaitu *customer service* sebanyak 4 *counter* dan kasir untuk pembayaran sebanyak 2 *counter*. Pada bagian *client* kemudian disetting oleh admin untuk memberikan ID pada *counter* seperti yang terlihat pada gambar 4.4 dan gambar 4.5.

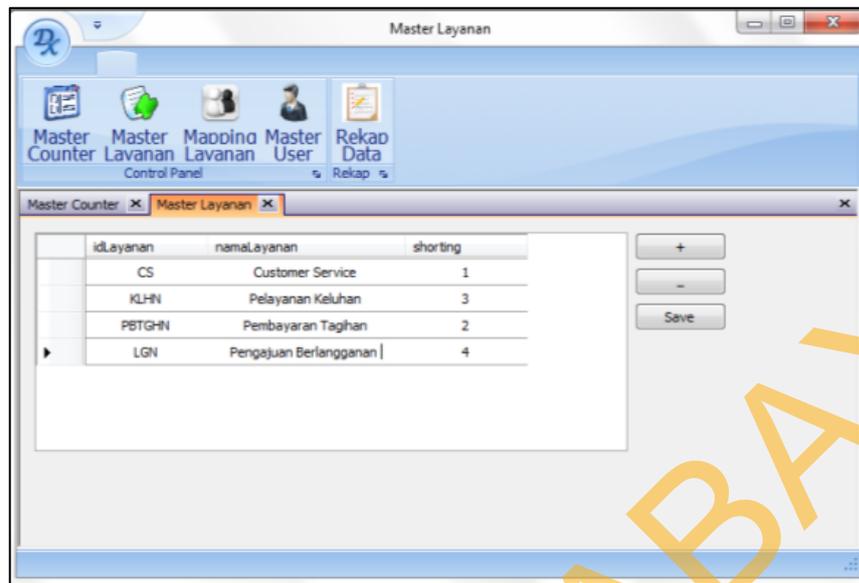


Gambar 4.4 Admin Masuk Menu Setting di Komputer Client



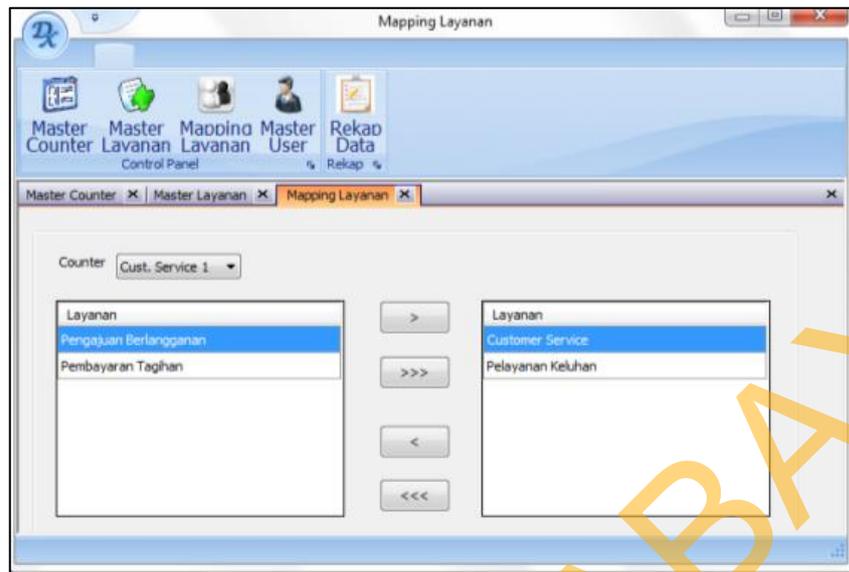
Gambar 4.5 Admin mengatur ID counter tertentu

2. Memiliki 4 layanan *customer service*, pelayanan keluhan, pengajuan berlangganan dan pembayaran tagihan. Kebutuhan ini dapat dipenuhi dengan menambahkan data layanan ke tabel master layanan melalui *form* admin Master Layanan. Hasil yang didapat adalah jenis layanan telah bertambah dan dapat diakses saat dibutuhkan seperti yang terlihat pada gambar 4.6.



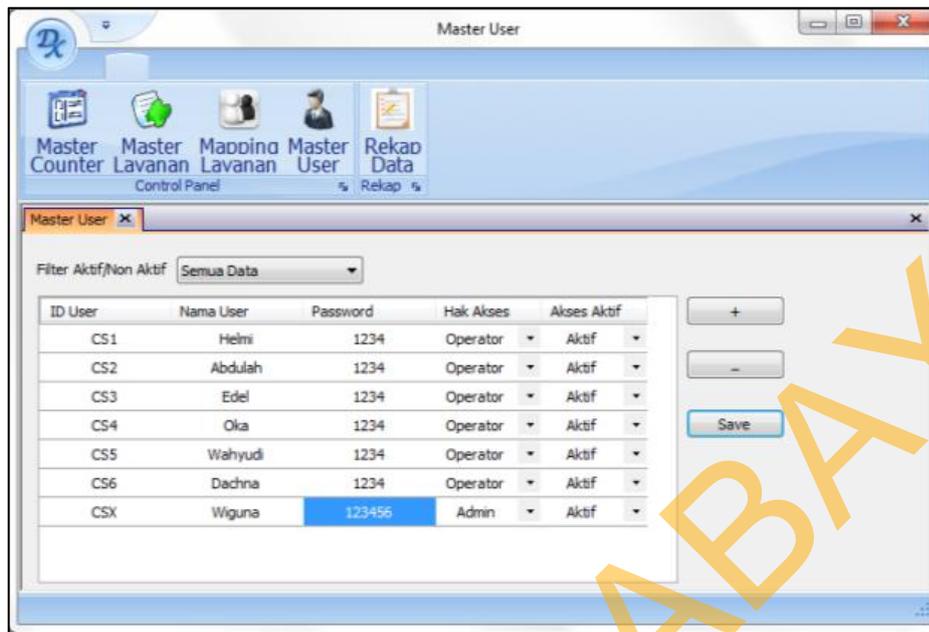
Gambar 4.6 menambahkan data layanan

3. *Mapping* layanan ke *counter*, sebuah counter melayani layanan tertentu saja. Kebutuhan ini dapat dipenuhi melalui form *admin mapping* layanan. *Counter* 1 dan 2 merupakan *customer service* dan pelayanan keluhan. *Counter* 3 dan 4 merupakan *customer service* dan pengajuan berlangganan. *Counter* 5 dan 6 untuk pembayaran tagihan. Kemudian didapatkan hasil layanan yang sudah termapping sesuai dengan kebutuhan. Pada gambar 4.7 ditampilkan hasil mapping yang dilakukan dengan menambahkan layanan ke kolom sebelah kanan dari layanan-layanan yang ada pada kolom sebelah kiri. Terlihat bahwa *counter* 1 melayani *Customer Service* dan Pelayanan Keluhan.



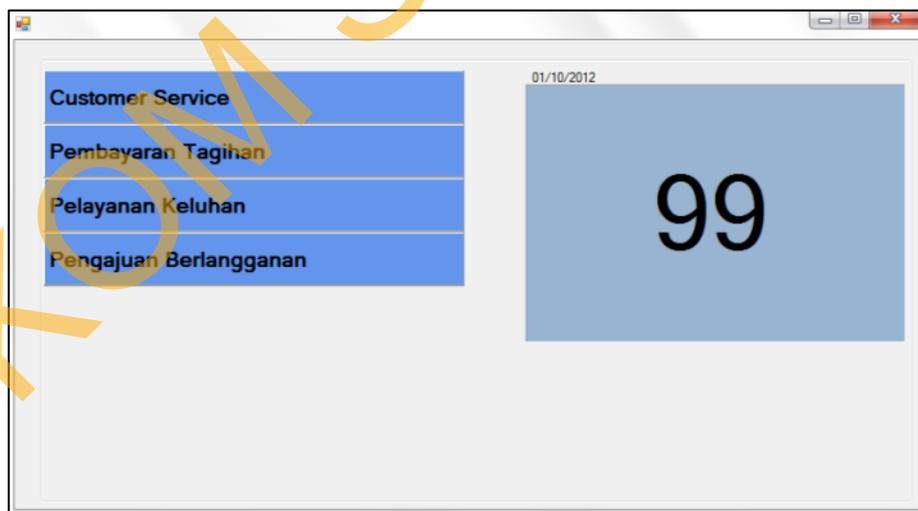
Gambar 4.7 Melakukan *mapping* Layanan

- Perusahaan telah memiliki enam pegawai yang perlu diberi hak akses ke aplikasi. Enam orang karyawan ini membutuhkan *username* dan *password* untuk *login* ke aplikasi. Hasil yang diperoleh adalah *user* yang telah terdaftar tersimpan di tabel *master user* dan memiliki hak akses masing-masing sehingga bisa *login* ke aplikasi. Gambar 4.8 menunjukkan *user* yang telah terdaftar beserta hak aksesnya masing-masing.



Gambar 4.8 User yang didaftarkan

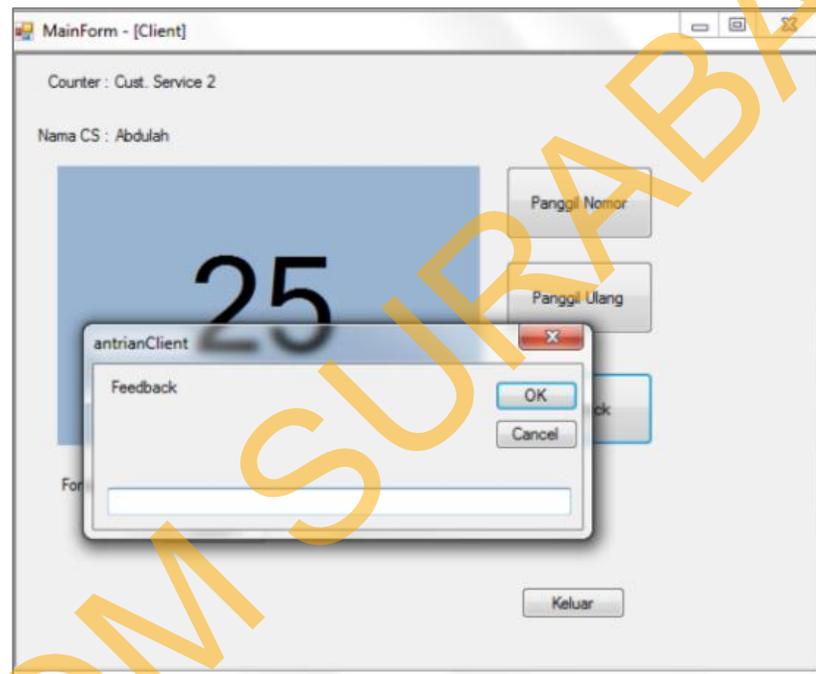
5. Saat aplikasi berjalan, *customer* mengambil nomor antrian pada komputer antrian.



Gambar 4.9 Nomor antrian yang diambil

Gambar 4.9 menunjukkan *form* ambil nomor yang berjalan pada komputer ambil nomor antrian dimana terdapat tampilan layanan-

layanan yang telah dibuat sebelumnya pada *form* master layanan. Nomor antrian kemudian dipanggil oleh *customer service* untuk dilayani. Dari nomor antrian yang sudah dilayani akan memberikan *feedback* berupa nilai 1 untuk puas dan 2 untuk tidak puas dan *customer* dapat di-*forward* jika diperlukan.



Gambar 4.10 *Customer* memeberikan *feedback*

Pada gambar 4.10 menampilkan *form client* saat menerima inputan *feedback* dari *customer* yang telah dilayani. Setelah memeberikan *input*, kemudian akan disimpan ke *database*. *Queue* diimplementasikan ke dalam *datatable* yang merupakan hasil dari proses *retrive* data dari *database* yang kemudian diurutkan berdasarkan nomor antrian yang paling kecil ke antrian yg lebih

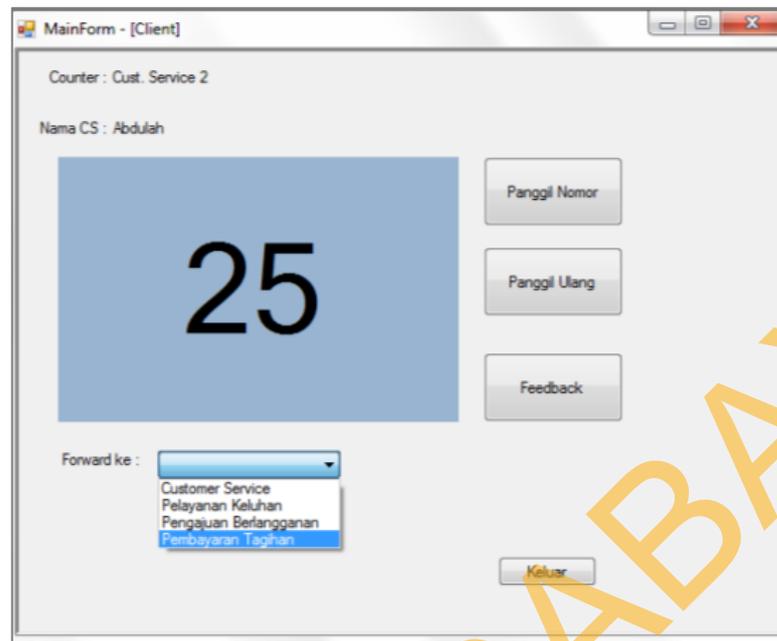
besar. Dimana data nomor yang paling kecil menjadi *front* dalam *queue* dan data yang paling besar menjadi *rear* dalam *queue*. Maka dari itu, nomor yang paling kecil akan dipanggil terlebih dahulu dan keluar dari antrian. Jika ada penambahan antrian, maka otomatis akan masuk di posisi paling belakang.

Nomor antrian yang sudah dipanggil juga akan tampil di *pole display*. Dimana data dikirim dengan komunikasi serial yang kemudian ditampilkan di layar display seperti gambar 4.11.



Gambar 4.11 Penggunaan *pole display*

Jika diperlukan *forwarding*, maka user/customer service dapat melakukan *forwarding* dengan memilih layanan yang dituju pada *combobox* yang disediakan seperti gambar 4.12.



Gambar 4.12 *Customer service* melakukan forwarding

Nomor antrian yang sudah dipanggil akan ditampilkan ke layar display seperti pada gambar 4.12 yang menampilkan nomor antrian, *counter* dan sisa antrian yang menunggu.

Counter	Nomor	Counter	Sisa Antrian
Cust. Service 1	14	Cust. Service	58
Cust. Service 2	25	Kasir	2
Cust. Service 3	21		
Cust. Service 4	35		
Kasir 1	36		
Kasir 2	97		

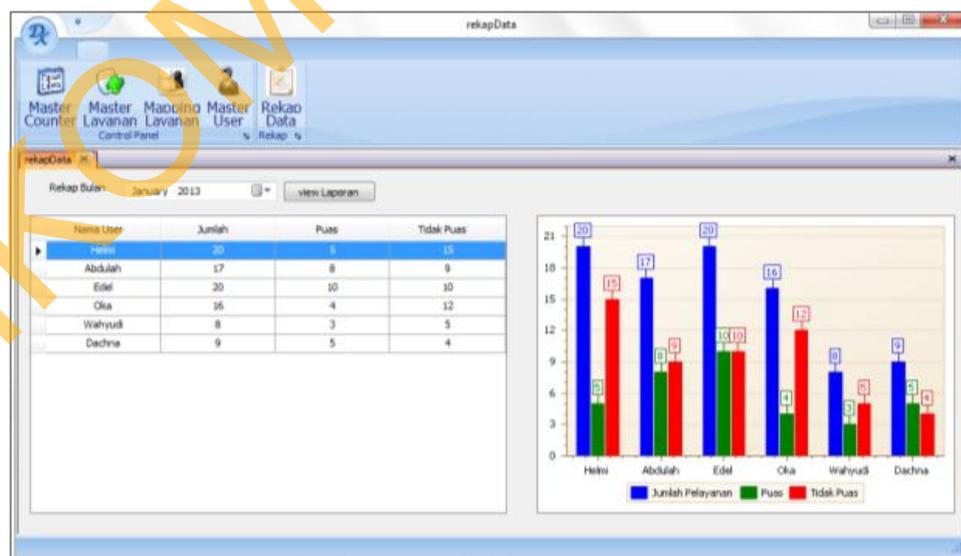
Gambar 4.13 Display nomor antrian

Selanjutnya adalah memastikan aplikasi berjalan normal ketika terjadi gangguan listrik padam. Untuk itu aplikasi dimatikan dan

komputer di-*restart*. Saat komputer aktif kembali aplikasi dijalankan, didapatkan kondisi dimana aplikasi menunjukkan nomor antrian terakhir sebelum komputer dimatikan.

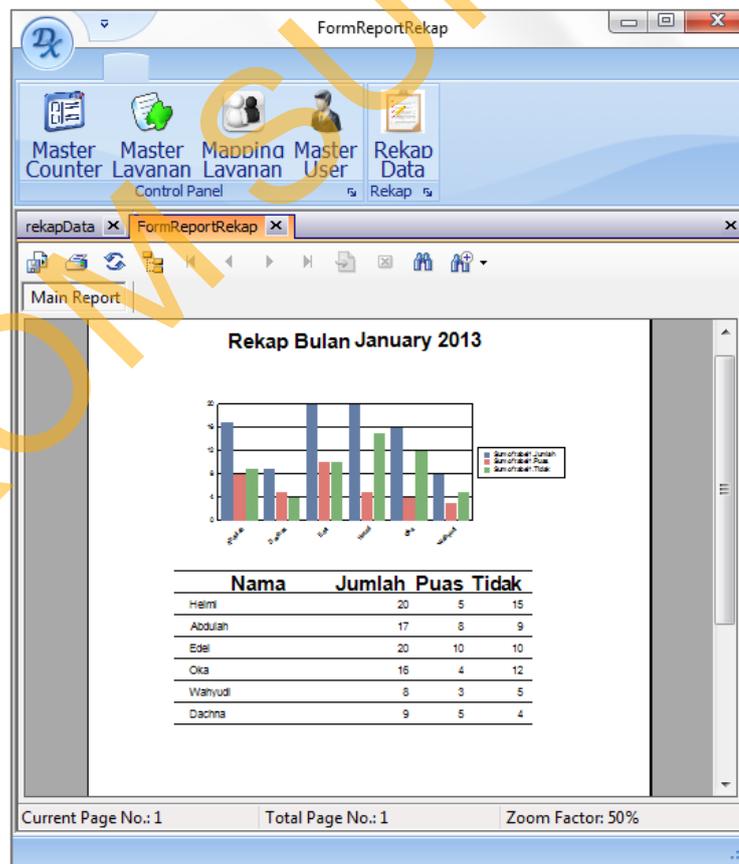
Hal ini teratasi karena *queue* yang berupa *datatable* selalu dipantau dan di-*refresh* setiap detik. Setiap perubahan selalu disimpan ke *database* pada saat itu juga sehingga mencegah kekacauan nomor antrian saat terjadi gangguan listrik.

6. Pada akhir bulan, pimpinan memeinginkan data kepuasan pelanggan sebagai penilaian terhadap kinerja pegawainya. Ini dapat diselesaikan melalui *form* admin oleh admin dengan membuka *form* rekap data. Dari rekap data tersebut kemudian dibuat *report* yang bisa dicetak sebagai laporan ke pimpinan.



Gambar 4.14 Rekap data

Gambar 4.14 menunjukkan rekap data berupa nama cs, jumlah *customer* yang dilayani, jumlah *customer* yang menyatakan puas dan jumlah *customer* yang menyatakan tidak puas. Data ditampilkan berupa tabel dan grafik yang akan dicetak dan diserahkan ke manajemen sebagai laporan yang ditampilkan berdasarkan bulan tertentu. Untuk mendapat laporan yang bisa dicetak dapat dilakukan dengan menekan tombol *view* laporan sehingga muncul *report* yang dapat dicetak seperti pada gambar 4.15.

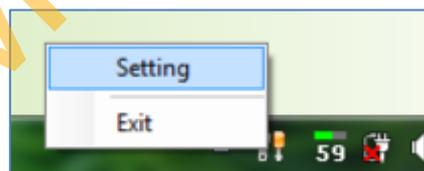


Gambar 4.15 *Report* pelayanan

Menambahkan logo perusahaan pada aplikasi. Penggunaan logo perusahaan membuat aplikasi terlihat mengusung nama perusahaan yang menyediakan layanan. Logo perusahaan dapat ditambahkan pada aplikasi dengan menggunakan *file* gambar berukuran tinggi max 100px dengan lebar menyesuaikan kebutuhan.

Merubah logo pada layar display nomor antrian dapat dilakukan dengan cara berikut :

1. Klik kanan pada *icon* aplikasi antrian pada *tray icon* seperti gambar 4.16 (a).
2. Pilih menu *setting*, kemudian muncul form *Set Logo*. Gunakan tombol *browse* untuk mengambil logo 4.14(b). Kemudian *apply*.



(a)



(b)

Counter	Nomor
Cust. Service 1	0
Cust. Service 2	0
Cust. Service 3	0
Cust. Service 4	0
Kasir 1	0
Kasir 2	0

(c)

Gambar 4.16 menambahkan logo nomor antrian

Dari pengujian yang telah dilakukan, telah didapatkan hasil pengujian seperti pada tabel 4.3. Dalam pengujian, semua fungsi aplikasi berjalan dengan baik sesuai dengan yang diharapkan. Aplikasi dapat melakukan tugas-tugas melayani antrian sesuai dengan tujuan yang diharapkan.

Tabel 4.3 hasil pengujian aplikasi

Test Case ID	Tes yang dilakukan	Pengujian										Berhasil	Gagal
		1	2	3	4	5	6	7	8	9	10		
1	a. Input data <i>counter</i>	V	V	V	V	V	V	V	V	V	V	10	-
	b. Menghapus data <i>counter</i>	V	V	V	V	V	V	V	V	V	V	10	-
	c. Mengedit data <i>counter</i>	V	V	V	V	V	V	V	V	V	V	10	-
2	a. Input data layanan	V	V	V	V	V	V	V	V	V	V	10	-
	b. Menghapus data layanan	V	V	V	V	V	V	V	V	V	V	10	-
	c. Mengedit data layanan	V	V	V	V	V	V	V	V	V	V	10	-
3	a. Menambahkan layanan ke counter	V	V	V	V	V	V	V	V	V	V	10	-
	b. Menghapus layanan dari counter	V	V	V	V	V	V	V	V	V	V	10	-
4	a. Input data <i>user</i>	V	V	V	V	V	V	V	V	V	V	10	-
	b. Menghapus data <i>user</i>	V	V	V	V	V	V	V	V	V	V	10	-
	c. Mengedit data <i>user</i>	V	V	V	V	V	V	V	V	V	V	10	-
5	a. Melakukan panggilan nomor antrian	V	V	V	V	V	V	V	V	V	V	10	-
	b. Melakukan forwarding antrian	V	V	V	V	V	V	V	V	V	V	10	-
	c. Menerima input feedback	V	V	V	V	V	V	V	V	V	V	10	-
	d. Menguji handling saat listrik mati (menutup dan me-restart komputer).	X	X	V	V	V	V	V	V	V	V	8	2
6	a. Membuat rekap	V	V	V	V	V	V	V	V	V	V	10	-
	b. Memilih rekap berdasarkan bulan tertentu	V	V	V	V	V	V	V	V	V	V	10	-

Keterangan :

V = Berhasil

X = Gagal

Tabel uji menampilkan hasil uji dengan menggunakan sebuah contoh kasus dan parameter pengujian pada tabel 4.2 yang meliputi uji *input data*, menghapus dan mengedit data *counter*, layanan serta data *user* yang memiliki hak akses ke aplikasi. Aplikasi mampu melakukan penambahan, pengurangan dan perubahan jumlah dan jenis layanan.

Simulasi dengan *counter* yang berbeda-beda telah dilakukan dengan melakukan proses pemanggilan nomor antrian berdasarkan jenis layanan yang dilakukan melalui jaringan komputer dengan tiga buah komputer dimana sebuah komputer bertindak sebagai *server* dan dua komputer lain bertindak sebagai *client*.

Pada pengujian dengan simulasi *client-server*, telah dilakukan percobaan dengan mematikan semua program, merestart komputer dan menghidupkan kembali komputer dan menjalankan aplikasi, didapatkan aplikasi mampu melakukan *handling* terhadap gangguan listrik padam. Pada percobaan pertama dan kedua masih terdapat *bug*. Setelah *bug* teratasi, kemudian pada percobaan ketiga dan seterusnya aplikasi mampu menangani permasalahan listrik mati sesuai dengan yang diharapkan dan tidak mengacaukan nomor antrian. Aplikasi juga telah mampu melakukan *forwarding* nomor antrian ke layanan lain tanpa perlu mengambil nomor antrian lagi.