
RANCANG BANGUN APLIKASI ANTRIAN LOKET UNTUK MENINGKATKAN KETERTIBAN PELAYANAN PELANGGAN (STUDI KASUS: KANTOR PDAM BEKASI)

Syamsul Afandi ¹, Didin Samsudin Adhuri ²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Attahiriyah
Jl. Kampung Melayu Kecil III No. 15, Tebet, Jakarta Selatan 12840
Email: syamsul08490@gmail.com¹, didin.samsudin59@gmail.com²

Abstrak

Pelayanan Perusahaan Daerah Air Minum (PDAM) terhadap masyarakat Bekasi dinilai kurang memuaskan terutama dari sektor sistem pelayanan dalam pembayaran. Pelanggan diminta untuk mengantri atau berbaris satu-persatu untuk menunggu mendapatkan layanan. Sering juga terjadi ketidakteraturan yang membuat petugas loket antrian akan kesulitan dalam menentukan orang pertama yang akan dilayani. Melalui program Visual Basic 2010 Express dapat membuat sistem aplikasi otomatis melalui komputer sebagai sarana kontrol. Aplikasi yang dibuat yaitu berupa sistem antrian. Untuk mencapai tujuan kegiatan seperti yang dikemukakan diatas dan memenuhi kaidah-kaidah melakukan riset yang baik, maka dilakukan beberapa langkah metode yaitu : Metode Analisis, *questionnaire*, *survey*, wawancara, dan metode selanjutnya adalah metode Perancangan yang dilakukan dalam pembuatan aplikasi ini meliputi: perancangan sistem dengan menggunakan pendekatan *Unified Modeling Language* (UML), seperti pembuatan *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram* dan Perancangan *interface*. Adapun tujuan dari penelitian ini adalah membuat sistem perangkat lunak digunakan untuk mengatur atau mengontrol pengambilan dan pemanggilan nomor antrian pada loket. Pembuatan aplikasi ini diharapkan dapat memberikan kemudahan aktivitas dan efektifitas petugas loket antrian dalam melakukan pelayanan transaksi di kantor PDAM sehingga antrian dapat berjalan tertib.

Kata kunci : *Aplikasi Antrian Loket, Visual Basic 2010.*

1. PENDAHULUAN

1.1. Latar Belakang

Teknologi pemrograman melalui komputer pada masa sekarang ini mengalami perkembangan yang pesat. Hal ini menunjukkan suatu kecenderungan yang mengarah kepada penciptaan maupun pengembangan suatu sarana teknologi otomatis yang lebih praktis, efisien dan ekonomis untuk mempermudah dan mempercepat segala aktivitas manusia. Peralatan otomatis banyak menggunakan komputer sebagai sarana kontrol. Sistem otomatis dengan bantuan komputer tidak akan berjalan apabila tidak ada pemrograman atau perangkat lunak yang berfungsi sebagai perintah kerja pada komputer. Visual Basic merupakan salah satu bahasa pemrograman yang bekerja di bawah lingkup sistem operasi *Windows* yang dapat digunakan untuk membuat sistem aplikasi otomatis melalui komputer. Saat ini beberapa sistem loket antrian masih menggunakan sistem manual. Salah satunya adalah pelayanan pembayaran rekening pada kantor Perusahaan Daerah Air Minum (PDAM) cabang Bekasi. Hal tersebut dapat membuat petugas loket antrian mengalami kesulitan saat melayani orang di depan loket antrian dalam jumlah banyak. Sekarang ini budaya tidak tertib masih ada pada masyarakat Indonesia, saat aktivitas loket antrian dilakukan secara manual orang yang melakukan antrian tidak ada aturan, sehingga setiap orang yang sedang melakukan antrian saling menyerobot ingin cepat dilayani, maka membuat petugas loket antrian akan kesulitan dalam menentukan orang pertama yang akan dilayani dalam antrian.

Berbeda dengan aktivitas loket antrian yang dilakukan secara otomatis. Orang yang sedang melakukan antrian harus berjalan sesuai teori antrian, teori antrian adalah teori yang mencakup studi matematis dari antrian-antrian atau baris-baris penunngguan, formasi baris-baris penunngguan ini tentu saja merupakan suatu fenomena yang lagi terjadi apabila kebutuhan akan suatu pelayanan melebihi kapasitas yang tersedia untuk menyelenggarakan pelayanan itu, keputusan-keputusan yang berkenaan

dengan jumlah kapasitas ini harus dapat dibuat suatu prediksi yang tepat mengenai kapan unit-unit yang membutuhkan dengan aturan, karena untuk melakukan antrian harus mempunyai nomor urut antrian terlebih dahulu.

Sistem antrian yang menggunakan program Visual Basic sebagai pembuat perintah pada komputer digunakan untuk mengontrol sistem antrian agar lebih efisien dan efektif. Selanjutnya komputer tersebut menghasilkan nomor urut antrian, bertujuan agar saat orang-orang melakukan antrian dapat berjalan dengan tertib, sesuai dengan kertas nomor urutan yang didapatkan setelah menekan tombol print out/cetak. Orang yang telah mengambil nomor antrian dari printer tersebut tinggal menunggu pemanggilan nomor yang dimiliki untuk melakukan transaksi pada loket antrian.

1.2. Rumusan Masalah

Adapun identifikasi masalah yang ada adalah

1. Bagaimana membuat rancang bangun aplikasi antrian loket untuk mengendalikan sistem antrian dengan menggunakan visual basic 2010 ?
2. Apakah aplikasi antrian loket ini dapat mempermudah petugas loket dalam melayani pelanggan ?

2. METODOLOGI

Metodologi pada penelitian ini menggunakan metode perancangan sistem dengan menggunakan pendekatan analisis berorientasi obyek dengan *Unified Modeling Language (UML)*, seperti pembuatan *Use Case Diagram, Activity Diagram, Class Diagram, Sequence Diagram* dan Perancangan *interface*.

3. HASIL DAN PEMBAHASAN

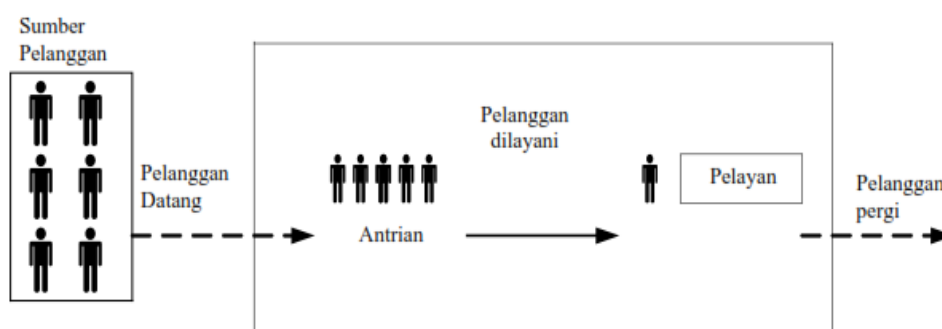
3.1. Teori Antrian

Teori Antrian merupakan studi matematika dari antrian atau kejadian garis tunggu, yakni suatu garis tunggu dari pelanggan yang memerlukan layanan dari sistem pelayanan yang ada (Kakiay, 2004).

Ada empat model struktur antrian dasar yang umum terjadi dalam seluruh sistem antrian (Bronson, 1998):

1. Single Chanel - Single Phase

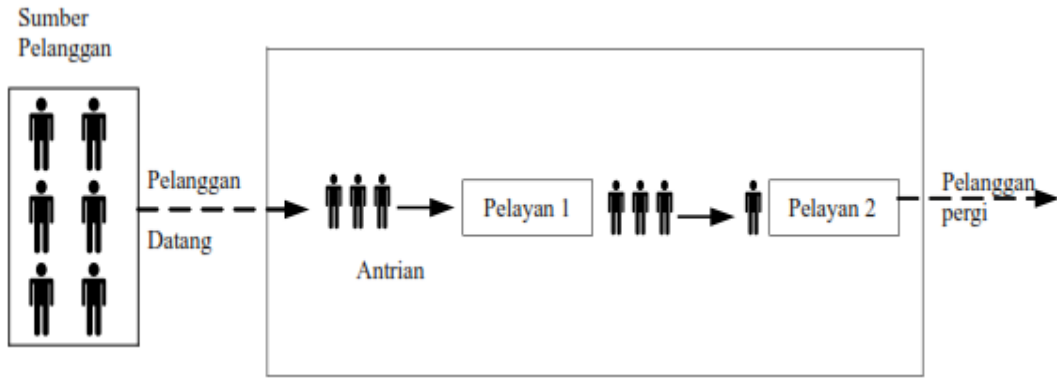
Single Chanel berarti bahwa hanya ada satu jalur untuk memasuki sistem pelayanan atau ada satu pelayanan. *Single phase* menunjukkan bahwa hanya ada satu stasiun pelayanan sehingga yang telah menerima pelayanan dapat langsung keluar dari sistem antrian. Contohnya adalah pada pembelian tiket bus yang dilayani oleh satu loket, seorang pelayan toko dan lain-lain.



Gambar 1. *Single Chanel - Single Phase*

2. Single Chanel - Multi Phase

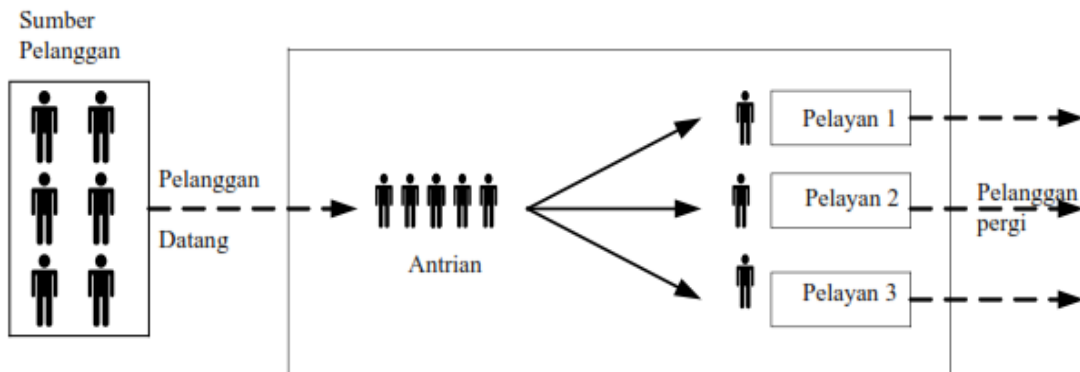
Multi phase berarti ada dua atau lebih pelayanan yang dilaksanakan secara berurutan dalam *phase*. Misalnya proses pencucian mobil.



Gambar 2. Single Chanel - Multi Phase

3. Multi Chanel - Single Phase

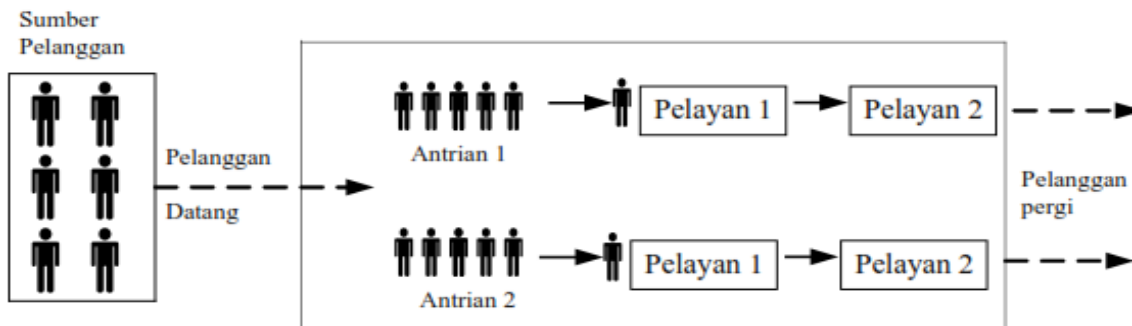
Sistem *multi chanel- single phase* terjadi jika ada dua atau lebih fasilitas pelayanan dialiri oleh suatu antrian tunggal. Sebagai contoh adalah pada pembelian tiket yang dilayani oleh lebih dari satu loket, pelayanan nasabah di Bank, dan lain-lain.



Gambar 3. Multi Chanel - Single Phase

4. Multi Chanel - Multi Phase

Sistem ini terjadi jika ada dua atau lebih fasilitas pelayanan dengan pelayanan pada lebih dari satu phase. Sebagai contoh adalah pada pelayanan kepada pasien di rumah sakit dari pendaftaran, diagnosa, tindakan medis sampai pembayaran. Setiap sistem-sistem ini mempunyai beberapa fasilitas pelayanan pada setiap tahap, sehingga lebih dari satu individu dapat dilayani pada suatu waktu.



Gambar 4. Multi Chanel - Multi Phase

3.2. Terminologi Dan Notasi

Terminologi dan notasi yang digunakan di sistem sebagai berikut :

1. Keadaan sistem adalah banyaknya pelanggan pada sistem
2. Panjang antrian adalah jumlah pelanggan yang menunggu pelayanan

Notasi yang digunakan adalah sebagai berikut :

- n = jumlah pelanggan dalam sistem antrian
- $P_n(t)$ = probabilitas bahwa tepat n pelanggan dalam sistem antrian pada saat (t)
- s = jumlah pelayanan pada sistem antrian
- λ = rata-rata kedatangan dalam satuan waktu
- μ = rata-rata pelayanan dalam satuan waktu
- ρ = probabilitas masa sibuk
- L_s = rata-rata jumlah pelanggan dalam sistem
- L_q = rata-rata jumlah pelanggan dalam antrian
- W_s = rata-rata waktu yang dihabiskan pelanggan dalam sistem
- W_q = rata-rata waktu yang dihabiskan pelanggan dalam antrian

Untuk kemudahan dalam memahami karakteristik suatu sistem antrian digunakan notasi Kendall Lee, notasi standar yang digunakan ditulis: $(a / b / c) ; (d / e / f)$

Notasi tersebut adalah unsur-unsur dasar dari model antrian sebagai berikut :

- a = distribusi kedatangan
- b = distribusi pelayanan
- c = jumlah fasilitas pelayanan $(s = 1, 2, \dots, \infty)$
- d = disiplin pelayanan
- e = jumlah pelanggan maksimum dalam sistem
- f = ukuran sumber pemanggilan

Notasi a dan b untuk distribusi kedatangan dan keberangkatan mempunyai kode sebagai berikut :

- M = Distribusi kedatangan Poisson atau distribusi pelayanan Eksponensial
- D = Waktu pelayanan konstan
- E_k = Distribusi waktu pelayanan Erlang dengan parameter k.

3.2.1. Rumus Perhitungan

Dalam melakukan perhitungan, penulis mengambil acuan dengan rumus yang digunakan antara lain sebagai berikut :

1. Menentukan probabilitas masa sibuk

Ketika λ menyatakan tingkat rata-rata kedatangan per satuan waktu dan μ menyatakan tingkat rata-rata pelayanan per satuan waktu di mana $\lambda > \mu$ menyertai sebagai asumsi maka tingkat kesibukan sistem dapat dinyatakan :

$$\rho = \frac{\lambda}{\mu}$$

2. Menentukan probabilitas semua pelayan menganggur

Tingkat kesibukan sistem paling sibuk adalah 100 % dan jika tingkat kedatangan λ dan semakin kecil pada tingkat pelayanan μ yang tidak berubah maka tingkat kesibukan akan menurun. Dengan demikian, peluang sistem yang sedang kosong sangat tergantung pada penggunaan fasilitas pelayanannya. Secara matematik dituliskan :

$$P_0 = \frac{\lambda}{\rho}$$

Secara umum P_0 merupakan peluang waktu menganggur berlaku untuk semua sistem pelayanan. Bila seorang yang berada dalam sistem, maka suatu pelayanan akan sibuk maka dinyatakan dengan rumus:

$$P_n = \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{\mu}\right)$$

3. Rata-rata jumlah pelanggan dalam antrian

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

4. Rata-rata jumlah pelanggan dalam system

$$L_s = \frac{\lambda}{\mu - \lambda}$$

5. Rata-rata waktu menunggu pelanggan dalam antrian

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)}$$

6. Rata-rata waktu pelanggan menunggu dalam system

$$W_s = \frac{1}{\mu - \lambda}$$

3.3. Analisis Berorientasi Obyek

Pengertian *Object-Oriented Analysis* adalah sebagai berikut :

“*Object-Oriented Analysis* adalah metode analisis yang memeriksa *requirements* (syarat / keperluan yang harus dipenuhi oleh suatu sistem) dari sudut pandang kelas-kelas dan obyek-obyek yang ditemui dalam ruang lingkup permasalahan.”

Metodologi adalah cara sistematis untuk mengerjakan pekerjaan analisis dan desain. Demikian pula didefinisikan oleh A. Suhendar dan Hariman G. “Metodologi Berorientasi Obyek adalah metode penyelesaian masalah dengan menggunakan pendekatan berorientasi obyek” Metodologi pengembangan sistem berorientasi obyek mempunyai tujuh karakteristik utama, yaitu :

1. *Abstraction*

Memfokuskan pada apakah objek dan apa yang dikerjakan oleh objek tersebut.

2. *Inheritance*

Sarana untuk menghilangkan penulisan ulang terhadap kode yang dapat digunakan berulang kali, yang dapat menyiaikan banyak waktu, ketidakkonsistenan dan meningkatkan resiko menimbulkan kesalahan.

3. *Polymorphisme*

Mendefinisikan ulang atribut-atribut yang bernama sama pada kelas-kelas yang berbeda.

4. *Encapsulation*

Pemisahan aspek-aspek eksternal objek, yang dapat diakses oleh objek-objek lain.

5. *Message Sending*

Cara untuk berhubungan antara suatu objek dengan objek lain.

6. *Association*

Hubungan relasi antar objek.

7. *Aggregation*

Bentuk khusus dari asosiasi.

Unified Modelling Language (UML) adalah salah satu alat bantu yang sangat handal dalam dunia pengembangan sistem berorientasi obyek karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka ke bentuk baku, mudah dipahami lengkap dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain. (Ladjamudin, 2006)

Unified Modelling Language (UML) mendefinisikan diagram-diagram sebagai berikut:

1. *Usecase diagram*

Use case merupakan salah satu metode dalam analisis dan desain sistem berorientasi objek (*Object Oriented Analysis and Design*). *Use case* juga merupakan bagian dari UML (*Unified Modelling Language*). *Use case modelling* digunakan untuk mendokumentasikan *system behaviour* dan *subsystem* pada saat pengembangan sistem, termasuk di dalamnya fungsi internal suatu sistem (*use case*), pengguna sistem (*user*) dan hubungan interaksi antara keduanya (*use case diagram*).

2. *Class diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

3. *Component diagram*

Component Diagram merupakan gambaran aspek fisik sistem berbasis objek dengan menunjukkan hubungan dan ketergantungan dalam serangkaian komponen. Menggambarkan komponen fisik *software* termasuk *source code*, *run time (binary) code*, *executable file*, *table*, *library*, dan dokumen. Meliputi komponen, *interface*, *dependency*, *generalization*, *association*, *realization*, *notes*, *constraint*, *packages*, *subsystem* dari sebuah model.

4. *Deployment diagram*

Deployment diagram menggambarkan sumber fisik dalam sistem, termasuk node, komponen dan koneksi (model implementasi sistem yang statistik). Dalam hal ini meliputi topologi *hardware* yang dipakai sistem.

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

5. *Statechart diagram*

Menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

6. *Sequence diagram*

Sequence diagram merupakan diagram yang menggambarkan pola hubungan diantara sekumpulan objek yang saling mempengaruhi menurut urutan waktu. Sebuah objek berinteraksi dengan objek lain melalui pengiriman pesan (*messages*). *Sequence diagram* biasanya digunakan untuk mengilustrasikan sebuah *use case*.

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

3.4. Pengenalan Visual Basic

Visual Basic adalah salah satu *development tools* untuk membangun aplikasi dalam lingkungan *Windows*. Dalam pengembangan aplikasi, Visual Basic menggunakan pendekatan Visual untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk kode sumbernya menggunakan dialek bahasa Basic yang cenderung mudah dipelajari. Visual Basic telah menjadi *tools* yang terkenal bagi para pemula maupun para developer dalam pengembangan aplikasi skala kecil sampai ke skala besar (Andi, 2010).

3.5. Analisis Antrian Locket

Dalam sistem ini dioperasikan satu buah loket dengan satu operator. Rata-rata tingkat kedatangan pelanggan mengikuti distribusi *poisson* yaitu 20 orang per jam. Operator dapat melayani rata-rata 25 orang per jam, dengan waktu pelayanan setiap orang mengikuti distribusi probabilitas eksponensial. Jika diasumsikan model sistem antrian yang digunakan operator tersebut adalah (M/M/1), maka dapat dihitung :

1. Tingkat intensitas (kegunaan) pelayanan (ρ)
2. Jumlah rata-rata pelanggan yang diharapkan dalam sistem
3. Jumlah pelanggan yang diharapkan menunggu dalam antrian.
4. Waktu pelanggan selama menunggu dalam sistem (menunggu pelayanan)
5. Waktu yang diharapkan oleh setiap pelanggan untuk menunggu dalam antrian

Dalam tabel digambarkan sebagai berikut

Tabel 1. Jumlah kedatangan

No	Waktu	Kedatangan
1	08.00-9.00	32
2	09.00-10.00	34
3	10.00-11.00	25
4	11.00-12.00	18
5	Istirahat	0
6	13.00-14.00	29
7	14.00-15.00	12
Σ	Jumlah	150
(λ)	Rata-rata	25

Tabel 2. Pelayanan Antrian

No	Waktu	Pelayanan
1	08.00-9.00	26
2	09.00-10.00	19
3	10.00-11.00	17
4	11.00-12.00	18
5	Istirahat	0
6	13.00-14.00	22
7	14.00-15.00	18
Σ	Jumlah	120
(μ)	Rata-rata	20

Diketahui

- $\lambda = 20$
- $\mu = 25$
- $S = 1$

- a. Tingkat intensitas (kegunaan) pelayanan (p). $p = 0,80$
Angka tersebut menunjukkan bahwa *teller* akan melayani selama 80% dari waktu selama bekerja
- b. Jumlah rata-rata pelanggan yang diharapkan dalam system (L). $L = 24$
Jumlah tersebut merupakan jumlah yang diharapkan dalam sistem yaitu 24 pelanggan dalam 1 jam
- c. Jumlah pelanggan yang diharapkan menunggu dalam antrian (Lq). $Lq = 3,2$
Angka tersebut menunjukkan jumlah pelanggan yang menunggu di dalam sistem
- d. Waktu pelanggan selama menunggu dalam sistem (W). $W = 0,20$ jam atau 12 menit
Angka tersebut menunjukkan rata-rata waktu menunggu di dalam *system*.
- e. Waktu yang diharapkan oleh setiap pelanggan untuk menunggu dalam antrian. $Wq = 0,6$ jam atau 9,6 menit.

3.6. Rancangan Sistem

1. Perancangan Use Case Diagram

Dalam perancangan *use case* diagram bertujuan untuk menspesifikasikan sistem dari aksi-aksi yang diharapkan oleh calon pengguna sistem yang akan diusulkan dan diterapkan dalam sistem tersebut.

a) Nama *use case* : **Pengambilan Slip Nomor Antrian.**

Pelanggan sebelum melakukan transaksi dihadapkan terlebih dahulu untuk mengambil karcis antrian, langkah-langkahnya adalah :

- 1) Pelanggan menuju tempat pengambilan karcis nomor antrian yang sudah disiapkan
- 2) Menekan tombol pengambilan karcis nomor antrian
- 3) Karcis nomor antrian keluar
- 4) Karcis nomor antrian disimpan

b) Nama *use case* : **Menunggu.**

Pelanggan mengantri tetapi dalam kondisi bisa relaksasi sedikit karena untuk mengantri tidak lagi dengan berdiri tetapi bisa sambil duduk di kursi yang sudah disiapkan di ruang tunggu antrian. Adapun yang perlu diperhatikan ketika mengantri adalah :

- 1) Menunggu pemanggilan nomor antrian dan suara panggilan nomor antrian yang ditampilkan pada layar monitor serta *speaker*.
- 2) Pelanggan yang merasa nomor antriannya di panggil segera menuju ke petugas *teller* yang di sebutkan.

c) Nama *use case* : **Panggilan.**

Untuk dimulainya suatu transaksi petugas teller menekan tombol pemanggilan pelanggan yang sudah disiapkan di tempat kerja *teller*, adapun langkah-langkahnya adalah:

- 1) Petugas teller yang sudah siap melayani pelanggan yang akan bertransaksi, diharuskan menekan tombol pemanggilan nomor antrian pelanggan yang sudah disiapkan.
- 2) Proses kondisi pemanggilan nomor antrian nasabah jika petugas *teller* sudah selesai melayani transaksi pelanggan dan adanya suatu pelanggan yang akan melakukan transaksi.

d) Nama *use case* : **Transaksi Dimulai.**

Pada mulainya suatu transaksi yaitu ketika petugas *teller* telah menekan tombol pemanggilan nomor antrian pelanggan dan pelanggan mengunjungi serta petugas *teller* telah siap melayani transaksi.

e) Nama *use case* : **Transaksi Selesai.**

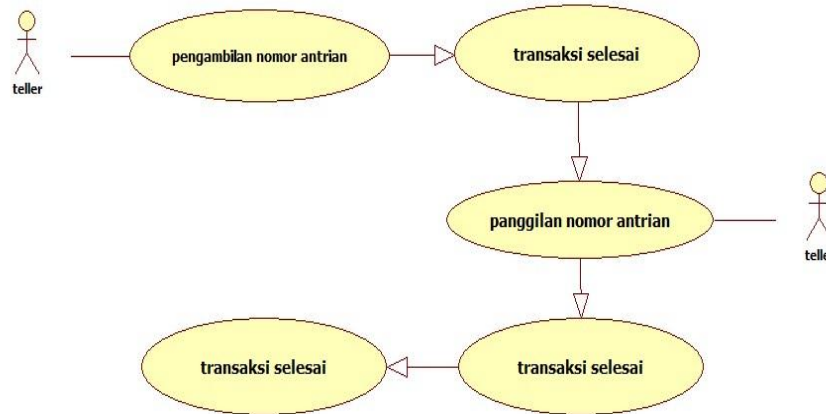
Transaksi pelanggan telah selesai jika transaksi pelanggan sudah selesai dilayani oleh petugas teller setempat dan petugas *teller* mengakhiri pelayanannya.

f) Nama *use case* : **Sistem Kontrol.**

Pada pengontrolan sistem dilakukan oleh seorang administrator yang tugasnya antara lain :

- 1) Menghidupkan dan mematikan komputer.
- 2) Menjalankan program yang telah dibuat.

Setelah merancang semua *use case*, maka dapat digambarkan dengan *use case diagram*.

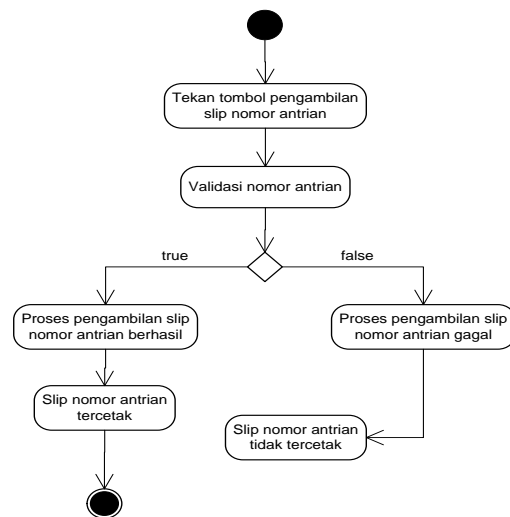


Gambar 5. Use Case Diagram

2. Perancangan Activity Diagram Yang Dilakukan

a. Perancangan Activity Diagram Pengambilan Slip Nomor Antrian

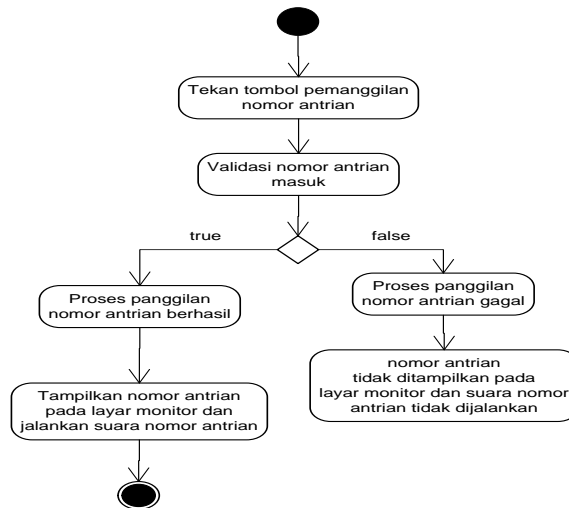
Untuk memulai suatu transaksi pelanggan diwajibkan mengambil slip nomor antrian, tinggal menekan tombol maka di proses oleh program sistem antrian jika berhasil maka slip akan keluar lewat *printer* jika gagal maka slip nomor antrian tidak tercetak.



Gambar 6. Diagram pengambilan nomor antrian

b. Perancangan Activity Diagram Pemanggilan Nomor Antrian

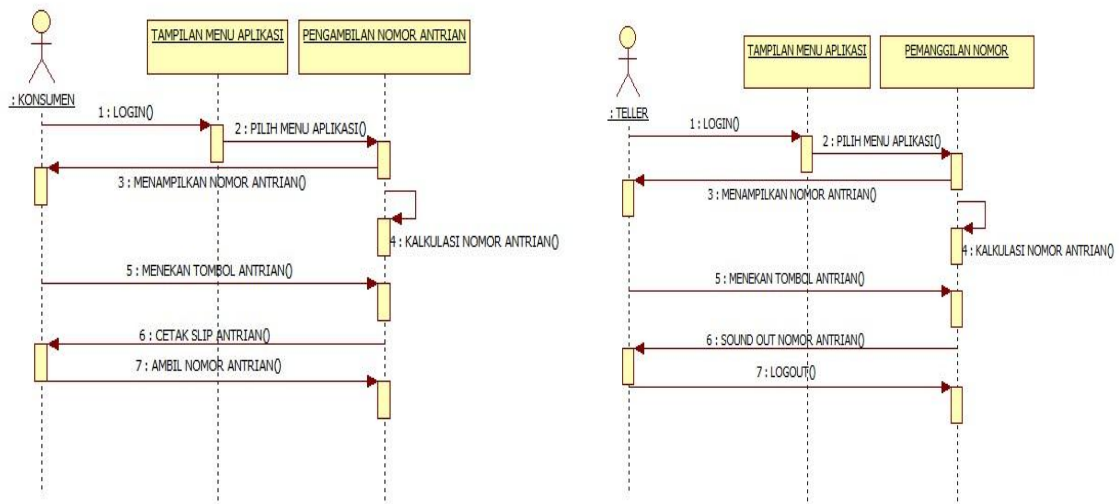
Pada proses pemanggilan nomor antrian dilakukan jika, ada pelanggan yang masuk dan nomor antrian tidak kurang dari batas yang ditentukan maka proses pemanggilan berhasil tetapi jika tidak, maka gagal.



Gambar 7. Diagram pengambilan nomor antrian

c. Squance Diagram

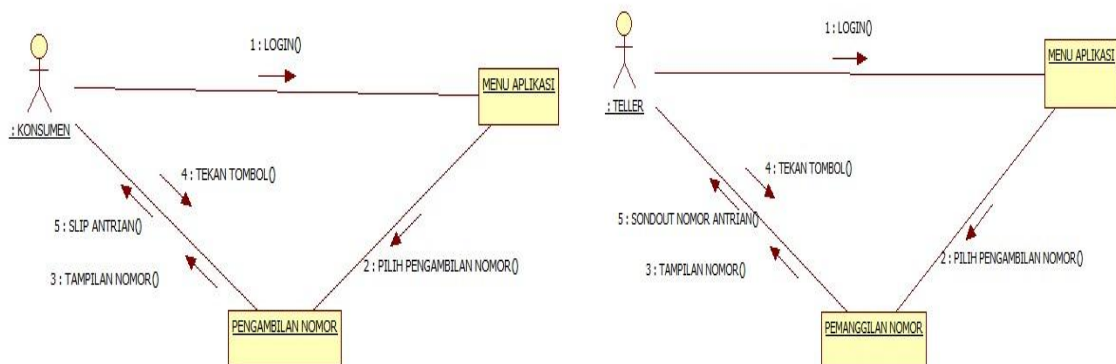
Perancangan *Sequence Diagram* Pengambilan Nomor Antrian



Gambar 8. Pengambilan dan pemanggilan Nomor

d. Perancangan Collaboratiion Diagram

Collaboration Pengambilan Nomor

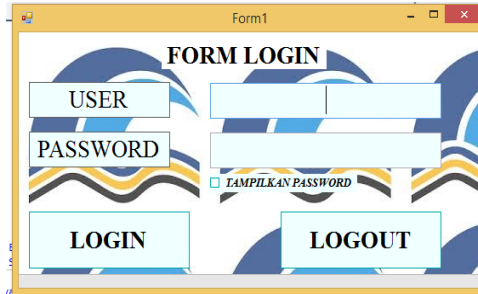


Gambar 9. Collaboration Pengambilan dan pemanggilan Nomor

3.7. Perancangan Tampilan

1. Tampilan Login

Form Login merupakan tampilan pertama kali muncul ketika program aplikasi antrian pertama kali dijalankan. Aplikasi *Login* ini berfungsi untuk pengamanan setiap aplikasi berikut adalah tampilan *Form Login* :



Gambar10. Form Login

Di dalam *Form* ini ada *User* dan *Password* yang harus diisi. Apabila proses *login* selesai ada dua kemungkinan yaitu proses *login* berhasil dan poses *login* gagal, dan akan muncul *textbox*.

2. Tampilan Pengambilan Nomor



Gambar 11. Pengambilan nomor

Keterangan :

- a. Nomor antrian nol : gambar ini menjelaskan, bahwa belum ada nomor antrian yang diambil sehingga maka akan muncul angka “0” (Nol) selama belum ada yang mengambil nomor antrian.
- b. Tombol Locket : Tombol ini berfungsi untuk mengambil nomor antrian apabila pelanggan akan menuju loket.
- c. Tombol Customer Servis : Tombol ini berfungsi untuk mengambil nomor antrian apabila pelanggan akan *customer servis*.
- d. Tampilan Pemanggilan Nomor



Gambar 12. Pemanggilan nomor

Keterangan :

- a. Nomor antrian nol : Gambar ini menjelaskan, bahwa belum ada nomor antrian yang dipanggil sehingga secara otomatis maka akan muncul angka “0” (Nol) selama belum ada pemanggilan nomor antrian.
- b. Tombol Locket : Tombol ini berfungsi untuk memanggil nomor antrian apabila pelanggan akan menuju loket.
- c. Tombol *Customer Servis* : Tombol ini berfungsi untuk memanggil nomor antrian apabila pelanggan akan *customer servis*.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dan disertai pembahasan sebelumnya maka diperoleh beberapa kesimpulan sebagai berikut :

1. Rancang bangun aplikasi Antrian Locket dibuat dengan menggunakan pemrograman Visual Basic 2010 untuk mengatasi antrian pelanggan dalam melakukan transaksi pembayaran tagihan air di kantor Perusahaan Daerah Air Minum(PDAM) cabang Bekasi.
2. Dengan adanya Aplikasi Antrian Locket ini dapat membantu dan mempermudah petugas melayani pelanggan dalam melakukan transaksi pembayaran tagihan air di kantor PDAM (Perusahaan Daerah Air Minum) cabang Bekasi.

DAFTAR PUSTAKA

- Andi. 2010. Visual Basic 2010 Source Code. Penerbit: Wahana Komputer.
- Bronson, R. 1988. Teori dan Soal-Soal Operation Research. Penerbit: Erlangga, Jakarta.
- Kakiay. Thomas J. 2004. Dasar Teori Antrian Untuk Kehidupan Nyata. Penerbit : Andi, Yogyakarta.
- Ladjamudin B, Al Bahra. 2006. Rekayasa Perangkat Lunak. Penerbit : Graha Ilmu, Jakarta.
- Roger S.Pressman. Ph.D. 2002. Rekayasa Perangkat Lunak (Buku Satu). Penerbit : Andi, Yogyakarta.
- Subari & Yuswanto, Pemrograman Visual Basic.Net. Penebit : Prestasi Pustaka
- Supardi. Yuniar. 2011. Semua Bisa Jadi Programmer VB 6 Hingga VB 2008. Penerbit : Elex Media Komputindo.
- Winarno. Edy. 2010. Dasar-Dasar Pemrograman dengan Visual Basic 2010. Penerbit : Gramedia, Jakarta.