

Program Aplikasi Sistem Antrian *Multiserver* Bank dengan *Vacation M/M/C*

Rakhmat Nurul Hakim¹, Rini Marwati², Dewi Rachmatin³

^{1), 2), 3)}Departemen Pendidikan Matematika FPMIPA UPI

*Surel: rakhmatnurulhakim93@gmail.com

ABSTRAK. Penelitian ini membahas mengenai perancangan program aplikasi antrian *multiserver M/M/C* pada Bank dengan melibatkan waktu *vacation*. Program aplikasi ini dirancang dengan tujuan untuk memberikan informasi kepada nasabah/*customer* tentang rata-rata waktu lama *customer* menunggu dalam antrian. Program aplikasi ini dikembangkan berdasarkan teori-teori antrian *multiserver M/M/C* dengan melibatkan prosedur *vacation*. Notasi M yang pertama menunjukkan nilai rata-rata jumlah kedatangan *customer* per jam (λ), di mana λ berdistribusi poisson. Notasi M yang kedua menunjukkan nilai rata-rata jumlah *customer* yang dilayani per jam (μ), di mana μ berdistribusi eksponensial. Notasi C menunjukkan jumlah *server* yang diteliti. Pada penelitian ini, jumlah *server* yang diteliti sebanyak 5 buah. *Vacation* adalah rata-rata waktu lama *server* berhenti per kejadian dalam melayani untuk sementara waktu. Dikarenakan studi kasus penelitian ini adalah sistem antrian Bank, maka sistem antrian yang digunakan adalah FIFO (*First In First Out*). Keluaran dari program aplikasi ini berupa nilai rata-rata atau nilai ekspektasi, di mana nilainya tidak akan selalu tepat dengan realitanya. Akan tetapi dengan menggunakan prosedur *vacation*, maka keakuratan hasilnya dapat terjamin.

Kata kunci: *M/M/C*, aplikasi antrian, antrian *vacation*, antrian *multiserver*.

ABSTRACT. In this paper we create the application program of $M/M/C$ Bank multisever queueing system with vacation. This application program is designed with the purpose of providing information to customer. The information is average of customer's waiting length time in queue. So, customers can find out how long they will be in queue and they can schedule their next activities. This application program is developed based on theories of $M/M/C$ multiserver queue involves vacation procedure. First M notation shows average of customers' arrival number per hour value (λ), where λ is distributed by poisson distribution. Second M notation shows average of customer's served number per hour value (μ), where μ is distributed by exponential distribution. C notation shows number of searched servers, which number of searched servers are 5. Multiserver shows that number of searched servers more than one. Vacation is average length of servers stop time per incident for awhile. Because of this essay case study is Bank system queueing, the queue discipline used is FIFO (First In First Out). Keep in mind that the output of this application program is average value or expected value, so that the results will not always be appropriate to the reality. However, because this application program using procedure vacation, then the result can be guaranteed for accuracy.

Keywords : $M/M/C$, queue application, vacation queue, multiserver queue

1. PENDAHULUAN

Menurut Panji (2015), program aplikasi adalah *software* atau perangkat lunak komputer yang dibuat untuk melakukan tugas tertentu. Salah satu teknologi yang dibutuhkan dewasa ini adalah program aplikasi, karena dapat memudahkan kegiatan sehari-hari dan mudah dalam penggunaannya. Menurut Kakiay (2004), antrian adalah suatu kejadian yang biasa dalam kehidupan sehari-hari. Antrian dapat terjadi di mana saja, salah satunya di Bank. Antrian terjadi karena kebutuhan akan layanan melebihi kapasitas pelayanan, sehingga pengguna fasilitas yang tiba tidak bisa segera mendapat pelayanan. Pelaku utama pada sistem antrian Bank terdapat pada nasabah / *customer* dan *teller* / *server*. Menurut Tian dan Zhang (2006), *vacation* dalam sistem antrian merupakan suatu keadaan di mana tidak terjadi pelayanan untuk beberapa waktu tertentu. Dalam antrian Bank, *vacation* dapat dianggap sebagai waktu istirahat *server*, waktu *server* ketika melakukan tugas sekunder, atau gangguan teknis saat melakukan pelayanan.

Pada dewasa ini, manusia sangat membutuhkan Bank untuk kegiatan terkait keuangan seperti transfer uang, menabung uang, menarik uang, dll. Pada umumnya, pihak Bank selalu meningkatkan fasilitas untuk memberikan pelayanan terbaik pada *customer*. Salah satu fasilitas yang ditingkatkan tersebut adalah sistem antrian, di mana pelayanan terbaiknya adalah tidak membiarkan *customer* mengantri terlalu lama atau memberikan *customer* info waktu lamanya antrian

sebelum memasuki sistem antrian sehingga *customer* dapat mengagendakan kegiatan lain di luar mengantri di Bank. Untuk mengetahui info waktu lamanya antrian, pihak Bank dapat menghitung secara manual. Akan tetapi, menghitung secara manual tidaklah efektif, sehingga dibutuhkan program aplikasi untuk sistem antrian tersebut.

Program aplikasi akan selalu membutuhkan data input untuk diproses sehingga menghasilkan data output. Data input pada program aplikasi untuk kasus ini adalah rata-rata jumlah kedatangan *customer* per jam (λ) yang diasumsikan berdistribusi Poisson, rata-rata jumlah *customer* yang dilayani per jam (μ) yang diasumsikan berdistribusi Eksponensial, dan rata-rata waktu dari total waktu *vacation* yang terjadi (θ) yang diasumsikan berdistribusi Eksponensial. Data output pada program aplikasi ini adalah rata-rata lamanya waktu mengantri ($W_v^{(c)}$). Perlu diketahui, output pada program aplikasi ini adalah ekspektasi waktu sehingga hasilnya tidak akan selalu tepat dengan realitanya. Akan tetapi, karena menggunakan waktu *vacation* dapat dijamin keakuratannya. Pelaku utama pada program aplikasi ini adalah *customer*, *server* dan operator.

Dari sekian banyaknya *software* untuk membuat program aplikasi, matlab merupakan salah satu *software* yang cocok untuk digunakan dalam pembuatan program aplikasi ini dikarenakan mudah dalam penggunaannya, bentuk hasil dan outputnya yang memuaskan, dan memiliki *user interface* atau GUI. Dikarenakan studi kasus untuk program aplikasi ini menggunakan studi kasus penelitian dari Puspita (2015) tentang analisis antrian *multiserver* dengan *vacation*, maka penggunaan program aplikasi ini dibatasi pada 5 *server* saja.

2. METODOLOGI

Masalah yang dianalisa dalam penelitian ini adalah teori antrian dan antrian dengan *vacation*. Teori antrian membutuhkan data awal yaitu λ , μ , dan θ . Model dasar yang digunakan pada penelitian ini adalah sistem informasi tentang program aplikasi antrian *multiserver* dengan *vacation* dan sistem antrian *multiserver* dengan *vacation* ($M/M/C$). Notasi M yang pertama menunjukkan nilai λ yang berdistribusi Poisson. Notasi M yang kedua menunjukkan nilai μ yang berdistribusi eksponensial. Notasi C menunjukkan jumlah *server* yang diteliti, di mana jumlah *server* yang diteliti sebanyak 5 *server*. Dikarenakan studi kasus penelitian ini adalah sistem antrian Bank, maka disiplin antrian yang digunakan adalah FIFO (*First In First Out*).

Pengembangan program aplikasi ini melalui 4 tahapan yaitu merancang tampilan pengguna, membuat algoritma, menuliskan dalam bahasa pemrograman, dan menampilkan hasil program. Setelah melakukan pengembangan program

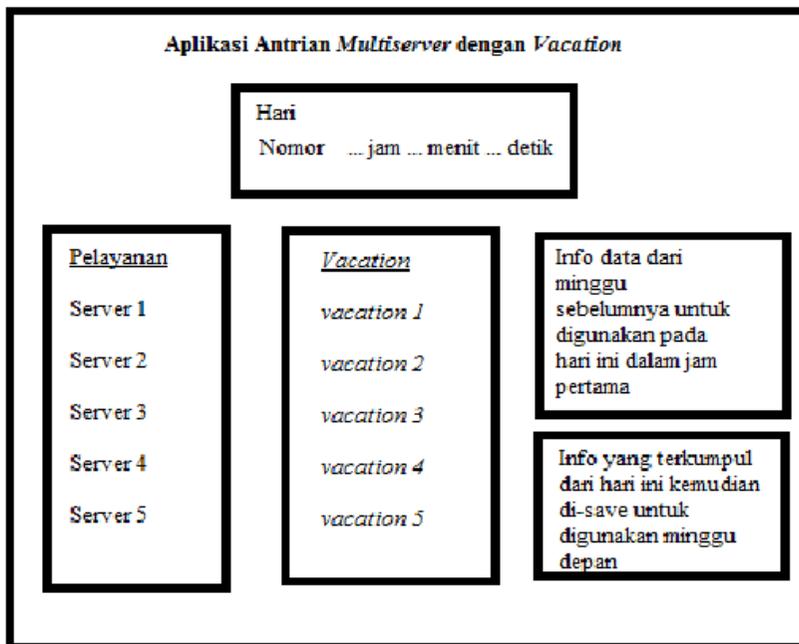
aplikasi, akan dilakukan validasi berupa pengujian dalam memasukkan data inputan secara manual yang sama pada penelitian yang digunakan sebelumnya.

3. PROGRAM APLIKASI SISTEM ANTRIAN *MULTISERVER* BANK DENGAN *VACATION M/M/C*

3.1 Pengembangan Program Aplikasi

a. Tampilan Pengguna

Pada rancangan tampilan tersebut terdapat 3 kolom yaitu kolom untuk *customer*, kolom untuk *server*, dan kolom untuk operator. Terdapat pula 5 panel yaitu panel untuk output dan nomor antrian, panel untuk pelayanan, panel untuk *vacation*, panel untuk data input, dan panel untuk kumpulan data dalam satu hari. Berikut adalah rancangan tampilan pengguna program aplikasi antrian *multiserver* dengan *vacation* :



Gambar 1. Rancangan Tampilan Penggunan

b. Algoritma dan Pemrograman

Model antrian *M/M/C* dengan *vacation* dikonstruksi ke dalam sub-sub prosedur sebagai berikut :

1) Prosedur Submatriks

Prosedur ini membutuhkan 3 data input yaitu λ , μ , dan θ . Setelah data input diproses dalam prosedur, maka akan menghasilkan 3 data output yaitu matriks A, matriks B, dan matriks C yang memiliki orde $c+1$ dengan $c=5$.

$$A = \begin{bmatrix} -h_0 & c\theta & 0 & 0 & 0 \\ 0 & -h_1 & (c-1)\theta & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -h_{c-1} & \theta \\ 0 & 0 & 0 & 0 & -h_c \end{bmatrix}$$

$$A = \begin{bmatrix} -(\lambda+5\theta) & 5\theta & 0 & 0 & 0 & 0 \\ 0 & -(\lambda+\mu+4\theta) & 4\theta & 0 & 0 & 0 \\ 0 & 0 & -(\lambda+2\mu+3\theta) & 3\theta & 0 & 0 \\ 0 & 0 & 0 & -(\lambda+3\mu+2\theta) & 2\theta & 0 \\ 0 & 0 & 0 & 0 & -(\lambda+4\mu+\theta) & \theta \\ 0 & 0 & 0 & 0 & 0 & -(\lambda+5\mu) \end{bmatrix}$$

$$B = \text{diag}(0, \mu, 2\mu, \dots, c\mu) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & (c-1)\mu & 0 \\ 0 & 0 & 0 & c\mu \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 3\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 4\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 5\mu \end{bmatrix}$$

$$C = \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda \end{bmatrix}$$

Prosedur ini membutuhkan 3 data input yaitu λ , μ , dan θ . Setelah data input diproses dalam prosedur, maka akan menghasilkan output matriks R, di mana elemen-elemen pada matriks R adalah kumpulan dari persamaan kuadrat *rate* (r).

$$R = \begin{bmatrix} r(0) & r(0,1) & r(0,2) & r(0,3) & r(0,4) & r(0,5) \\ 0 & r(1) & r(1,2) & r(1,3) & r(1,4) & r(1,5) \\ 0 & 0 & r(2) & r(2,3) & r(2,4) & r(2,5) \\ 0 & 0 & 0 & r(3) & r(3,4) & r(3,5) \\ 0 & 0 & 0 & 0 & r(4) & r(4,5) \\ 0 & 0 & 0 & 0 & 0 & r(5) \end{bmatrix}$$

Di mana untuk : $0 \leq k \leq 5$ (entri diagonal) :

$$r_1^* = \frac{\lambda + \mu + 4\theta + \sqrt{(\lambda + \mu + 4\theta)^2 - 4\mu\lambda}}{2\mu}$$

$$r_1 = \frac{\lambda + \mu + 4\theta - \sqrt{(\lambda + \mu + 4\theta)^2 - 4\mu\lambda}}{2\mu}$$

$$r_0 = \frac{\lambda}{(\lambda + c\theta)}$$

Untuk: $0 \leq k \leq 4$ (entri non-diagonal)

$$r_{k,k+1} = \left(\frac{c-k}{k+1}\right) \left(\frac{\theta}{\mu}\right) \frac{r_k}{r_{k+1}^* - r_k}$$

Untuk: $0 \leq k \leq 3$ (entri non-diagonal)

$$= \frac{(c-k)(c-k-1)}{(k+1)(k+2)} \left(\frac{\theta}{\mu}\right)^2 \frac{r_{k,k+2}}{(r_{k+2}^* - r_{k+1})(r_{k+2}^* - r_k)(r_{k+1}^* - r_k)}$$

Untuk: $0 \leq k \leq 2$ (entri non-diagonal)

$$r_{k,k+3} = \left[\frac{(c-k)(c-k-1)(c-k-2)}{(k+1)(k+2)(k+3)} \left(\frac{\theta}{\mu}\right)^3 \cdot \frac{r_k r_{k+3}^* (r_{k+2}^* r_{k+3}^* - r_k r_{k+1})}{(r_{k+3}^* - r_{k+2})(r_{k+3}^* - r_{k+1})(r_{k+3}^* - r_k)(r_{k+2}^* - r_{k+1})(r_{k+2}^* - r_k)(r_{k+1}^* - r_k)} \right]$$

Untuk: $0 \leq k \leq 1$ (entri non-diagonal)

$$r_{k,k+4} = \frac{[(c-k-3)\theta + (k+4)\mu r_{k+3,k+4}] r_{k,k+3} + (k+4)\mu (r_{k,k+1} r_{k+1,k+4} + r_{k,k+2} r_{k+2,k+4})}{(k+4)\mu (r_{k+4}^* - r_k)}$$

Untuk: $k = 0$ (entri non-diagonal)

$$r_{k,k+5} = \frac{[(c-k-4)\theta + (k+5)\mu r_{k+4,k+5}] r_{k,k+4}}{(k+5)\mu (r_{k+5}^* - r_k)} + \frac{(k+5)\mu (r_{k,k+1} r_{k+1,k+5} + r_{k,k+2} r_{k+2,k+5} + r_{k,k+3} r_{k+3,k+5})}{(k+5)\mu (r_{k+5}^* - r_k)}$$

2) Prosedur Matriks *Generator* B[R]

Prosedur ini merupakan lanjutan dari prosedur submatriks. Output dari prosedur ini adalah matriks $B[R]$, di mana elemen-elemennya adalah matriks-matriks dari prosedur submatriks dan matriks R .

$$B[R] = \begin{bmatrix} A_0 & C_0 & 0 & 0 & 0 & 0 \\ B_1 & A_1 & C_1 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & B_{c-1} & -A_{c-1} & C_{c-1} & 0 \\ 0 & 0 & 0 & B_c & A_c & AR + B \end{bmatrix}$$

$$B[R] = \begin{bmatrix} A_0 & C_0 & 0 & 0 & 0 & 0 \\ B_1 & A_1 & C_1 & 0 & 0 & 0 \\ 0 & B_2 & A_2 & C_2 & 0 & 0 \\ 0 & 0 & B_3 & A_3 & C_3 & 0 \\ 0 & 0 & 0 & B_4 & A_4 & C_4 \\ 0 & 0 & 0 & 0 & B_5 & A + RB \end{bmatrix}$$

3) Prosedur Penyelesaian Matriks

Prosedur ini merupakan dari prosedur matriks *generator* $B[R]$. Beberapa elemen pada matriks *generator* $B[R]$ merupakan persamaan. Maka di prosedur ini dilakukan penyelesaian persamaan matriks-matriks tersebut. Output dari prosedur ini adalah matriks β , di mana elemen-elemennya adalah kumpulan dari hasil penyelesaian persamaan matriks. Di mana :

$$\pi = [\pi_{00} \pi_{10} \pi_{11} \pi_{20} \pi_{21} \pi_{22} \pi_{30} \pi_{31} \pi_{32} \pi_{33} \pi_{40} \pi_{41} \pi_{42} \pi_{43}$$

$$\pi_{44} \pi_{50} \pi_{51} \pi_{52} \pi_{53} \pi_{54} \pi_{55}]$$

$$\pi \cdot B[R] = 0 ; \pi_{00} = K \text{ dan } \pi_k = K \cdot \beta_k ;$$

$$\beta = [\beta_{00} \beta_{10} \beta_{11} \beta_{20} \beta_{21} \beta_{22} \beta_{30} \beta_{31} \beta_{32} \beta_{33} \beta_{40} \beta_{41} \beta_{42} \beta_{43} \beta_{44}$$

$$\beta_{50} \beta_{51} \beta_{52} \beta_{53} \beta_{54} \beta_{55}]$$

4) Prosedur Keefektifan Sistem

Prosedur ini merupakan lanjutan dari prosedur-prosedur sebelumnya. Data yang digunakan pada prosedur ini adalah data dari output prosedur-prosedur sebelumnya. Output dari prosedur ini adalah $W_v^{(c)}$ (rata-rata lamanya *customer* mengantri).

$$\text{Di mana : } \delta = [\beta_{50} \quad \beta_{51} \quad \beta_{52} \quad \beta_{53} \quad \beta_{54}] ; \eta = \begin{bmatrix} r_{05} \\ r_{15} \\ r_{25} \\ r_{35} \\ r_{45} \end{bmatrix} ; \rho = \frac{\lambda}{c\mu}$$

$$H = \begin{bmatrix} r_0 & r_{01} & r_{02} & r_{03} & r_{04} \\ 0 & r_1 & r_{12} & r_{13} & r_{14} \\ 0 & 0 & r_2 & r_{23} & r_{24} \\ 0 & 0 & 0 & r_3 & r_{34} \\ 0 & 0 & 0 & 0 & r_4 \end{bmatrix} ; L_d = \frac{1}{\sigma} \delta (I - H)^{-2} \eta ; \sigma = \beta_{55} + \delta (I - H)^{-1} \eta$$

$$L_v^{(c)} = \frac{\rho^2}{1-\rho} + L_d \quad \text{dan} \quad W_v^{(c)} = \frac{L_v^{(c)}}{\lambda}$$

5) Prosedur Program Utama

Prosedur ini merupakan akhir dari semua prosedur. Pada prosedur ini hanya mengatur penampilan output secara benar.

c. PASIAV (Program Aplikasi Sistem Info Antrian *Vacation*)

Pembuatan *coding* untuk program aplikasi ini dibagi dua tahapan, yaitu *coding* untuk prosedur utama dan *coding* untuk *user interface*. Prosedur utama pada program aplikasi ini adalah prosedur perhitungan menggunakan model antrian *vacation M/M/C*. Sebagaimana telah dijelaskan pada bagian 3.1, penulisan *coding* untuk prosedur utama dibuat berdasarkan perhitungan sub-sub prosedur pada bagian tersebut,.

Dikarenakan program aplikasi ini tentang waktu, maka yang paling penting pada *coding* untuk *user interface* adalah menggunakan fungsi *timer*. Fungsi *timer* berguna untuk merekam waktu ketika tombol tertentu ditekan dan akan mengeluarkan hasil waktu jika dibutuhkan. Dari banyaknya sintaks yang disediakan oleh matlab, program aplikasi ini hanya menggunakan beberapa sintaks, di antaranya yaitu *string* untuk penulisan, *get* untuk membaca data (misal untuk input λ , μ , dan θ), *set* untuk menulis data (misal untuk output $W_v^{(c)}$), *xlsread* untuk membaca data yang berasal dari microsoft excel (misal mengambil data hari senin minggu lalu pada microsoft excel), *xlswrite* untuk menulis data pada microsfot excel (misal mengumpulkan dan menyimpan data hari senin untuk digunakan minggu depan), *switch case* untuk menentukan pemilihan hari, dll. Variabel umum yang digunakan pada *coding* ini adalah *handles* supaya fungsi-

fungsi pada program aplikasi dapat berjalan. Salah satu yang penting juga adalah menuliskan “guidata(hObject, handles)” pada setiap akhir prosedur, untuk memperbaharui kinerja dari *handles*.

Berikut adalah petunjuk dalam menggunakan PASIAV :

1) *Load Data*

Persiapkan data input pada *file excel*. Pada panel *pop up* menu dipilih hari, misal hari yang digunakan adalah hari Senin. Sebelum menekan tombol “*load data*”, terlebih dahulu mengisi alamat *file excel* di kolom “lokasi file”, kemudian tekan tombol “load data”.

2) *Input Data*

Misal input data manual yang diambil adalah data untuk hari senin yaitu $\lambda = 39$, $\mu = 10$, $\theta = 0,335$, maka pada panel “data yang digunakan” pun akan terinput datanya.

3) *Panel Pelayanan*

Pada penelitian ini terdapat 5 *server*, sehingga pada aplikasi ini terdapat 5 tombol “mulai” pelayanan. Tombol “mulai” pelayanan ditekan di setiap *server* untuk setiap *customer* yang mulai dilayani. Kemudian jumlah pelayanan yang terbaru akan tercatat pada “panel data hari ini”.

4) *Panel Vacation*

Pada penelitian ini terdapat 5 *server*, sehingga pada aplikasi ini terdapat 5 tombol “mulai” dan “berhenti” *vacatioin*. Tombol “mulai” *vacation* ditekan ketika terjadinya *vacation*, dan tombol “berhenti” ditekan ketika *vacation* tersebut sudah berhenti dan berlaku untuk setiap *server*. Kemudian waktu *vacation* yang terbaru akan tercatat pada panel “data hari ini”.

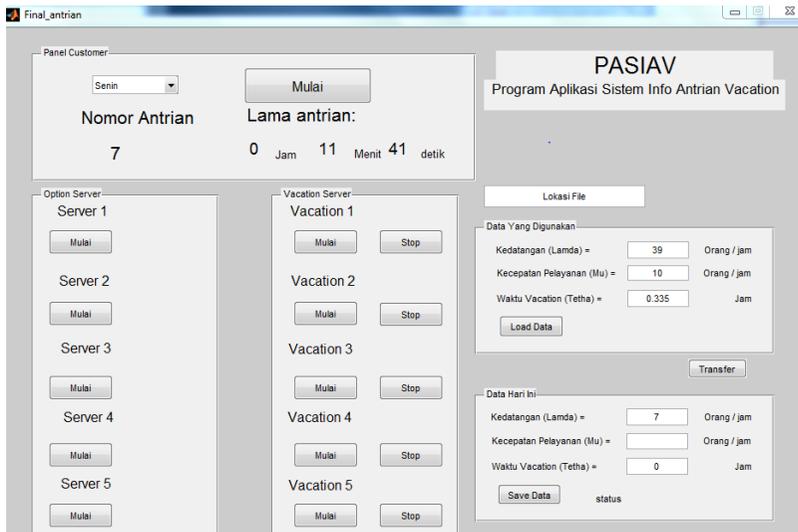
5) *Pembaharuan Data*

Untuk pembaharuan data yaitu dalam setiap 1 jam sejak tombol “mulai” antrian dimulai. Sehingga tombol “transfer” harus ditekan agar data yang digunakan akan berubah dengan data hari ini.

6) *Save Data*

Setelah waktu pelayanan pada Bank tutup atau selesai, maka data yang terdapat pada hari ini harus di-*save* dengan menekan tombol “save data”, supaya (misal) data hari senin minggu ini dapat digunakan pada hari senin minggu setelahnya. Ketika data telah di-*save* maka terdapat status berupa “tersimpan” di sebelah tombolnya yang artinya data telah tersimpan dalam *file excel*.

Setelah melakukan tahapan-tahapan sebelumnya, maka akan ditampilkan tampilan dan nilai output dari program aplikasi ini (menggunakan *software matlab*) menggunakan data dari studi kasus yang diambil. Tampilannya menjadi seperti gambar 2.



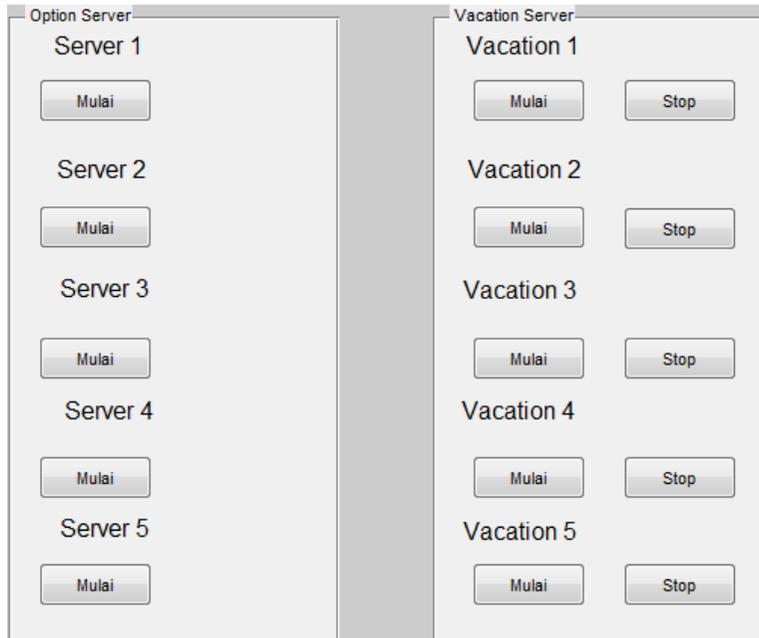
Gambar 2. Tampilan Program Aplikasi pada Matlab

Pada kolom customer terdapat satu panel. *Customer* hanya membutuhkan nomor antrian dan waktu *customer* berada dalam antrian. Berikut adalah tampilan untuk kolom *customer* :



Gambar 3. Kolom *Customer*

Pada kolom *server* terdapat 2 panel yaitu panel pelayanan dan panel *vacation*. Panel pelayanan digunakan ketika *server/teller* mulai melayani *customer*. Panel *vacation* digunakan ketika terjadi *vacation*. Berikut adalah tampilan untuk kolom *server* :



Gambar 4. Kolom *Server*

Pada kolom operator terdapat 2 panel yaitu panel “data yang digunakan” (data input) dan panel “data hari ini” (pembaharuan data). Operator berbeda dari *customer* dan *server*. Operator bertugas menginputkan data minggu lalu pada panel “data yang digunakan” dan menyimpan data yang diperoleh pada hari tersebut di panel “data hari ini” untuk digunakan minggu depan. Berikut adalah kolom untuk operator :

The image shows a software interface for a queue simulation. It is divided into two main sections: "Data Yang Digunakan" (Data Used) and "Data Hari Ini" (Today's Data). The top section, "Data Yang Digunakan", contains three input fields: "Kedatangan (Lamda) =" with the value 39, "Kecepatan Pelayanan (Mu) =" with the value 10, and "Waktu Vacation (Tetha) =" with the value 0.335. Below these fields is a "Load Data" button. The bottom section, "Data Hari Ini", contains three input fields: "Kedatangan (Lamda) =" with the value 6, "Kecepatan Pelayanan (Mu) =" which is empty, and "Waktu Vacation (Tetha) =" with the value 0. Below these fields is a "Save Data" button and a "status" label. A "Transfer" button is located between the two sections.

Gambar 5. Kolom Operator

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

- a. Pengembangan algoritma program aplikasi antrian *multiserver* dengan *vacation* membutuhkan 3 informasi data input yaitu data rata-rata jumlah kedatangan *customer* per jam (λ), data rata-rata jumlah *customer* yang dilayani per jam (μ), dan data rata-rata lamanya waktu *vacation* (θ), sehingga program aplikasi ini memberikan informasi data output rata-rata lamanya *customer* menunggu dalam antrian ($W_v^{(c)}$).
- b. Pengujian studi kasus program aplikasi antrian *multiserver* dengan *vacation* menggunakan data antrian dari salah satu Bank di Bandung. Berikut adalah hasilnya :

- 1) Hari Senin
 Input :
 Laju kedatangan (λ) = 39 orang/jam.
 Kecepatan pelayanan (μ) = 10 orang/jam.
 Waktu *vacation* (θ) = 0,335 jam.
 Output :
 Rata-rata waktu lamanya mengantri = 1,003 jam atau 1 jam 10 detik.
- 2) Hari Kamis
 Input :
 Laju kedatangan (λ) = 40 orang/jam.
 Kecepatan pelayanan (μ) = 12 orang/jam.
 Waktu *vacation* (θ) = 0,395 jam.
 Output :
 Rata-rata waktu lamanya mengantri = 0,7167 jam atau 43 menit.
- 3) Hari Jum'at
 Input :
 Laju kedatangan (λ) = 39 orang/jam.
 Kecepatan pelayanan (μ) = 11 orang/jam.
 Waktu *vacation* (θ) = 0,321 jam.
 Output :
 Rata-rata waktu lamanya mengantri = 0,8831 jam atau 52 menit 59 detik.

4.2 Saran

Program aplikasi antrian *multiserver* dengan *vacation* ini merupakan penerapan dari penelitian sistem antrian *multiserver* Bank *M/M/C* dengan *vacation* dengan batasan 5 *server*. Pada penelitian selanjutnya diharapkan dapat membuat program aplikasi antrian Bank *multiserver* dengan *vacation* dengan jumlah *server* yang diinginkan oleh operator sehingga dapat digunakan di seluruh Bank, baik itu pusat atau cabang.

5. DAFTAR PUSTAKA

- [1] Code, V. (2015). *Pengenalan GUI Matlab*. [Online] 2017. Tersedia: <http://vvrongcode.blogspot.co.id/2015/05/gui-matlab.html>.
- [2] Puspita, L. (2015). *Analisis Efektivitas Sistem Antrian Multiserver Bank BCA cabang Jamika Bandung dengan Vacation menggunakan Visualisasi Promodel*. Bandung: UPI.
- [3] Elyzabeth. (2014). *Aplikasi Model Antrian Multiserver dengan Vacation pada Sistem Antrian di Bank BCA cabang Ujung Berung*. Bandung: UPI.
- [4] Hardani, A. (2013). *Matlab*. [Online] 2016. Tersedia: <http://ameliaadz.blogspot.co.id/2013/03/pengertian-matlab.html>
- [5] Herrhyanto, N., & Gantini, T. (2009). Dalam N. Herrhyanto, & T. Gantini, *Pengantar Statistik Matematis*. Bandung: Yrama Widya.
- [6] Kakiay, T. J. (2004). Dalam T. J. Kakiay, *Dasar Teori Antrian untuk Kehidupan Nyata*. Yogyakarta: Penerbit Andi.
- [7] Nurjamil, J. (2014). *Pengertian Pemrograman Dasar*. [Online] 2017. Tersedia: <http://jajangnurjamiltkj.blogspot.co.id/2014/08/pengertian-pemograman-dasar.html>.
- [8] Panji, P. (2015). *Pengertian dan Jenis-jenis Program Aplikasi*. [Online] 2016. Tersedia: <https://pramadanapanji.wordpress.com/2015/06/03/pengertian-dan-jenis-jenis-program-aplikasi/>.
- [9] Siagian. (1987). *Operation Management*. [Online] 2016. Tersedia: <https://sites.google.com/site/operasiproduksi/teori-antrian>.
- [10] Sugiharto, A. (2006). *Pemrograman GUI dengan Matlab*. Yogyakarta: Penerbit ANDI.
- [11] Tian, & Zhang. (2006). Dalam N. Z. Tian, *Vacation Queueing Models Theory and Applications*. New York: Springer Science & Business Media.