ANALISIS PERFORMANSI SERVER SISTEM INFORMASI AKADEMIK UNIVERSITAS MERCU BUANA DENGAN OPEN QUEUEING NETWORK

Desi Ramayanti

Program studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jl. Raya Meruya Selatan, Kembangan, Jakarta, 11650
e-mail : desiramayanti@yahoo.com

ABSTRAK

SIA (Sistem Informasi Akademik) merupakan sebuah sistem informasi penting yang berkaitan dengan nilai yang sudah diperoleh untuk setiap matakuliah yang telah diambil, status dari mahasiswa pada semester terkait, dll. SIA ini akan selalu diakses oleh mahasiswa dan juga dosen di lingkungan UMB, baik pada setiap awal semester, pertengahan semester atau pada akhir semester. SIA merupakan fitur penting yang harus tetap available setiap saat, dan mampu melayani ribuan user dengan baik. Keterlambatan data (delay), kehilangan data (lost) dan kerusakan data (error), harus menjadi point penting dalam sebuah system, karena jika ketiga hal tersebut terjadi, maka nilai informasi akan berkurang atau menjadi tidak berarti lagi, dan hal ini akan sangat merugikan, khususnya kepada mahasiswa sebagai konsumen yang wajib mendapatkan pelayanan terbaik. Maka dikarenakan hal tersebut, maka penulis melakukan penelitian berkaitan dengan performansi SIA saat ini. Dalam penelitian ini, penulis melakukan analisis performansi SIA yang terkait dengan arrival rate resource, service time resource utilisasi resource, Rata-rata jumlah kunjungan ke resource, service demand resourcedan arrival rate maksimum dari server. Dari hasil analisis maka diperoleh kesimpulan bahwa performansi dari server SIA yang ada saat ini kurang maksimum dengan jumlah mahasiswa yang ada. Server Sistem Informasi Akademik saat ini hanya mampu melayani request per detik sebanyak 7.278 request. Sedangkan arrival rate yang ada yaitu 8.88 request per detik, sehingga kemungkinan lost request adalah 21.98%.

Kata Kunci: *Performansi Server, SIA, UMB.*

PENDAHULUAN

Latar Belakang

Universitas Mercu Buana (UMB) sebagai perguruan tinggi swasta yang terus berkembang menjadikan Teknologi Informasi sebagai kata kunci pada Visi Misi nya. Hal ini diimplementasikan dengan website www.mercubuana.ac.id. Dimana fitur-fitur yang ada pada website tersebut, hampir mencakup semua bagian penting, seperti:

1. Link Home, berisi semua informasi umum dan link ke bagian lain dalam Universitas Mercu Buana, seperti link untuk melihat struktur organisasi, link informasi pendaftaran mahasiswa baru, link ke fakultas dan program studi yang ada.
2. Link SIA Reguler, berisi informasi yang berkaitan dengan nilai dan status mahasiswa regular.
3. Link SIA Karyawan, berisi informasi yang berkaitan dengan nilai dan status mahasiswa Kelas Karyawan.
4. Link Umb Webmail, link ke email internal UMB
5. Dll.

Dari sekian banyak fitur yang tersedia, yang paling sering diakses dan sifatnya paling penting adalah link SIA (Sistem Informasi Akademik) Reguler ataupun Karyawan. Karena link ini berisi semua informasi penting tentang setiap mahasiswa UMB, mulai dari nilai yang sudah diperoleh untuk setiap matakuliah yang telah diambil, status dari mahasiswa pada semester terkait, dll. SIA ini akan selalu diakses oleh mahasiswa dan juga dosen di lingkungan UMB, baik pada setiap awal semester, pertengahan semester atau pada akhir semester.

Sehingga SIA merupakan fitur penting yang harus tetap *available* setiap saat, dan mampu melayani ribuan *user* dengan baik.

Dimana keterlambatan data (*delay*), kehilangan data (*lost*) dan kerusakan data (*error*), harus menjadi point penting dalam sebuah system, karena jika ketiga hal tersebut terjadi, maka nilai informasi akan berkurang atau menjadi tidak berarti lagi, dan hal ini akan sangat merugikan, khususnya kepada mahasiswa sebagai konsumen yang wajib mendapatkan pelayanan terbaik.

Performansi suatu sistem harus memenuhi suatu ukuran yang telah ditetapkan, seperti *service demand*, utilisasi, *residence time*, *queue length* dan *throughput*[3]. Untuk melakukan analisis terhadap performansi, salah satu metoda yang dapat digunakan adalah memodelkan sistem sebagai jaringan antrian. Jaringan antrian ini dapat terdiri dari sebuah *resource* atau beberapa *resource* (misalnya CPU, *disk* dan lain-lain), dan antrian dari *request* yang menunggu untuk menggunakan *resource*. Jaringan antrian juga merupakan kumpulan dari *service center* yang mewakili *resource* sistem, dan *customer* atau *request* yang mewakili *user* atau transaksi .

Perumusan Masalah

Berdasarkan uraian yang telah disampaikan sebelumnya, maka masalah yang akan berusaha dijawab dalam penelitian ini adalah

1. bagaimana memodelkan Sistem Informasi Akademik menjadi sebuah jaringan antrian, untuk melakukan analisis terhadap resource server seperti CPU dan harddisk
2. bagaimana menganalisa performansi system berdasarkan model jaringan antrian Sistem Informasi Akademik Universitas Mercu Buana sehingga diperoleh kemampuan maksimum server dalam menerima request.

Batasan Masalah

Agar pembahasan dalam membuat sistem ini tidak terlalu luas, maka ruang lingkup dibatasi pada :

1. Analisis Arrival rate resource (λ)
2. Analisis Service time resource (S_i)
3. Analisis Utilisasi resource (U_i)
4. Analisis Rata-rata jumlah kunjungan ke resource (V_i)
5. Analisis Service demand resource(D_i)
6. Analisis Arrival rate maksimum dari server (λ_{Sat})

Tujuan Penelitian

Penelitian ini bertujuan untuk memperoleh hasil analisis performansi Sistem Informasi Akademik UMB yang sudah ada saat ini.

Metodologi Penelitian

Metode penelitian yang akan dipergunakan adalah sebagai berikut:

1. Identifikasi masalah
   Pada tahap ini akan dilakukan identifikasi terhadap masalah yang akan dibahas, yaitu bagaimana melakukan perancangan model jaringan antrian dan menganalisa resource maupun server secara utuh, sehingga diperoleh bagaimana performansi system yang ada
2. Studi Literatur
   Pada tahap ini akan dilakukan studi literature tentang teori-teori yang berkaitan denan performansi suatu system computer, analisa performansi menggunakan metoda jaringan antrian dan performansi resource yang digunakan
3. Analisis Performansi Resource

Pada tahap ini dilakukan analisis terhadap performansi resource dari server, dimana server dimodelkan menjadi sebuah jaringan antrian open queueing network, dengan populasi infinite dan kapasitas buffer finite

4. Tabel dan grafik hasil
   Pada tahap ini hasil dari analisa akan direkap kedalam sebuah table dan diproses menjadi bentuk grafik

LANDASAN TEORI

Model Jaringan Antrian

Jaringan antrian merupakan suatu jaringan yang dihubungkan dengan suatu antrian, yang dapat menggambarkan suatu sistem computer. Antrian dalam jaringan antrian ini dapat terdiri dari; sebuah resource atau beberapa resource (CPU, disk, dan lain-lain) dan antrian dari request yang menunggu untuk menggunakan resource. Jaringan antrian merupakan kumpulan dari service center yang mewakili resource sistem, dan customer atau request yang mewakili user atau transaksi.

Pada model jaringan antrian dengan service center tunggal seperti pada Gambar. 1, memiliki 2 parameter yaitu:

1. Menentukan intensitas workload
2. Menentukan service demand [Lazowska, Edward, D., Zahorjan, John, Graham, G. Scoot, and Sevcik, Kenneth C. (1984)], yaitu rata-rata layanan yang dibutuhkan oleh seorang customer.

Dengan menetapkan nilai dari kedua parameter ini, maka dapat digunakan untuk mengevaluasi model dengan menyelesaikan beberapa persamaan yang menghasilkan ukuran-ukuran kinerja seperti utilisasi (ukuran waktu server sibuk), residence time (rata-rata waktu yang dibutuhkan pada service center untuk satu customer, baik ketika menunggu atau menerima layanan), queue length (rata-rata jumlah customer pada service center yang menunggu layanan) dan throughput (kecepatan dimana customer meninggalkan service center).

Sedangkan pada model dimana terdapat beberapa service center, seperti pada Gambar 2, sama dengan parameter pada model service tunggal, yaitu menentukan workload yang merupakan kecepatan dari kedatangan customer dan menentukan service demand, tetapi pada service demand, ditetapkan untuk setiap service center. Intensitas workload pada model ini, berhubungan dengan kecepatan user memberikan sebuah transaksi ke sistem, dan service demand pada setiap service center yang berhubungan dengan kebutuhak total layanan per transaksi, yang berhubungan dengan resource dalam sistem. pada sistem ini customer datang, beredar diantara service center dan kemudian keluar.

Hukum-hukum dasar performansi

Pada subbab ini dijelaskan sejumlah kuantitas yang penting dan diperkenalkan notasi-notasi yang akan digunakan untuk kuantitas ini, memperoleh berbagai hubungan aljabar antara kuantitas ini dan dikenal dengan fundamental law [Gunther, Neil, J. (2005)]

A. Hukum Utilisasi

Gambar 1 menunjukkan *service center* tunggal, yang dilihat sebagai antrian *resource* tungal, sehingga utilisasi (U_i) dari *resource* didefenisikan sebagai persentase waktu dimana *resource* sibuk. Jika antrian *i* tersebut dimonitor selama τ detik, dan menemukan *resource* sibuk selama B_i detik, maka diperoleh utilisasi, $U_i = \frac{B_i}{\tau}$. Diasumsikan bahwa selama *interval* τ yang sama, C_0 transaksi telah selesai dari antrian. Hal ini berarti bahwa rata-rata *throughput* dari antrian adalah

$$X_i = \frac{C_0}{\tau}$$

Dengan menggabungkan hubungan rata-rata *throughput* ini dengan defenisi utilisasi, maka diperoleh persamaan berikut:

$$U_i = \frac{B_i}{\tau} = \frac{B_i}{C_0} \times X_i = S_i + X_i$$

Persamaan di atas menyatakan bahwa rata-rata waktu *resource* sibuk per transaksi, merupakan rata-rata *service time* (S_i) per transaksi, yang diperoleh dari total waktu *resource* sibuk (B_i), dibagi dengan jumlah

transaksi yang dilayani selama perioda waktu yang dimonitor. Pada kondisi *equilibrium* $\lambda_i = X_i$, dan *service time* $S_i = \frac{1}{\mu}$, maka diperoleh persamaan

$$U_i = S_i \times X_i = S_i \times \lambda_i = \lambda_i \times \frac{1}{\mu_i} = \frac{\lambda_i}{\mu_i}$$

Utilisasi dapat juga diinterpretasikan sebagai jumlah rata-rata transaksi dalam *resource*, karena ada satu transaksi yang menggunakan *resource* selama U_i persen waktu dan nol transaksi selama (1- U_i) persen waktu.

Untuk kasus *multiple service center* seperti yang ditunjukkan pada Gambar 2, yang memiliki banyak antrian *resource*, maka utilisasi didefenisikan sebagai jumlah rata-rata transaksi yang menggunakan suatu *resource*, dan dinormalisasikan dengan jumlah *resource*, maka diperoleh persamaan beriiikut untuk utilisasi dari antrian *m* resource.

$$U_i = S_i \times \frac{X_i}{m}$$

B. Hukum Forced Flow

Dengan defenisi dari rata-rata jumlah kunjungan V_i, dimana setiap penyelesaian transaksi harus melalui V_i kali, pada rata-rata, dari antrian *i*. Sehingga, jika X_0 transaksi selesai per satuan waktu, maka $V_0 \times X_i$ transaksi akan mengunjungi antrian *i* per satuan waktu. Sehingga, diperoleh persamaan untuk rata-rata *throughput* dari antrian *i*.

$$X_i = V_i \times X_0$$

Persamaan di atas dikenal dengan hukum *Forced Flow*.

C. Hukum Service Demand

Total waktu yang dibutuhkan oleh sebuah *request* pada suatu *server* dapat dibagi menjadi 2 jenis *interval*; yaitu *service time* dan *waiting time*. *Service time* (S_i), merupakan perioda waktu selama *request* menerima layanan dari suatu *resource* (misalnya, CPU dan *disk*). Sedangkan waktu yang dibutuhkan oleh sebuah *request* untuk menunggu mendapatkan layanan dari *resource i*, disebut dengan *waiting time* (W_i).

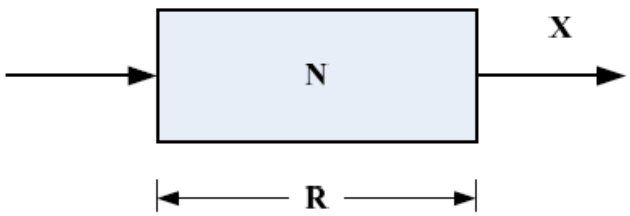
Jumlah dari *service time* (S_i), untuk sebuah *request* pada *resource i*, dikalikan dengan rata-rata jumlah kunjungan (V_i) disebut dengan *service demand* dan dilambangkan dengan (D_i), sehingga diperoleh persamaan,

$$D_i = V_i \times S_i$$

Dengan persamaan di atas, jika dihubungakan dengan *throughput* dan utilisasi sistem, dan menggabungkan hukum Utilisasi dengan *Forced Flow*, maka diperoleh persamaan berikut;

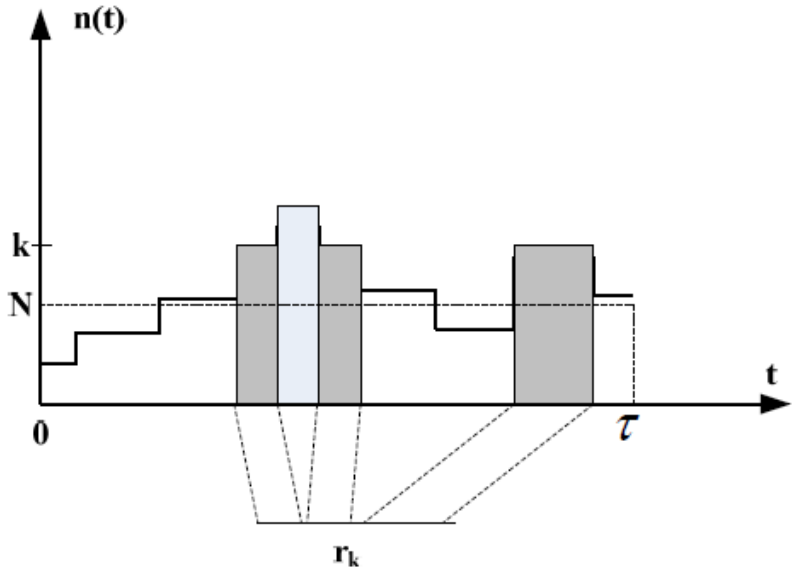
$$D_i = V_i \times S_i = \left(X_i/X_0\right) + \left(U_i/X_i\right) = U_i/X_0$$

D. Hukum Little's



Gambar 1. Kotak Little's [5]

Kotak pada Gambar 1 di atas, dapat berisi apa saja, contoh yang paling sederhana adalah *disk*, atau yang kompleks adalah *internet*. Diasumsikan bahwa *customer* datang ke *black box* tersebut menggunakan rata-rata *R* detik dalam *black box*, dan kemudian meninggalkan *black box* tersebut. Rata-rata kecepatan keberangkatan yaitu *throughput* dari *black box* adalah *X customer*/detik dan jumlah rata-rata *customer* dalam *black box* adalah *N*.



Gambar 2. Grafik hubungan n(t) terhadap t[5].

Pada Gambar 2 terlihat sebuah grafik jumlah *customer* *n*(*t*) di dalam *black box* pada waktu *t*. Misalkan, aliran *customer* dimulai dari waktu 0 sampai waktu τ . Maka, jumlah rata-rata *customer* selama *interval* tersebut adalah sama dengan jumlah semua hasil perkalian dari *k* × *fk* , dimana *k* merupakan jumlah *customer* dalam *black box*, dan *fk* merupakan persentase waktu dimana *k* customer ada dalam *black box*.

$f_k = \left({}^{r_k}/_{\tau}\right)$, dimana r_k adalah total waktu dimana ada *k* customer dalam *black box*. Sehingga diperoleh persamaan;

$$N = \sum_k k \times f_k = \sum_k k \times {}^{r_k}/_{\tau}$$

Dengan mengalikan dan membagi sisi bagian kanan dari persamaan diatas dengan jumlah *customer* C_0, maka diperoleh persamaan berikut untuk keberangkatan dari *black box* dalam *interval* [0,τ].

$$N = \frac{C_0}{\tau} \times \frac{\sum_k k \times r_k}{C_0}$$

$\frac{C_0}{\tau}$ merupakan *throughput* X, maka persamaan di atas menjadi jumlah total *customer* (*customer* × detik) yang diakumulasi dalam sistem. Jika dibagi jumlah ini dengan jumlah total *customer* C_0 yang selesai, maka diperoleh rata-rata waktu *R*,

setiap *customer* dalam *black box*, sehingga diperoleh persamaan; $N = X \times R$

Jika hukum *Little's* ini diaplikasikan untuk sekumpulan *m resource*, maka diperoleh

Persamaan; $N_i = X_i \times R_i$

Input Parameter Performansi Model [Manasce, Daniel, A., and Almeida, Virgilio, A.F. (1998)]

Input Parameter untuk performansi model menggambarkan konfigurasi *hardware*, *software* dan *workload* sistem. Parameter-parameter ini terdiri dari 4 kelompok informasi, yaitu sebagai berikut:

1. *queue* atau *device*,
2. kelas *workload*,
3. intensitas *workload*,
4. *service demand*.

Teorema Jackson[Gunther, Neil, J. (2005)]

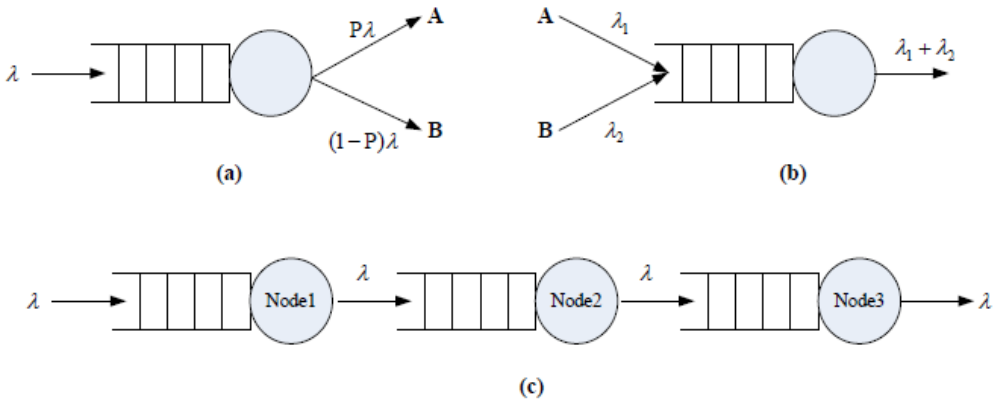
Teorema Jackson dapat digunakan untuk menganalisis suatu jaringan antrian. Teoreman ini didasarkan pada 3 asumsi yaitu:

1. jaringan antrian terdiri dari *m node*, dimana masing-masing *node* memberikan layanan eksponensial yang bebas,
2. sebuah *request* datang dari luar sistem ke suatu *node*, dengan *arrival rate* adalah Poisson,
3. setelah dilayani pada satu *node*, *request* tersebut akan menuju ke *node* yang lain dengan suatu probabilitas atau keluar dari sistem.

State pada Jackson *theorem* merupakan suatu jaringan antrian, dimana masing-masing *node* adalah sistem antrian yang *bebas*, dengan input Poisson yang ditentukan dengan antrian *partitioning*, *merging* atau *tandem.*

1. Antrian *partitioning*.
   Request datang ke suatu antrian dengan rata-rata *arrival rate* λ , dan ada 2 kemungkinan tujuan dari *request* selanjutnya yaitu A dan B (Gambar 5a). Pada saat *request* telah selesai dilayani dan keluar dari jalur A dengan probabilitas *p*, dan dari jalur B dengan probabilitas 1-*p*. Secara umum, distribusi aliran *request* A dan B akan berbeda dengan distribusi *incoming*. Namun, jika distribusi *incoming* merupakan Poisson, maka 2 aliran keberangkatan *request* juga distribusi Poisson, sehingga *arrival rate*-nya adalah $p\lambda$ dan $(1-p)\lambda$.
2. Antrian Merging.
   Jika 2 distribusi Poisson dengan rata-rata *arrival rate* adalah λ_1 dan λ_2 digabung, maka menghasilkan distribusi Poisson dengan rata-rata *arrival rate* adalah $\lambda_1+\lambda_2$, seperti yang terlihat pada Gambar 5b.
3. Antrian Tandem.
   Gambar 5c merupakan contoh rangkaian dari antrian *server tandem* tunggal. Masukan untuk setiap antrian kecuali antrian yang pertama, merupakan keluaran untuk antrian selanjutnya. Jika diasumsikan masukan untuk antrian pertama adalah Poisson, maka *service time* untuk setiap antrian adalah eksponensial dan *waiting line* adalah *infinite*. Keluaran dari masing-masing antrian adalah distribusi Poisson yang identik dengan masukan.

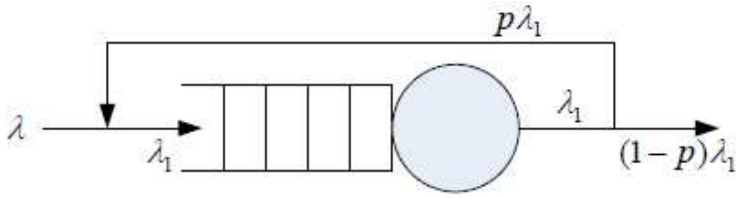
Pada saat suatu *request* menuju ke antrian berikutnya, *delay* pada antrian kedua adalah sama, jika *request* orisinil sudah melalui antrian yang pertama, dan lansung menuju antrian kedua. Jadi, antrian adalah *independent*, dan akan dianalisis satu kali pada satu waktu. Sehingga, rata-rata total *delay* dalam sistem *tandem* ini adalah sama dengan jumlah rata-rata *delay* pada setiap *stage*.



Gambar 3. Elemen Jaringan

Pada bagian ini yang akan dibahas lebih detil adalah antrian *partitioning*. Pada Gambar 6 terlihat suatu *open* jaringan antrian dengan *partitioning* dan *feedback*. Dimana eksternal *request* datang dengan *arrival rate* λ . *Request* yang sudah menyelesaikan layanan akan keluar dari sistem, atau akan kembali ke antrian untuk memperoleh layanan lagi, dengan probabilitas *p*, dengan kecepatan 1 $p\lambda$, dan digabungkan dengan *request* eksternal, sehingga *arrival rate* pada antrian ini ditunjukkan pada persamaan.

$$\lambda_i = \lambda + p\lambda_i = \frac{\lambda}{1-p}$$



Gambar 4. Open antrian dengan partitioning dan feedback [Gunther, Neil, J. (2005)];

Batasan Performansi

Penggunaan beberapa komputasi untuk menentukan batas atas dan batas bawah pada *throughput* sistem dan *response time*, sebagai fungsi dari intensitas *workload* sistem (jumlah atau *arrival rate customer*). Teknik untuk menghitung batasan performansi ini adalah *asymptotic bound* dan *balanced system bound*. Yang akan dijelaskan pada bagian ini hanya untuk *asymptotic bound*.

Untuk model dengan tipe *transaction workload*, batasan *throughput* mengindikasikan maksimum *arrival rate customer* yang dapat diproses oleh sistem, sementara batasan *response time*, menggambarkan *response time* yang terbesar dan terkecil, dimana *customer* dapat mengalami sebagai suatu fungsi dari *arival rate* sistem. Untuk model dengan tipe *batch* atau *terminal workload*, batasan mengindikasikan kemungkinan maksimum dan minimum *throughput* sistem, dan *response time* sebagai fungsi dari jumlah *customer* dalam sistem. Untuk batas atas *throughput* dan batas bawah *response time*, dihubungkan dengan batasan yang optimistik, karena mengindikasikan kemungkinan performansi yang terbaik, dan untuk batas bawah *throughput* dan batas atas *response time*, dihubungkan dengan batasan yang pesimistis, karena mengindikasikan kemungkinan performansi yang terburuk.

Aysmptotic bound analisis memberikan batasan yang optimistik dan pesimistik dari *throughput* sistem, dan *response time* pada *single class* jaringan antrian. Seperti namanya, analisis ini diperoleh dengan mempertimbangkan (*asymptotically*) kondisi kondisi ekstrim dari *load* yang ringan dan *load* yang berat. Validitas dari batasan ini hanya tergantung pada sebuah asumsi tunggal yaitu *service demand* dari *customer* pada *service center*, dan tidak tergantung pada berapa banyak *customer* lain ada di dalam sistem atau pada *service center*.

Jenis informasi yang disediakan oleh *asymptotic bound* tergantung pada *workload* sistem yaitu *open* untuk *transaction workload*, atau *closed* untuk *terminal* dan *batch workload*. Dan yang akan dijelaskan hanya untuk *transaction workload*. Untuk *transaction workload*, batasan *throughput* mengindikasikan kemungkinan maksimum *arrival rate* dari *customer*, dimana sistem dapat memproses secara sukses. Jika *arrival rate* melebihi batasan ini, maka suatu *backlog* dari *customer* yang tidak diproses akan meningkat secara kontinyu mengikuti kedatagan *job*.

Sehingga, suatu *job* yang datang akan menunggu untuk jangka waktu yang tidak tentu, karena mungkin ada beberapa *job* yang sudah ada dalam antrian, ketika *job* yang baru tersebut datang. Dalam hal ini dikatakan bahwa sistem saturasi. Batasan *throughput* merupakan *arrival rate* yang pemrosesannya mungkin terpisah dari saturasi.

Untuk menentukan batasan *throughput* adalah menggunakan hukum Utilisasi, $U_i = X_i \times S_i$ untuk setiap *i*. Jika *arrival rate* ke sistem dinotasikan sebagai λ , maka $X_i = \lambda_i \times V_i$ dan hukum Utilisasi menjadi $U_i = \lambda_i \times D_i$ yaitu *service demand* pada *resource i*. Untuk mendapatkan batasan *throughput*, jika selama semua *service center* memiliki kapasitas yang tidak digunakan, dimana utilisasi kurang dari 1, maka peningkatan *arrival rate* dapat diakomodasi. Namun, pada saat setiap *service center* menjadi saturasi, yaitu memiliki utilisasi sama dengan 1, maka seluruh sistem menjadi saturasi, karena peningkatan *arrival rate customer* tidak dapat ditangani secara sukses. Sehingga batasan *throughput* adalah *arrival rate* yang terkecil *sat* λ pada setiap saturasi *service center*. *Service center* yang saturasi pada *arrival rate* yang terendah merupakan *bottleneck service center*, yaitu *service center* dengan *service demand* terbesar. Dan *max* merupakan indeks dari *bottleneck service center* maka diperoleh[3, 5]

$$U_{max}(\lambda) = \lambda D_{max} \leq$$

$$\lambda_{sat} = \frac{1}{D_{max}}$$

$$X(\lambda) \leq \frac{1}{D_{max}}$$

$$D \leq R(\lambda)$$

Untuk *arrival rate* yang lebih besar atau sama dengan $\frac{1}{D_{max}}$, jmaka sistem adalah saturasi, sementara sistem mampu memproses *arrival rate* adalah yang kurang dari $\frac{1}{D_{max}}$

1. ANALISA PERFORMANSI SISTEM

Spesifikasi Hardware

Spesifikasi hardware yang digunakan oleh Server SIA Mercubuana adalah sebagai berikut:

1. Server HP Proliant ML370G5 (Spesifikasi pada Lampiran 1)
2. Memori 4GB

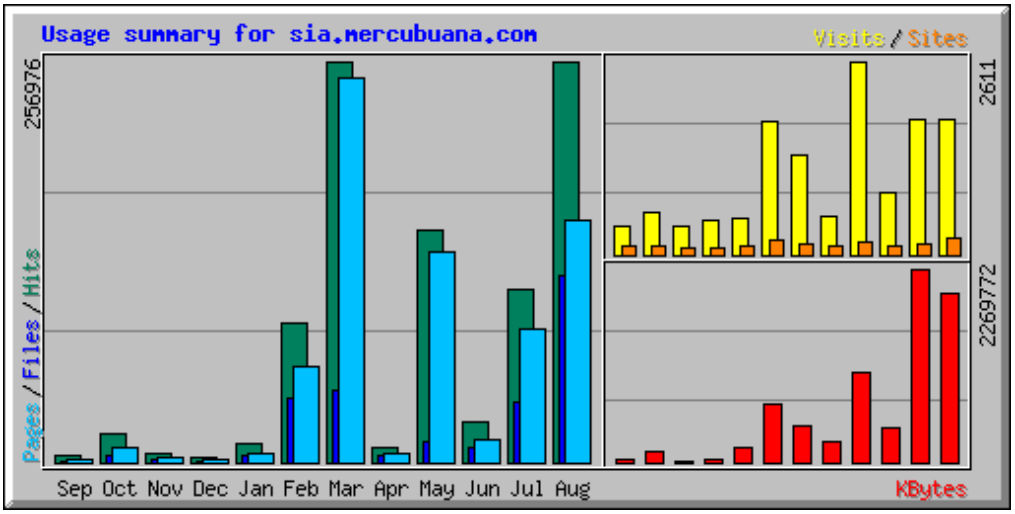
Analisis Jumlah Mahasiswa Mercubuana

Jumlah mahasiswa yang digunakan dalam analisis ini adalah rekapitulasi jumlah mahasiswa semester genap tahun akademik 2010/2011 pertanggal 6 April 2011, jam 09:00. Dimana dalam rekadpitulasi tersebut (dapat dilihat pada Lampiran 2) jumlah mahasiswa mercubuana adalah jumlah total mahasiswa program kelas karyawan ditambah jumlah total mahasiswa program kelas regular sehingga totalnya adalah: 10664 + 7947 =18611

Analisis statistik penggunaan SIA Mercubuana

Data statistik penggunaan SIA mercubuana diambil untuk Perioda 12 bulan terakhir yaitu dari September 2010 sampai Agustus 2011. Dimana data ini diambil pada tanggal 11 Agustus 2011 pada pukul 11:58 WIB dengan alamat 10.1.1.9/usage.

Summary Period: Last 12 Months
Generated 11-Aug-2011 04:11 WIT



Gambar 5. Analisis statistik penggunaan SIA Mercubuana

Tabel 1. Statistik Penggunaan SIA

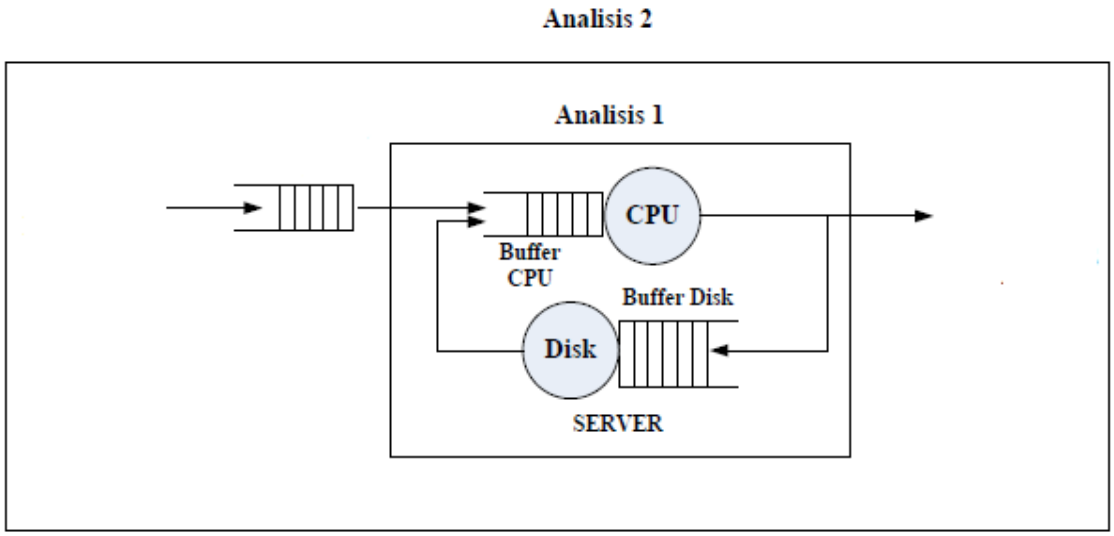
	Daily Avg						Monthly Totals			
	Hits	**Files**	**Pages**	**Visits**	**Sites**	**KBytes**	**Visits**	**Pages**	**Files**	**Hits**
Aug 2011	25586	12010	15559		219	1983958	1834	155592	120106	255864
Jul 2011	3579	1262	2774	58	145	2269772	1826	86013	39149	110964
Jun 2011	850	319	470	27	131	400568	833	14121	9589	25510
May 2011	4798	426	4348	84	164	1044284	2611	134815	13230	148759
Apr 2011	317	138	187	17	115	242639	512	5626	4150	9537
Mar 2011	8289	1498	7939	43	149	426517	1350	246120	46440	256976
Feb 2011	3185	1483	2182	63	207	692965	1790	61123	41548	89202
Jan 2011	408	141	213	17	124	163198	493	6186	4089	11834
Dec 2010	94	26	61	15	91	24828	470	1920	828	2932
Nov 2010	186	43	121	13	101	19015	393	3643	1308	5580
Oct 2010	599	158	317	18	110	120144	563	9838	4898	18578
Sep 2010	130	41	83	13	110	46146	392	2511	1246	3907
Totals						**7434034**	**13067**	**727508**	**286581**	**939643**

Perancangan Model Jaringan Antrian

Dalam perancangan model jaringan antrian ini, terdapat beberapa asumsi yang akan digunakan yaitu sebagai berikut:

1. Jumlah request adalah infinite
2. Jumlah server yang digunakan adalah 1 server
3. Proses kedatangan atau arrival rate (λ) request adalah random, dengan distribusi poisson. Nilai arrival rate ini merupakan suatu asumsi yang diperoleh dari jumlah traffic 1 tahun terakhir. Dimana nilai dari parameter *arrival rate* (λ) request didasarkan pada **usage statistics for SIA mercubuana.com** pada kondisi tersibuk yaitu pada bulan Agustus 2011 yaitu 25586 hit per hari. Untuk jumlah request per detiknya adalah sebagai berikut:
   - Diasumsikan yang mengakses server SIA adalah selama jam kerja yaitu 8 jam, sehingga request/ jam =25586/ 8 = 3198.25
   - Request per 1 detik = 3198.25/360 = 8.88

   Sehingga diperoleh nilai *arrival rate* (λ) = 8.88 request /detik.
4. Karena kedatangan request merupakan distribusi poisson, maka service rate (μ) merupakan distribusi eksponensial dan nilai ini diperoleh dari jumlah total mahasiswa mercubuana semester genap tahun akademik 2010/2011 pertanggal 6 April 2011, jam 09:00.
5. Prioritas layanan yang digunakan adalah first come first serve (FCFS)
6. Kapasitas buffer antrian server adalah finite, sehingga ada request yang akan diblok atau hilang apabila buffer penuh.
7. Model jaringan antrian adalah open queueing network model, karena request yang 252ating akan mendapatkan pelayanan 252ating atau menunggu dalam buffer jika layanan belum tersedia dan kemudian meninggalkan 252ating jika sudah mendapatkan layanan, pada penelitian ini hanya dilakukan analisis 1.



Gambar 6. Model Analisis

Analisis Performansi Resource

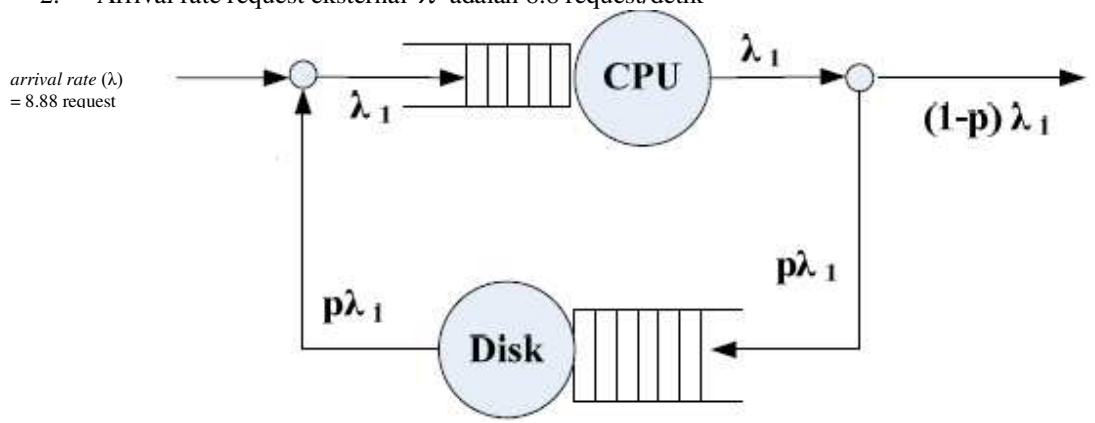
Pada analisis ini, terdapat 2 buah resource yang digunakan dalam server, yaitu CPU dan harddisk. Scenario yang terjadi dalam model jaringan antrian ini adalah sebagai berikut:

1. Terdapat eksternal request yang 253ating ke CPU dengan arrival rate adalah λ.
2. Request yang sudah dilayani oleh CPU akan memiliki 2 percabangan yaitu menuju ke harddisk untuk melaksanakan operasi I/O harddisk, dengan probabilitas *p* atau keluar dari server karena telah menyelesaikan layanan, dengan probabilitas 1-*p*.
3. Setelah menyelesaikan operasi I/O pada harddisk, request akan kembali ke antrian CPU, untuk meminta layanan berikutnya, sehingga CPU akan memiliki 2 masukan request yaitu dari eksternal dan harddisk. Arrival rate untuk CPU dinotasikan dengan λ_1. Arrival rate ke CPU merupakan gabungan dari kedua masukan tersebut, yaitu arrival rate request dari harddisk $\lambda_1 p$ dan eksternal arrival rate request λ, sehingga untuk memperoleh arrival rate pada CPU, dapat menggunakan persamaan berikut:

$$\lambda_1 = \lambda_1 p + \lambda = \frac{\lambda}{1-p}$$

Dari kondisi tersebut diatas, terlihat bahwa terdapat suatu feedback yang terjadi dalam model jaringan antrian ini. Sehingga model jaringan antrian seperti ini dapat dianalisis dengan menggunakan teorema Jackson. Deskripsi detil dari model jaringan antrian ini, dapat dilihat pada Gambar berikut. Dalam analisis model jaringan antrian ini terdapat beberapa asumsi yang digunakan yaitu sebagai berikut:

1. Analisis dilakukan dalam periode waktu 1 detik
2. Arrival rate request eksternal λ adalah 8.8 request/detik



Gambar 7. Model Analisis resource

Analisis Arrival rate (λ) resource

Dalam analisis arrival rate (λ) dari setiap resource ini, selain 2 asumsi pada sub bab 3.3 diatas, terdapat 1 asumsi lagi yang digunakan dalam analisis ini yaitu asumsi terhadap probabilitas dari percabangan yang terjadi setelah request memperoleh layanan dari CPU. Dimana dalam hal ini diasumsikan bahwa probabilitas request yang meminta layanan kembali untuk operasi I/O pada harddisk (*p*) dan request yang telah selesai diproses dari server (1-*p*) adalah variatif.

Tabel 2. Analisis Arrival rate (λ) resource

Probabilitas request meminta layanan I/O harddisk (*p*)	Arrival rate menuju CPU ($\lambda_1 = \frac{\lambda}{1-p}$)	arrival rate menuju harddisk ($p\,\lambda_1$)
10%	9.8	1.0
20%	11.0	2.2
30%	12.6	3.8
40%	14.7	5.9
50%	17.6	8.8
60%	22.0	13.2
70%	29.3	20.5
80%	44.0	35.2
90%	88.0	79.2

Analisis service time (S_i) resource

Dari gambar 3.5 diatas, terlihat bahwa terdapat 2 resource yang digunakan yaitu CPU dan harddisk. Dimana untuk melakukan analisis lebih lanjut terhadap model jaringan antrian resource ini, diperlukan parameter service time (S_i) untuk setiap resource. Service time merupakan perioda waktu dimana sebuah request menerima layanan dari suatu resource.

Analisis service time processor (S_{CPU})

Spesifikasi dari processor yang digunakan pada server adalah Dual-Core Intel® Xeon® Processor 5150 (2.66 GHz, 65 Watts, 1333 FSB). Dari spesifikasi ini diperoleh nilai parameter sebagai berikut:

a. clock rate (*f*) adalah 2.66 GHz.
b. Kecepatan transfer data adalah 10.66 GBytes/s

Performansi dari sebuah processor dapat dilihat dari waktu yang digunakan oleh processor untuk mengeksekusi sebuah program, (*T* detik/program). Untuk memperoleh nilai *T* tersebut adalah sebagai berikut:

T=ukuran rata-rata file yang diakses (Kbyte) / kecepatam data transfer processor (Kbyte/s).

Untuk rata-rata file yang diakses diambil dari **d**ata statistik penggunaan SIA mercubuana pada bulan Agustus 2011 yaitu 1983958 Kbyte. Sehingga untuk rata-rata perdetiknya adalah sebagai berikut:

a. Rata-rata per hari : 1983958/11 = 180359.8182 ➔ karena data diambil pada tanggal 11 Agustus 2011
b. Rata-rata per jam = 180359.8182 / 8 =22544.97727 ➔ diasumsikan pada jam kerja (8 jam)
c. Rata-rata per detiknya adalah 62.62493687 KByte

Sehingga diperoleh nilai T adalah

T=62.62493687 KByte / 10.66 GByte/s
T=62.62493687 KByte / 11177820.16 KByte/s
T=5.60261E-06 s = 5.6 µs

Analisis service time Harddisk (S_{Disk})

Magnetic disk merupakan komponen penting untuk setiap sistem computer. Jumlah akses informasi yang disimpan pada magnetic disk, lebih banyak dibandingkan jumlah akses informasi pada RAM. Yang menjadi ukuran performansi pada harddisk adalah service time $(\overline{S}_d)$ yaitu merupakan rata-rata waktu yang dibutuhkan oleh controller ditambah disk untuk mengakses satu blok data dari disk. Persamaan berikut dapat digunakan untuk memperoleh nilai ini.

$$\overline{S}_d = ControllerTime + P_{miss} \times (SeekTime + RotationalLatency + TransferTime)$$

Dari persamaan diatas dapat dilihat parameter yang mempengaruhi service time harddisk adalah :

a. Controller time, merupakan waktu yang diperlukan oleh sebuah controller memproses sebuah I/O request (termasuk waktu mencek cache, ditambah waktu untuk read/write sebuah blok dari/ke cache).
b. Pmiss, merupakan probabilitas dimana blok yang dimaksud tidak ada pada disk cache.
c. Seek time, merupakan waktu rata-rata yang diperlukan untuk menempatkan arm pada cylinder yang tepat
d. Transfer time adalah waktu transfer sebuah blok dari disk ke disk controller.

Spesifikasi dari harddisk yang digunakan adalah HP 500GB 3G SATA 7.2K rpm SFF (2.5-inch) Midline:

a. Transfer rate = 3 GB/s
b. Kapasitas = 500 GB
c. Seek Time = 8.2 ms
d. Spindle Time = 7200 rpm
e. Controller Time = 0.1 ms

1. Menentukan Transfer Time
2.

$$TransferTime = \frac{BlockSize}{TransferRate}$$

Untuk ukuran block size diambil dari rata-rata file yang diakses pada bulan Agustus 2011 yaitu 1983958 Kbyte. Sehingga untuk rata-rata perdetiknya adalah sebagai berikut 62.62493687 Kbyte. Dan untuk transfer rate adalah 3 GB/s. Maka diperoleh nilai transfer time nya adalah:

$$TransferTime = \frac{62.62493687\text{KB}}{3GB/s}$$

$$= \frac{0.000059724\text{GB}}{3\text{GB/s}} = 0.000019908 = 19.908\mu\text{s}$$

3. P_{miss}

Untuk nilai P_{miss} digunakan nilai untuk random workload yaitu 1

4. Seek Time

Untuk nilai SeekTime adalah 8.2 ms

5. Rotational Latency

$$RotationalLatency = 1/2 \times Disk\,\text{Re}volutionTime$$

$$Disk\,\mathrm{Re}\,volutionTime = \frac{60}{DiskSpeed}$$

$$= \frac{60}{7200} = 0.0083s = 8.3ms$$

Sehingga diperoleh,

$$RotationalLatency = \tfrac{1}{2} \times Disk\,\mathrm{Re}\,volutionTime$$

$$RotationalLatency = \tfrac{1}{2} \times 8.3ms = 4.17ms$$

Service time $\overline{S}_d$ harddisk adalah:

$$\overline{S}_d = ControllerTime + P_{miss}$$
$$\times (SeekTime + RotationalLatency + TransferTime)$$
$$\overline{S}_d = 0.1 + 1 \times (8.3ms + 4.17ms + (19.908\mu s)$$
$$\overline{S}_d = 0.1 + 1 \times (8.3ms + 4.17ms + 0.019908\mathrm{ms})$$
$$\overline{S}_d = 0.1 + 1 \times (12.489908)$$
$$\overline{S}_d = 1.1 \times 12.489908$$
$$\overline{S}_d = 13.7388988\mathrm{ms}$$

Analisis Utilisasi (U_i)Resource

Setelah diperoleh service time (*S*) untuk setiap resource yaitu 5.6µs CPU dan 13.74ms harddisk, maka utilisasi dari masing-masing resource dalam perioda waktu 1 detik, dapat diperoleh sebagai berikut.

a. Utilisasi CPU

Arrival rate CPU (λ_i) merupakan arrival rate untuk probabilitas request meminta layanan I/O harddisk (*p*) adalah 90% sampai dengan 10% dan service time CPU

$S_{CPU} = 5.6\mu\mathrm{s} = 5.6 \times 10^{-6}\,s$ sehingga diperoleh,

Tabel 3. Analisis Utilisasi CPU (U_{CPU})

Probabilitas request meminta layanan I/O harddisk (*p*)	Arrival rate menuju CPU ($\lambda_1 = \frac{\lambda}{1-p}$)	Utilisasi CPU ($U_{CPU} = \lambda_i \times S_{CPU}$)	
0%	9.8	0.00005488	0.005488%
0%	11.0	0.0000616	0.00616%
30%	12.6	0.00007056	0.007056%
40%	14.7	0.00008232	0.008232%
50%	17.6	0.00009856	0.009856%
60%	22.0	0.0001232	0.01232%
70%	29.3	0.00016408	0.016408%
80%	44.0	0.0002464	0.02464%
90%	88.0	0.0004928	0.04928%

b. Utilisasi Harddisk

Arrival rate Harddisk ($p\lambda_i$) merupakan arrival rate untuk probabilitas request meminta layanan I/O harddisk (*p*) adalah 10% sampai dengan 90% dan service time Harddisk $S_{Disk} = 13.74\text{ms} = 13.74 \times 10^{-3}\ s$ sehingga diperoleh,

Tabel 5. Analisis rata-rata jumlah kunjungan (V_i) *ke resource.*

Probabilitas request meminta layanan I/O harddisk (*p*)	Rata-rata kunjungan ke resource $V = \frac{1}{1-p}$
10%	1.11
20%	1.25
30%	1.43
40%	1.67
50%	2.00
60%	2.50
70%	3.33
80%	5.00
90%	10.00

Analisis service demand resource (D_i)

Persamaan untuk memperoleh service demand untuk setiap resource diperoleh dari hasil perkalian antara service time (S), dengan rata-rata jumlah kunjungan (V), sehingga diperoleh service demand untuk setiap resource adalah sebagai berikut:

$$D_i = V_i \times S_i$$

Untuk service time CPU dan harddisk adalah sebagai berikut:

$$S_{CPU} = 5.6\mu\text{s} = 5.6 \times 10^{-6}\ s$$

$$S_{Disk} = 13.74\text{ms} = 13.74 \times 10^{-3}\ s$$

Tabel 6. Analisis service demand resource (D_i)

Rata-rata kunjungan ke resource $V = \frac{1}{1-p}$	Service demand CPU	Service demand CPU
1.11	$6.22 \times 10^{-6}\ s$	$1.525 \times 10^{-2}\ s$
1.25	$7.00 \times 10^{-6}\ s$	$1.718 \times 10^{-2}\ s$
1.43	$8.01 \times 10^{-6}\ s$	$1.965 \times 10^{-2}\ s$
1.67	$9.35 \times 10^{-6}\ s$	$2.295 \times 10^{-2}\ s$
2.00	$1.12 \times 10^{-5}\ s$	$2.748 \times 10^{-2}\ s$
2.50	$1.40 \times 10^{-5}\ s$	$3.435 \times 10^{-2}\ s$
3.33	$1.86 \times 10^{-5}\ s$	$4.575 \times 10^{-2}\ s$
5.00	$2.80 \times 10^{-5}\ s$	$6.870 \times 10^{-2}\ s$
10.00	$5.60 \times 10^{-5}\ s$	$1.374 \times 10^{-1}\ s$

Dari nilai service demand diatas, maka dapat ditentukan maksimum arrival rate $\lambda_{sat} = \frac{1}{D_{\max}}$ ke server dalam perioda 1 detik, dimana diperoleh service demand max $D_{\max}$ adalah service demand harddisk. Dalam hal ini diambil untuk jumlah kunjungan ke resource adalah 10 yaitu $1.374 \times 10^{-1} s$. sehingga diperoleh nilai saturasi server adalah:

$$\lambda_{sat} = \frac{1}{D_{\max}} = \frac{1}{1.374 \times 10^{-1} s} = 7.278$$

Nilai diatas mengindikasikan bahwa server mampu memproses arrival rate request dalam perioda 1 detik adalah maksimal 7.278 request/detik.

Analisis Throughput (X)

Untuk mendapatkan nilai throughput dari sistem adalah dengan menggunakan persamaan $X = (1-p)\lambda_1$

Tabel 7. Analisis Throughput (X)

Probabilitas request meminta layanan I/O harddisk (p)	Arrival rate menuju CPU ($\lambda_1 = \frac{\lambda}{1-p}$)	$X = (1-p)\lambda_1$
10%	9.8	8.82
20%	11.0	8.8
30%	12.6	8.82
40%	14.7	8.82
50%	17.6	8.8
60%	22.0	8.8
70%	29.3	8.79
80%	44.0	8.8
90%	88.0	8.8

PENUTUP

Dari analisis yang telah dilakukan maka dapat ditarik kesimpulan yaitu server Sistem Informasi Akademik saat ini mampu melayani request per detik sebanyak 7.278 request. Dibandingkan dengan arrival rate yang ada yaitu 8.88 request per detik, maka Sistem Informasi Akademik saat ini sudah kurang maksimum, sehingga kemungkinan lost request adalah 21.98%.

DAFTAR PUSTAKA

[1] Adan, Ivo and Resing, Jacques (2001). *Queueing Theory*. Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven Netherlands

[2] Bai, Ying-Wen, and Cheng, Chien-Yung (2004). The Performance Estimation by Queueing Network Models for a Web-based Medical Information System, *Proceding of the 17th IEEE Symposium on Computer-Based Medical System (CBMS'04)*.

[3] Breuer, L. and Baum, D. (2005). *An Introduction to Queueing Theory and Matrix-Analytic Methods*, Springer, AA Dordrecht, Netherlands.

[4] Daigle, John, N (2005). Queueing Theory with Applications to Packet Telec*ommunication*, Springer, Boston.