BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian tentang antrian yang telah dilakukan menyebutkan bahwa terdapat beberapa model antrian yang dapat digunakan untuk menyelesaikan permasalahan terhadap antrian. Penelitian yang dilakukan oleh Hideaki Takagi dari Universitas Tsukuba, Jepang, menjelaskan bahwa selama terdapat pengunjung dan *server /* pelayan yang berwujud manusia, teori antrian akan selalu dipakai. Namun teori baru tentang antrian sangat dibutuhkan untuk mengatasi permasalahan terhadap antrian terhadap manusia (Takagi, 2014).

Salah satu teori baru tentang antrian untuk manusia yakni menerapkan teori antrian yang digunakan pada cara kerja komputer. Penelitian ini dilakukan pada pelayanan rumah sakit. Dengan melakukan metode perkiraan lama perawatan untuk setiap pasien, pasien baru dapat mengetahui kapan kamar rawat di rumah sakit akan kosong dan bisa dipakai. Metode perkiraan yang dilakukan berdasarkan pada data jenis penyakit yang dialami pasien. Dengan menerapkan model antrian *multi channel* dengan server berupa kamar rawat dan *pengunjung* berupa pasien. Pengunjung akan dipisahkan berdasarkan jenis penyakit untuk kemudian dilayani oleh server yang berbeda pula sesuai dengan jenis penyakit (Takagi, 2014).

Penelitian tentang antrian diterapkan pula pada studi kasus yang berbeda, yakni pada pusat pelayanan bank. Universitas Science dan Teknologi Nanjing melakukan penelitian tentang antrian di pusat pelayanan bank menggunakan metode *Bussiness Process Reengineering* (BPR). BPR melakukan optimasi kerja server untuk melayani pengunjung. Optimasi yang dilakukan dalam BPR yakni dengan menentukan jumlah server dan tujuan server yang berbeda-beda untuk setiap pengunjung. Para pengunjung akan dilayani oleh server yang berbeda-beda berdasarkan jenis pelayanan yang akan pengunjung inginkan. Inilah tujuan dari metode BPR yakni menentukan server mana yang digunakan untuk setiap jenis pelayanan yang berbeda, misalnya: pengunjung yang ingin menabung akan mengantri dan dilayani di server nomor 1,2,3 dan 4. Pengunjung yang ingin ke

pengunjung service akan dilayani di server nomor 5. Kunci dari BPR yakni menentukan nomor *server* yang akan digunakan untuk pengunjung dengan kebutuhan yang berbeda dan menentukan jumlah server yang harus aktif berdasarkan jumlah pengunjung dan panjang antrian (Hao dan Yifei, 2011).

Penelitian lain tentang pemanfaatan teori penjadwalan untuk digunakan pada masalah publik adalah penelitian yang dilakukan oleh Cleiton Taufemback dan Sergio Da Silva. Penelitian tersebut meneliti tentang teori penjadwalan digunakan untuk mengoptimalkan *management* sebuah bank. Penelitian ini memanfaatkan teknologi telepon selluler untuk mengoptimalkan antrian dan meningkatkan produktivitas pelayanan (Taufemback dan Da Silva, 2012).

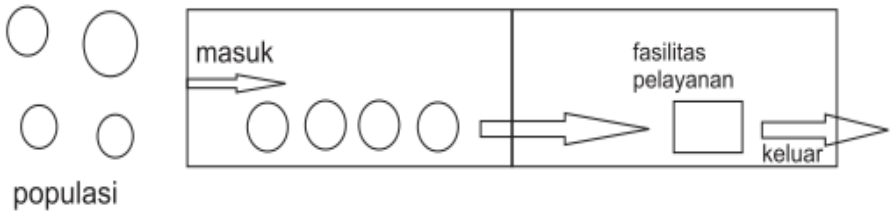
Permasalahan yang akan terjadi jika metode *first come first served* dikombinasikan dengan *priority scheduling* adalah bagaimana cara menyelipkan antrian dengan prioritas pada antrian *first come first served.* Namun, ada penelitian yang dapat melakukan hal ini, yakni penelitian dari Walraefens. Penelitian ini meneliti tentang penjadwalan kedatangan dan keberangkatan kereta api. Kedatangan dan keberangkatan kereta api sebenarnya sudah dilakukan sesuai dengan jadwalnya, namun suatu saat bisa terjadi sebuah kereta api diharuskan untuk datang atau berangkat terlebih dahulu. Hal ini dilakukan dengan cara men-*delay* jadwal kereta api yang akan disisipkan (Walraefens dan Bruneel, 2017).

2.2. Dasar Teori

2.2.1. Sistem Antrian

Sistem antrian merupakan suatu himpunan yang terdiri dari pengunjung, pelayanan dan suatu aturan yang mengatur kedatangan pengunjung. Terdapat lima buah komponen ciri-ciri sistem antrian yaitu : pola kedatangan para pengunjung, pola waktu pelayanan, jumlah layanan, kapasitas fasilitas untuk menampung para pengunjung dan aturan pengunjung dilayani. Sedangkan struktur umum dari model antrian memiliki dua komponen utama yaitu: garis tunggu atau antrian (*Queue*), dan Fasilitas pelayanan (*Service Facility*). Pengunjung menunggu di fasilitas pelayanan, menerima pelayanan hingga selesai. Model antrian diperlukan untuk memperkirakan kinerja sistem antrian. Dalam antrian, ukuran kinerja dapat diukur

dengan beberapa parameter yakni utilisasi pelanggan dan server, panjang antrian, dan waktu tunggu pelanggan. Adapun model struktur antrian secara umum ditunjukkan pada Gambar 2.1 (Jay dan Barry, 2005).



Gambar 2.1. Struktur umum model antrian

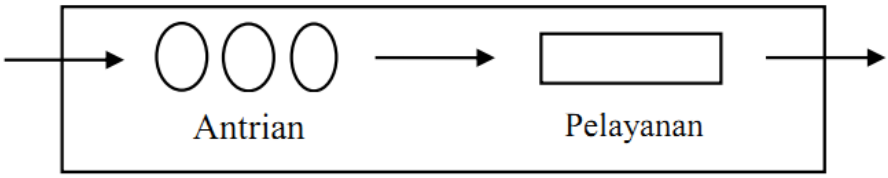
Gambar 2.1 adalah model antrian secara umum. Model antrian terdiri dari 3 komponen utama, yaitu kedatangan atau masukan sistem, antrian, dan fasilitas pelayanan (Aminudin, 2005). Komponen kedatangan memiliki beberapa karakteristik kedatangan yang berbeda-beda. Karakteristik kedatangan tersebut diantaranya adalah ukuran populasi kedatangan bisa terbatas atau tak terbatas, pola kedatangan bisa teratur atau berpola dan bisa tidak teratur atau secara acak, dan perilaku kedatangan contohnya ada pelanggan yang memiliki perilaku bersabar menunggu antrian hingga dilayani atau pelanggan yang memiliki perilaku tidak bersabar dengan meninggalkan sistem antrian karena antrian yang panjang.

Komponen antrian memiliki beberapa karakteristik antrian diantaranya adalah panjang antrian dan aturan pelayanan. Panjang antrian biasanya tak terbatas kecuali oleh waktu, namun dapat terbatas misalnya seperti panjang antrian di restaurant terbatas karena pengaruh jumlah meja pada restauran tersebut. Sedangkan untuk aturan antrian biasanya menggunakan aturan *first in first out* (FIFO), pelanggan yang pertama mengantri akan mendapat layanan pertama. Namun suatu saat aturan antrian bisa menggunakan aturan lain, seperti antrian di rumah sakit yang memprioritaskan situasi darurat.

Fasilitas pelayanan biasanya dikelompokkan dalam jumlah jalur, jumlah *server* atau pemberi layanan, dan jumlah tahapan. Fasilitas pelayanan biasanya digambarkan dalam sebuah struktur antrian. Struktur antrian adalah cara menyimpan atau merepresentasikan antrian (Herjanto, 2014). Terdapat beberapa struktur antrian yakni seperti skema berikut.

a. Satu jalur dan satu tahap pelayanan *(Single Channel Single Phase)*

Single Channel berarti hanya ada satu jalur yang memasuki sistem pelayanan atau ada satu fasilitas pelayanan. *Single Phase* berarti hanya ada satu pelayanan. Skema antrian *Single Channel Single Phase* ditunjukkan pada Gambar 2.2.

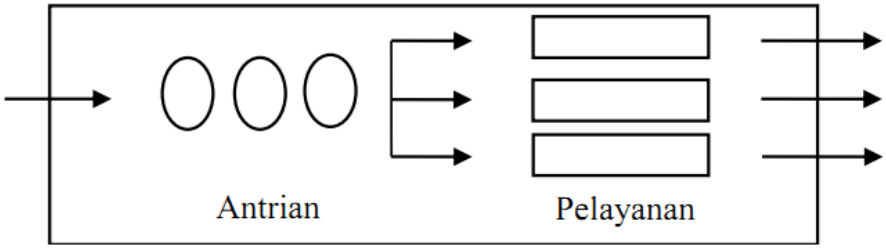


Gambar 2.2 Skema antrian *Single Channel Single Phase*

Gambar 2.2 menggambarkan fasilitas layanan tunggal, yakni hanya terdapat satu server. Contoh fasilitas seperti ini dapat ditemui di antrian anjungan tunai mandiri (ATM) yang hanya bisa dioperasikan secara bergantian.

b. Banyak jalur dengan satu tahap pelayanan *(Multi Channel Single Phase)*

Istilah *multi phase* menunjukkan ada dua atau lebih pelayanan yang dilaksanakan secara berurutan. Skema antrian *Multi Channel Single Phase* ditunjukkan pada gambar 2.3.

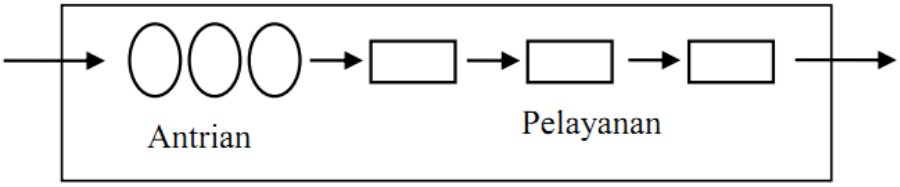


Gambar 2.3 Skema antrian *Multi Channel Single Phase*

Gambar 2.3 menggambarkan fasilitas layanan dengan jalur ganda, yakni terdapat beberapa *server*, paralel, dan identik, baik dalam satu jalur atau dalam beberapa jalur. Contoh antrian ini adalah antrian pencucian mobil, antrian pelayanan *customer service* atau *teller* di bank.

c. Satu jalur dengan banyak tahap pelayanan *(Single Channel Multi Phase)*

Sistem *Multi Channel – Single Phase* terjadi kapan saja di mana ada dua atau lebih fasilitas pelayanan dialiri oleh antrian tunggal. Skema antrian *Single Channel Multi Phase* ditunjukkan pada Gambar 2.4.

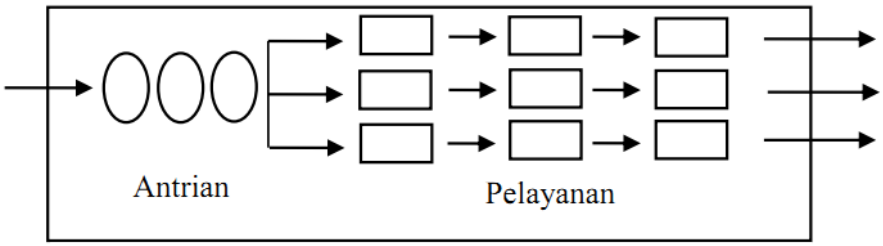


Gambar 2.4 Skema antrian *Single Channel Multi Phase*

Gambar 2.4 menggambarkan fasilitas layanan bertahap, pelanggan harus melalui beberapa tahap atau tempat layanan. Contoh model ini adalah antrian pada kantor pembayaran pajak bumi dan bangunan (PBB), pelanggan harus melakukan tahapan peenyerahan formulir, pembayaran pajak, dan penerimaan bukri pembayaran atau kwitansi dalam tempat yang berbeda atau secara bertahap.

d. Banyak jalur dan banyak tahap pelayanan *(Multi Channel Multi Phase)*

Sistem *Multi Channel – Multi Phase*, yakni model antrian lebih lanjut dari antrian *single channel multi phase*. Pada antrian ini terdapat tahapan antrian dan juga jenis antrian yang berbeda. Skema antrian *Multi Channel Multi Phase* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Skema antrian *Multi Channel Multi Phase*

Gambar 2.5 menggambarkan fasilitas layanan banyak *channel* dan bertahap. Sebagai contoh, *heregistrasi* para mahasiswa di universitas, pelayanan kepada pasien di rumah sakit mulai dari pendaftaran, diagnosa, penyembuhan sampai pembayaran. Setiap sistem mempunyai beberapa fasilitas pelayanan pada setiap

tahapnya dan juga terdapat lebih dari satu jenis antrian seperti antrian poli penyakit dalam, poli anak, poli THT, dan sebagainya.

2.2.2. Algoritma Penjadwalan CPU

Penjadwalan adalah sebuah proses pengambilan keputusan yang banyak digunakan oleh perusahaan dan industri. Penjadwalan mengatur banyaknya sumber daya untuk bekerja menghasilkan manfaat yang optimal (tidak banyak menganggur dan tidak banyak antrian). Penjadwalan CPU adalah proses dimana menentukan antrian yang siap untuk dieksekusi oleh CPU (Pinedo, 2016). Terdapat beberapa algoritma penjadwalan CPU diantaranya:

1. Algoritma penjadwalan *First Come First Served* (FCFS)

Proses yang pertama kali meminta jatah waktu untuk menggunakan CPU akan dilayani terlebih dahulu. Metode FCFS ini adalah metode yang umum digunakan untuk menyelesaikan permasalahan antrian publik (Platz dan Osterdal, 2015). Contoh proses dalam komputer beserta waktu pelayanannya ditunjukkan pada tabel 2.1.

Tabel 2.1 Daftar proses antrian FCFS

Proses	Burst Time
P1	24
P2	3
P3	3

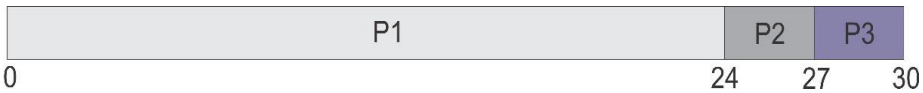
dengan :

P1= Proses ke-1 masuk

P2= Proses ke-2 masuk

P3= Proses ke-3 masuk

Kemudian proses antrian akan dilayani / dieksekusi dengan algoritma *first come first served*. Proses dapat dilihat pada *gant chart* yang ditunjukkan pada Gambar 2.6.



Gambar 2.6 *Gant chart* proses antrian FCFS

Gambar 2.6 menunjukkan *gant chart* dari proses FCFS, proses P1 datang terlebih dahulu dengan beban proses sepanjang 24. Langkah kedua, proses P2 yang datang setelah P1 akan dilayani setelah P1 selesai dilayani. Proses ini akan terus dilakukan hingga semua proses selesai. Penerapannya dalam sistem ini adalah pengunjung yang datang terlebih dahulu dan meminta pelayanan secara langsung akan mendapat jatah pelayanan terlebih dahulu.

2. Algoritma Penjadwalan *Shortest Job First*(SJF)

Algoritma ini digunakan ketika proses CPU yang mempunyai waktu terpendek akan mendapat mendapat prioritas. Seandainya dua proses atau lebih mempunyai waktu yang sama maka algoritma FCFS digunakan untuk menyelesaikan masalah tersebut. Prinsip algoritma penjadwalan ini adalah, proses yang memiliki CPU *burst* paling kecil dilayani terlebih dahulu. Contoh proses dalam komputer beserta waktu pelayanannya ditunjukkan pada Tabel 2.2.

Tabel 2.2 Daftar proses antrian SJF

Proses	Arrival Time	Burst Time
P1	0	10
P2	2	2

Kemudian proses antrian akan dilayani / dieksekusi dengan algoritma *shortest job first*. Proses dapat dilihat pada *gant chart* yang ditunjukkan pada Gambar 2.7.

Gambar 2.7 *Gant chart* proses antrian SJF

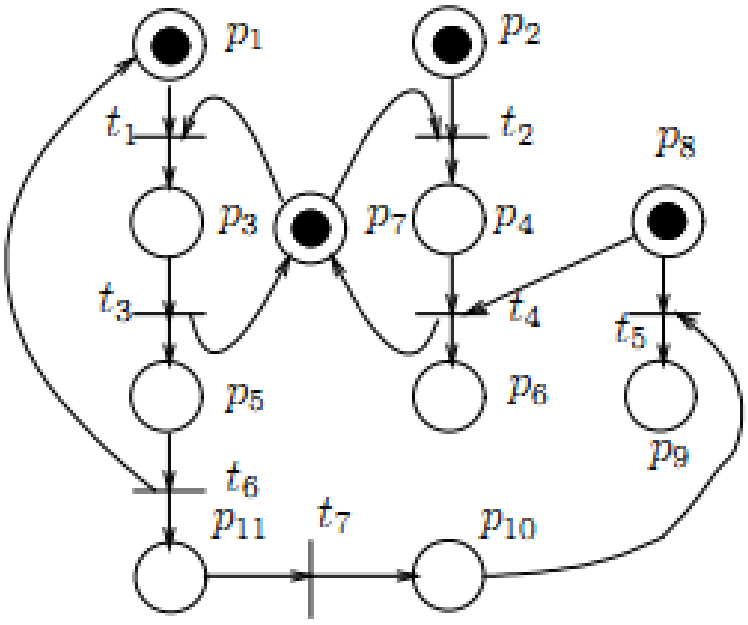
Gambar 2.7 menunjukkan *gant chart* dari SJF, proses P1 datang pada waktu ke-0 akan dilayani hingga ada proses lain datang yakni proses P2. Langkah berikutnya adalah membandingkan sisa dari beban proses P1 dengan proses yang baru datang yakni proses P2. Pilih proses yang memiliki beban paling sedikit yakni proses P2, kemudian proses P2 akan dilayani terlebih dahulu menyisipkan dari proses P1. Jika tidak ada proses lagi yang memiliki beban lebih sedikit, maka lanjutkan dan selesaikan proses P1 dan proses lain yang belum selesai.

Algoritma ini optimal jika digunakan, tetapi sulit untuk diimplementasikan karena sulit mengetahui CPU *burst* selanjutnya. Ada dua skema dalam SJFS ini yaitu:

- *Non preemptive*, ketika CPU memberikan kepada proses itu tidak bisa ditunda hingga selesai.
- *Preemptive*, bila sebuah proses datang dengan waktu proses lebih rendah dibandingkan dengan waktu proses yang sedang dieksekusi oleh CPU maka proses yang waktunya lebih rendah mendapatkan prioritas. Skema ini disebut juga *Short Remaining Time First* (SRTF).

3. Algoritma Penjadualan *Priority Schedulling* (Jadwal Prioritas).

Penjadualan SJF adalah kasus khusus untuk algoritma penjadualan Prioritas. Prioritas dapat diasosiasikan masing-masing proses dan CPU dialokasikan untuk proses dengan prioritas tertinggi. Untuk proritas yang sama dilakukan dengan FCFS. Ilustrasi jadwal prioritas dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi Pemodelan Antrian dengan *Priority Scheduling*

Pada gambar 2.8 dapat dilihat bahwa antrian p7 dapat mendahului antrian yang lain yang sudah ada dalam antrian. Hal ini dapat terjadi jika antrian p7 memiliki prioritas antrian yang lebih besar. Dalam penelitian ini antrian prioritas akan diberikan kepada pengunjung yang memilih layanan pemesanan waktu tertentu, namun antrian prioritas harus ikut mengantri dalam antrian FCFS yang sudah ada.

4. Algoritma Penjadwalan *Round Robin.*

Algoritma *Round Robin* (RR) dirancang untuk sistem time sharing. Algoritma ini mirip dengan penjadual FCFS, namun *preemption* ditambahkan untuk switch antar proses. Antrian *ready* diperlakukan atau dianggap sebagai antrian sirkular. CPU mengelilingi antrian *ready* dan mengalokasikan masing-masing proses untuk interval waktu tertentu sampai satu *time slice*/ *quantum*

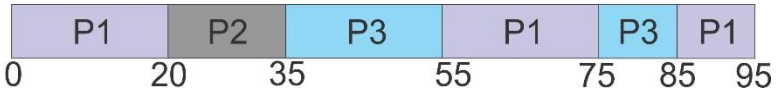
Berdasarkan beberapa algoritma penjadualan, algoritma yang mungkin diterapkan untuk antrian manusia dan untuk sistem ini adalah algoritma FCFS dan SJF. Contoh proses dalam komputer beserta waktu pelayanannya ditunjukkan pada tabel 2.3.

Tabel 2.3 Daftar proses antrian *round robin*

Proses	**Burst Time**
P1	50
P2	15
P3	30

Keterangan : *time slice* = 20 mili detik

Kemudian proses antrian akan dilayani / dieksekusi dengan algoritma *round robin*. Proses dapat dilihat pada *gant chart* yang ditunjukkan pada Gambar 2.9.



Gambar 2.9 *Gant chart* proses antrian *round robin*

Gambar 2.9 menunujukkan *gant chart* dari proses *round robin*. Proses *round robin* memiliki *time slice* yakni waktu untuk bergantian melayani setiap proses. Proses P1 memiliki beban 50 akan dilayani 20 terlebih dahulu bergantian dengan

setelahnya. Demikian proses *round robin*, bergantian melayani sebuah proses berdasarkan besarnya *time slice.*

2.2.3. Notasi Kendall

Untuk memudahkan memahami karakteristik dan memperjelas bagaimana sistem antrian dimodelkan suatu sistem antrian digunakan Notasi *Kendall* (Gross dan Harris, 1998). Berikut adalah persamaan karakteristiknya :

$$\boldsymbol{A / S / m / B / K / D} \tag{2.1}$$

dengan :

A : fungsi distribusi dari waktu antar kedatangan *(distribution function of inter**A**rrival times)*

S : fungsi distribusi dari waktu layanan *(distribution function of **S**ervice times)*

m : banyaknya pemberi layanan *(number of servers)*

B : kapasitas *buffer (**B**uffer capacity)/* kapasitas sistem, merupakan banyak pelanggan maksimal yang dapat berada pada sistem, termasuk yang sedang dilayani

K : ukuran populasi, banyak pelanggan yang mungkin datang

D: disiplin/ jenis/ cara pemberian layanan (contohnya first-come-first-serve, dan lainnya).

Misalkan suatu model antrian dideskripsikan ***M / M / 4 / 30 / 200 / FCFS***, maka model antrian tersebut memiliki karakteristik distribusi jeda waktu antar kedatangan pelanggan adalah eksponensial, distribusi waktu layanan untuk masing-masing pelanggan adalah eksponensial, model antrian memiliki 4 pemberi layanan (4 buah *server*), model antrian memiliki kapasitas 30 pelanggan, yang terdiri atas: 4 yang dilayani (dari m = 4) dan 26 pengantri (kapasitas antrian adalah 26), model dapat melayani 200 pelanggan dalam satu *cycle*, pelanggan dilayani berdasarkan aturan *first-come-first-served* (Averill dan David, 2014).

2.2.4. Formula Model Antrian

Langkah selanjutnya adalah memasukkan perhitungan ke formula model sistem antrian. Formula model sistem antrian ditunjukkan pada persamaan 2.1 hingga 2.7 (Kakiay, 2004).

Langkah 1 penunjukan variabel jenis data, ditunjukkan pada persamaan 2.1 dan 2.2.

$$\lambda = \frac{1}{T1/N} \tag{2.2}$$

$$\mu = \frac{1}{T2/N} \tag{2.3}$$

dengan:
λ = tingkat kedatangan pengunjung
μ = tingkat pelayanan
T1 = total waktu antar kedatangan
T2 = total waktu pelayanan pengunjung
N = total pengunjung

Tingkat kedatangan pengunjung adalah nilai yang didapat dari satu dibanding dengan jumlah dari selisih waktu kedatangan antar pengunjung yang dibagi dengan jumlah pengunjung dalam satu rangkaian antrian. Sedangkan tingkat pelayanan adalah nilai dari satu dibanding dengan jumlah dari lama setiap pengunjung dilayani dibagi dengan jumlah pengunjung dalam satu rangkaian antrian.

Langkah II menentukan nilai peluang masa menganggur (P_0)

$$P_0 = (\sum_{n=0}^{c=1} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!\left(1 - \left(\frac{\lambda}{c.\mu}\right)\right)})^{-1} \tag{2.4}$$

dengan:

P_0 = peluang masa menganggur

λ = tingkat kedatangan pengunjung

μ = tingkat pelayanan

c = jumlah fasilitas pelayanan

Peluang masa menganggur adalah nilai dari peluang seorang atau sebuah pelayan / *server* menganggur atau *idle time*. Dari data ditunjukkan semakin banyak jumlah pengunjung dalam antrian maka nilai dari Po semakin kecil, begitu sebaliknya semakin sedikit jumlah pengunjung dalam antrian maka nilai Po semakin besar.

Langkah III menentukan L_q , yaitu jumlah rata - rata pengunjung dalam antrian.

$$\boldsymbol{L_q} = \frac{\left(\frac{\lambda}{\mu}\right)^c . (\frac{\lambda}{c.\mu})}{c!\,(1 - (\frac{\lambda}{c.\mu}))^2} P_0 \tag{2.5}$$

dengan:

L_q = jumlah rata-rata pengunjung dalam antrian

P_0 = peluang masa menganggur

λ = tingkat kedatangan pengunjung

μ = tingkat pelayanan

c = jumlah fasilitas pelayanan

Jumlah rata-rata pengunjung dalam antrian adalah nilai rata-rata untuk jumlah pengunjung yang berada dalam setiap antrian dalam waktu tertentu.

Langkah IV menentukan L_s , yaitu jumlah rata - rata pengunjung dalam sistem.

$$L_s = L_q + \frac{\lambda}{\mu} \tag{2.6}$$

dengan:

L_s = jumlah rata-rata pengunjung dalam sistem

L_s = jumlah rata-rata pengunjung dalam antrian

λ = tingkat kedatangan pengunjung

μ = tingkat pelayanan

Jumlah rata-rata pengunjung dalam sistem adalah nilai rata-rata untuk jumlah pengunjung yang berada dalam setiap antrian dalam waktu tertentu ditambah dengan pengunjung yang sedang dilayani dalam sistem.

Langkah V menentukan W_q , yaitu waktu rata - rata pengunjung dalam antrian.

$$W_q = \frac{L_q}{\lambda} \tag{2.7}$$

dengan :

W_q = waktu rata-rata pengunjung dalam antrian

L_q = jumlah rata-rata pengunjung dalam antrian

λ = tingkat kedatangan pengunjung

μ = tingkat pelayanan

Waktu rata-rata pengunjung dalam antrian adalah waktu rata-rata lama mengantri dari setiap pengunjung di dalam antrian.

Langkah VI menentukan W_s , yaitu waktu rata-rata pengunjung dalam sistem.

$$W_s = W_q + \frac{1}{\mu} \tag{2.8}$$

dengan :

W_s = waktu rata-rata pengunjung dalam sistem

W_q = waktu rata-rata pengunjung dalam antrian

μ = tingkat pelayanan

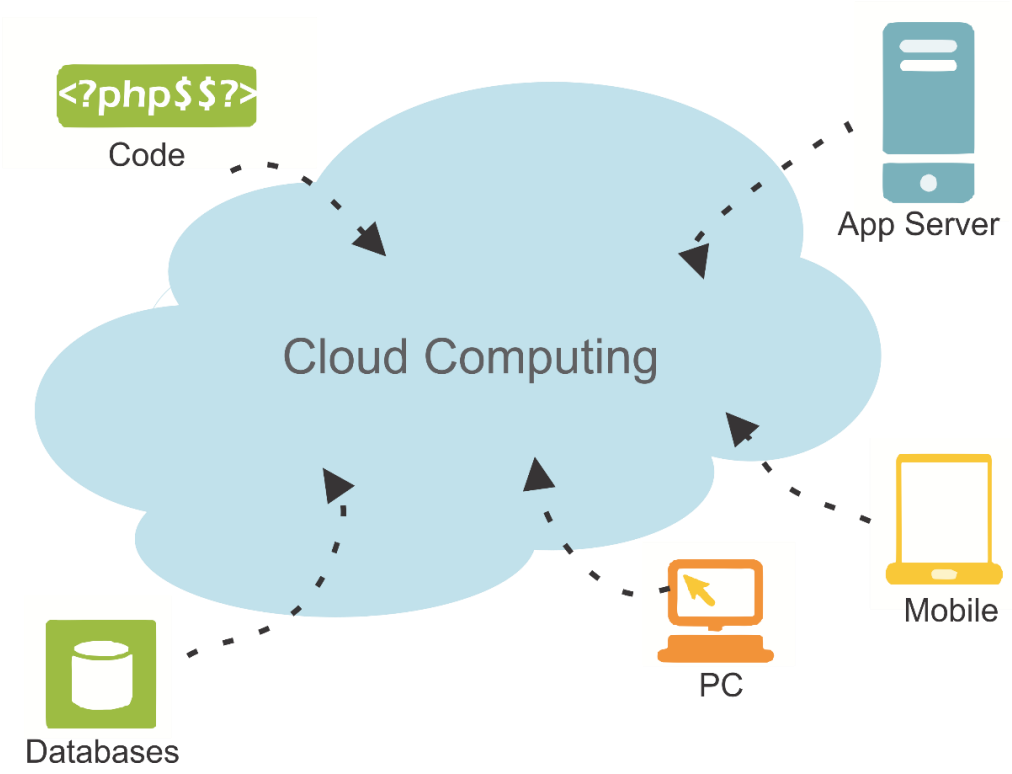
Waktu rata-rata pengunjung dalam sistem adalah waktu rata-rata lama mengantri dari setiap pengunjung di dalam antrian ditambah dengan waktu lama dilayani dalam sistem.

2.2.5. Cloud Computing

Cloud adalah sebuah inovasi platform teknologi untuk menyediakan layanan berskala tinggi kepada pengguna. Penggunaan *cloud* dapat dikendalikan dari jarak jauh karena teknologi ini menggunakan internet sebagai aksesnya. Oleh karena itu sebuah sistem seharusnya dibangun menggunakan teknologi ini sehingga pengguna dapat menggunakan sistem tersebut dimanapun dan kapanpun (Erl dan Puttini, 2013).

Cloud computing merupakan gabungan pemanfaatan antara teknologi komputasi dalam suatu jaringan dengan pengembangan berbasis internet yang berfungsi untuk menjalankan aplikasi melalui komputer. Tidak semua sistem atau aplikasi yang terkoneksi ke internet adalah sistem *cloud computing*.

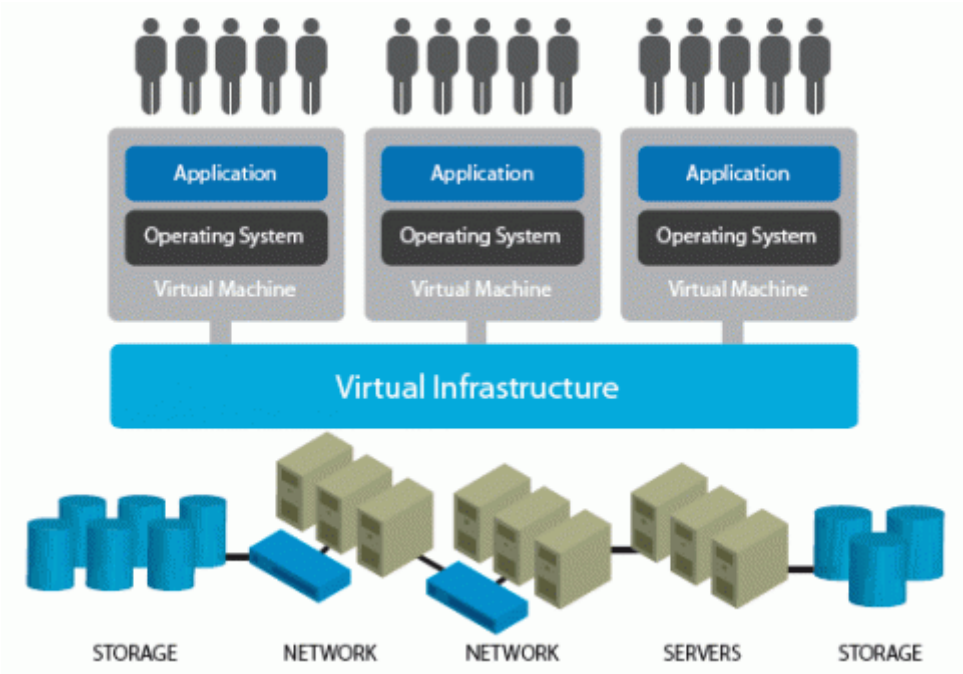
Sistem antrian akan diimplementasikan menggunakan metode *cloud computing* sebagai metode untuk membangun jaringan, *server*, penyimpanan, aplikasi dan layanan dalam membangun sistem. Sistem ini dikonfigurasikan dengan basis data *MySql* dan bahasa pemrograman *PHP*. Skema cloud computing ditunjukkan pada Gambar 2.10.



Gambar 2.10 Arsitektur c*loud computing*

Gambar 2.10 menunjukkan arsitektur *cloud computing*. Semua perangkat terhubung ke internet untuk saling terhubung. Perangkat *client* berupa PC dan mobile dapat mengakses layanan dimanapun dengan terhubung ke internet. Perangkat *server* akan selalu *on-line* untuk bisa diakses kapanpun, juga harus terhubung dengan internet agar bisa diakses dimanapun. Implikasinya adalah aplikasi atau sistem di *server* berupa *database, content, code* akan bisa diakses oleh *client* melalui konsep *cloud computing* kapanpun dan dimanapun.

Inti dari teknologi *cloud computing* adalah adanya virtualisasi, dimana virtualisasi tersebut merupakan *emulator* yang dapat digunakan untuk menjalankan beberapa sistem operasi diatas sistem operasi yang sedang digunakan (Valls dkk., 2014). Penjelasan *cloud computing* dapat dilihat dalam Gambar 2.11.



Gambar 2.11 Arsitektur *virtual infrastructure*

Gambar 2.11 menjelaskan tentang *virtual infrastructure* yang merupakan kunci dari sebuah *cloud*. Di dalam sebuah *server* pengguna dapat berbagi sumber daya untuk mengakses *server*. Sebuah *server* dapat membagi media penyimpanan, jaringan dan juga sistem operasi dengan menggunakan *virtual machine*. Hal ini memungkinkan

sebuah *server* dapat digunakan untuk melayani lebih dari satu pengguna atau aplikasi.