BAB II

TEORI PENUNJANG

2.1 *Distribusi Probabilitas*

2.1.1 Proses *Poisson*

Proses *Poisson* merupakan salah satu contoh dari proses-proses stokastik sebagai proses perhitungan untuk suatu kejadian. Proses perhitungan adalah suatu fungsi yang menunjukkan jumlah kumulatif dari kejadian yang telah terjadi setiap saat. Proses *Poisson* merupakan suatu jenis proses perhitungan yang dapat diaplikasikan untuk menghitung kedatangan-kedatangan pelanggan.

Jika X(t) adalah proses perhitungan maka, proses *Poisson* didasarkan pada asumsi-asumsi berikut:

1. Suatu peristiwa dapat terjadi secara acak (random) dan pada waktu atau titik dalam ruang yang mana saja.
2. Banyaknya kedatangan/kejadian selama selang waktu tertentu tidak tumpang tindih *(non-overlapping)*.
3. Banyaknya kejadian selama selang waktu tertentu adalah variabel random yang bebas.
4. a. Probabilitas bahwa lebih dari satu kedatangan pada selang yang kecil dt dapat diabaikan, atau dapat diartikan sebagai P[dua atau lebih kedatangan dalam selang dt] = 0(dt).

   b. Probabilitas bahwa secara tepat satu kedatangan akan terjadi dalam sebuah selang yang lebarnya dt adalah sebanding dengan λdt atau dapat diartikan sebagai P[satu kedatangan dalam selang dt] = λdt + 0(dt), dimana λ adalah laju kedatangan.

c. Probablitas bahwa secara tepat nol kedatangan akan terjadi dalam selang waktu dt adalah sebanding dengan 1-λdt atau dapat diartikan P[tidak ada kedatangan dalam selang dt] = 1- [λdt + 0(dt)], dengan λ sebagai laju kedatangan.

2.1.2 Distribusi *Poisson*

Distribusi *Poisson* menyatakan fungsi probabilitas untuk sejumlah kejadian-kejadian dari suatu peristiwa di dalam suatu selang waktu atau ruang tertentu.

Theorema 2.1

Jika X(t) adalah proses perhitungan yang menunjukkan kejadian-kejadian dari suatu peristiwa dalam selang waktu t dan memenuhi asumsi-asumsi pada proses *Poisson*, maka :

$$P[X(t) = x] = \frac{e^{-\lambda t}(\lambda t)^x}{x!} \quad , \text{untuk } x = 0,1,2,...$$

Bukti:

Misal t adalah suatu titik dalam selang waktu setelah waktu 0, maka selang waktu (0, t] mempunyai lebar t dan dalam selang waktu (t, t + dt] mempunyai lebar dt.

Misal $P_n(dt) = P[\, X(dt) = n\,]$

$P[\, X(dt) = n\,]$ = P[secara tepat n kejadian di dalam selang dt]

maka

$P_o(\, t + dt\,)$ = P[tidak ada kejadian dalam selang (0, t +dt]]

= P[tidak ada kejadian dalam selang (0, t] dan tidak ada kejadian dalam selang (t, t + dt]].

Dengan menggunakan asumsi nomor 3 maka

$P_o(t + dt)$ = P[tidak ada kejadian dalam selang (0, t]] P[tidak ada kejadian dalam selang (t, t + dt]]

Sehingga di dapat persamaan

$$P_o(t+dt) = P_o(t)\ P_o(dt) \qquad \text{............ (2.1)}$$

Selanjutnya untuk

$P_o(dt)$ = P[tidak ada kejadian dalam selang (t, t+dt]]

= 1-P[satu atau lebih kejadian dalam selang (t, t+dt]]

= 1 - P[satu kejadian dalam selang (t, t + dt]] - P[lebih dari satu kejadian dalam selang (t, t + dt]]

$= 1 - \lambda dt - 0(dt) - 0(dt)$

Jadi :

$$P_o(t+dt) = P_o(t)[1 - \lambda dt - 0(dt) - 0(dt)]$$

$$= P_o(t) - \lambda P_o(t)dt - P_o(t)0(dt) - P_o(t)0(dt)$$

$$\frac{P_0(t+dt) - P_0(t)}{dt} = -\lambda P_0(t) - P_0(t)\frac{0(dt)+0(dt)}{dt} \qquad \text{...... (2.2)}$$

Dengan menggunakan asumsi 4.a, maka dengan mencari limit dari persamaan (2.2) didapat :

$$\lim_{dt\to 0}\frac{P_0(t+dt) - P_0(t)}{dt} = \lim_{dt\to 0}\left[-\lambda P_0(t) - P_0(t)\frac{0(dt)+0(dt)}{dt}\right]$$

atau

$$P_o'(t) = -\lambda P_o(t) \qquad \text{............... (2.3)}$$

Solusi dari persamaan diferensial (2.3) adalah :

$$P_o(t) = e^{-\lambda t} \qquad \text{............... (2.4)}$$

Dengan menggunakan cara yang sama dan menggunakan persamaan (2.1) dicari untuk n = 1, maka didapat:

$$P_1(t+dt) = P_1(t)\,P_o(dt) + P_o(t)\,P_1(dt)$$

$$= P_1(t)[1-\lambda dt-0(dt)]+P_o(t)[\lambda dt+0(dt)]$$

$$\frac{P_1(t+dt)-P_1(t)}{dt} = -\lambda P_1(t) - P_0(t)$$

$$P_1'(t) = -\lambda P_1(t) + \lambda P_o(t) \qquad \ldots\ldots\ldots\ldots\ldots \quad (2.5)$$

Dengan cara yang sama untuk $n = 2$ diperoleh :

$$P_2'(t) = -\lambda P_2(t) + \lambda P_1(t)$$

analog untuk $n = 3$, maka diperoleh :

$$P_3'(t) = -\lambda P_3(t) + \lambda P_2(t)$$

Sehingga untuk setiap $n = 1,2, \ldots$ diperoleh :

$$P_n'(t) = -\lambda P_n(t) + \lambda\, P_{n-1}(t)$$

Persamaan (2.5) diselesaikan dengan metode faktor integral sebagai berikut:

$$P_1'(t) + \lambda P_1(t) = \lambda\, P_o(t)$$

Subtitusi untuk nilai $P_0(t)$ dari persamaan (2.4), maka diperoleh :

$$P_1(t)\,e^{\lambda t} = \int e^{\lambda t}\,\lambda\, e^{-\lambda t} dt$$

$$= \lambda \int dt$$

$$= \lambda\, t$$

$$P_1(t) = \lambda\, t\, e^{-\lambda t} \qquad \ldots\ldots\ldots\ldots\ldots \quad (2.6)$$

Analog dengan cara diatas untuk $n = 2$ didapat :

$$P_2(t) = \frac{(\lambda t)^2 e^{-\lambda t}}{2!}$$

Untuk $n = 3$ didapat:

$$P_3(t) = \frac{(\lambda t)^3 e^{-\lambda t}}{3!}$$

Sehingga untuk setiap n didapat :

$$P_n(t) = \frac{e^{-\lambda t}(\lambda t)^n}{n!} \quad n = 0,1,2,...$$

Jadi untuk suatu proses perhitungan X(t), dalam selang waktu t dan untuk x > 0, memiliki probabilitas sebesar:

$$P[X(t) = x] = \frac{e^{-\lambda t}(\lambda t)^x}{x!} \qquad \text{..............} \quad (2.7)$$

Persamaan diatas disebut sebagai fungsi probabilitas untuk sebuah variabel random *Poisson* dengan mean λt.

Definisi 2.1

Suatu variabel random X mempunyai distribusi *Poisson* jika :

$$f(x) = \begin{cases} \dfrac{(\lambda t)^x}{x!} e^{-\lambda t} & \text{untuk } x = 0,1,2,3, \ldots \\ 0 & \text{untuk } x < 0 \end{cases}$$

dimana λt = parameter > 0.

Fungsi diatas dikenal sebagai fungsi probabilitas *Poisson* dengan mean λt.

Definisi 2.2

Fungsi distribusi probabilitas dengan variabel random X yang berdistribusi *Poisson* adalah :

$$F(x) = \sum_{n=0}^{x} f(n) = \sum_{n=0}^{x} \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \quad x = 0,1,2, \ldots$$

dimana λt = parameter > 0.

Theorema 2.2

Jika variabel random X berdistribusi *Poisson* dengan parameter λt, maka :

$$E[X] = \lambda t$$

$$Var[X] = \lambda t$$

Bukti:

$$E[X] = \sum_{x=0}^{\infty} x f(x)$$

$$= \sum_{x=0}^{\infty} x \frac{e^{-\lambda t}(\lambda t)^x}{x!}$$

$$= e^{-\lambda t} \sum_{x=0}^{\infty} \frac{x(\lambda t)^x}{x!}$$

$$= e^{-\lambda t} \sum_{x=0}^{\infty} \frac{(\lambda t)^x}{(x-1)!}$$

$$= \lambda t \sum_{x=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^{x-1}}{(x-1)!}$$

substitusi untuk x-1 = m, jika x = 1 maka m = 0 sehingga:

$$E[X] = \lambda t \sum_{m=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^m}{m!}$$

$$= \lambda t$$

$$E[X^2] = \sum_{x=0}^{\infty} x^2 f(x)$$

$$= \sum_{x=0}^{\infty} x^2 \frac{e^{-\lambda t}(\lambda t)^x}{x!}$$

$$= e^{-\lambda t} \sum_{x=0}^{\infty} x^2 \frac{(\lambda t)^x}{x(x-1)!}$$

$$= e^{-\lambda t}\sum_{x=0}^{\infty} x\frac{(\lambda t)^x}{(x-1)!}$$

$$= \lambda t\sum_{x=0}^{\infty} x\frac{e^{-\lambda t}(\lambda t)^{x-1}}{(x-1)!}$$

substitusi untuk $x-1 = m$

$x = m + 1$

jika $x = 1$ maka $m = 0$ sehingga :

$$E[X^2] = \lambda t\sum_{m=0}^{\infty}(m+1)\frac{e^{-\lambda t}(\lambda t)^m}{m!}$$

$$= \lambda t\sum_{m=0}^{\infty} m\frac{e^{-\lambda t}(\lambda t)^m}{m!} + \lambda t\sum_{m=0}^{\infty}\frac{e^{-\lambda t}(\lambda t)^m}{m!}$$

$$= (\lambda t)^2 + \lambda t$$

$$Var[X] = E[X^2] - [E[X]]^2$$

$$= (\lambda t)^2 + \lambda t - (\lambda t)^2$$

$$= \lambda t$$

2.1.3 Distribusi Eksponensial

Distribusi eksponensial adalah suatu distribusi kontinu. Pada distribusi kontinu suatu variabel random memiliki fungsi probabilitas yang disebut fungsi kepadatan probabilitas *(probability density function).*

Definisi 2.3

Suatu random variabel X berdistribusi Eksponensial jika fungsi densitas dari X adalah :

$$f(x) = \lambda e^{-\lambda x} \qquad \text{untuk } x > 0$$

dimana λ parameter > 0

Definisi 2.4

Fungsi distribusi probabilitas dengan variabel random X yang berdistribusi Exponensial dan parameter λ adalah :

$$F(x) = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x} \quad , x \geq 0$$

Theorema 2.3

Jika X berdistribusi Eksponensial dengan parameter λ maka :

$$E[X] = \frac{1}{\lambda}$$

$$Var[X] = \frac{1}{\lambda^2}$$

Bukti:

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx$$

$$= \int_0^{\infty} x \lambda e^{-\lambda x} dx$$

$$= -\int_0^{\infty} x d e^{-\lambda x}$$

$$= -[x e^{-\lambda x} \Big|_0^{\infty} - \int_0^{\infty} e^{-\lambda x} dx]$$

$$= -[x e^{-\lambda x} \Big|_0^{\infty} + \frac{1}{\lambda} e^{-\lambda x} \Big|_0^{\infty}]$$

$$= \frac{1}{\lambda}$$

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f(x)dx$$

$$= \int_{0}^{\infty} x^2 \lambda e^{-\lambda x} dx$$

$$= -\int_{0}^{\infty} x^2 de^{-\lambda x}$$

$$= -[x^2 e^{-\lambda x}\Big|_0^{\infty} - \int_{0}^{\infty} 2xe^{-\lambda x} dx]$$

$$= -[x^2 e^{-\lambda x}\Big|_0^{\infty} + \frac{2}{\lambda}\int_{0}^{\infty} xde^{-\lambda x}]$$

$$= -[x^2 e^{-\lambda x}\Big|_0^{\infty} + \frac{2}{\lambda}(xe^{-\lambda x}\Big|_0^{\infty} - \int_{0}^{\infty} e^{-\lambda x} dx)]$$

$$= -[x^2 e^{-\lambda x}\Big|_0^{\infty} + \frac{2}{\lambda}(xe^{-\lambda x} + \frac{1}{\lambda}e^{-\lambda x})\Big|_0^{\infty}]$$

$$= \frac{2}{\lambda^2}$$

$$Var[X] = E[X^2] - [E[X]]^2$$

$$= \frac{2}{\lambda^2} - \frac{1}{\lambda^2}$$

$$= \frac{1}{\lambda^2}$$

2.1.4 Keterkaitan antara Distribusi *Poisson* dan Distribusi Eksponensial

Jika X adalah variabel random yang lebih besar dari t, oleh distribusi eksponensial diartikan sebagai:

$P[\ X > t] = P[$ tidak ada kejadian dalam selang waktu t $]$

$= 1\text{-}F(t)$

$$= 1 - \int_0^t \lambda e^{-\lambda x} dx$$

$$= 1 - (1 - e^{-\lambda t})$$

$= e^{-\lambda t}$, dengan λ sebagai laju kejadian.

Probabilitas tidak ada kejadian dalam selang waktu t oleh distribusi *Poisson* diartikan sebagai :

P[X(t)=0] = P[secara tepat 0 kejadian didalam selang waktu t]

$$= \sum_{n=0}^{0} \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

$= e^{-\lambda t}$, dengan λ sebagai laju kejadian.

Sehingga probabilitas untuk variabel random X yang lebih besar t identik terhadap probabilitas 0 kejadian selama selang t. Kemudian dapat disimpulkan bahwa selang waktu kedatangan untuk proses *Poisson* dengan laju λ adalah variabel random eksponensial yang independen dengan mean $1/\lambda$

2.2 Teori Antrian

Saat ini teori antrian banyak diterapkan dalam bidang bisnis, industri, transportasi dan lain-lain. Pada prinsipnya sistem operasi juga menerapkan teori antrian. Dimana tiap-tiap *job* yang akan diproses dimasukkan dalam suatu antrian. Tujuan penggunaan teori antrian adalah untuk merancang fasilitas pelayanan, untuk mengatasi permintaan pelayanan yang berfluktuasi secara random dan menjaga keseimbangan antara biaya (waktu tunggu) pelayanan dan biaya (waktu) yang diperlukan selama antri. Komponen-komponen dasar proses antrian adalah diterangkan pada bagian dibawah ini.

2.2.1 Kedatangan

Setiap masalah antrian melibatkan kedatangan. Unsur ini sering dinamakan proses input. Proses input meliputi sumber kedatangan *(calling population)*, dan cara terjadinya kedatangan yang umumnya proses random.

Definisi 2.5

Waktu kedatangan adalah waktu pada saat suatu *job* tiba / datang ke dalam antrian.

2.2.2 Pelayanan

Pelayanan atau mekanisme pelayanan dapat terdiri dari satu atau lebih pelayan, atau satu atau lebih fasilitas pelayanan. Mekanisme pelayanan dapat hanya terdiri dari satu pelayan dalam satu fasilitas pelayanan, hal ini biasa ditemui dalam sistem operasi yang menangani suatu proses yang menggunakan *processor* tunggal.

Definisi 2.6

Waktu keberangkatan adalah waktu pada saat *job* menyelesaikan pelayanan dan meninggalkan *processor.*

Definisi 2.7

Waktu keberangkatan dari antrian adalah waktu pada saat *job* meninggalkan antrian untuk dilayani oleh *procesor.*

Definisi 2.8

Waktu pelayanan adalah waktu keberangkatan *job* dari *processor* dikurangi dengan waktu keberangkatan *job* dari antrian.

2.2.3 Antri

Timbulnya antrian terutama tergantung dari sifat kedatangan dan proses pelayanan, sebagai penentu dalam proses antrian adalah disiplin antri. Disiplin antri merupakan aturan keputusan yang menjelaskan cara melayani pengantri, misalnya datang lebih awal akan dilayani lebih dulu atau FCFS.

Definisi 2.9

Waktu tunggu adalah jumlah waktu yang dihabiskan *job-job* dalam antrian menunggu giliran untuk diproses.

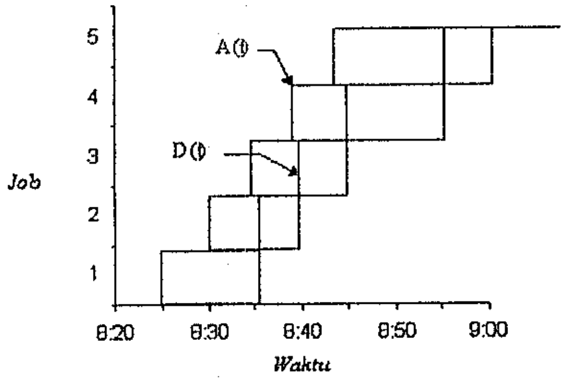
2.2.4 Teorema Little

Teorema Little adalah teorema yang banyak diaplikasikan terhadap disiplin antrian, khususnya bagaimana untuk menghitung waktu tunggu rata-rata dalam antrian. Teorema ini di formulasikan oleh **J.D.C Little** pada tahun 1961. Untuk memahami tentang teorema Little, maka sebelumnya diberikan beberapa pengertian dasar yang digunakan dalam antrian.

Sebagai contoh seorang operator komputer di sebuah bank mulai menyalakan komputer pada pukul 8.20 WIB, kemudian seorang karyawan mulai memakai komputer dengan melakukan proses-proses sebagai berikut :

Job	Waktu kedatangan	Waktu keberangkatan dari antrian	Waktu keberangkatan
1.	8:25	8:25	8:35
2.	8:30	8:35	8:40
3.	8:34	8:40	8:45
4.	8:39	8:45	8:55
5.	8:43	8:55	9:00

Gambar 2.1 merupakan diagram untuk kedatangan kumulatif *job* ke dalam antrian dan waktu keberangkatan *job* dari antrian, yaitu waktu pada saat *job* mulai dilayani. Panjang antrian adalah arah vertikal pada kurva D(t) dan waktu tunggu adalah arah horisontal. Proses pelayanan diasumsikan bersifat FCFS.



Gambar 2.1
Diagram kedatangan job

Definisi 2.10

A(t) def jumlah kedatangan kumulatif selama waktu t.

D(t) def jumlah keberangkatan dari antrian, yaitu waktu pada saat *job* mulai dilayani.

Dari contoh diatas *job* datang pertama kali pada pukul 8:25 dan langsung dilayani. Kemudian *job* ke dua datang sebelum *job* pertama selesai dilayani. Kemudian pada saat *job* pertama selesai dilayani 2 *job* sudah berada dalam antrian. Pada pukul 8:40 maka A(t) = 4 *job* dan D(t) = 3 *job*.

Definisi 2.11

$A^{-1}(n)$ def waktu kedatangan *job* ke n.

$D^{-1}(n)$ def waktu keberangkatan *job* ke n dari antrian.

Dari contoh waktu kedatangan *job* ke 2 atau $A^{-1}(2)$ = 8:30, dan waktu keberangkatan *job* ke ke 2 atau $D^{-1}(2)$ = 8:40.

Definisi 2.12

W(n) def waktu tunggu yaitu waktu yang dihabiskan *job* ke n berada dalam antrian.

Dari contoh waktu tunggu *job* ke 4 atau W(4) = 6 menit.

Waktu tunggu rata-rata dapat diartikan sebagai jumlah rata-rata waktu yang digunakan seluruh *job* untuk menunggu dalam antrian. Dari gambar 2.2 (a) jumlah seluruh waktu yang digunakan *job* pertama sampai *job* ke lima untuk menunggu dalam antrian adalah luas seluruh persegi panjang yang horisontal.

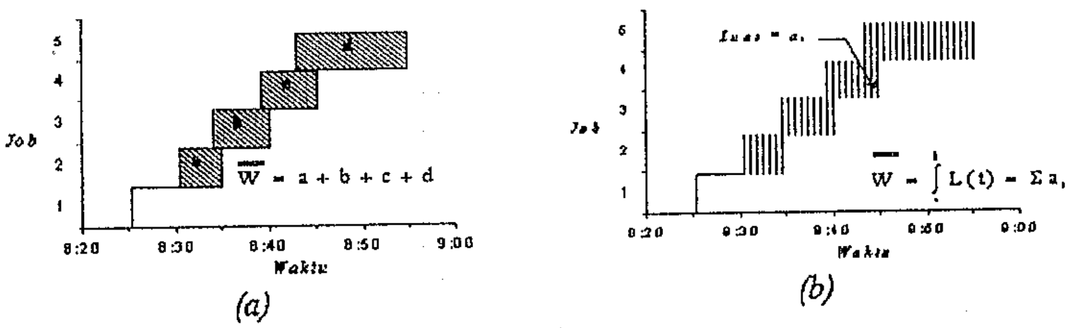
Misalnya jumlah *job* yang datang dari pukul 08:25 sampai dengan 09:00 adalah N, karena *job* berjalan secara diskrit, maka dapat didefinisikan :

Definisi 2.13

W def waktu tunggu rata-rata seluruh *job* berada dalam antrian atau dapat dituliskan:

$$W = \frac{\sum_{n=1}^{N} W(n)}{N} \qquad \text{.........................} \quad (2.8)$$

Dari contoh, banyaknya *job* atau N=5 dan waktu tunggu rata-rata untuk ke lima *job* adalah W=(0+5+6+6+12)/5 = 5.8 menit.



Gambar 2.2

Dua metode untuk menghitung waktu tunggu total semua job dalam antrian

Definisi 2.14

L(t) <u>def</u> banyaknya *job* yang berada dalam antrian pada saat waktu t atau dapat dituliskan :

$$L(t) = A(t)-D(t) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.9)$$

Pada pukul 8:40 dimana A(t) = 4 *job* dan D(t) = 2 *job*, maka L(t)=2 *job*.

Rata-rata panjang antrian dapat diartikan sebagai jumlah rata-rata *job* dalam antrian selama interval waktu yang diberikan. Dari gambar 2.2 (b) jumlah seluruh *job* yang berada dalam antrian dari pukul 08:25 sampai dengan 09:00 adalah luas seluruh persegi panjang kecil-kecil yang diarsir vertikal. Misalkan a adalah awal interval dalam satuan waktu dan b adalah akhir interval juga dalam satuan waktu, karena waktu berjalan secara kontinu, maka dapat didefinisikan :

Definisi 2.15

L <u>def</u> rata-rata banyaknya *job* yang berada dalam antrian atau dapat dituliskan :

$$L = \frac{\int_a^b L(t)dt}{b-a} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.10)$$

Dari contoh didapat data sebagai berikut :

Interval waktu	L(t)
08:25 - 08:30	0
08:30 - 08:34	1
08:34 - 08:35	2
08:35 - 08:39	1
08:39 - 08:40	2
08:40 - 08:43	1
08:43 - 08:45	2
08:45 - 08:55	1
08:55 - 09:00	0

L(t) = 0 terjadi selama 10 menit, L(t) = 1 terjadi selama 21 menit dan L(t) =2 terjadi selama 4 menit. Integral dari L(t) sama dengan 1 x 21 + 2 x 4 = 29 *job*-menit. Rata-rata panjang antrian selama interval waktu 08:25 - 09:00 adalah 29/35 = 0,83 *job*.

Berikut ini akan diberikan ilustrasi untuk gambar 2.1 dan gambar 2.2. Gambar 2.1 adalah luasan yang dibentuk oleh kurva A(t) dan D(t) dalam hal ini kurva A(t) dan D(t) sesuai dengan data pada contoh. Luasan yang dibentuk oleh kurva A(t) dan D(t) akan berlaku untuk semua disiplin antrian, termasuk FCFS. Hal ini cukup beralasan karena setiap *job* yang akan dilayani pada semua disiplin antrian selalu mengalami status antri, sehingga setiap *job* akan memiliki waktu tunggu sebelum akhirnya *job* tersebut dilayani. Pada gambar 2.1 luasan yang dibentuk merupakan jumlah total waktu tunggu semua *job* dalam antrian. Untuk *job* pertama tidak membentuk luasan karena langsung dilayani. Pada gambar 2.2 (a) jumlah total waktu tunggu dihitung secara vertikal, sesuai dengan penambahan *job* yang datang, sehingga total waktu tunggu semua *job* sama dengan jumlah seluruh luasan a, b, c, dan d. Pada gambar 2.2 (b) jumlah total waktu tunggu dihitung secara horisontal, sesuai dengan perubahan waktu, sehingga jumlah total waktu tunggu sama dengan jumlahan luasan kecil a_i , karena waktu berjalan secara kontinu, maka jumlahan semua luasan a_i dapat dihitung dengan integral. Arah vertikal pada gambar 2.2 (b) adalah banyaknya *job* yang sedang menunggu selama interval waktu yang kecil a_i, sehingga jika dihitung untuk seluruh interval waktu yang diberikan a sampai b, maka jumlah semua luasan a_i adalah $\int_a^b L(t)dt$. Luasan yang dihasilkan pada gambar 2.2 (a) maupun 2.2 (b) akan memberikan nilai yang sama yaitu jumlah total waktu tunggu semua *job* dalam antrian.

Teorema 2.4 **(Teorema Little)**

Rata-rata *job* yang berada dalam antrian sebanding dengan waktu tunggu rata-rata dalam antrian dan laju kedatangan selama interval waktu t.

Bukti :

Jika $\overline{\overline{W}}$ adalah jumlah total waktu tunggu semua *job* dalam antrian, maka dari gambar 2.2 (a) diperoleh :

$$\overline{\overline{W}} = a+b+c+d$$

dan dari gambar 2.2 (b) diperoleh :

$$\overline{\overline{W}} = \int_a^b L(t)\,dt$$

Dari definisi dasar tentang waktu tunggu rata-rata W (definisi 2.13), waktu tunggu rata-rata *job* dalam antrian dapat dituliskan:

$$W = \frac{\overline{\overline{W}}}{N} \qquad \text{.....................} \quad (2.11)$$

dan dari definisi dasar tentang panjang antrian rata-rata L (definisi 2.15), panjang antrian rata-rata *job* selama interval [a,b] dapat dituliskan:

$$L = \frac{\overline{\overline{W}}}{b-a} \qquad \text{...................} \quad (2.12)$$

Dengan mensubtitusikan nilai $\overline{\overline{W}}$ dari persamaan (2.11) ke (2. 12) diperoleh L(b-a) = WN atau dapat dituliskan:

$$\frac{L}{W} = \frac{N}{b-a} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.13)$$

Jika didefinisikan λ sebagai laju kedatangan selama interval [a,b] atau λ = N/(b-a), maka persamaan 2.13 dapat ditulis :

$$L=\lambda W \qquad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2.14)$$

Persamaan 2.14 dikenal sebagai teorema **Little** yang dapat diartikan bahwa rata-rata *job* yang berada dalam antrian sebanding dengan waktu tunggu rata-rata dalam antrian dan laju kedatangan selama interval waktu t. Dari contoh waktu tunggu rata-rata semua *job* adalah 5,8 menit, dan λ = 5/35 = 0,143 *job* tiap menit, sehingga L = 0,143 x 5,8 = 0,83 *job*.

2.3 Sistem Operasi

Definisi 2.16

Sistem operasi adalah suatu perangkat lunak yang mengontrol komponen/peralatan lain dalam suatu sistem komputer sehingga *user* (pemakai komputer untuk selanjutnya disebut *user*) dapat melakukan pekerjaan-pekerjaan menyimpan data, mengolah data, mencetak dan lain-lain.

Pada sebuah komputer banyak sekali terdapat perlengkapan-perlengkapan yang digunakan untuk memudahkan suatu aplikasi seperti *processor, printer, disk drive, memory, network interface* dan lain sebagainya. Salah satu tugas dari sistem operasi adalah melakukan pengontrolan terhadap *processor*, memori, dan perlengkapan I/O lainnya.

Bila pada waktu yang bersamaan terdapat dua program yang sedang dijalankan pada sebuah komputer dan secara bersamaan pula ingin melakukan proses mencetak pada printer yang sama, kemungkinan yang terjadi adalah baris pertama mungkin tercetak dari program I, kemudian baris kedua akan tercetak dari program II. Untuk menghindari kejadian tersebut maka sistem operasi mengadakan pengaturan terhadap kerja *processor*, memori, dan peralatan I/O lainnya.

Pada skripsi ini penulis akan membahas tentang bagaimana sistem operasi melakukan pengontrolan terhadap kerja peralatan I/O, yang salah satunya dengan melakukan penjadualan terhadap suatu program yang akan menggunakan *processor*. Karena tidak mungkin dua program atau lebih menggunakan alat yang sama pada waktu yang bersamaan, maka dengan adanya penjadualan pada sistem operasi tersebut program-program yang akan diproses akan membentuk suatu antrian sebelum menggunakan *processor*. Pelayanan terhadap program-program tersebut akan dilakukan berdasarkan prioritas, sehingga akan ada satu program yang menunggu giliran untuk diproses.

2.3.1 Komponen Penjadualan Pada Sistem Operasi

Penjadualan adalah fungsi yang mendasar dalam sistem operasi, karena hampir semua komponen dasar komputer dijadual terlebih dahulu sebelum digunakan / diproses. *Processor* adalah salah satu komponen yang utama dari sebuah komputer. Selain digunakan untuk menjalankan *job-job* atau program, *processor* juga digunakan untuk menginisialisasikan kegiatan sistem lainnya. *Processor* juga harus tanggap terhadap kesalahan-kesalahan, permintaan program, maupun pada interupsi I/O.

2.3.1.1 Status Proses

Definis 2.17

Proses adalah program yang sedang diproses yang terdiri dari program yang diproses, data dan stack dari program , program counter dan semua informasi lain yang diperlukan untuk menjalankan program.

Sebuah program yang sedang diproses dapat berubah-ubah statusnya. Status dari proses tersebut ditentukan oleh kegiatan yang dilakukan paling akhir. Sehingga setiap proses tersebut mungkin saja berada dalam status :

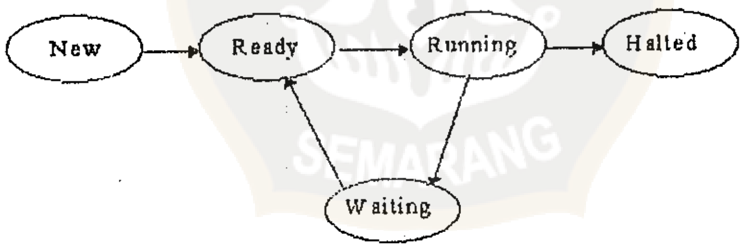
new : status proses yang baru diberikan oleh user

active : status proses yang sedang diproses

waiting : status proses yang sedang menuggu untuk menggunakan processor.

halted : status proses yang sudah selesai diproses.

Pada kenyataannya jika *processor* digunakan oleh beberapa proses, maka sebuah proses yang sedang aktif dapat dalam keadaan menunggu atau dalam keadaan diproses. Proses dalam keadaan menunggu untuk menggunakan processor disebut *ready*. Sedangkan proses yang sedang menggunakan processor disebut *running*., Gambar 2.3 memperlihatkan status-status proses pada sistem operasi..



Gambar 2.3
Status proses dalam sistem operasi

2.3.1.2 *Proses Control Blok (PCB)*

Definisi 2.18

Process Control Block adalah struktur data yang memuat informasi yang mengijinkan sistem operasi meletakkan (menentukan lokasi) semua informasi kunci tentang suatu proses.

Informasi-informasi yang dimuat oleh PCB antara lain :

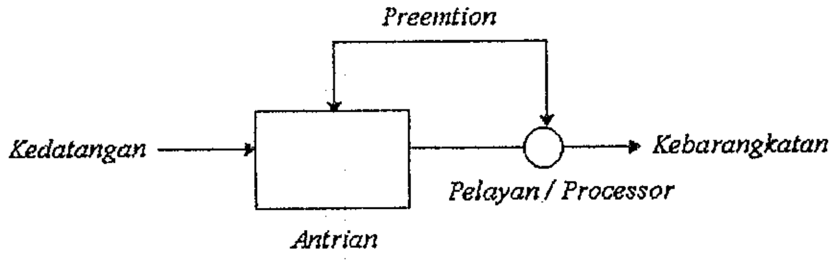
- status proses (*new* , *ready* , *running* , *waiting* , *atau halted*)
- *program counter* (menunjukkan alamat instruksi selanjutnya dari suatu proses untuk dijalankan / diproses)
- *processor register.*
- identifikasi (informasi manajemen memori , perhitungan , status I/O)
- informasi penjadualan *processor* (termasuk prioritas suatu proses, pointer untuk antrian penjadualan dll)

2.3.2 Antrian Penjadualan

Sasaran utama dalam *multiprogramming* adalah supaya proses dapat dijalankan sekaligus dengan maksud memaksimalkan utilitas *processor.* Pada sistem *processor* tunggal tidak pernah lebih dari satu proses yang dijalankan pada waktu yang bersamaan. Jika ada, maka proses lainnya harus menunggu sampai *processor* dalam keadaan bebas dan kemudian proses tersebut dijadual ulang kembali. Proses yang dalam keadaan siap dan sedang menunggu gilirannya untuk diproses oleh *processor*, akan membentuk suatu barisan dinamakan *ready queue.*

2.3.3 Model Sistem Antrian Pada Sistem Operasi

Gambar 2.4 menunjukkan model sistem antrian pada sebuah komputer dengan *processor* tunggal. *Job-job* akan datang pada sistem ketika *job* tersebut siap untuk diproses. Jika sebuah *job* datang pada saat *processor* sedang melayani *job* yang lain, maka *job* tersebut kemudian akan menunggu sampai *processor* selesai melayani *job* sebelumnya.



Gambar 2.4

Model sistem antrian pada processor tunggal

Pada suatu sistem komputer yang dipakai secara bersama-sama, contohnya komputer server pada *Local Area Network (LAN)*, *job-job* akan memiliki waktu pelayanan yang berbeda-beda dan datang secara tidak teratur. Setiap saat *job* datang walaupun belum tentu diproses pada saat itu. Sehingga akan dibentuk suatu sistem antrian walaupun *processor* memiliki kapasitas yang cukup untuk melayani semua *job* dalam waktu yang lama.

2.3.4 Pola Kedatangan Pada Sistem Operasi

Kedatangan suatu *job* pada sistem ditetapkan sebagai kejadian yang acak (*random*) dan saling bebas. Jika pada suatu saat *job-job* datang dalam jumlah yang banyak dan saling bebas, maka dapat diambil asumsi-asumsi sebagai berikut :

i. Jumlah beberapa kedatangan selama selang waktu tertentu tergantung pada panjang selang waktu tersebut.

ii. Untuk suatu selang waktu yang sangat kecil (t, t + dt), probabilitas dari suatu kedatangan tunggal adalah λdt , dimana λ suatu konstanta dan probabilitas adanya kedatangan yang lebih dari satu dapat diabaikan.

Asumsi -asumsi tersebut menunjukkan bahwa kedatangan *job* pada sistem adalah berdistribusi *Poisson*. Sehingga probabilitas tidak ada kedatangan yang terjadi selama selang waktu $t + dt$ yaitu $P_o(t + dt)$ adalah sama dengan perkalian antara probabilitas tidak ada kedatangan yang terjadi selama selang waktu t yaitu $P_o(t)$ dengan probabilitas tidak ada kedatangan yang terjadi selama selang dt yaitu $P_o(dt)$. Asumsi 4.c pada proses *Poisson* menunjukkan bahwa $P_o(dt) = 1\text{-}\lambda dt$.

Secara matematis keadaaan diatas dapat dituliskan sebagai berikut :

$$P_o(t + dt) = P_o(t)(1-\lambda dt)$$

$$= P_o(t) - P_o(t)\lambda dt$$

$$P_o(t + dt) - P_o(t) = -\lambda P_o(t)\, dt$$

$$\frac{P_0(t + dt) - P_0(t)}{dt} = -\lambda P_0(t)$$

$$\frac{dP_0}{dt} = -\lambda P_0(t)$$

Penyelesaian PD diatas adalah :

$$P_o(t) = e^{-\lambda t} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2.14)}$$

Waktu antara dua kedatangan yang secara berturut-turut disebut selang waktu kedatangan *(interarrival time)* dan konstanta λ adalah laju kedatangan *(arrival rate)*. Probabilitas selang waktu kedatangan antara t dan $t + dt$ yaitu $dF(t)$ adalah :

$$dF(t) = P_0(t)\lambda dt$$

atau

$$dF(t) = \lambda e^{-\lambda t}\, dt \qquad \ldots\ldots\ldots \text{(2.15)}$$

Fungsi distribusi $F(t)$ didefinisikan sebagai probabilitas dimana selang waktu kedatangannya lebih kecil atau sama dengan t atau $0 < x \leq t$, sehingga didapat :

$$F(t) = \int_{x=0}^{t} dF(x)$$
$$= \int_{0}^{t} \lambda e^{-\lambda x} dx$$
$$= 1 - e^{-\lambda t}$$

$$F(t) = 1 - e^{-\lambda t} \qquad \text{......................} \quad (2.16)$$

Persamaan (2.16) menunjukkan bahwa selang waktu kedatangan berdistribusi Eksponensial dengan nilai mean :

$$E(t) = \int_{t=0}^{\infty} t dF(t) = \frac{1}{\lambda} \qquad \text{............} \quad (2.17)$$

2.3.5 Pola Pelayanan Pada Sistem Operasi

Waktu pelayanan yang dibutuhkan oleh *job-job* juga ditetapkan sebagai kejadian yang random, dengan variabel-variabel yang saling bebas. Asumsi-asumsi untuk pola kedatangan juga berlaku untuk pola pelayanan, yaitu :

i. Jumlah waktu pelayanan selama waktu pelayanan yang diberikan pada suatu *job* hanya tergantung pada panjangnya pelayanan yang dibutuhkan *job* tersebut.

ii. Untuk suatu interval waktu yang sangat kecil (t,t+dt), probabilitas dari suatu pelayanan tunggal adalah μdt, dimana μ suatu konstanta dan probabilitas pelayanan *job* yang lebih dari satu dapat diabaikan

Hal ini menunjukkan bahwa waktu pelayanan juga berdistribusi Eksponensial dengan probabilitas waktu pelayanan dari suatu *job* diantara selang t dan t + dt yaitu:

$$dF(t) = \mu e^{-\mu t} dt \qquad \text{................} \quad (2.18)$$

Probabilitas F(t) adalah probabilitas bahwa waktu pelayanan suatu *job* adalah lebih kecil atau sama dengan t dengan kata lain $0 < x \leq t$, sehingga didapat :

$$F(t) = \int_{t=0}^{t} dF(x)$$

$$F(t) = 1 - e^{-\mu t} \qquad \text{................ (2.19)}$$

konstanta μ disebut laju pelayanan *(service rate)*, sedangkan kebalikannya $1/\mu$ adalah rata-rata waktu pelayanan.

Theorema 2.5

1. Probabilitas bahwa *job* yang berakhir selama selang waktu yang sangat kecil yaitu dt adalah independen / bebas dari sejumlah waktu pelayanan t_1, dengan t_1 adalah waktu yang telah digunakan oleh *job* sebelum selang waktu dt.
2. Jika sebuah *job* diinterupsi setelah t_1 detik pelayanan, sisa waktu pelayanannya yaitu dari $t - t_1$ akan berdistribusi eksponensial dengan nilai rata-rata yang sama yaitu $1/\mu$.

Bukti :

1. Pernyataan pertama dapat dibuktikan dengan mengambil kondisi probabilitas dari *job* yang sudah diproses selama selang waktu (t_1, $t_1 + dt$), asumsi bahwa waktu pelayanan telah melewati t_1 adalah:

   Prob *(job dilayani pada interval waktu $t_1 < T \leq t_1 + dt$)*

$$= \frac{\text{Prob}(t_1 < T \leq t_1 + dt)}{\text{Prob}(T > t_1)}$$

$$= \frac{\mu e^{-\mu t_1} dt}{e^{-\mu t_1}}$$

$$= \mu \, dt$$

2. Untuk membuktikan pernyataan kedua, dibutuhkan distribusi probabilitas dari sisa waktu pelayanannya yaitu dari $t - t_1$, asumsi bahwa waktu pelayanan telah melewati t_1 adalah :

Prob (sisa waktu pelayanan $\leq \Delta t$)

$$= \frac{\mathrm{Prob}(t_1 < T \leq t_1 + \Delta t)}{\mathrm{Prob}(T > t_1)}$$

$$= \frac{e^{-\mu t_1} - e^{-\mu(t_1+\Delta t)}}{e^{-\mu t_1}}$$

$$= 1 - e^{-\mu\Delta t}$$

Teorema 2.5 menunjukkan bahwa algoritma penjadualan tidak dapat meramalkan sisa waktu pelayanan dari *job-job* berdasarkan waktu pelayanan yang telah dilewatinya. Konsekuensinya, prioritas penjadualan harus berdasarkan pada perkiraan/estimasi yang akurat dari waktu pelayanan yang disediakan oleh *user* atau pada periode pengaturan prioritas selama proses sebagai sebuah fungsi dari waktu pelayanan sebelumnya.

2.3.6 Pengertian-Pengertian Lain Dalam Sistem Antrian Pada Sistem Operasi

2.3.6.1 Utilitas Processor

Definisi 2.19:

Jika *job-job* datang pada sistem antrian dengan pelayanan tunggal dengan masukkan *Poisson* dan distribusi waktu pelayanannya F(t), maka rata-rata banyaknya kedatangan selama waktu pelayanan suatu *job* tunggal adalah :

$$\rho = \int_{t=0}^{\infty} \lambda t dF(t)$$

Jika persamaan pada definisi (2.20) diselesaikan, maka didapat :

$$\rho = \frac{\lambda}{\mu} \quad \text{.................................} \quad (2.20)$$

ρ disebut sebagai utulitas *processor*, jika $\rho < 1$, maka rata-rata banyaknya kedatangan lebih kecil dari rata-rata banyaknya keberangkatan, yang berarti laju pelayanan lebih besar dibanding dengan laju kedatangan, sehingga *processor* akan dapat menangani semua *job* yang ada dalam antrian. Sebaliknya jika $\rho > 1$, maka rata-rata laju kedatangan lebih besar daripada rata-rata laju pelayanan sehingga antrian akan bertambah setiap saat.

2.3.6.2 Kondisi Steady State

Dalam sistem operasi proses antrian yang digunakan adalah antrian yang akan selalu mencapai kondisi *steady state*. Keadaan ini akan terjadi jika distribusi probabilitas untuk kedatangan maupun pelayanan adalah bebas untuk setiap saat, dengan catatan sistem sudah beroperasi untuk beberapa waktu dengan $\rho < 1$.

2.3.6.3 Asumsi - Asumsi Dalam Antrian

Asumsi-asumsi yang digunakan dalam antrian pada sistem operasi adalah sebagai berikut:

1. Sistem antrian terdiri dari sebuah *processor* tunggal dimana secara terus-menerus selalu dalam keadaan sibuk selama ada *job-job* pada sistem.
2. Semua *job* tetap berada pada sistem sampai pelayanannya selesai.
3. Penempatan terlebih dahulu (*preemtion*), jika ada, tidak akan menurunkan utilitas *processor*.
4. Pola kedatangan dari *job-job* merupakan proses *Poisson*.
5. Waktu kedatangan dan waktu pelayanan merupakan variabel random yang bebas.
6. Sistem dalam keadaan *steady state* yang setimbang.

Definisi 2.20

Rasio menunggu dari sebuah *job* adalah perbandingan dari rata-rata waktu menunggu W terhadap waktu pelayanan sebenarnya.

Definisi 2.21 :

Rasio tanggapan dari sebuah *job* adalah perbandingan dari rata-rata waktu tanggapan R terhadap waktu pelayanan sebenarnya.

2.3.6.4 Load Function u(t)

Definisi 2.22 :

Load function (*fungsi muatan*) *u(t)* adalah suatu fungsi waktu, yaitu waktu yang diperlukan *processor* untuk mengosongkan sistem antrian dari semua *job* yang ada pada waktu t , jika setelah waktu tersebut tidak ada *job* baru yang datang.

Setiap kedatangan *job* akan menyebabkan *u(t)* menyela sejumlah waktu pelayanan yang diperlukan oleh *job* tersebut. Diantara kedatangan, *u(t)* akan berkurang secara linear sebesar satu detik (dari waktu proses) setiap satu detik (waktu sebenarnya) sampai akhirnya mecapai nilai nol. Pada saat *job-job* dilayani secara kontinu untuk diselesaikan, dengan mengabaikan *preemtion*, algoritma penjadualan tidak akan mempengaruhi rata-rata pengurangan *u(t)* diantara kedatangan. Sehingga *u(t)* hanya tergantung pada pola kedatangan dan pola pelayanan.