

BAB II

TINJAUAN PUSTAKA

Tinjauan pustaka berisi konsep dan teori yang mendukung penelitian dan meletakkan dasar untuk metode pemecahan masalah. Kajian pustaka secara sistematis menggambarkan hasil penelitian yang diperoleh peneliti sebelumnya dan hasil penelitian terkait dengan penelitian yang akan dilakukan. Tinjauan pustaka dapat berupa deskripsi kualitatif, model matematika atau persamaan yang berkaitan langsung dengan masalah yang diteliti. Hipotesis juga diajukan, yang berisi pernyataan singkat yang diambil dari tinjauan pustaka atau tinjauan pustaka, yang merupakan jawaban sementara atas pertanyaan saat ini dan masih perlu dibuktikan.

1.1 Teori Antrian

Teori antrian merupakan teori yang mencakup riset matematik dari antrian (Dimiyati dan Tarliah, 2004 dalam Purnawan *et al*, 2013). Antrian terjadi pada saat kedatangan pelanggan melebihi kapasitas yang dibutuhkan oleh layanan, oleh karena itu jika persentase penganggur layanan kecil maka pelanggan membutuhkan waktu tunggu yang lama, artinya layanan tersebut tidak memiliki waktu luang sama sekali. Perhitungan kedua kondisi dalam sistem antrian ini menunjukkan keseimbangan, dan upaya harus selalu dilakukan untuk menjaganya dalam keadaan yang sesuai (Ilhamsyah, 2017).

Antrian muncul karena adanya tuntutan layanan yang melebihi kapasitas, fasilitas layanan atau pelayanan, mengakibatkan pengguna fasilitas yang datang tidak dapat langsung mendapatkan layanan akibat kepadatan layanan (Indriyani, 2010 dalam Nengsih, 2016). Dalam banyak kasus, fasilitas layanan lain dapat disediakan untuk mengurangi atau mencegah antrian. Namun, biaya untuk menyediakan layanan tambahan akan menghasilkan keuntungan yang lebih rendah, yang mungkin lebih rendah dari tingkat yang dapat diterima. Sebaliknya, antrian panjang yang sering akan mengakibatkan hilangnya pelanggan atau pengguna jasa (Anisah, 2015). Tujuan dasar dari model antrian adalah meminimalkan jumlah dua biaya yaitu biaya langsung penyediaan fasilitas pelayanan dan biaya tidak langsung karena harus menunggu pelayanan (Nengsih, 2016).

1.1.1 Karakteristik Sistem Antrian

Ada bermacam berbagai komentar tentang ciri sistem antrian, berikut ialah ciri sistem antrian yang dijelaskan oleh Heizer dan Render, 2006 dalam Nengsih, 2016.

1. Karakteristik Kedatangan

Kehadiran mempunyai ciri semacam dimensi populasi, distribusi kehadiran, serta sikap kehadiran.

A. Ukuran Populasi

- Populasi tidak terbatas: jumlah pelanggan yang datang atau pada saat waktu tertentu hanyalah sebagian kecil dari semua potensi kedatangan.
- Populasi terbatas: merupakan sebuah antrian dalam situasi dan kondisi pengguna pelayanan potensial terbatas.

B. Distribusi Kedatangan

Bentuk kedatangan ini dianggap sebagai kedatangan pelanggan satu per satu. Asumsikan kedatangan pelanggan mengikuti proses dengan distribusi probabilitas tertentu. Distribusi yang sering digunakan adalah distribusi Poisson. Secara umum rumus distribusi probabilitas Poisson sebagai berikut (Mulyono, 2004 dalam Ilhamsyah, 2017):

$$P(x) = \frac{e^{-\lambda} \lambda^x}{x!}, \text{ untuk } x = 0, 1, 2, 3, 4 \dots \text{dst.} \quad (1)$$

dimana:

$P(x)$ = probabilitas kedatangan sejumlah x

x = banyaknya kedatangan persatuan waktu

λ = rata-rata tingkat kedatangan

C. Perilaku Kedatangan

Berikut merupakan karakteristik perilaku kedatangan :

- Pelanggan yang sabar adalah mesin atau orang-orang (pelanggan) yang dengan sabar menunggu dalam suatu kondisi antrian sampai mereka mendapatkan pelayan dan tidak melakukan berpindahan dari antrian.
- Pelanggan menolak yang tidak ingin bergabung dalam suatu antrian sebab merasa proses antrian terlalu lama dalam memenuhi kebutuhan

mereka.

2. Karakteristik atau Disiplin Antrian

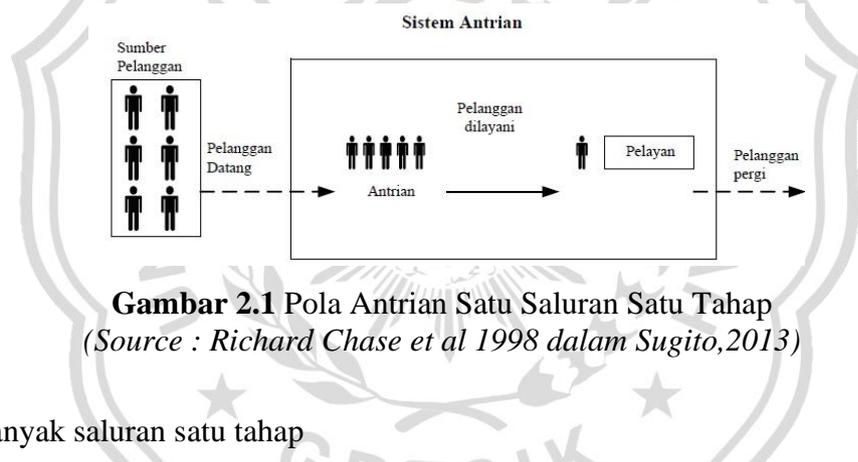
Setiap pelayanan mempunyai peraturan yang biasanya digunakan untuk menentukan pelanggan mana yang dipilih dari antrian untuk memulai pelayanan. Peraturan pelayanan disini biasanya disebut dengan disiplin antrian. Adapun disiplin antrian terdiri dari:

- a. *First Come First Served (FCFS)*
- b. *Last Come First Served (LCFS)*
- c. *Shortest Operation Times (SOT)*
- d. *Service in Random Order (SIRO)*

3. Karakteristik Pelayanan

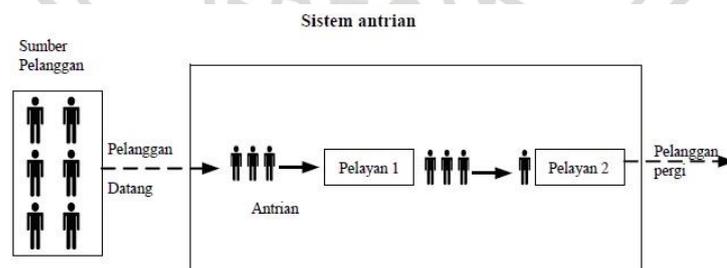
a. Desain Proses Antrian

1) Satu saluran satu tahap



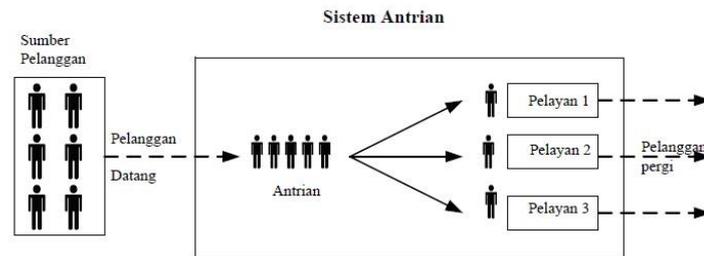
Gambar 2.1 Pola Antrian Satu Saluran Satu Tahap
(Source : Richard Chase et al 1998 dalam Sugito,2013)

2) Banyak saluran satu tahap



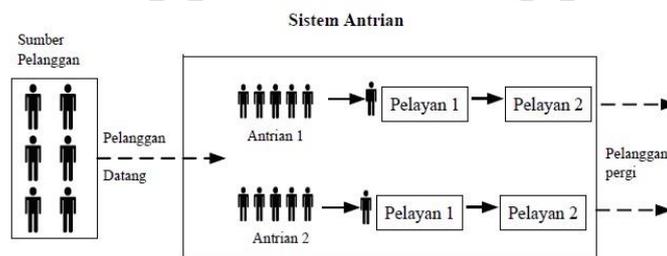
Gambar 2.2 Pola Antrian Banyak Saluran Satu Tahap
(Source : Richard Chase et al 1998 dalam Sugito,2013)

3) Satu saluran banyak tahap



Gambar 2.3 Pola Antrian Satu Saluran Banyak Tahap
(Source : Richard Chase et al 1998 dalam Sugito,2013)

4) Banyak saluran banyak tahap



Gambar 2.4 Pola Antrian Banyak Saluran Banyak Tahap
(Source : Richard Chase et al 1998 dalam Sugito,2013)

b. Distribusi Waktu Pelayanan

Asumsi umum biasanya yang digunakan bagi distribusi waktu pelayanan adalah distribusi eksponensial. Rumus umum probabilitas distribusi eksponensial adalah:

$$f(t) = \mu e^{-\mu t} \quad (2)$$

dimana:

$f(t)$: probabilitas yang berhubungan dengan t

t : waktu pelayanan

μ : rata-rata waktu pelayanan

Naman kadangkala dapat ditemukan proses Poisson pada proses pelayanan (*services process*), dengan demikian distribusi poisson juga berlaku pada proses pelayanan.

1.2 Model

Sebelum melakukan pembuatan model, terdapat juga aktivitas siklus diagram

atau *activity cycle diagram* (ACD). Menurut Subagyo dan Frylie (2018), *activity cycle diagram* merupakan metode yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem. Diagram tersebut memberikan notasi pemodelan grafis untuk menerangkan serangkaian aktivitas dalam bermacam suasana kehidupan nyata.

Model didefinisikan sebagai representasi kualitatif dan kuantitatif dari proses atau peristiwa sistem, dan dapat dengan jelas menggambarkan interaksi antara berbagai faktor penting yang akan diamati. (Sultan, 2007). Sedangkan model sistem merupakan tiruan dari sistem yang memuat semua komponen dan perilaku yang berkaitan dengan realisasi tujuan penelitian sistem. (Siswanto, 2018). Sebelum melakukan pembuatan model, terdapat juga aktivitas siklus diagram atau *activity cycle diagram* (ACD). Menurut Subagyo dan Frylie (2018), *activity cycle diagram* adalah metode untuk menggambarkan interaksi antar objek dalam suatu sistem. Diagram ini memberikan notasi pemodelan grafis untuk menjelaskan serangkaian kegiatan dalam berbagai situasi kehidupan nyata.

Modelnya mirip dengan sistem yang diwakilinya, tetapi lebih sederhana daripada sistem tersebut. Salah satu tujuan dari model ini adalah untuk membantu menganalisis dan memprediksi efek dan perubahan sistem. Model dapat didefinisikan sebagai proses yang menggambarkan operasi sistem yang sebenarnya untuk menjelaskan atau menunjukkan hubungan penting yang terlibat. Sistem aktual dapat dikelola sebagai model. Pada dasarnya model simulasi terbagi menjadi tiga dimensi yaitu (Law and Kelton, 1991 dalam Sidik, 2018) :

- a. Model Simulasi Statis: Model simulasi statik digunakan untuk merepresentasikan sistem pada momen tertentu atau sistem yang tidak terpengaruh oleh perubahan waktu.
- b. Model Simulasi dinamis: Digunakan apabila sistem yang sedang diteliti dipengaruhi oleh perubahan waktu.
- c. Model Simulasi Deterministik: Apabila model simulasi yang akan dibentuk tidak memuat variabel random maka model simulasi disebut simulasi deterministik.
- d. Model Simulasi Stokastik: Biasanya sistem yang dimodelkan dalam simulasi berisi beberapa masukan acak, sehingga pada sistem seperti ini model simulasi

yang dibangun pada umumnya disebut model simulasi stokastik

- e. Model Simulasi Kontinu: sistem dapat dikatakan kontinu apabila perubahan variabel sistem berlangsung secara berkelanjutan bersamaan dengan perubahan waktu.
- f. Model Simulasi Diskret: Suatu sistem dapat dikatakan diskrit apabila variabel sistem yang menggambarkan status sistem berubah pada titik waktu tertentu.

1.3 Sistem

Mendefinisikan sistem sebagai kumpulan entitas yang saling berinteraksi satu sama lain untuk mencapai tujuan logis yang tertentu (Siswanto, 2018). Sedangkan menurut Ekoanindiyo, 2011, suatu sistem pada umumnya didefinisikan sebagai sekumpulan komponen atau entitas yang melakukan tindakan dan berinteraksi antara satu entitas dengan entitas lainnya untuk mencapai tujuan akhir yang logis. Entitas adalah bagian dasar dari sistem yang menyusun sistem. Sistem mempunyai beberapa karakter yaitu:

- a) Perilaku sasaran (*purposive behaviour*)
- b) Keseluruhan (*wholism*)
- c) Keterbukaan (*openness*)
- d) Transformasi (*transformation*)
- e) Keterhubungan (*interrelatedness*)
- f) Mekanisme kontrol (*control mechanism*)

Tidak ada sistem yang benar-benar independen, karena sistem pasti dipengaruhi oleh faktor-faktor di luar sistem. Suatu sistem atau subsistem dapat diidentifikasi sebagai sistem khusus, dan jika dimaksudkan untuk pembelajaran, diasumsikan untuk digabungkan dengan sistem lain (Khosnevis, 1994 dalam Sidik 2018). Oleh karena itu, relatif terhadap tujuan penelitian. Penelitian yang dilakukan hanya mengamati keadaan atau kondisi suatu sistem yang tersusun dari sekumpulan entitas pada waktu tertentu (Law dan Kelton, 2007 dalam Sidik 2018)).

1.3.1 Elemen-Elemen Sistem

Berikut ini adalah deskripsi dari elemen sistem : (Arifin, 2009)

- a. Entitas dan Atribut

Entitas adalah item yang akan diproses oleh sistem. Karakteristik unik dari

suatu entitas adalah biaya, bentuk, prioritas, dan lain sebagainya. Misalnya produk paku, nasabah bank dan lain-lain. Pada saat yang sama, atribut adalah semua atribut dari suatu entitas. Misalnya, operator mesin adalah entitas, jadi keahlian operator adalah atribut. Oleh karena itu, entitas dibagi menjadi beberapa bentuk, antara lain : (Arifin, 2009)

- 1) Bernyawa, misalnya pelanggan, pasien, dan lain sebagainya.
- 2) Tidak bernyawa, misalnya part, dokumen, dan lain sebagainya.
- 3) Tidak dapat diraba, misalnya kemajuan teknologi informasi yang pesat, dan lain sebagainya.

b. Aktivitas dan *Delay*

Aktivitas adalah peristiwa yang dijalankan sistem secara langsung atau tidak langsung di entitas pemrosesan. Misalnya memperbaiki peralatan, memberikan layanan kepada pasien, dan sebagainya. Pada saat yang sama, penundaan adalah suatu kondisi dimana durasi proses tidak diketahui. Misalnya menunggu servis, menunggu produk (*work in process*) diproses dalam proses manufaktur, dan lain-lain (Arifin, 2009).

c. Sumber Daya dan Kontrol

Sumber daya diartikan sebagai segala sesuatu yang dapat membantu melaksanakan kegiatan atau dapat dikatakan fasilitas penunjang, seperti peralatan, personel, dan sebagainya. Sumber daya ini memiliki banyak karakteristik, seperti kapasitas, kecepatan, waktu siklus, dll. Oleh karena itu, sumber daya dapat diklasifikasikan sebagai berikut: (Arifin, 2009)

- 1) Manusia atau yang bernyawa, meliputi dokter, operator, dan sebagainya.
- 2) Yang tidak bernyawa, meliputi peralatan, rantai produksi, dan sebagainya.
- 3) Tidak dapat diraba, meliputi informasi, tenaga elektrik, dan sebagainya.

1.3.2 Karakteristik Sistem

Karakteristik-karakteristik sistem antara lain sebagai berikut : (Sultan, 2007)

- ✓ Perilaku sasaran (*purposive behaviour*)

Setiap sistem berusaha untuk mencapai satu atau lebih tujuan sehingga tujuan tersebut menjadi penggerak (motivasi) dari sistem untuk mencapai tujuan tersebut.

✓ Keseluruhan (*wholism*)

Suatu teori pada umumnya menyatakan bahwa faktor-faktor yang menentukan kesatuan tidak dapat direduksi lagi.

✓ Keterbukaan (*openness*)

Tunjukkan kesamaan akhir (*quifinality*), yang berarti bahwa keadaan akhir sistem dapat diperoleh dari berbagai keadaan awal.

✓ Transformasi (*transformation*)

Hal tersebut menunjukkan bahwa sistem memiliki kemampuan untuk mengubah nilai dari status sumber daya (*input*) menjadi keluaran (*output*) melalui proses transformasi.

✓ Keterhubungan (*interrelatedness*)

Termasuk interaksi internal dan ketergantungan antara berbagai bagian atau elemen yang menyusun sistem, serta interaksi antara sistem dan lingkungan.

✓ Mekanisme kontrol (*control mechanism*)

Ini adalah proses pengawasan yang digunakan oleh sistem untuk memperbaiki penyimpangan yang terjadi

1.3.3 Variabel-Variabel Sistem

Berikut adalah penjelasan mengenai variabel-variabel sistem : (Arifin, 2009)

a. Variabel Keputusan

Variabel keputusan adalah sebagai variabel independent (*non dependen*). Contohnya ialah tingkat layanan, tingkat kedatangan pesanan, dan lain sebagainya.

b. Variabel Respon

Variabel respon adalah variabel yang mengukur kinerja sistem, yaitu memberikan respon terhadap variabel keputusan tertentu. Misalnya jumlah entitas yang diproses dalam periode waktu tertentu, tingkat pemanfaatan rata-rata sumber daya, dan sebagainya. Seperti tingkat produksi, waktu tunggu rata-rata, dan lain sebagainya.

c. Variabel *State*

Variabel keadaan adalah variabel yang menandai keadaan sistem pada saat tertentu. Seperti kapasitas, jumlah server, kecepatan konveyor, dan lain sebagainya.

1.4 Pengertian Simulasi

Simulasi adalah proses perencanaan model dari sistem aktual dan bereksperimen dengan model yang bertujuan untuk memahami perilaku sistem atau mengevaluasi strategi dari berbagai sistem operasi (Sultan, 2007).

Simulasi adalah teknologi yang meniru operasi atau proses yang terjadi di sistem dengan bantuan komputer dan peralatan. berdasarkan asumsi tertentu sehingga sistem dapat dipelajari secara ilmiah (Sentia, 2016).

Tujuan Simulasi. Simulasi pada umumnya dilakukan dengan dua tujuan yaitu (Law and Kelton, 1991 Dalam Siswanto, 2018):

1. Sesuai dengan kondisi input yang berbeda, nilai parameter dan konfigurasi model, tentukan karakteristik variabel (rata-rata, minimum, maksimum, varians, dan sebagainya.) Sehingga analisis sistem dan pemahaman perilakunya dapat dilakukan.
2. Membandingkan karakteristik (rata-rata, nilai maksimal, nilai minimal, variansi, dan sebagainya) variabel berdasarkan kondisi input, nilai parameter, dan konfigurasi model yang berbeda sehingga diketahui performansi setiap skenario dan memilih alternatif yang memiliki performansi terbaik.

Dengan demikian berikut adalah kelebihan menggunakan metode simulasi dengan *software* yang sesuai, meliputi : (Siswanto, 2018)

- a. Mampu memodelkan keterkaitan dan ketergantungan antar elemen dalam sistem sistem yang tidak dapat diperoleh melalui metode analitis.
- b. Fleksibel untuk berbagai jenis model sistem meskipun belum tentu merupakan metode yang paling efektif, metode ini dapat secara fleksibel disesuaikan dengan berbagai jenis model sistem.
- c. Dapat menampilkan perilaku sistem selama periode waktu tertentu.
- d. Menawarkan hemat waktu dan biaya, terutama untuk masalah skala besar.
- e. Tidak akan merusak atau mengganggu sistem yang sebenarnya.
- f. Tidak hanya memberikan informasi tentang satu metrik kinerja sistem, tetapi

juga memberikan informasi tentang beberapa metrik sekaligus.

- g. Hasil simulasi mudah untuk dipahami dan dikomunikasikan, sehingga orang yang tidak paham teori pun dapat memahaminya dengan baik.
- h. Menekankan terhadap detail sistem

Langkah-langkah Simulasi adalah sebagai berikut :

Menurut (Suryani, 2006 dalam Sidik, 2018), beberapa langkah yang perlu dilakukan dalam melakukan simulasi, antara lain :

1. Pendefinisian sistem. Langkah ini termasuk menentukan batas sistem dan mengidentifikasi variabel penting.
2. Formulasi model : Merumuskan hubungan antara berbagai komponen model.
3. Pengambilan data : Sesuai dengan tujuan pembentukan model, identifikasi data yang dibutuhkan oleh model.
4. Pembuatan model : Dalam perumusan model, perlu dilakukan penyesuaian jenis bahasa simulasi yang akan digunakan.
5. Verifikasi model : proses pengecekan pada model, apakah sudah benar terbebas dari eror atau belum.
6. Validasi model adalah proses pengujian apakah model tersebut konsisten dengan sistem yang sebenarnya.
7. Untuk menentukan jumlah replika yang dibutuhkan, dilakukan perhitungan jumlah replika. Dalam kasus ini metode yang digunakan untuk menghitung jumlah replika adalah metode absolute error. Sedangkan seang kepercayaan yang digunakan adalah 95%. Perhitungannya sebagai berikut:

$$\begin{aligned}
 n &= 30 \\
 n-1 &= 29 \\
 \alpha &= 0,05 \\
 hw &= (n-1, \alpha/2) \times S\sqrt{n} \\
 &= (2,04523) \times S \sqrt{30}
 \end{aligned}$$

8. Nilai hw atau Absolute error (β) adalah banyaknya error yang ditoleransi yang terjadi dalam sistem tersebut dan dinyatakan dalam bentuk jumlah eror. Selanjutnya menghitung nilai n' . yaitu nilai half width yang didapatkan dari perhitungan sebelumnya dengan menggunakan rumus sebagai berikut :

$$n' = ((Z_{0.05/2}) \times S hw)^2$$

$$= (1,96 \times S \text{ hw})^2$$

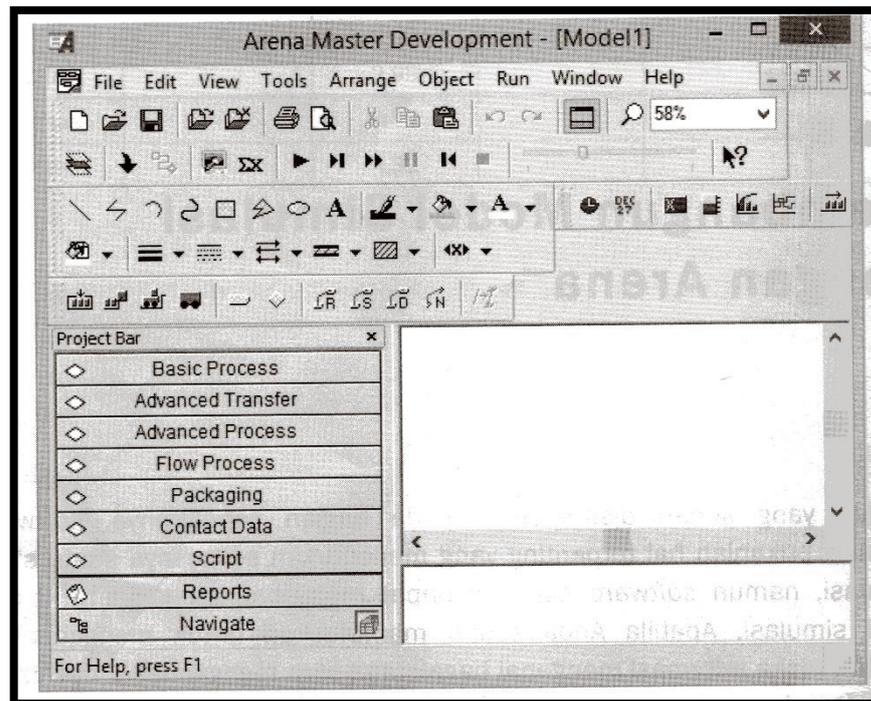
9. Skenarioisasi : menyusun rencana berdasarkan model. Setelah model valid, langkah selanjutnya adalah membuat beberapa skenario (eksperimen) untuk memperbaiki kinerja sistem sesuai kebutuhan. Secara umum, ada dua jenis program :
 - a. Skenario parameter : Diselesaikan dengan mengubah nilai parameter model. Jenis skenario ini relatif mudah diterapkan karena kita hanya melakukan perubahan pada keluaran model.
 - b. Skenario struktur : Diselesaikan dengan mengubah struktur model. Jenis skenario ini memerlukan pemahaman sistem yang cukup sehingga usulan atau percobaan struktur baru dapat meningkatkan kinerja sistem.
10. Interpretasi model : proses ini merupakan penarikan kesimpulan dari hasil output model simulasi.
11. Implementasi adalah penerapan model pada sistem.
12. Dokumentasi adalah proses penyimpanan hasil output model.

1.5 Software Simulasi Arena

Arena adalah paket simulasi general purpose, dengan kemampuan memodelkan sistem manufaktur dan non manufaktur. Sistem manufaktur *flow lines*, *assembly lines*, *job shop*, *conveyors*. Sistem non manufaktur *health care*, *maintenance sistem*, *computer network* (Taslim, 2015).

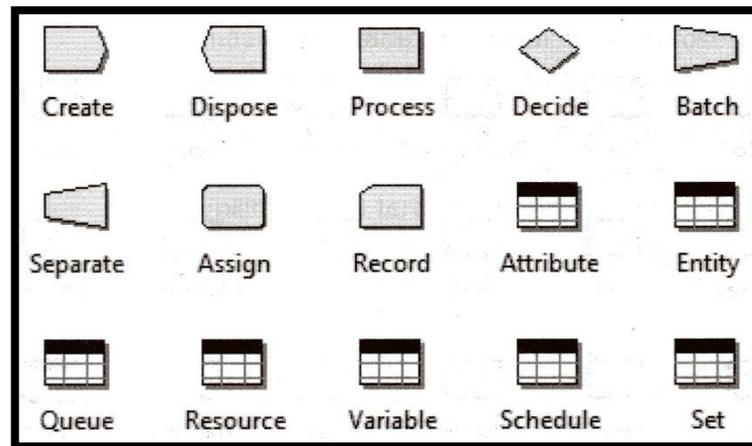
Didalam jendela Arena terdapat berbagai fungsi umum pada bagian atas seperti halnya jendela pada *software* di bawah OS Windows lainnya. Dibagian kiri jendela *software* terdapat menu atau project bar yang menampilkan beberapa panel, setiap panel tersusun dari modul-modul khusus. (Siswanto *et al.*, 2018).

Panel adalah sekumpulan modul yang dikumpulkan sesuai dengan kategori fungsi dan tingkat detail dari model simulasi yang akan dibangun. Sedangkan modul adalah blok dasar di Arena yang digunakan untuk membangun model, sama seperti kita membuat diagram alur (flowchart) dengan memasukkan data di dalamnya (Siswanto *et al.*, 2018).

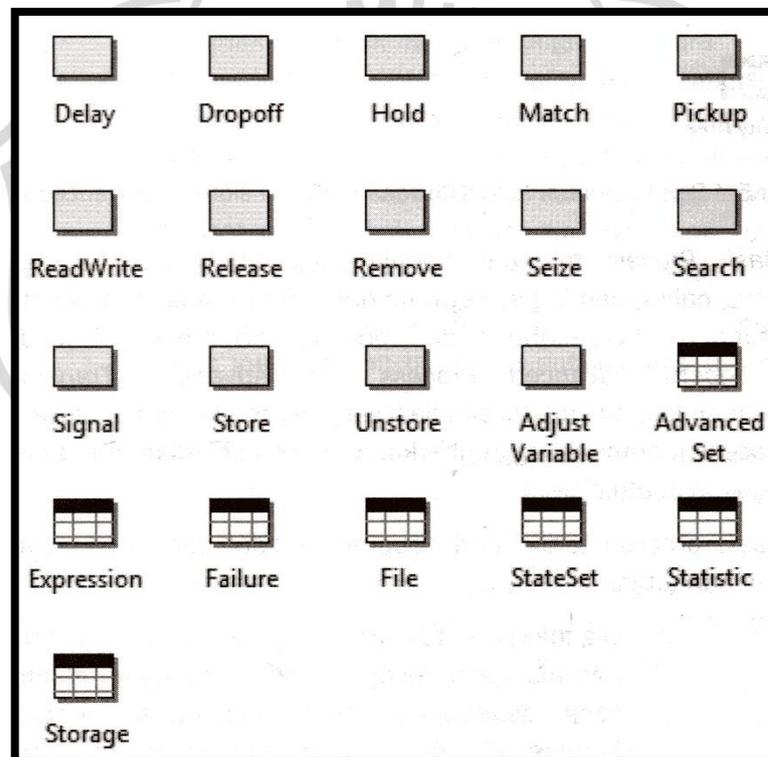


Gambar 2.5 Jendela Utama Software Arena
Sumber : (Siswanto *et al.*, 2018)

Modul pada umumnya terdiri dari dua jenis yaitu *flowchart* dan data. Flowchart dari modul mendefinisikan proses dalam simulasi, dan data adalah tabel yang digunakan untuk mengelola informasi rinci dari proses serta elemen yang terlibat dalam model. Modul-modul yang tersedia di Arena memiliki bentuk yang berbeda-beda dan mirip dengan bentuk dasar flowchart klasik. Setiap modul di panel memiliki fungsi khusus saat memodelkan sistem. Panel utama di Arena mencakup *Basic Process*, *Advanced Process*, dan *Advanced Transfer*. Panel lainnya yakni *packaging*, *flow process*, *blocks*, *statistics*, *elements*, *report*, dan *negative* (Siswanto *et al.*, 2018).

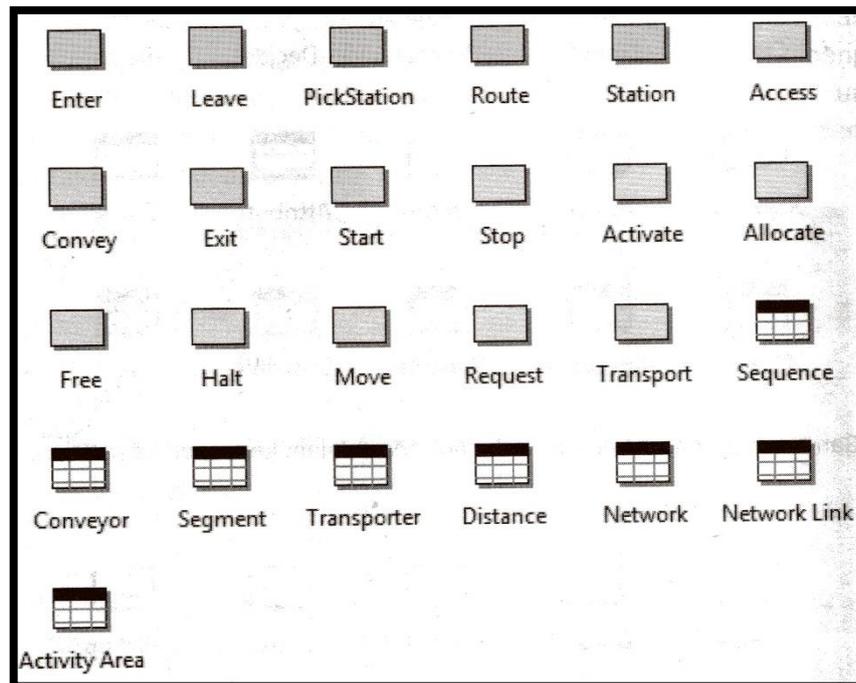


Gambar 2.6 Bentuk-bentuk Modul Flowchart Dalam Kelompok Basic Process
Sumber : (Siswanto *et al.*, 2018)



Gambar 2.7 Bentuk-bentuk Modul Flowchart Dalam Kelompok Advanced Process

Sumber : (Siswanto *et al.*, 2018)



Gambar 2.8 Bentuk-bentuk Modul Flowchart Dalam Kelompok Advanced Transfer
Sumber : (Siswanto *et al.*, 2018)

Panel basic process mencakup modul dan tabel data sebagai berikut (Siswanto *et al.*, 2018) :

- a. *Create* : Ini digunakan untuk menghasilkan atau membuat entitas dan dianggap sebagai kedatangan di sistem yang sebenarnya.
- b. *Process* : digunakan untuk menampilkan proses dalam model simulasi dan menentukan detail proses.
- c. *Decide* : Digunakan ketika entitas menghadapi beberapa kondisi dan memilih kondisi tertentu dengan aturan.
- d. *Dispose* : digunakan untuk mengeluarkan entitas dari sistem.
- e. *Batch* : digunakan untuk menggabungkan sementara atau permanen antara dua entitas atau lebih karena alasan tertentu.
- f. *Separate* : digunakan untuk membuat duplikasi dari suatu entitas atau memisahkan beberapa entitas yang digabungkan melalui modul *batch*.
- g. *Assign* : digunakan untuk menetapkan nilai baru ke variabel atau atribut berdasarkan terjadinya peristiwa tertentu.
- h. *Record* : digunakan untuk pencatatan statistik yang lebih rinci dari suatu

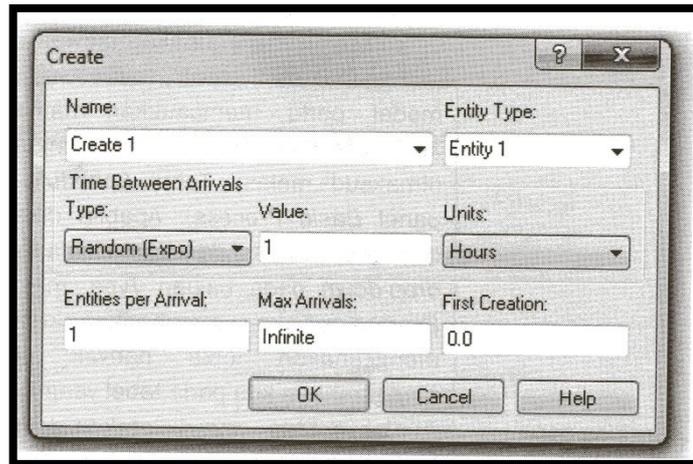
atribut tertentu atau menghitung entitas pada suatu titik tertentu dalam simulasi.

- i. *Tabel entity* : digunakan untuk menentukan berbagai informasi tentang entitas dalam model simulasi, termasuk nama, gambar, dan biaya terkait.
- j. *Tabel recourse* : digunakan untuk menentukan sumber daya yang digunakan dalam model simulasi, termasuk biaya, kapasitas, kuantitas, dan sebagainya.
- k. *Tabel set* : Digunakan untuk menentukan berbagai jenis set yang digunakan dalam model simulasi dan informasi tentang sumber daya di dalamnya.
- l. *Tabel attribute* : digunakan untuk mendefinisikan atribut yang dibuat dengan sengaja atau otomatis oleh sistem yang dibuat di perangkat lunak arena untuk membedakan entitas dan mempermudah catatan statistik.
- m. *Tabel schedule* : digunakan untuk menentukan jadwal operasi atau perubahan kapasitas sumber daya terdaftar atau jadwal kedatangan.
- n. *Tabel queue* : digunakan untuk mendefinisikan nama antrian dan jenis yang digunakan dalam antrian, misalnya : *first in first out, last in first out*, dan lain sebagainya.
- o. *Tabel variable* : digunakan untuk menjelaskan nilai dan dimensi variabel yang digunakan dalam pencatatan data.

Untuk mempermudah proses penggunaan *software* simulasi Arena, berikut ini adalah penjelasan modul-modul dasar yaitu *create, process, decide*, dan *dispose* (Siswanto *et al.*, 2018) :

1.5.1 Modul Create

Modul *create* termasuk dalam kategori modul *flowchart* dan digunakan sebagai titik awal dimana suatu entitas diciptakan pada suatu model simulasi. Tingkat kedatangan suatu entitas dikendalikan oleh suatu distribusi waktu antar kedatangan (*time between arrivals*) atau berupa jadwal. Entitas yang dibuat dalam modul *create* ini adalah persyaratan utama sistem untuk mulai berjalan. Modul *create* direpresentasikan dengan bentuk dua trapezium siku-siku yang digabungkan pada bagian alasnya. Ketika modul tersebut di *double klik* menunjukkan kotak dialog box sebagai berikut (Siswanto *et al.*, 2018):



Gambar 2.9 Dialog Box Dari Modul Create
Sumber : (Siswanto *et al.*, 2018).

Tabel 2.1 Keterangan Data Input Modul Create

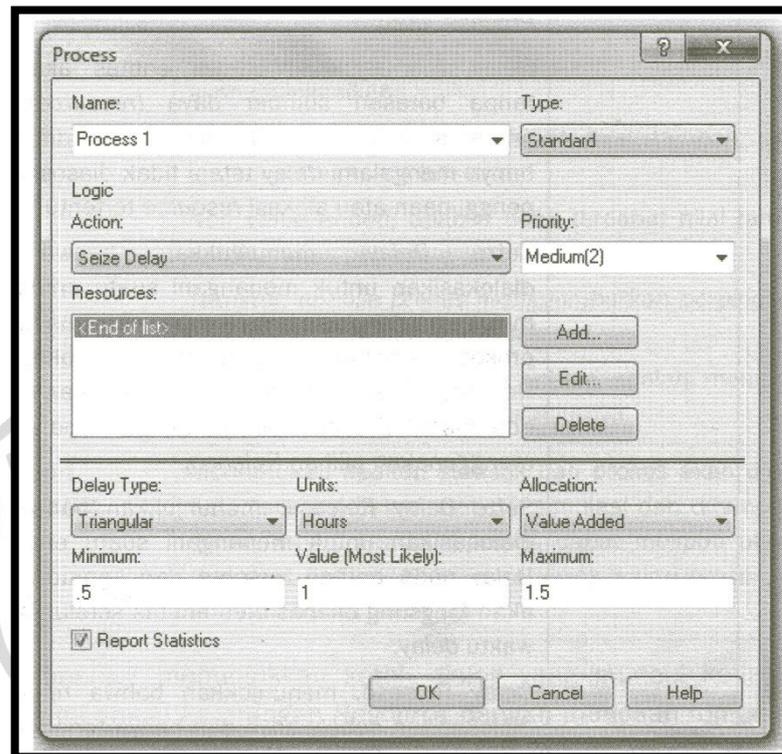
Label	Keterangan
<i>Name</i>	Nama spesifik yang muncul untuk suatu model <i>create</i> . Jika suatu model menggunakan lebih dari satu <i>create</i> , maka nama yang digunakan harus berbeda.
<i>Entity Type</i>	Deskripsi jenis entitas yang akan dibuat dari pembuatan. Sistem akan otomatis menggunakan urutan entity 1 dan setelahnya berurutan, namun pengguna dapat memasukkan sendiri nama lain, seperti: car, child, adult, dan lain-lain. Memberikan nama yang berbeda akan memudahkan kami dalam melakukan penyesuaian khusus untuk setiap jenis.

<i>Time Between Arrivals</i>	Mendeskripsikan jenis distribusi antar kedatangan (<i>type</i>) dengan nilai parameter dari distribusi yang dipilih (<i>value</i>) dan unit waktu (satuan) yang digunakan, seperti jam atau menit. Jenis distribusi waktu tidak selalu probabilistik, tetapi juga dapat dimodelkan dengan determinisme konstan atau ditentukan dengan jadwal tertentu. Jika jadwal digunakan, pabrikan model perlu memasukkan nama jadwal dan kemudian membuat atau mengelola jadwal yang direferensikan di schedule (masih di panel Proses Dasar). Jika versi rilis tertentu diperlukan dan tidak terdaftar dalam opsi drop-down di bagian Type, opsi Build Expression akan mengarahkan kita untuk menggunakan tabel yang lebih lengkap, memungkinkan kita untuk menggunakan lebih banyak opsi secara fleksibel.
<i>Entities Per Arrival</i>	Jumlah entitas yang dibuat dalam waktu kedatangan tertentu. Misalnya: di pom bensin, setiap mobil selalu datang, sedangkan di restoran, kedatangan fisik lebih cenderung dua orang atau sekelompok besar orang pada satu waktu..
<i>Max Arrivals</i>	Jumlah maksimum entitas yang dibuat oleh modul create ini. Jika jumlah maksimum telah tercapai, maka create akan bisa menghentikan keluarnya entitas.
<i>First Creation</i>	Posisi waktu atau jam simulasi ketika entitas pertama dimunculkan.

Saat menggunakan model *create*, maka secara otomatis pada modul data tabel *entity* akan muncul data entitas yang kita inputkan pada modul *create* beserta dengan atribut-atribut yang menyertainya, yaitu tipe, distribusi waktu antar kedatangan, unit waktu yang digunakan jumlah entitas per kedatangan, jumlah maksimum kedatangan, dan waktu entitas pertama diciptakan (Siswanto *et al.*, 2018).

1.5.2 Modul *Process*

Modul *process* termasuk dalam kategori *flowchart* dan untuk digunakan mewakili suatu proses termasuk sumber daya yang digunakan, antrian proses, dan waktu proses. *Double click* pada modul *process* akan membuka kotak dialog yang ditampilkan pada gambar berikut (Siswanto *et al.*, 2018).



Gambar 2.10 Kotak Dialog Modul *Process*
Sumber : (Siswanto *et al.*, 2018).

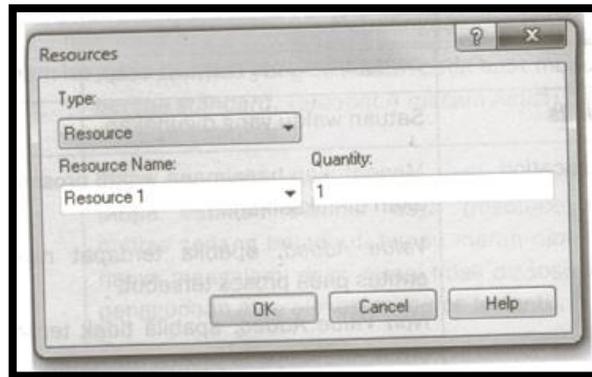
Tabel 2.2 Keterangan Data Input Modul *Process*

Label	Keterangan
<i>Name</i>	Nama spesifik yang ditampilkan untuk modul <i>process</i> . Jika suatu model menggunakan lebih dari satu <i>process</i> , jadi nama yang digunakan harus berbeda.
<i>Type</i>	Adalah tipe logika yang digunakan dalam model, terdapat dua opsi yaitu <i>standard</i> dan sub model. Pilihan pertama berarti semua

	<p>logika disimpan dalam modul proses dan didefinisikan pada <i>action</i>. Pilihan kedua berarti logika disusun secara hierarki pada sub model yang terdiri dari beberapa modul. Ketika kita membuka suatu model process baru, kolom type akan otomatis terisi standard.</p>
<i>Action</i>	<p>Jenis aktivitas yang dilakukan pada saat modul process bertipe standard. Terdapat 4 macam action pada modul process, yaitu :</p> <ol style="list-style-type: none"> 1) Delay Menunjukkan bahwa entitas akan diproses tanpa batasan sumber daya (resource), misalnya entitas dalam keadaan lampu merah dimana entitas hanya mengalami delay tetapi tidak ada hubungannya dengan penggunaan atau alokasi resource tertentu. 2) Seize delay Menunjukkan bahwa resource telah dialokasikan untuk memproses entitas (setelah melewati antrian) dan terjadi aktivitas (delay) terjadi dalam proses selama durasi tertentu. Resource tidak langsung dilepas dan akan dilepas saat ada modul process lainnya dengan tipe action yang menggunakan pilihan release. 3) Seize delay release Menunjukkan bahwa resource telah dialokasikan untuk memproses entitas, diikuti delay pada entitas tersebut dan kemudian resource akan langsung dilepas oleh entitas setelah berakhirnya waktu delay. 4) Delay release Menunjukkan bahwa resource telah dialokasikan sebelumnya dan hanya terjadi delay pada entitas tersebut dan kemudian resource akan langsung dilepas.
<i>Priority</i>	<p>Nilai prioritas yang menunggu untuk entitas resource tertentu. Digunakan ketika terdapat satu atau lebih entitas dari modul lain menunggu untuk resource yang sama (shared resources).</p>
<i>Resources</i>	<p>Digunakan untuk menambahkan resource atau set yang digunakan dalam proses, namun kotak resource menjadi tidak aktif ketika yang dipilih dalam action atau delay, atau yang dipilih dalam type adalah sub model. Definisi lebih jelas mengenai resource dapat diisikan pada</p>

	kotak dialog (gambar x.x) yang muncul dengan Add.
<i>Delay Type</i>	Jenis distribusi probabilitas waktu proses yang digunakan, umumnya pada pilihan drop down hanya akan terdapat pilihan yang terbatas, yaitu normal, triangular, exponential. Selain itu juga terdapat pilihan constant atau build expression dimana pengguna dapat memilih ekspresi matematis lainnya.
<i>Units</i>	Satuan waktu yang digunakan.
<i>Allocation</i>	Menentukan bagaimana waktu proses dan biaya proses akan dialokasikan. Value added, apabila terdapat nilai tambah untuk entitas pada proses tersebut. Non value added, apabila tidak terdapat nilai tambah untuk entitas pada proses tersebut. Transfer, apabila proses menggambarkan perpindahan entitas dari satu titik ke titik lainnya. Wait, apabila yang dialami entitas adalah menunggu untuk melakukan aktivitas selanjutnya.
<i>Report Statistics</i>	Menentukan apakah statistik dari proses akan dicatat dan disimpan dalam statistik simulasi dan ditampilkan dalam report. Statistik proses dapat meliputi utilisasi resource, panjang antrian, rata-rata waktu proses, dan rata-rata waktu mengantri.

Berikut ini menunjukkan gambar kotak dialog yang muncul ketika menambahkan keterangan *resource* dengan menekan tombol *add* (Siswanto *et al.*, 2018).



Gambar 2.11 Kotak dialog Untuk Menambahkan Resource Pada Modul Process
Sumber : (Siswanto *et al.*, 2018).

Kolom *type* memberikan alternatif untuk menggunakan *resource* atau *set*. Ketika yang dipilih adalah *resource* maka yang akan digunakan oleh entitas dalam modul *process* hanyalah *resource* yang namanya tertera dalam kolom *resource name* sejumlah yang tertulis pada *quality*. Namun jiks yang dipilih pada *type* adalah *set*, maka entitas dapat menggunakan *resource* mana saja yang terdapat dalam *set* tersebut dengan aturan yang telah ditentukan. Kolom *quality* pada pilihan *set*, tidak menunjukkan jumlah *resource* yang digunakan oleh entitas, tetapi menunjukkan jumlah *set* yang ada (Siswanto *et al.*, 2018).

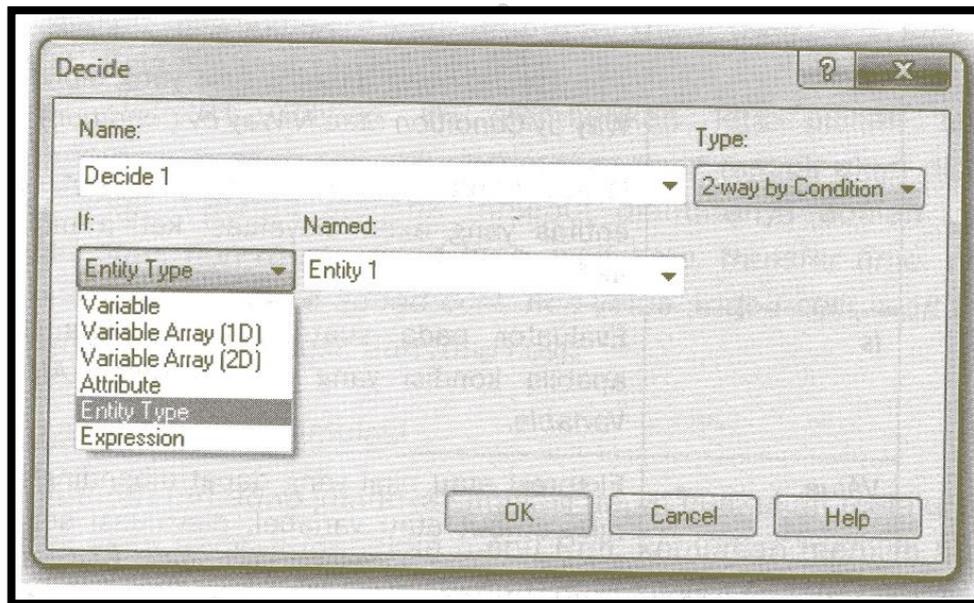
Apabila jenis *resource* yang dignakan adalah *set*, maka selanjutnya kita perlu memilih *selection rule* untuk mengatur bagaimana entitas akan memilih *resource* yang ada dalam *set*. Pilihan dalam *selection rule* sebagai berikut (Siswanto *et al.*, 2018) ::

- *Cyclical*, sesuai urutan *resource* yang ada dalam suatu *set* misalnya : *resource 1*, *resource 2*, dan seterusnya.
- *Random*, artinya pemilihan *resource* oleh entitas dilakukan secara acak.
- *Preferred order*, mengutamakan anggota *resource* yang tersedia, contohnya *resource 1* akan terpilih jika tersedia, jika tidak maka berikutnya akan diarahkan ke *resource 2* jika tersedia, dan seterusnya
- *Specific member*, berarti anggota dari *set* yang dipilih sesuai dengan *input atribut* (yang sebelumnya telah disimpan dikolom *save attribute*).
- *Largest remaining capacity*, mengalokasikan *resource* dalam *set* berdasarkan sisa kapasitas *resource* terbesar.

- *Smallest number busy*, mengalokasikan *resource* berdasarkan tingkat kesibukan (rangkuman status *busy*) yang terkecil.

1.5.3 Modul *Decide*

Modul *decide* termasuk dalam kategori modul *flowchart*, digunakan ketika alur proses dihadapkan pada situasi yang terjadi berikutnya pada entitas ditentukan pada suatu kondisi tertentu. Kondisi spesifik ini dapat diekspresikan dalam modul *decide* (Siswanto *et al.*, 2018).



Gambar 2.12 Kotak Dialog Modul *Decide*
Sumber : (Siswanto *et al.*, 2018).

Dalam modul *decide*, pilihan dapat ditentukan berdasarkan beberapa kondisi atau berdasarkan peluang. Kondisi pemilihan dapat didasarkan nilai atribut, variabel, tipe entitas atau berupa ekspresi lainnya (Siswanto *et al.*, 2018).

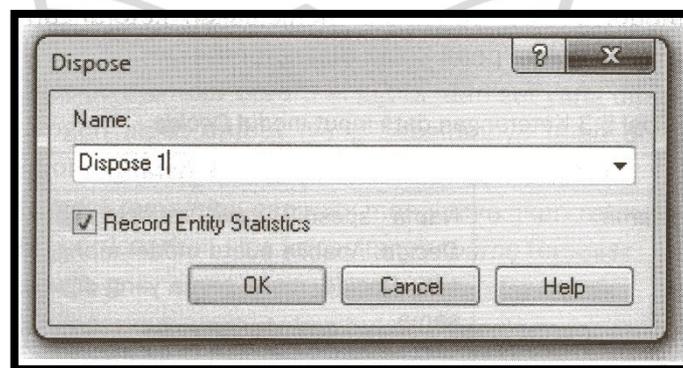
Tabel 2.3 keterangan Data Input Modul *Decide*

Label	Keterangan
<i>Name</i>	Nama spesifik yang muncul untuk suatu model <i>decide</i> . Jika model menggunakan lebih dari satu <i>decide</i> , nama yang digunakan harus berbeda satu sama lain.

<i>Type</i>	Mendefinisikan apakah keputusan akan dibuat berdasarkan suatu kondisi atau suatu peluang. Untuk setiap situasi dan peluang terdapat pilihan apakah 2 atau lebih dari 2 keadaan.
<i>Precent true</i>	Yaitu nilai % yang digunakan untuk kondisi yang bernilai benar, otomatis sisanya adalah kondisi lainnya (100%- <i>precent true</i>)
<i>If</i>	Jenis kondisi yang tersedia (jika yang dipilih adalah <i>2-way by condition</i> atau <i>N-way by condition</i>).
<i>Named</i>	Bisa berupa nama variabel, atribut, atau jenis entitas yang akan dievaluasi ketika modul <i>decide</i> aktif.
<i>Is</i>	Evaluator pada suatu kondisi, digunakan hanya apabila kondisi yang dipilih adalah <i>atributte</i> atau <i>variable</i> .
<i>Value</i>	Ekspresi atau nilai yang dapat dibandingkan dengan nilai atribut atau variabel, dievaluasi sebagai suatu ekspresi yang menentukan kondisi benar atau salah. <i>Value</i> tidak digunakan ketika <i>input</i> pada <i>type</i> adalah <i>entity type</i> . Apabila <i>type</i> yang dipilih adalah <i>expression</i> , maka selanjutnya evaluator harus dituliskan pada kolom <i>value</i> , mialnya <i>color = = red</i> .

1.5.4 Modul *Dispose*

Modul *dispose* digunakan sebagai titik akhir entitas dari model simulasi. Modul *dispose* berfungsi mengeluarkan entitas dari sistem. Statistik dari suatu entitas dapat direkam terlebih dahulu sebelum entitas tersebut keluar.



Gambar 2.13 Modul *Dispose* Beserta Kolom Input Data
Sumber : (Siswanto *et al.*, 2018).

Pada modul *dispose* terdapat satu kolom isian, yaitu nama modul (*name*). Nama modul seharusnya adalah nama spesifik yang tidak boleh sama antara satu modul *dispose* dengan modul *dispose* lainnya. Kolom nama ini tidak harus diisi karena ketika mengambil modul *dispose* dari panel, nama akan otomatis disediakan oleh *software* dan berurutan.

1.6 Metode Analisis Investasi

Dalam mengevaluasi suatu investasi, diperlukan suatu metode analisis untuk menentukan kelayakan suatu investasi. Penilaian tersebut mencakup semua aspek, dan pertimbangan tertentu diperlukan untuk menentukan kelayakan investasi. Selain itu, untuk menilai kelayakan suatu investasi diperlukan beberapa metode untuk menganalisis kelayakan suatu investasi.: (Prihastono, dkk.,2015)

a. Metode *Payback Period*

Metode ini digunakan untuk mengukur kecepatan dana yang terkumpul dalam pendapatan investasi atau arus kas bersih dikembalikan secara penuh, dengan penerimaan atau aliran kas netto (*net cash flow*). Sehingga satuan hasilnya bukan prosentase melainkan satuan waktu, seperti tahun, *Payback method* bukan merupakan pengukur tingkat keuntungan (*profitability*) suatu investasi. Dengan kata lain *payback period* merupakan suatu periode yang diperlukan untuk dapat menutup kembali pengeluaran investasi dengan menggunakan *proceeds* atau aliran kas netto (*net cash flow*). Jika periode *payback* ini waktu pengembalian modal lebih pendek dari waktu yang dibutuhkan, investasi bisa menguntungkan sedangkan Jika lebih lama dari proposal investasi, maka ditolak. Karena metode ini mengukur seberapa cepat investasi dapat dipulihkan, ini didasarkan pada arus kas, bukan laba. Metode *payback period* ini sangat sederhana. Dan mudah dalam perhitungannya sehingga sudah banyak digunakan oleh perusahaan.

Namun cara ini memiliki kekurangan yaitu:

1. Metode ini mengabaikan penerimaan investasi yang diperoleh sesudah *payback period* tercapai.
2. Metode ini mengabaikan *time value of money* (nilai waktu uang). Rumus yang digunakan adalah :

$$\text{Payback Period} = \frac{\text{Cost}}{\text{Uniform annual benefit}}$$

b. Metode Profitability Index (PI)

Metode ini menghitung rasio antara nilai sekarang pendapatan kas bersih masa depan dan nilai sekarang investasi. Jika nilai indeks profitabilitas lebih besar dari ($PI > 1$), proyek dikatakan menguntungkan, tetapi nilai $PI > 1$ dianggap kurang menguntungkan. Seperti halnya metode NPV, metode ini juga membutuhkan penentuan suku bunga yang ditunjukkan sebelumnya. Rumus yang digunakan adalah sebagai berikut:

$$Probability\ Index = \frac{present\ value\ benefit}{present\ value\ cost}$$

c. Metode Net Present Value (NPV)

Metode ini menghitung selisih antara nilai sekarang dari investasi masa depan dan nilai sekarang dari pendapatan kas bersih (operasi dan terminal cash flow). Jika nilai sekarang dari pendapatan tunai bersih masa depan lebih besar dari nilai sekarang dari investasi, proyek tersebut dianggap menguntungkan. Namun jika proyeknya kecil (disebut NPV negatif atau $NPV < 0$), maka proyek tersebut akan ditolak karena dianggap tidak menguntungkan. Rumus yang digunakan sebagai berikut :

$$NPV = PV\ benefit - PV\ cost$$

d. Metode Internal Rate of Return (IRR)

Metode internal rate of return didefinisikan sebagai tingkat bunga yang membuat total nilai sekarang dan pengembalian yang diharapkan sama dengan nilai sekarang dan proceeds. Nilai sebenarnya dapat diperoleh dengan cara coba-coba (trial and error), kemudian dilakukan interpolasi untuk menentukan besarnya yang digunakan. Jika nilai r (tingkat bunga) lebih besar dari suku bunga bersangkutan (tingkat keuntungan yang diisyaratkan) maka investasi dianggap menguntungkan, dan jika nilai r kecil disebut rugi. Rumus sebagai berikut:

$$r = P_1 - C_1 \left[\frac{P_2 - P_1}{C_2 - C_1} \right]$$

Keterangan :

R = ROR yang dicari

P1 = Tingkat bunga ke-1

P2 = Tingkat bunga ke-2

C1 = NPV ke-1

C2 = NPV ke-2

1.7 Penelitian Terdahulu

Tabel 2.4 keterangan Data Input Modul *Decide*

No	Nama Peneliti	Objek Penelitian	Judul Penelitian	Hasil Penelitian
1	Muhammad Arsyad Sulistiono, Nicolas Ananto S.W., Bambang Riyanto, Kami Hari Basuki. JURNAL KARYA TEKNIK SIPIL, Volume 3, Nomor 3, Tahun 2014, Halaman 617 – 629	Antrian Angkutan Barang Pada Jembatan Timbang	Analisis Antrian Angkutan Barang Pada Jembatan Timbang Dengan Metode Simulasi Multiple Channel (Studi Kasus Pada Jembatan Timbang Sarang)	Berdasarkan perhitungan didapat bahwa dengan sistem single channel Jembatan Timbang Sarang memiliki kinerja sebesar 1,032 yang berarti bahwa jembatan timbang tersebut memiliki kekurangan 3,2% dari waktu operasionalnya. Dengan sistem double channel pada Jembatan Timbang Sarang memiliki kinerja sebesar lebih baik yaitu 0,503 yang berarti memiliki kelebihan waktu 50% dari waktu operasionalnya. 3. Dari hasil analisa yang dilakukan terdapat perbedaan hasil kinerja dari masing masing sumber data. Perbedaan hasil dari beberapa sumber data tersebut masih bisa diterima JURNAL KARYA TEKNIK SIPIL, Volume 3, Nomor 3, Tahun 2014, Halaman 629 629 karena selisihnya masih berkisar antara 0,003-0,129 sehingga tingkat kepercayaannya pun masih tinggi.
2.	B.J. Camerling ARIKA, Vol.	Kinerja Sistem Antrian Kapal	Model Simulasi Untuk	Hasil penelitian ini dengan mengambil data tbm wayame ambon, maka dapat dilihat untuk hasil awal atau

	11 No. ISSN: 1978-1105, 1 Pebruari 2017	Tanker Pada Dermaga Pt. Pertamina Tbbm Wayame Ambon	Menganalisis Kinerja Sistem Antrian Kapal Tanker Pada Dermaga Pt. Pertamina Tbbm Wayame Ambon	kenyataannya bahwa untuk hasil utilitas menunjukkan bahwa pada server dermaga II tidak terlalu sibuk dengan utilitas adalah 55.69% dengan kapal yang dilayani selama sebanyak 152 kapal loading dan untuk dermaga kapal backloading, dermaga I mempunyai utilitas 99.94% dan dermaga III mempunyai utilitas 99.76% dengan banyak kapal yang dilayani sebanyak 463 kapal backloading, dari hasil tersebut maka penulis memberikan alternative perbaikan untuk mempercepat pelayanan dimana kapal backloading dapat melakukan kegiatan bongkar muat pada dermaga kapal loading apabila tidak ada kegiatan bongkar muat, dapat digunakan untuk kapal backloading. Hasil utilitas dermaga I adalah 91.80% , utilitas dermaga II adalah 87.80% dan utilitas dermaga III adalah 82.86%, untuk pelayan kapal backloading sebanyak 492 kapal dan kapal loading 152 kapal, dengan menggunakan alternative perbaikan ini maka dapat mengurangi waktu pelayanan kapal.
3	Tommy Iskandar, Jurnal Matrik p- Volume XVI	Sistem Pelayanan Antrian	Analisis Sistem Pelayanan Antrian	Dari hasil simulasi, dengan melihat tingkat kegunaan fasilitas / utilitas pada proses pelayanan yang sangat tinggi, maka perlu adanya suatu penambahan fasilitas jembatan timbang / loket.

<p>No.1, September 2015, p.61- 68. Gresik (2015) ISSN : 1693-5128</p>	<p>Di Jembatan Timbang</p>	<p>Di Jembatan Timbang Pt. Sadp Gresik</p>	<p>Dapat dilihat tingkat kegunaan fasilitas setelah adanya dua rancangan usulan, yang pertama dengan penambahan 1 fasilitas jembatan timbang, rata-rata tingkat kegunaan fasilitas pada model simulasi di jembatan timbang A adalah 46,3 % dan rata-rata tingkat kegunaan fasilitas pada model simulasi di jembatan timbang B adalah 46,1 %. Dan yang kedua dengan penambahan 2 fasilitas jembatan timbang, rata-rata tingkat kegunaan fasilitas pada model simulasi di jembatan timbang A adalah 33,2 % dan rata-rata tingkat kegunaan fasilitas pada model simulasi di jembatan timbang B adalah 29,2 % dan di jembatan timbang C adalah 31,7 %. Dari keterangan diatas dengan memperhitungkan rata-rata waktu menunggu di jembatan timbang dan tingkat kegunaan fasilitas dapat diambil kesimpulan bahwa model usulan skenario 1 dengan penambahan 1 fasilitas jembatan timbang, adalah model yang bisa memberikan hasil yang lebih baik.</p>
---	--------------------------------	--	--