BAB 2
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Pengendalian persediaan adalah aktivitas yang dilakukan oleh industri manufaktur atau jasa untuk memastikan persediaan yang mereka miliki berada pada level yang optimal. *Retailer* sebagai salah satu industri dalam bidang jasa yang memiliki persediaan perlu melakukan pengendalian persediaan dengan baik. Tantangan yang dialami oleh *retailer* dalam mengendalikan persediaan salah satunya adalah persediaan produk yang bersifat *perishable* atau terdeteriorasi.

Penelitian tentang pengendalian persediaan pada objek *retailer* sudah banyak dilakukan oleh peneliti terdahulu. Pengendalian persediaan pada *retailer* memiliki beberapa tujuan seperti minimasi total biaya *retailer* yang memiliki *uncertain supply lead time* oleh Sazvar dkk (2013), penentuan *order size* optimal pada dua produk *perishable* yang bersifat subtitusi oleh Mishra (2017), penentuan *order size* optimal produk *perishable* dengan penawaran *trade credit* dari supplier oleh Annadurai dan Uthayakumar (2015), minimasi biaya persediaan dua retail dengan *lateral transhipment* oleh Haji dkk (2014), penentuan panjang periode *review* yang optimal dengan model simulasi oleh Sezen (2006), minimasi *loss* produk *perishable* dengan model simulasi oleh Ibrahim dkk (2020), dan penentuan *order size* optimal dengan metode EOQ untuk produk terdeteriorasi oleh Tripathi (2020). Tujuan pengendalian persediaan di retail umumnya berfokus pada penentuan *order size* optimal, penentuan periode *review* optimal, dan minimasi biaya persediaan, namun metode yang digunakan untuk mencapai tujuan bisa berbeda – beda tergantung pada situasi permasalahan.

Sazvar dkk (2013) dan Haji dkk (2014) melakukan penelitian di objek retail dengan tujuan minimasi biaya persediaan. Sazwar dkk (2013) mengembangkan model *inventory* berdasarkan teknik *continous reviews* (r,Q) untuk persediaan produk *perishable* yang memiliki *lead time* stokastik. Penelitian ini menyimpulkan produk yang memiliki laju deteriorasi rendah dan biaya persediaan tinggi akan lebih baik bila distok lebih banyak pada *reorder point* untuk menghindari *shortage*. Haji dkk (2014) meneliti permasalahan suatu retail yang mengalami kehabisan persediaan. Metode pengendalian yang digunakan Haji dkk (2014) adalah model persediaan *one-for-one-period* (1,T) dengan *unidirectional lateral transhipment*. Metode ini memperbolehkan retailer dengan permintaan berskala kecil yang kehabisan persediaan, untuk meminta suplai *emergency* kepada retailer yang memiliki skala permintaan besar untuk memenuhi permintaannya.

Sezen (2006) dan Ibrahim dkk (2020) keduanya melakukan penelitian menggunakan model simulasi. Penelitian Sezen (2006) memiliki tujuan mengetahui periode optimal dalam melakukan *review* persediaan. Penelitian ini menilai bahwa panjang periode untuk melakukan *review* persediaan sangat bergantung pada variansi permintaan. Produk yang memiliki permintaan tinggi akan membutuhkan periode *review* yang singkat dan begitu sebaliknya. Sedangkan penelitian Ibrahim dkk (2020) berfokus untuk minimasi *loss* produk *perishable* berupa ikan mentah dengan implementasi kebijakan proses pembaharuan pada model simulasinya. Kedua peneliti di atas menggunakan *tools* yang berbeda walaupun keduanya menggunakan model simulasi. Sezen (2006) menggunakan *tools* berupa *spreadsheet*, sedangkan Ibarhim (2020) menggunakan aplikasi *Arena Simulation*.

Tujuan penelitian pengendalian persediaan dengan fokus penentuan *order size* produk optimal dilakukan oleh Annadurai dan Uthayakumar (2015), Mishra (2017), dan Tripathi (2020). Annadurai dan Uthayakumar (2015) mengembangkan model persediaan untuk produk terdeterioasi dengan permintaan yang bersifat *credit-period dependant*. Hal ini terjadi ketika supplier memberikan periode *trade credit* kepada retailer dan retailer juga memberikan periode *trade credit* kepada *customer*. Penelitian yang dilakukan oleh Mishra (2017) menggunakan model persediaan EOQ untuk dua produk terdeteriorasi yang bersifat saling substitusi. Pemesanan kedua produk ini akan dilakukan secara *joint order* dengan siklus *order* yang sama. Tripathi (2020) mengembangkan model *inventory* deterministik untuk produk terdeteriorasi dengan permintaan yang bersifat *stock sensitive*. Model oleh Tripathi (2020) juga dapat digunakan pada permintaan yang bersifat *credit sensitive*, stokastik, dan *price dependent*.

Permasalahan yang dialami oleh PT. X sekarang adalah belum adanya metode pengendalian persediaan yang baik, khususnya untuk menentukan persediaan maksimum, persediaan minimum, dan siklus produk *fresh food*. Penelitian ini akan menggunakan model matematis *inventory* yang sudah ada dari penelitian terdahulu. Cara mengamati level persediaan pada penelitian ini dilakukan secara *periodic reviews* seperti penelitian Sezen (2006) dengan *tools* aplikasi *Arena Simulation* seperti penelitian Ibrahim dkk (2020) untuk menentukan *order size* dan periode pembelian produk buah alpukat di PT. X.

2.2. Dasar Teori

Dasar teori yang digunakan pada penelitian ini dijabarkan pada sub bab di bawah ini.

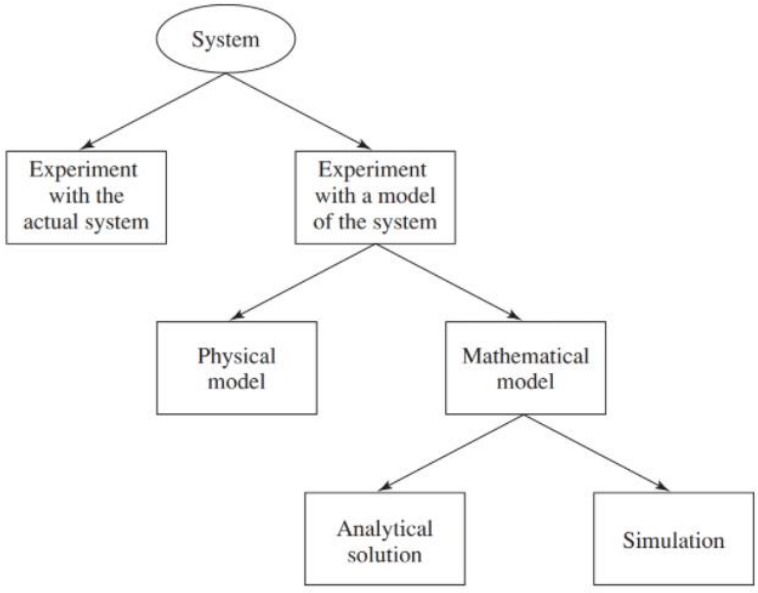
2.2.1. Pendekatan Sistem, Model, dan Simulasi

Pendekatan sistem, model, dan simulasi yang digunakan pada penelitian ini dijabarkan pada sub bab di bawah ini.

a. Sistem

Sistem adalah sekumpulan entitas seperti manusia, mesin, metode, modal, energi, informasi dan lain-lain yang berinteraksi dan bekerja sama untuk mencapai suatu tujuan tertentu. Law (2015) menyatakan bahwa definisi sistem dapat dibagi menjadi dua yaitu diskret dan kontinyu. Sistem diskret merupakan sistem yang keadaan variabelnya berubah secara instan dalam titik waktu yang terpisah. Sedangkan sistem kontinyu adalah sistem yang keadaan variabelnya berubah secara kontinyu seiring berjalannya waktu.

Law (2015) menyatakan pendekatan untuk mempelajari suatu sistem penting karena manusia atau pelaku sistem perlu untuk mengetahui hubungan antara banyak komponen sistem sehingga mampu memprediksi performansi sebuah sistem dalam kondisi yang berbeda-beda. Cara untuk mempelajari sistem menurut Law (2015) dapat digambarkan seperti *tree diagram* pada Gambar 2.1 di bawah ini:



Gambar 2.1 Diagram Cara Mempelajari Sistem (Law, 2015)

b. Model

Pendekatan mempelajari sebuah sistem dengan model dapat dibagi menjadi dua yaitu model fisik dan model matematis. Model fisik adalah model berwujud fisik atau *tangible*, sedangkan model matematis adalah model yang merepresentasikan sistem dalam logika dan hubungan kuantitatif yang bisa diubah.

Model matematis akan menghasilkan dua cara untuk menjawab pertanyaan dari sebuah sistem yaitu solusi analitik dan simulasi. Solusi analitik memungkinkan untuk didapatkan dari model matematis yang sederhana, sedangkan simulasi akan digunakan ketika sistem yang ditiru sangat kompleks sehingga tidak dimungkinkan mendapatkan solusi analitik. Model matematis yang dibuat dengan tujuan simulasi juga dapat disebut sebagai model simulasi.

c. Simulasi

Model simulasi dapat dibagi dalam 3 dimensi yang berbeda yaitu:

i. *statis vs dynamic*: model simulasi statis merepresentasikan sistem dalam suatu waktu tertentu sedangkan model simulasi dinamis merepresentasikan sistem seiring berjalannya waktu

ii. *deterministic vs stochastic*: model simulasi deterministik merupakan model yang didalamnya tidak memiliki unsur probabilitas atau *randomness* sama sekali, sedangkan model stokastik adalah kebalikannya yaitu model yang memiliki unsur probabilitas.

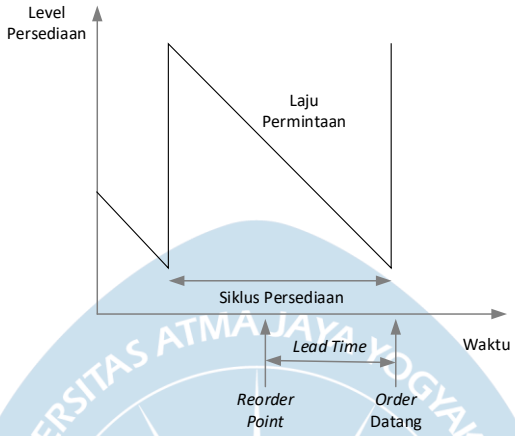
iii. *discrete vs continuous*: memiliki pengertian yang sama dengan sistem diskrit dan kontinyu. Model yang keadaan variabelnya berubah secara instan dalam titik waktu yang terpisah disebut diskrit. Sedangkan model kontinyu adalah model yang keadaan variabelnya berubah secara kontinyu seiring berjalannya waktu.

Maka model simulasi persediaan buah di PT. X yang akan dibuat oleh penulis merupakan model simulasi persediaan yang bersifat *discrete, stochastic,* dan *dynamic.*

2.2.2. Sistem Persediaan Secara Umum

Definisi stok adalah segala jenis produk atau material yang disimpan oleh organisasi yang digunakan untuk kepentingan di masa depan, sedangkan persediaan atau *inventory* adalah *list* yang mencatat stok tersebut. Namun pemahaman umum istilah persediaan pada beberapa tahun terakhir adalah persediaan dianggap sebagai stok dan juga *list* itu sendiri. Persediaan ada karena organisasi membutuhkan material atau produk namun tidak langsung menggunakannya.

Persediaan akan mengalami peningkatan ketika pesanan produk dari supplier datang dan sebaliknya akan mengalami penurunan ketika *customer* memesan produk tersebut (Waters, 2009). Kedua aktivitas ini akan terus berjalan seiring berjalannya waktu sehingga membentuk siklus persediaan. Gambaran siklus persediaan secara umum dapat dilihat pada Gambar 2.2.



Gambar 2.2 Siklus Persediaan (Waters, 2009)

Jenis persediaan yang dimiliki oleh organisasi bermacam - macam sesuai produk yang dimiliki organisasi tersebut. Jenis persediaan berdasarkan jenis produknya antara lain *raw materials*, *work in progress*, *spare part*, dan produk *perishable*. Persediaan juga dapat dibagi berdasarkan tujuannya antara lain *cycle stock* untuk kegiatan operasi normal, *safety stock* untuk kegiatan darurat, dan *seasonal stock* yang bertujuan menjaga stabilitas persediaan berdasar musimnya.

Performansi yang menunjukkan baik buruknya sebuah persediaan salah satunya adalah biaya persediaan. Jenis – jenis biaya persediaan menurut Waters (2009) antara lain:

a. *Unit Cost*

Unit cost atau biaya per unit adalah biaya yang diberikan oleh *supplier* untuk pembelian 1 unit produk.

b. *Reorder Cost*

Reorder cost atau biaya pesan ulang adalah semua biaya dari segala aktivitas yang dibutuhkan ketika melakukan pemesanan ulang atau *reorder* produk ke supplier. Biaya ini bisa termasuk seperti transportasi, kontrol kualitas, dan *delivery*.

c. *Holding Cost*

Holding cost atau biaya simpan adalah biaya yang digunakan untuk menyimpan satu unit produk dalam satu periode waktu tertentu. Biaya simpan bisa dikategorikan dalam beberapa faktor seperti faktor tempat penyimpanan, faktor deteriorasi produk, dan faktor administrasi.

d. *Shortage Cost*

Shortage cost atau biaya kekurangan adalah biaya yang dihasilkan ketika terjadi permintaan pelanggan yang tidak terpenuhi akibat persediaan produk habis.

2.2.3. Rasio *Turnover Inventory*

Ukuran performansi yang dipakai oleh PT. X untuk menilai baik buruknya sistem persediaan adalah rasio *turnover inventory*. Definisi rasio ini oleh PT. X adalah perbandingan antara *inventory on hand* dengan pendapatan tahunan. Cara PT. X melakukan perhitungan rasio *turnover inventory* dapat dilihat pada Persamaan 2.1.

$$R = \frac{AOH}{AS} \tag{2.1}$$

Keterangan dari rumus di atas adalah *R* adalah rasio *turnover inventory*, *AOH* adalah *average on hand*, dan *AS* adalah *annual sales*. Satuan *AOH* dan *AS* adalah rupiah. Rumus yang digunakan PT. X di atas menunjukkan bahwa semakin kecil rasio maka sistem persediaan akan dinilai semakin baik.

Namun penggunaan rumus oleh PT. X ini memiliki perbedaan dari teori yang disampaikan oleh Swamiddas (1996) yaitu rasio *inventory turnover* adalah perbandingan antara pendapatan tahunan dengan jumlah *inventory on hand.* Rasio *inventory turnover* akan semakin baik apabila nilai rasio semakin besar. Swamiddas (1996) menggunakan cara perhitungan rasio *inventory turnover* yang terbalik dengan PT. X yang dapat dilihat pada Persamaan 2.2.

$$R = \frac{AS}{AOH} \tag{2.2}$$

PT. X mempraktikan perhitungan rasio *inventory turnover* yang berbeda dengan penelitian Swamiddas (1996). Penelitian ini akan menggunakan rumus perhitungan rasio *inventory turnover* dari Swamiddas (1996) dengan tujuan maksimasi rasio.

2.2.4. Jenis Persediaan Perishable: Buah Alpukat

Produk yang dijadikan objek penelitian di PT. X adalah persediaan produk *fresh food* khususnya buah alpukat. Penelitian terdahulu dilakukan oleh Burg pada tahun 2004 yang kemudian disitasi oleh Thompson (2015) menyatakan bahwa tingkat daya tahan buah alpukat setelah dipanen dipengaruhi oleh suhu dan tekanan ruang penyimpanan. Hal ini dibuktikan dari salah satu jenis alpukat yaitu *Choquette* yang akan matang sepenuhnya selama 14 hari dengan ruang penyimpanan bersuhu 14,4 derajat Celcius dan bertekanan atmosfer. Namun jenis ini memiliki proses pematangan yang lebih lama yaitu 25 hari apabila dimasukkan dalam ruang penyimpanan bertekanan 40-101 mmHg. Penelitian ini menyatakan bahwa PT. X harus mempertimbangkan suhu dan tekanan dalam melakukan penyimpanan buah alpukat. PT. X belum menghitung suhu dan tekanan yang ada di *supermarket*nya, namun kebijakan yang telah dilakukan oleh PT. X mengenai persediaan buah alpukat adalah menentukan batas siklus umur hidup buah alpukat sebesar 7 hari.

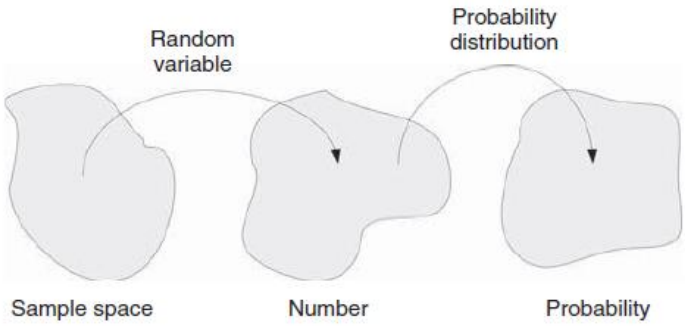
2.2.5. Pendekatan Model Simulasi

Teori pendekatan model simulasi yang digunakan pada penelitian ini dijabarkan pada sub sub sub bab di bawah ini.

a. Teori Pembangkitan Bilangan Random Dalam Simulasi

Input dalam suatu model simulasi yang bersifat *stochastic* pasti akan ada yang bersifat *random* atau acak. Aktivitas untuk membangkitkan nilai random dalam model simulasi dapat disebut sebagai pembangkitan nilai random atau *modelling randomness in simulation*.

Bentuk nilai random atau acak dalam data *input* model simulasi dapat mengikuti dua jenis distribusi yaitu diskrit dan kontinyu. Rosetti (2015) memberikan ilustrasi untuk proses penentuan distribusi probabilitas yang dapat dilihat pada Gambar 2.3 di bawah ini:



Gambar 2.3 Tahapan Penentuan Probabilitas Data *Input* (Rosetti, 2015)

Tahapan koleksi data mentah akan memberikan penulis *sample space* atau *sample data* yang kemudian akan diubah menjadi distribusi probabilitas yang paling *fit* atau sesuai.

i. Pembangkitan Distribusi Probabilitas Diskret (*Negative Binomial*)

Distribusi probabilitas diskret sederhana contohnya adalah model distribusi Bernoulli. Distribusi ini memodelkan perilaku variabel random yang mengikuti aturan percobaan Bernoulli. Karakteristik dari percobaan Bernoulli adalah percobaan yang memiliki dua hasil *outcome* yaitu sukses atau gagal. Probabilitas atau "p" sukses dari setiap percobaan adalah sama dan sifat dari setiap percobaan adalah independen.

Distribusi probabilitas diskret lainnya adalah distribusi geometrik. Distribusi ini memodelkan bilangan random yang merepresentasikan jumlah percobaan sampai kesuksesan yang pertama berdasarkan urutan percobaan Bernoulli. Pembangkitan bilangan random dengan distribusi geometrik di aplikasi *Arena Simulation* menurut Rosetti (2015) dapat dilakukan dengan Persamaan 2.3 di bawah ini:

$$1 + \text{AINT}(\text{LN}(1 - \text{UNIF}(0, 1)) / \text{LN}(1 - p)) \qquad (2.3)$$

p adalah probabilitas kesuksesan dalam percobaan Bernoulli.

Distribusi geometrik merupakan contoh kasus spesial dari distribusi *negative binomial*. Montgomery (2016) menyatakan distribusi *negative binomial* adalah generalisasi dari distribusi geometrik dimana variabel randomnya adalah jumlah percobaan Bernoulli yang dibutuhkan untuk mencapai *r'* kesuksesan. Apabila *X* adalah sebuah variabel random negative binomial dengan parameter *r'* dan *p*, maka *mean* dapat dihitung dengan Persamaan 2.4.

$$\mu = E(X) = \frac{r'}{p} \quad (2.4)$$

Pembangkitan distribusi *negative binomial* di aplikasi *Arena Simulation* dapat dilakukan dengan membuat *loop* distribusi geometrik yang akan berhenti ketika jumlah kesuksesan mencapai *r'*. Maka distribusi *negative binomial* juga dapat dibangkitkan dengan akumulasi penjumlahan sebanyak *r'* distribusi geometrik (Persamaan 2.3) dengan probabilitas kesuksesn *p*.

ii. Pembangkitan Distribusi Probabilitas Kontinyu (*LogNormal*)

Distribusi probabilitas kontinyu *lognormal* menurut Montgomery (2016) terjadi ketika *W* memiliki distribusi normal pada rumus random variabel eksponensial X = exp (*W*). Apabila *W* memiliki distribusi normal dengan *mean* θ dan variansi ω^2, maka *X* = exp (*W*) memiliki *mean* dengan perhitungan Persamaan 2.5 di bawah:

$$E(X) = e^{\theta+\omega^2/2} \quad (2.5)$$

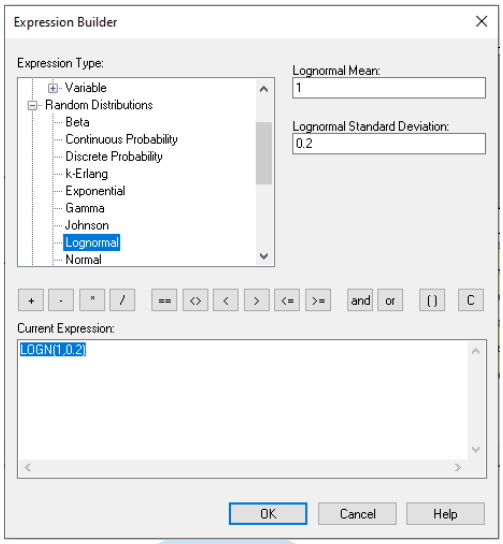
Variansi *X* dapat dihitung dengan Persamaan 2.6 di bawah:

$$V(X) = e^{2\theta+\omega^2}(e^{\omega^2} - 1) \quad (2.6)$$

Pembangkitan distribusi *lognormal* dalam aplikasi *Arena Simulation* dapat dituliskan dalam bentuk ekspresi yang ada pada Persamaan 2.7 di bawah ini:

LOGN(LogMean, LogStd) (2.7)

Parameter *LogMean* adalah rata-rata dari data sampel sedangkan *LogStd* adalah standar deviasi dari data sampel. Tampilan pembangkitan distribusi lognormal di *Arena Simulation* dapat dilihat pada Gambar 2.4 di bawah ini:



Gambar 2.4 Tampilan Pembangkitan Distribusi LogNormal di *Arena Simulation*

Cara lain membangkitkan distribusi lognormal berdasarkan Gambar 2.4 adalah yang pertama memilih fitur *expression builder* pada *Menu Bar*. Setelah *expression builder* terbuka maka pilih *random distributions* kemudian pilih distribusi *Lognormal*. Parameter yang perlu diisi adalah rata-rata dan standar deviasi dari data sampel.

b. *Input Modelling*

Model simulasi yang digunakan untuk menirukan sebuah sistem memerlukan sebuah data *input* atau masukan. Banks (2014) menyatakan pengembangan model data *input* memiliki beberapa tahap yaitu koleksi data mentah, membuat hipotesis distribusi data *input*, melakukan estimasi parameter dari hipotesis, dan menguji hipotesis distribusi data tersebut. Penjelasan dari setiap tahapan ini dapat dilihat di bawah:

i. Tahap Koleksi Data Mentah

Tahapan yang pertama ini adalah proses mengambil data yang diperlukan dari sistem nyata untuk digunakan sebagai *input* simulasi. Koleksi data saja tidak cukup untuk digunakan sebagai simulasi karena data *input* bisa saja memiliki banyak *error* atau data sudah terlalu lama. Maka selain koleksi diperlukan tahapan analisis data *input*.

ii. Membuat Hipotesis Distribusi Data *Input*

Langkah pertama yang dapat dilakukan pada tahapan ini adalah membuat histogram data *input* yang sudah dikoleksi dari tahap sebelumnya. Histogram berguna untuk mengidentifikasi bentuk dari distribusi data *input*. Histogram dapat dibuat dengan

bantuan aplikasi simulasi seperti *Arena Simulation: Input Analyzer* dan bahasa pemrograman R.

Langkah kedua adalah memilih *family* distribusi yang tersedia. Pemilihan *family* ini bisa disesuaikan dari bentuk distribusi data *input* yang sudah dibangkitkan di langkah sebelumnya. *Family* distribusi yang dipilih juga harus disesuaikan dengan proses natural data *input* yang sudah dikoleksi yaitu apakah data tersebut bersifat diskrit atau kontinyu.

Law (2015) memberikan contoh *family* distribusi dari jenis data diskrit dan kontinyu. Contoh *family* distribusi data yang bersifat kontinyu seperti uniform, normal, lognormal, eksponensial, Gamma, Weibull, Beta, dan triangular. Sedangkan contoh *family* distribusi data yang bersifat diskrit adalah Binomial, Negatif Binomial, Bernoulli, *discrete uniform*, *geometric*, dan Poisson.

iii. Melakukan Estimasi Parameter Hipotesis

Tahap setelah memilih *family* distribusi adalah mengestimasi parameter dari distribusi tersebut. Statistik dasar seperti rata-rata *sample* dan rata-rata variansi dapat digunakan untuk mengestimasi parameter distribusi yang dihipotesiskan. Perumusan rata-rata *sampe* ($\overline{X}$) dapat dilihat pada Persamaan 2.8.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{2.8}$$

Perumusan rata-rata variansi (S^2) dapat dilihat pada Persamaan 2.9.

$$S^2 = \frac{\sum_{i=1}^{n} X_1{}^2 - n\left(\overline{X}\right)^2}{n-1} \tag{2.9}$$

n adalah ukuran *sample* dan X_i adalah data observasi ke-*i*. Persamaan 2.8 dan Persamaan 2.9 di atas dapat digunakan pada jenis data mentah baik kontinyu atau diskrit.

Cara mengestimasi parameter distribusi selain statistika dasar adalah menggunakan rekomendasi estimator dari distribusi yang dihipotesiskan. Penulis menggunakan dua jenis distribusi dalam melakukan simulasi persediaan di PT. X yaitu *Negative Binomial* dan *Lognormal*. Parameter dan rekomendasi estimator dari kedua distribusi ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Parameter dan Rekomendasi Estimator Distribusi

Distribusi	Parameter	Rekomendasi Estimator
Negative Binomial	p, r	$p \in (0,1)$ s = *positive integer*
Lognormal	μ, σ^2	$\hat{\mu} = \overline{X}$ $\hat{\sigma}^2 = S^2$

iv. Menguji Hipotesis Data Distribusi

Proses pengujian hipotesis data distribusi ini juga biasa disebut sebagai *Goodness-of-Fit Tests.* Ujian ini digunakan untuk mengevaluasi sustainabilitas dari model *input* yang akan digunakan peneliti dalam membuat simulasi. *Goodness-of-Fit Tests* akan membantu peneliti untuk mendapatkan sebagian bukti untuk menolak atau tidak menolak *family* distribusi terpilih.

Salah satu prosedur jenis uji hipotesis untuk data distribusi dengan *random sample* berukuran n dari *random variable* X yang mengikuti jenis distribusi tertentu adalah *chi-square goodness-of-fit test*. Uji ini dilakukan dengan cara mengatur *n* observasi ke dalam *k* set interval. Rumus *chi-square goodness-of-fit test* adalah seperti pada Persamaan 2.10.

$$X_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \qquad (2.10)$$

O_i adalah frekuensi yang diamati pada kelas ke-*i* dan E_i adalah frekuensi *expected* pada kelas interval. X_0^2 atau biasa disebut *chi-square test statistic* mengikuti distribusi *chi-square* dengan derajat kebebasan sebesar *k-s-1*, dimana *s* merepresentasikan jumlah paramater dari *family* distribusi yang dihipotesiskan. Hipotesis pada uji *chi-square* ini adalah sebagai berikut:

H_0: Variabel random X mengikuti asumsi distribusi dengan parameter *s* yang diberikan oleh parameter estimasi (*s*)

H_1: Variabel random X tidak sesuai dengan asumsi distribusi.

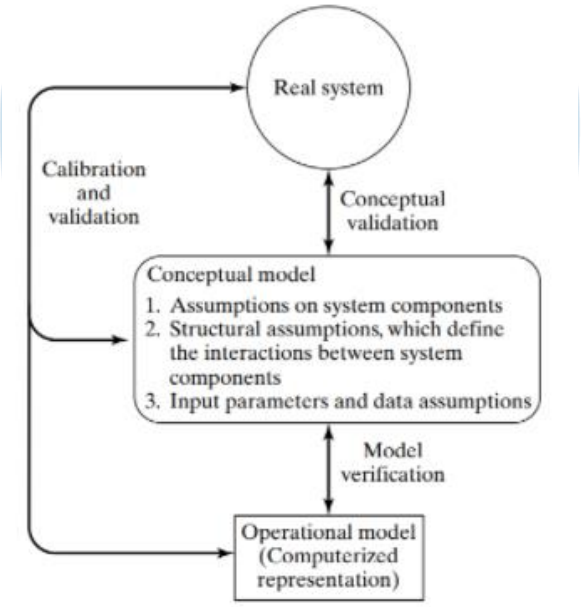
Hipotesis *null* akan ditolak apabila X_0^2 > *critical value* $X_{\alpha,k-s-1}$. Nilai *critical value* dapat dicari dengan bantuan tabel distribusi *Chi-square* yang ada pada Lampiran 4.

c. Verifikasi Model Simulasi

Proses verifikasi model simulasi dilakukan untuk memastikan bahwa model dirancang atau dibuat secara benar. Banks (2014) menyatakan bahwa proses verifikasi digunakan untuk membandingkan model konseptual dengan representasi model dari komputer yang akan mengimplementasikan konsep tersebut. Verifikasi berguna untuk memastikan bahwa struktur logika, parameter *input*, dan implementasi model secara keseluruhan sudah benar.

d. Validasi Model Simulasi

Proses validasi model simulasi menurut Banks (2014) dilakukan untuk memastikan bahwa peneliti merancang atau membuat model yang benar atau secara akurat mampu merepresentasikan sistem nyata secara akurat. Validasi model simulasi bisa dicapai dengan kalibrasi model simulasi yang dilakukan secara iteratif hingga model simulasi bisa diterima. Ilustrasi pembuatan model simulasi, verifikasi, dan validasi menurut Banks (2014) dapat dilihat pada Gambar 2.5 di bawah ini.

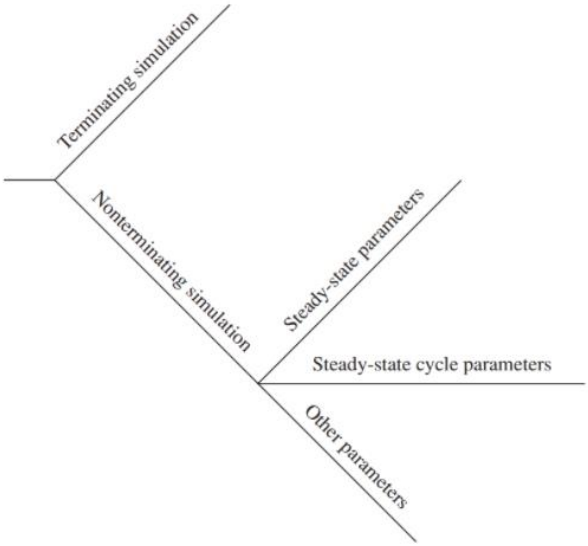


Gambar 2.5 Ilustrasi Pembuatan Model, Verifikasi, dan Validasi (Banks, 2014)

e. Analisis *Output* Model Simulasi

Simulasi berdasarkan hasil analisis *Output*nya dapat dibagi menjadi beberapa jenis simulasi yaitu *terminating simulation* dan *non-terminating simulation*. Law (2015) membagi *non-terminating simulation* menjadi 3 jenis yaitu *steady-state parameters*,

steady-state cycle parameters, dan parameter lain yang dapat dilihat pada Gambar 2.6 di bawah ini.



Gambar 2.6 Jenis Simulasi Berdasarkan Hasil *Output* (Law, 2015)

Terminating simulation adalah simulasi yang memiliki sebuah *event* khusus yang menentukan panjang dari setiap replikasi simulasi. *Event* khusus ini adalah variabel yang akan menentukan kapan berhentinya sebuah simulasi. Sedangkan *non-terminating simulation* merupakan kebalikan *terminating simulation* dimana tidak ada *event* khusus yang akan menentukan kapan berhentinya simulasi.

Faktor yang menentukan *Output* simulasi selain panjang replikasi adalah jumlah replikasi. Rosetti (2015) menyatakan rumus untuk menentukan jumlah replikasi dengan Persamaan 2.11 di bawah ini:

$$n \cong n_0\left(\frac{h_0}{h}\right)^2 \qquad (2.11)$$

Formula X.x menggunakan metode *half-width ratio*, dimana *n* adalah jumlah replikasi yang diperlukan, n_0 adalah jumlah replikasi awal, h_0 adalah nilai *half width* awal, dan *h* adalah nilai *half width* yang diinginkan.

f. Teknik Perbandingan Dua *Output* Simulasi

Aktivitas membandingkan dua *output* simulasi sama dengan aktivitas membandingkan suatu ukuran performansi dari dua *sample*. Rosetti (2015) membagi dua jenis

perbandingan *output* simulasi menjadi perbandingan dua sampel independen dan dua sampel dependen.

Uji statistik yang digunakan untuk membandingkan dua sampel independen dalam aplikasi *Arena Simulation: Output Analyzer* adalah *two sample t-test*. Montgomery (2016) telah membuat langkah prosedur untuk pengetesan dua hipotesis sesuai *two sample t-test* yang dapat dilihat di bawah ini:

i. *Null hypotesis* atau H_0 adalah $\mu_1 = \mu_2$

ii. *Alternative hypotesis* atau H_1 adalah $\mu_1 \neq \mu_2$

iii. *Test statistic* seperti pada Persamaan 2.12 di bawah:

$$T = \frac{\left(\overline{X}_1\right) - \left(\overline{X}_2\right)}{\sqrt{\left(\frac{S_1^2}{n_1}\right) + \left(\frac{S_2^2}{n_2}\right)}} \tag{2.12}$$

Dimana $\left(\overline{X}\right)$ adalah rata-rata sampel, *s* adalah standar deviasi sampel, dan *n* adalah jumlah sampel.

iv. *Reject null hypotesis* atau H_0 apabila $T > t_{(1-\alpha/2,v)}$. Dimana $t_{(1-\alpha/2,v)}$ adalah *critical value* dari distribusi *t* dengan derajat kebebasan *v* yang dapat dicari dengan tabel distribusi *t*. Persamaan 2.13 menunjukkan formula derajat kebebasan *v*.

$$v = \frac{\left(\left(\frac{S_1^2}{n_1}\right) + \left(\frac{S_2^2}{n_2}\right)\right)^2}{\left(\frac{\left(S_1^2/n_1\right)^2}{n_1 - 1}\right) + \left(\frac{\left(S_2^2/n_2\right)^2}{n_2 - 1}\right)} \tag{2.13}$$

2.2.6. Jenis Model Simulasi *Inventory Periodic Reviews*

Model *inventory periodic reviews* adalah model yang memperbolehkan variansi jumlah pemesanan produk pada interval waktu yang tetap. Waters (2009) mengatakan terdapat 2 jenis model *inventory periodic reviews* yaitu *periodic reviews* dengan *reorder level (r, S)* dan *periodic reviews* dengan *reorder level* dan *target stock level* (*r, s, S*).

Model *inventory* (*r, S*) adalah model yang melakukan pengecekan persediaan dengan interval waktu yang tetap dan hanya akan melakukan pemesanan ulang produk ketika

level persediaan berada di bawah *reorder point*. Selama level persediaan masih di atas *reorder point* maka pemesanan akan ditunggu hingga periode selanjutnya.

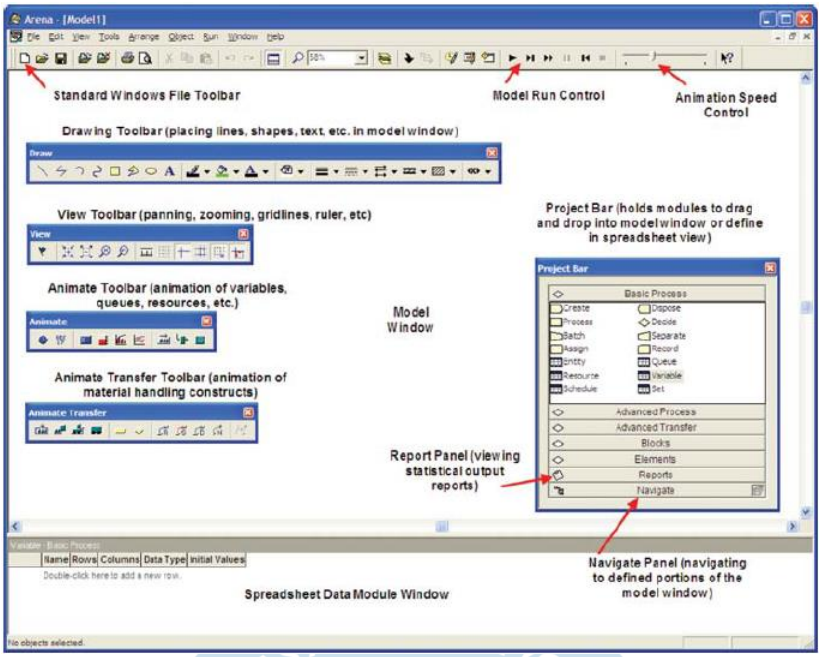
Model *inventory* (*r, s, S*) memiliki sifat yang mirip dengan model (*r, S*) namun ada perbedaan jumlah dalam pemenuhan persediaannya. Model (*r, s, S*) melakukan pengecekan persediaan dengan interval waktu yang tetap dan akan melakukan pemesanan ketika level persediaan berada di bawah *reorder point* atau *s*. Jumlah produk yang di pesan pada model (*r, s, S*) adalah untuk memenuhi *target stock level* atau *S* yang ditentukan. Penentuan jumlahnya adalah dari selisih *target stock level* dengan level persediaan akhir.

2.2.7. Tools Penelitian

a. *Arena Simulation*

Alat atau *tools* memiliki peranan penting dalam penelitian khususnya dalam pengolahan data. *Tools* untuk membantu penelitian dapat berupa aplikasi, *software*, algoritma, dan lain-lain. Penelitian teknik industri khususnya pada bidang *operations research* akan membutuhkan bantuan *tools* khususnya dalam melakukan perhitungan - perhitungan rumus.

Arena Simulation menurut Rosetti (2015) adalah sebuah *software* program komputer komersial yang memiliki fungsi untuk mengembangkan dan mengeksekusi model simulasi. Rosetti (2015) menjelaskan bahwa aplikasi ini memiliki panel yang menyediakan *toolbar* dan *menu* ke aktivitas simulasi sederhana seperti animasi model simulasi, menjalankan simulasi, dan mengamati hasil simulasi. Gambaran lingkungan atau *environment* aplikasi *Arena Simulation* serta kegunaannya dapat dilihat pada Gambar 2.7 di bawah ini:



Gambar 2.7 Lingkungan *Arena Simulation* (Rosetti, 2015)

Project bar terdiri dari berbagai macam *project templates* seperti *advanced transfers, advanced process, basic process,* dan *flow process*. Penulis dalam menyusun model simulasi persediaan akan menggunakan dua *template* yaitu *basic process* dan *advanced process*.

Template basic process dalam *Arena Simulation* memiliki banyak modul pemrograman. Modul pemrograman *basic process* yang akan digunakan penulis dalam membuat model simulasi adalah *Create, Record, Assign, Decide, Variable* dan *Dispose*. Fungsi dasar dari masing-masing modul pemrograman ini adalah:

i. *create*: modul ini akan membangkitkan entitas untuk melewati model simulasi sehingga logika model akan tereksekusi
ii. *assign*: modul yang akan memberikan *value* pada variabel yang ada di model
iii. *record*: modul ini berfungsi untuk merekam statistik dalam bentuk ekspresi
iv. *dispose*: modul yang akan membuang entitas setelah selesai dieksekusi oleh model simulasi
v. *decide*: modul yang digunakan untuk membagi jalur aliran entitas berdasarkan probabilitas atau kondisi yang telah dibangkitkan
vi. *dispose*: modul ini akan membuang entitas dari modul setelah logikanya selesai tereksekusi.

vii. *variable*: modul ini memiliki fungsi untuk membangkitkan variabel dalam model dan juga menginisiasi nilainya.

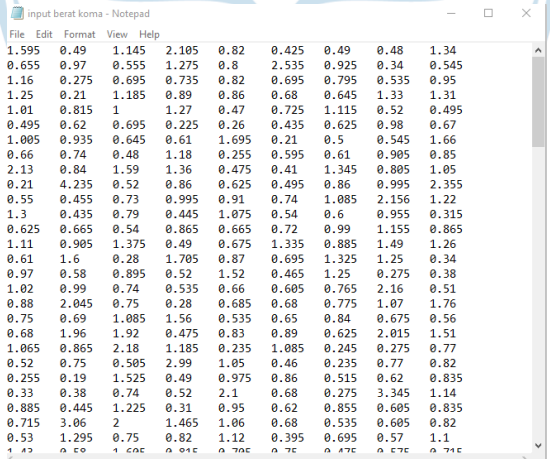
Template kedua yang digunakan oleh penulis dalam menyusun model simulasi adalah *advanced process*. Modul pemrograman dari template ini salah satunya adalah *statistic*. Modul ini memiliki fungsi merekam hasil statistik dari satu percobaan eksekusi model simulasi menjadi file berbeda. File yang dihasilkan dari modul statistik akan digunakan pada analisis *Output* menggunakan aplikasi *Arena: output analyzer*.

b. *Input Analyzer*

Input Analyzer merupakan *tools* yang digunakan untuk menentukan ekspresi *fit* distribusi terbaik dari sebuah data *input*. Penelitian ini menggunakan *Input Analyzer* untuk mengetahui ekspresi *fit* distribusi berat penjualan buah dari setiap pelanggan. Tahapan yang harus dilakukan untuk menggunakan *Input Analyzer* adalah sebagai berikut:

i. Mengubah Bentuk Data Menjadi .txt atau .dst

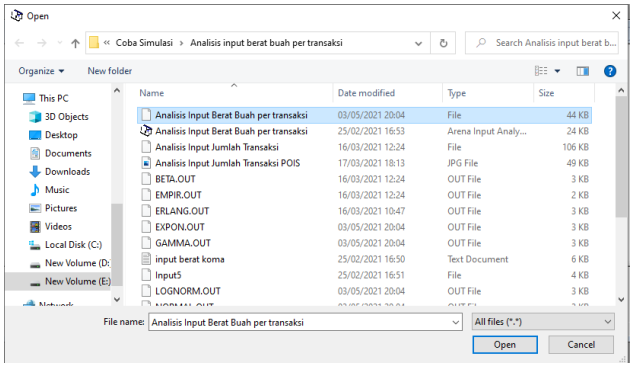
Data yang akan diolah oleh *Input Analyzer* harus diubah terlebih dahulu menjadi .txt atau .dst. Hal ini bisa dilakukan dengan bantuan notepad seperti contoh pada Gambar 2.8 di bawah ini:



Gambar 2.8 Data *Input* Mentah

ii. Memasukkan Data ke Input Analyzer

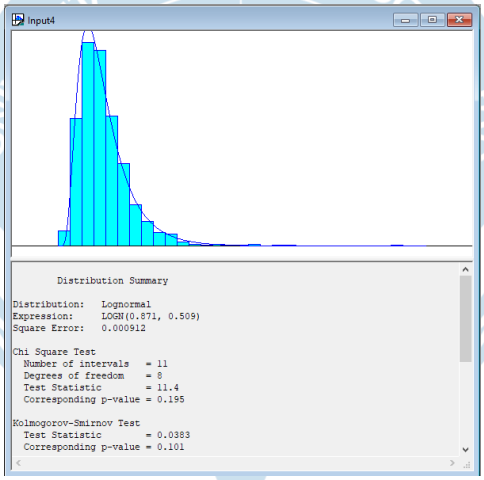
Langkah selanjutnya adalah membuka aplikasi *Input Analyzer* dan pilih menu "*File"* lalu "*Data File"* dan klik "*existing file"* untuk memilih file .dst atau .txt yang telah disimpan seperti pada Gambar 2.9.



Gambar 2.9 Pilih Data *Input*

iii. Melakukan Fit Distribusi

Langkah terakhir adalah melakukan *fit* distribusi untuk data *input* yang telah dimasukkan menggunakan fungsi "*Fit*" lalu "*Fit All*" sehingga muncul hasil seperti pada Gambar 2.10



Gambar 2.10 Hasil *Fit* Distribusi

Data yang diperlukan dari hasil *fit* distribusi adalah ekspresi dan hasil statistik *Chi-Square.* Ekspresi akan digunakan dalam pembangkitan data di model simulasi sedangkan hasil statistik *Chi-Square* digunakan untuk menguji apakah distribusi terpilih dapat diterima atau tidak secara statistik.

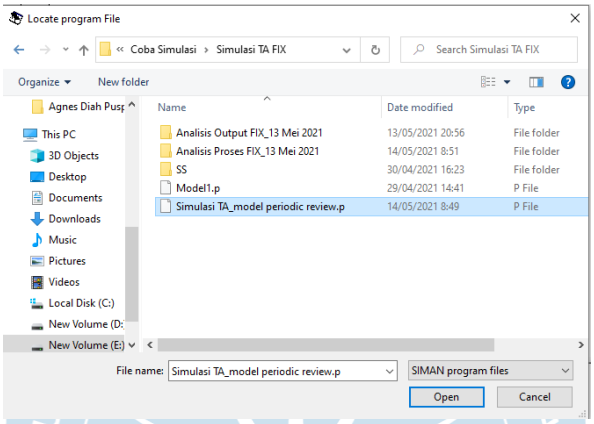
c. *Process Analyzer*

Process Analyzer merupakan *tools* yang digunakan untuk menjalankan banyak alternatif skenario model simulasi. *Tools* ini mampu membedakan alternatif skenario secara langsung dengan mengatur parameter kontrol (variabel keputusan) dan

parameter response yang diinginkan. Tahapan yang harus dilakukan untuk menggunakan *Process Analyzer* adalah sebagai berikut:

i. Memilih Skenario dari Arena Simulation

Pemilihan skenario dilakukan dengan cara terlebih dahulu membuat *file* baru pada *process analyzer* menggunakan menu "*File"* dan klik "*New".* Skenario dipilih dengan cara *double click* pada *work space* sehingga muncul tampilan seperti pada Gambar 2.11.



Gambar 2.11 Pilih Skenario *Process Analyzer*

Pilih *file* yang memiliki .p atau berjenis SIMAN *program files* lalu klik *Open*.

ii. Membangkitkan dan Menjalankan Skenario

Langkah selanjutnya adalah membangkitkan alternatif skenario dengan mengatur parameter kontrol yang berbeda-beda untuk setiap skenario sehingga akan dihasilkan respon yang berbeda juga. Setelah skenario sudah diatur maka dilanjutkan menjalankan skenario dengan fungsi "*Go".* Hasil *Process Analysis* dapat dilihat pada Gambar 2.12.

	Scenario Properties				Controls			Response
	S	Name	Program File	Reps	SiklusAlpukat	MaxStok	MinStok	Tally PROFIT
1		S1_M25	79 : Simulasi	14	1.0	25.0	10.0	922599.41
2		S1_M30	79 : Simulasi	14	1.0	30.0	10.0	1045337.19
3		S1_M35	79 : Simulasi	14	1.0	35.0	10.0	1107570.84
4		S1_M40	79 : Simulasi	14	1.0	40.0	10.0	1147765.11
5		S1_M50	79 : Simulasi	14	1.0	50.0	10.0	1165525.99
6		S1_M60	79 : Simulasi	14	1.0	60.0	10.0	1137837.13
7		S1_M70	79 : Simulasi	14	1.0	70.0	10.0	1101279.94
8		S1_M80	79 : Simulasi	14	1.0	80.0	10.0	1063175.01
9		S2_M25	79 : Simulasi	14	2.0	25.0	10.0	805393.45
10		S2_M30	79 : Simulasi	14	2.0	30.0	10.0	927440.15
11		S2_M35	79 : Simulasi	14	2.0	35.0	10.0	1042106.98
12		S2_M40	79 : Simulasi	14	2.0	40.0	10.0	1076376.90
13		S2_M50	79 : Simulasi	14	2.0	50.0	10.0	1115463.95
14		S2_M60	79 : Simulasi	14	2.0	60.0	10.0	1113738.60
15		S2_M70	79 : Simulasi	14	2.0	70.0	10.0	1085723.71
16		S2_M80	79 : Simulasi	14	2.0	80.0	10.0	1079462.95

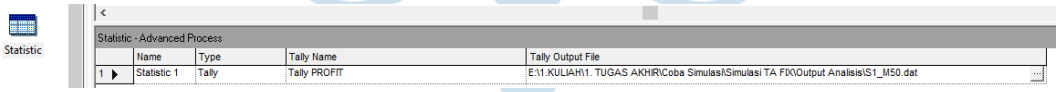
Gambar 2.12 Hasil *Process Analysis*

d. *Output Analyzer*

Output Analyzer merupakan *tools* atau aplikasi yang digunakan untuk membandingkan dua *output* berupa *statistic* dari model simulasi *Arena* Simulation. Tahapan yang harus dilakukan untuk menggunakan *Output Analyzer* adalah sebagai berikut:

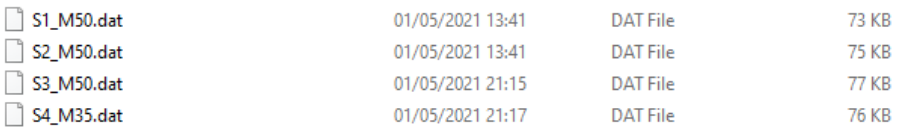
i. Aktifkan Modul *Statistic*

Modul *statistic* pada *Bar: Advanced Process* perlu diaktifkan untuk menghasilkan *file output* bertipe .dat yang akan digunakan pada *Output Analyzer*. Tipe yang dipilih adalah *tally Profit* sebagai *output* skenario yang akan dibandingkan seperti pada Gambar 2.13 di bawah ini.



Gambar 2.13 Modul *Statistic* Simulasi

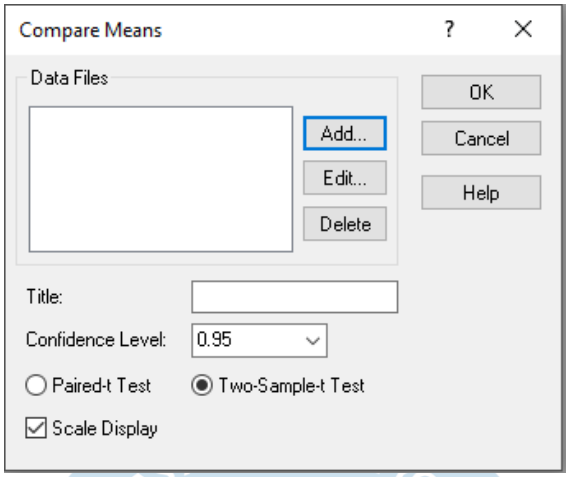
Setelah *file output* sudah tersimpan lakukan *run* simulasi, dan lakukan lagi menggunakan skenario lain dengan mengganti nama *file output* sebelum melakukan *run* simulasi selanjutnya. Maka akan tersimpan 4 *file output* bertipe .dat seperti pada Gambar 2.14



Gambar 2.14 *File Output* Skenario

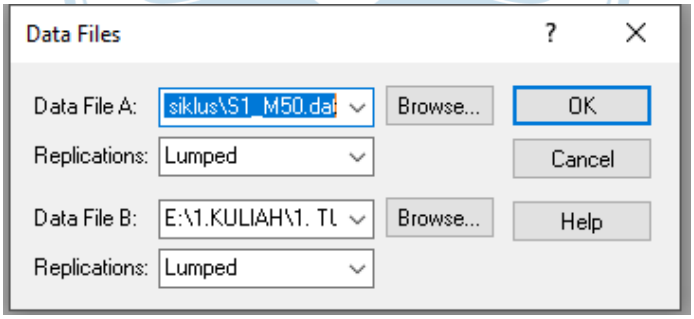
ii. *Compare Means* dengan *Output Analyzer*

Langkah pertama adalah membuka aplikasi *Output Analyzer* dan pilih fitur *Compare Means* seperti pada Gambar 2.15.



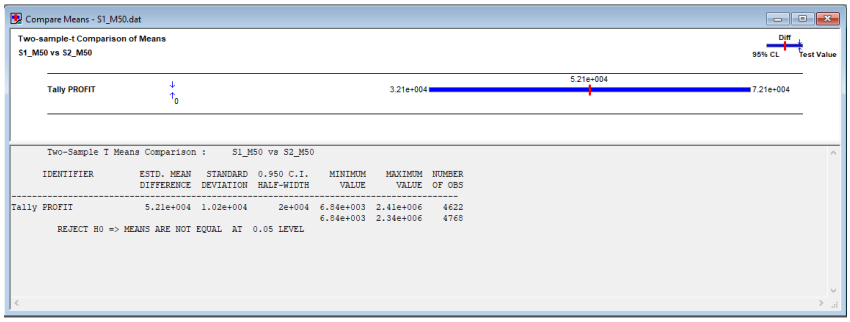
Gambar 2.15 *Compare Means Output Analyzer*

Two sample t-test akan menggunakan *confidence level* sebesar 0,95. Langkah selanjutnya adalah memilih dua *file* data *output* skenario yang akan dibandingkan seperti pada Gambar 2.16.



Gambar 2.16 Pemilihan Dua Data *Output* Skenario

Terakhir klik OK dan akan muncul hasil *output analyzer* seperti pada Gambar 2.17



Gambar 2.17 Hasil *Output Analyzer*

Hasil pada Gambar 2.17 menunjukkan *"Means Are Not Equal At 0,05 Level"* yang berarti kedua skenario simulasi menghasilkan profit yang berbeda.

e. *Software R Programming + Package "fitditsrplus"*

Package "*fitdistrplus"* pada bahasa pemrograman R digunakan untuk melakukan *fitting* distribusi *univariate* pada setiap jenis data yang berbeda. *Package* ini juga memiliki fungsi membandingkan estimasi parameter distribusi seperti uji statistik *Chi-Square.* Penelitian ini menggunakan *tools software R programming* dan *package* "fitdistrplus" untuk mengetahui *fit* distribusi kedatangan pelanggan per hari.

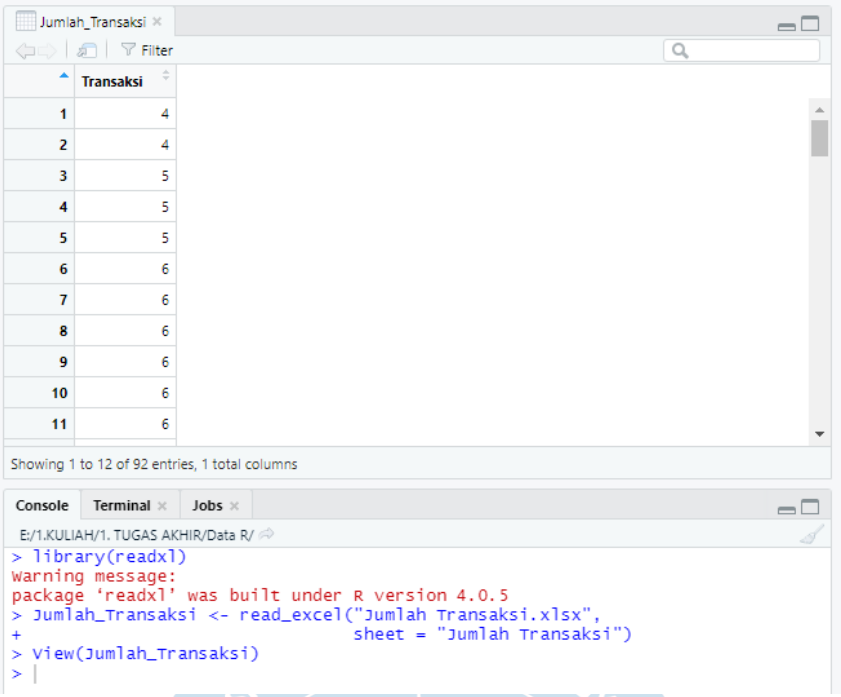
Tahapan penentuan *fit* distribusi probabilitas diskret terbaik adalah sebagai berikut:

i. *Install Package "fitdistrplus"*

Langkah pertama adalah menginstall *package "fitditstrplus"* dengan cara mengetik *install.packages (fitdistrplus)*. Selanjutnya adalah mengaktifkan *package* tersebut dengan mengetik *library (fitdistrplus*).

ii. *Insert* Data Jumlah Pelanggan ke *R*

Cara memasukkan data dari *Excel* ke *R* adalah menggunakan bantuan *package "readxl"*. Penulisan koding lengkap dapat dilihat pada Gambar 2.18 di bawah.



Gambar 2.18 *Insert* Data *Excel* ke *R*

iii. Membangkitkan Variabel Jumlah Transaksi

Data jumlah pelanggan per hari dari *file Excel* harus kita bangkitkan menjadi variabel agar bisa diolah oleh *R*. Cara membangkitkan variabel dilakukan dengan cara mengetik "x <- (Jumlah_Transaksi$Transaksi)

iv. Membandingkan *Fit* Distribusi Probabilitas Diskret

Fungsi yang digunakan untuk melakukan *fit* distribusi adalah *"fitdist"*. Perbandingan distribusi juga akan menggunakan fungsi *"gofstat"* untuk mengetahui *chi-square p value*, *chi-square degree of freedom*, dan *chi-square test statistic* yang akan digunakan untuk perbandingan distribusi probabilitas diskret.