

BAB II

KAJIAN LITERATUR

Penelitian mengenai kualitas proses telah dilakukan beberapa kali. Diantara penelitian-penelitian tersebut adalah penelitian yang dilakukan oleh Bayu (2007), melakukan penelitian tentang kualitas proses dan disimpulkan bahwa penambahan inspeksi *in-process* mampu mengurangi jumlah produk cacat dan biaya kualitas secara keseluruhan. Dalam penelitian tersebut, dilakukan pengumpulan data atribut untuk mengetahui jenis cacat terbanyak. Selain itu dilakukan juga penelitian terhadap biaya kualitas dan penggunaan metode FMEA dan PICA untuk mengetahui penyebab cacat yang paling dominan dan bagaimana usulan perbaikannya.

Isnawati (2008), melakukan penelitian tentang peningkatan kualitas proses dengan menggunakan *six sigma*. Dimana penelitian ini memberikan bukti empiris bahwa tentang pengaruh produk cacat terhadap biaya kualitas. Penelitian dilakukan dengan mengumpulkan data atribut dan variabel untuk mencari jumlah cacat terbanyak, dan juga meneliti tentang biaya kualitasnya.

Febriani (2006), dalam penelitian ini menggunakan metode *six sigma* dengan *tool* DMAIC. Dan didapat kesimpulan bahwa penggunaan *six sigma* dapat mengurangi produk cacat dan pemborosan dengan waktu relatif cepat.

Muchtari dan Noviyarsi (2007), menggunakan implementasi 5S ke dalam *lean sigma* dan didapat kesimpulan bahwa implementasi 5S dalam *lean sigma* dapat meminimalisasi pemborosan seperti meminimalisasi waktu untuk mencari alat, waktu tunggu dan waktu transportasi.

Penelitian serupa juga dilakukan oleh Kania (2007) dimana penelitian ini berusaha mengidentifikasi kualitas proses dengan menggunakan metode *six sigma* dengan *tool* DMAIC dan didapat kesimpulan bahwa faktor penyebab terjadinya pemborosan (*waste*) adalah adanya penambahan pengerjaan ulang (*rework*).

Mason et.al. (2008) menggunakan simulasi untuk mengoptimalkan alur proses manufaktur. Selain itu, simulasi juga dapat menganalisis pengukuran performansi, seperti utilitas mesin, utilitas tenaga kerja dan alur kerja berlebih yang tidak dapat diperkirakan lebih awal.

Kuhl et.al. (2005) menggunakan pendekatan simulasi dalam *Design For Six Sigma* (DFFS) *tools* untuk mengurangi produk cacat, proses *rework*, dan ketepatan waktu dalam memenuhi target produksi. Selain itu simulasi dapat juga mengidentifikasi *robust* dengan lebih cepat dan mudah.

Dari kajian literatur yang telah dilakukan, dapat diketahui bahwa penelitian sebelumnya hanya menentukan *baseline* kerja dan kapabilitas proses, tanpa memperhatikan adanya pemborosan yang terdapat pada proses. Penelitian yang dilakukan oleh Kuhl et.al. (2005) membuktikan bahwa hasil dari analisis *six sigma* dapat dijadikan input simulasi dalam menyelesaikan permasalahan kualitas.

Dalam penelitian ini akan dilakukan suatu rekomendasi perbaikan kualitas yang tidak hanya didasarkan pada *baseline* kerja dan kapabilitas proses tapi juga memperhatikan adanya pemborosan yang terdapat pada proses serta dampak bagi proses selanjutnya dan memperhitungkan biaya total produksinya dengan menggabungkan metode analisis *lean sigma* dan pendekatan simulasi. Dimana metode *lean sigma* dapat mengidentifikasi dan mengeliminasi pemborosan pada proses, dan output dari analisis ini akan digunakan sebagai input dalam membangun model

simulasi. Sehingga dari hasil penelitian bisa menjadi bahan pertimbangan manajemen perusahaan dalam mengambil keputusan yang berkaitan dengan perbaikan kualitas.

Sedangkan parameter yang akan dijadikan dasar dalam penelitian ini adalah total produk baik, jumlah produk cacat, beban kerja, dan biaya total produksi per unit. Dimana jumlah produk cacat dan biaya kualitas setelah usulan perbaikan diterapkan dalam model simulasi diharapkan dapat berkurang secara signifikan. Sehingga perusahaan dapat melakukan kebijakan perbaikan kualitas secara lebih tepat dan terarah karena dengan kualitas proses produksi yang baik maka akan mampu menurunkan produk cacat.

2.1 Pengertian Kualitas

Dalam mendefinisikan “kualitas” banyak ahli yang mendefinisikan secara berbeda-beda hal ini dikarenakan kualitas memiliki pengertian yang bervariasi. Berikut ini adalah pengertian tentang definisi kualitas (Dorothea, 2003) :

1. Menurut Juran, “kualitas adalah kesesuaian dengan tujuan atau manfaatnya.”
2. Menurut Crosby, “kualitas adalah kesesuaian dengan kebutuhan yang meliputi *availability, delivery, reliability, maintainability, dan cost effectiveness.*”
3. Menurut Deming, “kualitas harus bertujuan memenuhi kebutuhan pelanggan sekarang dan di masa mendatang.”
4. Menurut Feigenbaum, “kualitas merupakan kebutuhan karakteristik produk dan jasa yang meliputi *marketing, engineering, manufacture, dan maintenance*, dalam mana produk dan jasa tersebut dalam pemakaiannya akan sesuai dengan kebutuhan dan harapan pelanggan.”

5. Menurut Scherkenbach, "kualitas ditentukan oleh pelanggan, pelanggan menginginkan produk dan jasa yang sesuai dengan kebutuhan dan harapannya pada suatu tingkat harga tertentu yang menunjukkan nilai produk tersebut."
6. Menurut Elliot, "kualitas adalah sesuatu yang berbeda untuk orang yang berbeda dan tergantung pada waktu dan tempat, atau dikatakan sesuai dengan tujuan."
7. Menurut Goetch dan Davis, "kualitas adalah suatu kondisi dinamis yang berkaitan dengan produk, pelayanan, orang, proses, dan lingkungan yang memenuhi atau melebihi apa yang diharapkan."
8. Perbendaharaan istilah ISO 8402 dan dari Standar Nasional Indonesia (SNI 19-8402-1991), kualitas adalah keseluruhan ciri dan karakteristik produk atau jasa yang kemampuannya dapat memuaskan kebutuhan, baik yang dinyatakan secara tegas maupun tersamar. Istilah kebutuhan diartikan sebagai spesifikasi yang tercantum dalam kontrak maupun kriteria-kriteria yang harus didefinisikan terlebih dahulu.

Dengan demikian kualitas didefinisikan sebagai konsistensi peningkatan atau perbaikan dan penurunan variasi karakteristik dari suatu produk yang dihasilkan, agar memenuhi kebutuhan yang telah dispesifikasikan, guna meningkatkan kepuasan pelanggan internal maupun eksternal.

Pada dasarnya performansi kualitas dapat ditentukan dan diukur berdasarkan karakteristik kualitas yang terjadi dari beberapa sifat atau dimensi berikut (Gaspersz, 2006) :

1. Fisik : panjang, besar, diameter, tegangan, kekentalan, dll.
2. *Sensory* (berkaitan dengan panca indera) : rasa, penampilan, warna, bentuk, model, dll.

3. Orientasi waktu: keandalan (*reliability*), kemampuan pelayanan (*serviceability*), kemudahan pemeliharaan (*maintenanceability*), ketetapan waktu penyerahan produk, dll.
4. Orientasi biaya : berkaitan dengan dimensi biaya yang menggambarkan harga atau ongkos dari suatu produk yang harus dibayarkan oleh konsumen.

2.1.1 Pengertian Tentang Pengendalian

Arti kendali dalam industri dapat didefinisikan sebagai suatu proses untuk mendelegasikan tanggung jawab dan wewenang untuk kegiatan manajemen sambil tetap menggunakan cara-cara untuk menjamin hasil yang memuaskan.

Berdasarkan pada waktu pelaksanaan pengendalian, dikenal tiga macam pengendalian yaitu :

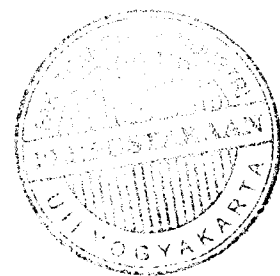
a. Pengendalian sebelum proses (*Preventive Control*)

Dimaksudkan agar produksi dapat berjalan sesuai dengan rencana, meliputi pemeriksaan terhadap :

1. Rencana produksi.
2. Desain produk.
3. Mesin/peralatan.
4. Bahan baku/penolong.
5. Tenaga kerja.

b. Pengendalian pada saat proses berlangsung

Hal ini bertujuan untuk mengendalikan apabila terjadi penyimpangan-penyimpangan terhadap standar yang telah ditetapkan oleh perusahaan dapat segera dilakukan koreksi.



c. Pengendalian setelah proses (*Repressive Control*)

Pengendalian ini dimaksudkan sebagai pencegahan terjadinya penyimpangan-penyimpangan yang telah terjadi selama proses produksi yang dapat dijadikan bahan pertimbangan untuk mencegah terjadinya penyimpangan yang akan datang.

2.1.2 Pengendalian Kualitas

Pengendalian kualitas pada umumnya ada empat langkah yaitu :

1. Menetapkan standar, yaitu menentukan standar mutu biaya, standar mutu prestasi kerja, standar mutu keamanan, dan standar mutu.
2. Menilai kesesuaian, yaitu membandingkan masalah dari produk yang dibuat, atau jasa yang ditawarkan terhadap standar-standar yang telah ditetapkan.
3. Bertindak bila diperlukan. Mengoreksi masalah dan penyebabnya melalui faktor-faktor yang meliputi pemasaran, perancangan, rekayasa, produksi dan pemeliharaan yang mempengaruhi kepuasan pelanggan.
4. Merencanakan perbaikan. Mengembangkan upaya yang kontinyu untuk memperbaiki standar-standar biaya, prestasi, keamanan dan keterandalan.

2.1.3 Tes Kecukupan dan Keseragaman Data

2.1.3.1 Tes Kecukupan Data

Tes kecukupan data adalah aktivitas yang dilakukan dengan tujuan untuk menentukan jumlah observasi yang harus dilakukan (N') untuk melakukan perhitungan ini terlebih dahulu harus ditetapkan tingkat kepercayaan (k) dan derajat ketelitian (s) serta jumlah pengamatan untuk elemen kerja yang telah diukur.

$$N' = \left[\frac{k / s \sqrt{N \sum Xi^2 - (\sum Xi)^2}}{\sum Xi} \right]^2$$

Dimana : Xi = waktu siklus

N = jumlah data pengamatan

k = harga indeks yang besarnya tergantung tingkat kepercayaan.

$k = 68\%$ maka nilai tingkat kepercayaan = 1

$k = 95\%$ maka nilai tingkat kepercayaan = 2

$k = 99\%$ maka nilai tingkat kepercayaan = 3

s = derajat ketelitian yang besarnya tergantung tingkat kepercayaan.

Misal untuk tingkat kepercayaan 95 % maka nilai tingkat kepercayaan = 2 dan $s = 5\%$ artinya sekurang-kurangnya 95 dari 100 rata-rata waktu yang diukur untuk suatu elemen kerja akan mempunyai penyimpangan tidak lebih dari 5 %.

Jika $N' < N$ maka data dapat dikatakan sudah mencukupi dan sebaliknya jika $N' > N$ maka data yang telah dikumpulkan belum mencukupi, sehingga perlu dilakukan penambahan data dan di uji kembali.

2.1.3.2 Tes Keseragaman Data

Dalam pengukuran waktu kerja tugas pengukuran adalah mendapatkan data yang seragam. Karena ketidak seragaman data tanpa disadari maka perlu sesuatu alat untuk “mendeteksinya”. Batas kontrol yang dibentuk dari data merupakan batas seragam tidaknya data. Data dikatakan seragam jika berada diantara kedua batas kontrol, dan tidak seragam jika berada diluar dua batas kontrol. Kedua batas tersebut adalah *Upper Control Limit* (UCL) dan *Lower Control Limit* (LCL).

$$\bar{X} = \frac{\sum Xi}{n}$$

$$SD = \frac{\sqrt{\sum (Xi - \bar{X})^2}}{n-1}$$

Dimana : \bar{X} = rata-rata

Xi = hasil pengamatan

SD = standar deviasi

\bar{X} = rata-rata hasil pengamatan

n = jumlah pengamatan

$$UCL = \bar{X} + k.SD$$

$$LCL = \bar{X} - k.SD$$

Apabila persebaran data masih dalam batas antara UCL dan LCL maka data dianggap seragam atau tidak menyimpang. Sebaliknya, jika data berada di luar batas UCL dan LCL maka data yang telah dikumpulkan belum seragam, sehingga data yang berada di luar kontrol di hilangkan. Kemudian dilakukan pengumpulan data dan di uji kembali untuk mendapatkan data yang seragam.

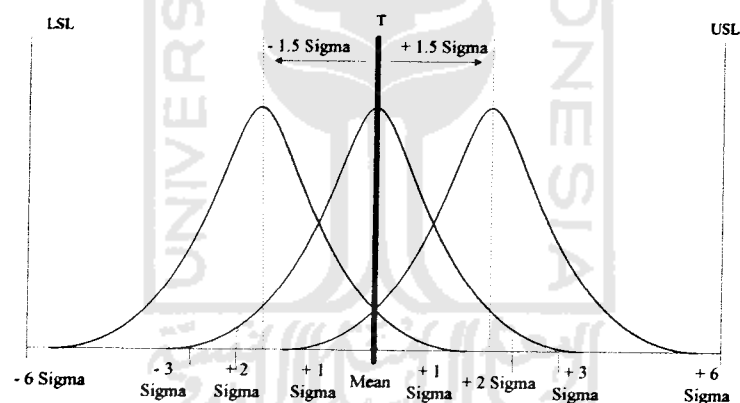
2.2 Pengertian Six Sigma

Six Sigma Motorola merupakan suatu metode atau teknik pengendalian dan peningkatan kualitas dramatik yang diterapkan oleh perusahaan Motorola sejak tahun 1986, yang merupakan terobosan baru dalam bidang manajemen kualitas. Prinsip-prinsip pengendalian dan peningkatan kualitas *Six Sigma* Motorola mampu menjawab tantangan ini, dan terbukti perusahaan Motorola selama kurang lebih 10 tahun setelah implementasi konsep *Six Sigma* telah mampu mencapai tingkat kualitas 3,4 DPMO (*defect per million opportunities*-kegagalan per sejuta kesempatan).

Beberapa keberhasilan Motorola yang patut dicatat dari aplikasi program *Six Sigma* adalah sebagai berikut :

1. Peningkatan produktivitas rata-rata 12,3% per tahun.
2. Penurunan COPQ (*cost of poor quality*) lebih daripada 84%.
3. Eliminasi kegagalan dalam proses sekitar 99,7%.
4. Penghematan biaya manufakturing lebih daripada \$11 miliar.
5. Peningkatan tingkat pertumbuhan tahunan rata-rata : 17% dalam penerimaan, keuntungan, dan harga saham Motorola.

Proses *Six Sigma* dengan distribusi normal yang mengizinkan nilai rata-rata (*mean*) proses bergeser 1,5 sigma dari nilai spesifikasi target kualitas (T) yang diinginkan oleh pelanggan, ditunjukkan pada gambar 2.1.



Gambar 2.1 Konsep *Six Sigma* Motorola Dengan Distribusi Normal Bergeser 1,5-sigma

Perlu dicatat dan dipahami sejak awal bahwa konsep *Six Sigma* Motorola dengan pergeseran nilai rata-rata (*mean*) dari proses yang diizinkan sebesar 1,5-sigma (1,5 standar deviasi maksimum) adalah berbeda dari konsep *Six Sigma* dalam distribusi normal yang umum dipahami selama ini yang tidak mengizinkan pergeseran dalam nilai rata-rata (*mean*) dari proses. Dengan demikian berdasarkan konsep *Six Sigma* Motorola, berlaku toleransi penyimpangan: (*mean* - Target) atau $(\mu - T) =$

$1,5\sigma$ atau $\mu = T \pm 1,5\sigma$. Di sini μ (baca: mu) merupakan nilai rata-rata (*mean*) dari proses, sedangkan σ (baca: *sigma*) merupakan ukuran variasi proses. Perbedaan ini ditunjukkan dalam tabel 2.1.

Tabel 2.1 Perbedaan *True Six Sigma* Dengan *Motorola Six Sigma*

<i>True Six Sigma Process</i>			<i>Motorola Six Sigma Process</i>		
Batas spesifikasi (LSL-USL)	Persentase yang memenuhi spesifikasi (LSL – USL)	DPMO	Batas spesifikasi (LSL-USL)	Persentase yang memenuhi spesifikasi (LSL – USL)	DPMO
± 1 -sigma	68,27%	317.300	± 1 -sigma	30,8538%	691.462
± 2 -sigma	95,54%	45.500	± 2 -sigma	69,1462%	308.538
± 3 -sigma	99,73%	2.700	± 3 -sigma	93,3193%	66.807
± 4 -sigma	99,9937%	63	± 4 -sigma	99,3790%	6.210
± 5 -sigma	99,99943%	0,57	± 5 -sigma	99,9767%	233
± 6 -sigma	99,999998%	0,002	± 6 -sigma	99,99966%	3,4

Dalam pendekatan *Six Sigma* terdapat konsep dasar yaitu pada dasarnya pelanggan akan puas apabila mereka menerima nilai yang mereka harapkan. Apabila produk diproses pada tingkat kinerja kualitas *Six Sigma*, perusahaan boleh mengharapkan 3,4 kegagalan per sejuta kesempatan (DPMO) atau bahwa 99,99966 persen dari apa yang diharapkan pelanggan akan ada dalam produk itu. Dengan demikian *Six Sigma* dapat dijadikan ukuran target kinerja proses industri tentang bagaimana baiknya suatu proses transaksi produk antara pemasok (industri) dan pelanggan (pasar). Semakin tinggi nilai target *sigma* yang dicapai, semakin baik kinerja proses industri. *Six Sigma* juga dapat dipandang sebagai pengendalian proses industri berfokus pada pelanggan, melalui penekanan pada kemampuan proses (*process capability*).

Apabila konsep *Six Sigma* akan diterapkan dalam bidang *manufacturing*, ada enam aspek yang perlu diperhatikan :

1. Identifikasi karakteristik produk sesuai ekspektasi pelanggan.

2. Klasifikasi karakteristik kualitas sebagai CTQ (*Critical-to-Quality*) individual.
3. Menentukan apakah setiap CTQ dapat dikendalikan melalui material, mesin, proses-proses kerja dan lain-lain.
4. Menentukan batas maksimum toleransi untuk setiap CTQ sesuai dengan ekspektasi pelanggan (menentukan nilai LCL dan UCL dari setiap CTQ).
5. Menentukan maksimum variasi proses untuk setiap CTQ (menentukan nilai maksimum standar deviasi untuk setiap CTQ).
6. Mengubah desain produk dan atau proses sedemikian rupa agar mampu mencapai nilai target *Six Sigma* yang memiliki indeks kemampuan proses minimum sama dengan dua ($C_{pm} \geq 2$).

2.2.1 Metodologi *Six Sigma*

Metodologi yang digunakan dalam upaya mendukung metode *Six Sigma* tersebut adalah DMAIC (*Define, Measure, Analyze, Improvement, Control*). DMAIC digunakan untuk memperbaiki produk dan proses yang ada. Proses DMAIC sebaiknya digunakan saat produk atau proses dapat ditingkatkan untuk memenuhi atau menambah persyaratan kebutuhan pelanggan sembari untuk mendukung tujuan bisnis. Harus juga diperhatikan bahwa metodologi *Six Sigma* tidaklah kaku. Pendekatan bervariasi, kadang-kadang secara signifikan. Salah satu variasi itu adalah dalam tahap-tahap tersebut. Sebagian pendekatan menggunakan kelima tahap di atas, sebagian lagi tidak memasukkan tahap *define*. Inti dari semua ini adalah seperangkat alat yang bertujuan membantu para manajer dan karyawan memahami dan memperbaiki proses-proses yang kritis sehingga dapat tercipta kepuasan pelanggan (*customer satisfaction*).

Konsep DMAIC merupakan sebuah *close loop* di mana output dari tiap fase akan menjadi *input* bagi fase selanjutnya bahkan output dari fase terakhir dalam satu

loop (fase control), akan menjadi input bagi rencana/proyek perbaikan (tahapan DMAIC) selanjutnya, ini akan menjamin dilakukannya peningkatan yang berkelanjutan. Siklus DMAIC dapat dilihat pada Gambar 2.2. Sedangkan fase-fase dalam DMAIC meliputi :

1. Tahap *Define*

Merupakan langkah operasional pertama dalam program peningkatan kualitas *Six Sigma*. Pada tahap ini perlu didefinisikan beberapa hal yang terkait dengan : (1) kriteria pemilihan proyek *Six Sigma*, (2) peran dan tanggung jawab dari orang-orang yang akan terlibat dalam proyek *Six Sigma*, (3) kebutuhan pelatihan untuk orang-orang yang terlibat dalam proyek *Six Sigma*, (4) proses-proses kunci dalam proyek *Six Sigma* beserta pelanggannya, (5) kebutuhan spesifik dari pelanggan, dan (6) pernyataan tujuan proyek *Six Sigma*.

2. Tahap *Measure*

Merupakan langkah operasional kedua dalam program peningkatan kualitas *Six Sigma*. Terdapat tiga hal pokok yang harus dilakukan yaitu : (1) memilih atau menentukan karakteristik kualitas (CTQ) kunci yang berhubungan langsung dengan kebutuhan spesifik dari pelanggan, (2) mengembangkan suatu rencana pengumpulan data melalui pengukuran yang dapat dilakukan pada tingkat proses, output, dan/atau *outcome*, dan (3) mengukur kinerja sekarang (*current performance*) pada tingkat proses, output, dan/atau *outcome* untuk ditetapkan sebagai *baseline* kinerja pada awal proyek *Six Sigma*.

3. Tahap *Analyze*

Merupakan langkah operasional ketiga dalam program peningkatan kualitas *Six Sigma*. Pada tahap ini perlu dilakukan beberapa hal berikut : (1) menentukan stabilitas (*stability*) dan kapabilitas/kemampuan (*capability*) dari proses, (2)

menetapkan target-target kinerja dari karakteristik kualitas kunci (CTQ) yang akan ditingkatkan dalam proyek *Six Sigma*, (3) mengidentifikasi sumber-sumber dan akar penyebab kecacatan atau kegagalan, dan (4) mengkonversikan banyak kegagalan ke dalam biaya kegagalan kualitas (*cost of poor quality*).

4. Tahap *Improvement*

Setelah sumber-sumber dan akar penyebab dari masalah kualitas teridentifikasi, maka perlu dilakukan penetapan rencana tindakan (*action plan*) untuk melaksanakan peningkatan kualitas *Six Sigma*.

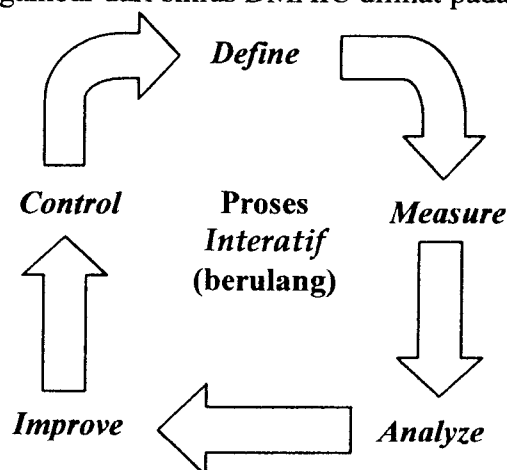
5. Tahap *Control*

Control merupakan tahap operasional terakhir dalam proyek peningkatan kualitas *Six Sigma*. Mengevaluasi solusi dan rencana, menjaga upaya-upaya yang telah dilakukan dengan menstandarisasi proses, dan juga memonitor langkah-langkah perbaikan.

Hasil dari tahap ini adalah :

- a. Analisa sebelum dan sesudah.
- b. Sebuah sistem monitoring.
- c. Dokumentasi hasil, pembelajaran, dan rekomendasi yang lengkap.

Berikut adalah gambar dari siklus DMAIC dilihat pada gambar 2.2.



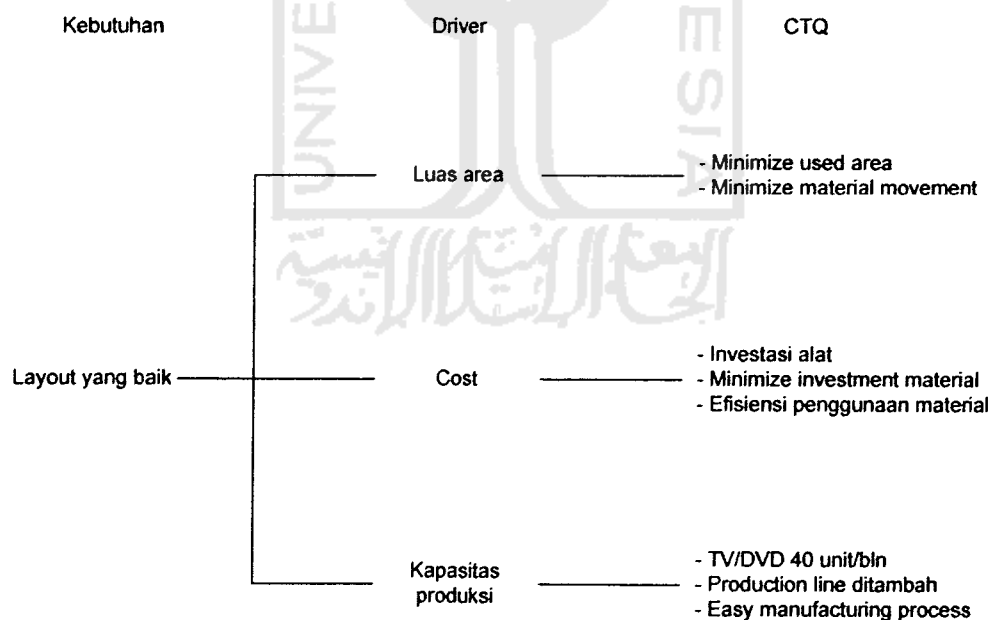
Gambar 2.2 Siklus Metode *Six Sigma* DMAIC

2.2.2 Tools Dalam Six Sigma

Salah satu dari pengertian *Six Sigma* adalah *Six Sigma* sebagai *tools*. Didalam *Six Sigma* banyak menggunakan *tools* perbaikan yang sebenarnya telah diterapkan pada program peningkatan kualitas sebelumnya. Akan tetapi ada beberapa *tools* dalam *Six Sigma* yang lebih komprehensif yang dapat digunakan untuk menganalisa masalah yang lebih kompleks. Berikut adalah beberapa *tools* yang digunakan :

1. CTQ (*Critical to Quality*) Tree

Tools ini digunakan untuk mengidentifikasi proses atau produk yang akan diperbaiki untuk menterjemahkan permintaan customer. Biasanya bentuknya hanya terdiri dari turunan masalah atau *breakdown* dari semua masalah sampai tercapai atau teridentifikasi masalah yang sesungguhnya guna memenuhi keinginan customer. Contoh *CTQ Tree* dapat dilihat pada gambar 2.3.



Gambar 2.3 Contoh *CTQ Tree*

2. SIPOC (*Supplier-Input-Proses-Output-Customer*)

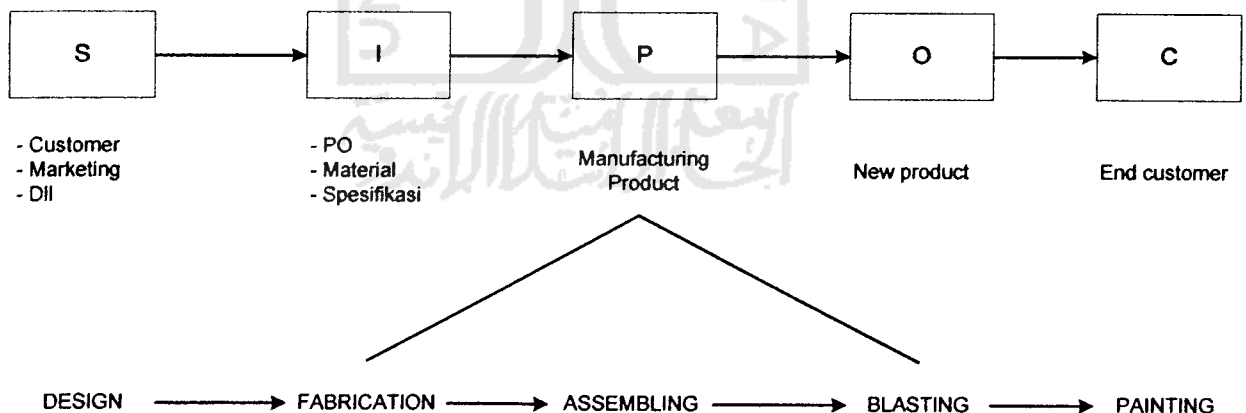
SIPOC adalah sebuah peta proses yang didalamnya teridentifikasi siapa pemasoknya, apa inputnya, bagaimana prosesnya, apa hasilnya dan siapa saja

pemakainya. Kualitas ditentukan oleh output yang dihasilkan, untuk itu output yang dihasilkan harus ditingkatkan dengan menganalisa input dan variabel-variabel yang ada didalamnya. SIPOC merupakan *tool* yang sangat efektif. SIPOC dapat memastikan bahwa semua orang akan melihat proses dalam cara pandang yang sama. Untuk itulah SIPOC harus ada pada tahap awal proyek. Proses dipetakan menjadi beberapa langkah, kemudian analisa mulai dijalankan dari kiri ke kanan, yaitu dari pelanggan (*customer*) ke pemasok (*supplier*).

Langkah-langkah proses *mapping* :

1. Menamakan proses.
2. Membuat batasan titik awal dan akhir proses.
3. Membuat daftar output dan pelanggan.
4. Membuat daftar input dan pemasok.
5. Identifikasi, beri nama dan urutkan langkah-langkah yang ada dalam proses.

Contoh Peta Proses dapat dilihat pada gambar 2.4.




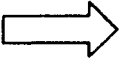




Gambar 2.4 Peta Proses

3. Diagram Operasi (*Operation Chart*)

Diagram operasi adalah alat untuk menggambarkan proses dalam bentuk yang ringkas, sehingga mudah untuk dimengerti. Diagram operasi akan memberi

gambaran secara grafis dari tiap-tiap kejadian dalam suatu pekerjaan. Simbol-simbol yang digunakan untuk membuat diagram proses antara lain adalah :

Tabel 2.2 Simbol Diagram Proses

	<i>Operation</i>	Operasi terjadi ketika suatu obyek dengan sengaja dirubah menjadi bentuk atau karakteristik lain
	<i>Transportation</i>	Transportasi terjadi ketika suayu obyek dipindahkan dari tempat satu ke tempat lain, tetapi bukan perpindahan yang masih terintegrasi dalam rangkaian kegiatan operasi
	<i>Inspection</i>	<i>Inspection</i> terjadi ketika suatu obyek diperiksa atau dibandingkan dengan standar baik dalam kuantitas maupun kualitas
	<i>Delay</i>	<i>Delay</i> terjadi ketika kegiatan selanjutnya yang mengikuti kegiatan sebelumnya tidak berjalan dengan segera (tertunda)
	<i>Storage</i>	<i>Storage</i> terjadi ketika suatu obyek disimpan dalam pengawasan, seperti misalnya pengawasan jumlah pengambilannya
	<i>Combined</i>	Dua simbol dapat dijadikan satu ketika menunjukkan aktifitas yang dikerjakan sekaligus bersamaan

Simbol-simbol tersebut berlaku seperti sebuah alat bantu dalam membuat langkah-langkah detail pada sebuah proses dengan cepat dan mudah, seperti urutan-urutan aktifitas seseorang, atau langkah-langkah yang dilalui oleh sebuah material. Penggunaan diagram proses secara benar akan dapat mendeteksi dan mengusulkan tindakan-tindakan perbaikan yang perlu dilakukan terhadap proses, misalnya pengurangan waktu, penggabungan aktifitas sehingga lebih efisien, dan lain-lain.

4. FMEA (*Failure Modes and Effect Analysis*)

Failure Modes and Effect Analysis adalah salah satu *tools* analisa yang sangat komprehensif. FMEA merupakan suatu prosedur yang digunakan untuk mengidentifikasi dan menilai resiko yang berhubungan dengan sumber potensial kegagalan produk atau proses. Dengan mengidentifikasi resiko,

sumber daya dapat dialokasikan untuk mengurangi atau menghilangkan kegagalan (*failure*). Langkah-langkah dalam membuat FMEA adalah :

1. Identifikasi *Potensial Failure Modes*.

Potensial Failure Modes adalah bentuk kegagalan dari produk, jasa atau proses.

2. Identifikasi *Potensial Modes Effect* dari tiap kegagalan dan dihitung *severity*.

a. *Potensial Effect* adalah dampak yang ditimbulkan bila *Failure Mode* tidak dicegah.

b. *Severity* adalah signifikan dampak yang ditimbulkan oleh *potensial effect* baik internal maupun eksternal.

3. Identifikasi *Causes* dan hitung *Occurence*.

a. *Causses* adalah kekurangan (*deficiency*) yang mengakibatkan kegagalan (*failure*).

b. *Occurance* adalah bagaimana kemungkinan sebab dari *failure* modes akan terjadi.

4. Hitung kemampuan untuk mendeteksi tiap *failure modes* (*detection*)

5. Kalikan ketiga angka yang ada, yaitu *Severity*, *Ocurance*, *Detection* untuk mendapatkan nilai RPN (*Risk Priority Number*)

$$RPN = SEV \times OCC \times DET$$

RPN adalah perhitungan numerik dari resiko relatif dari suatu kegagalan.

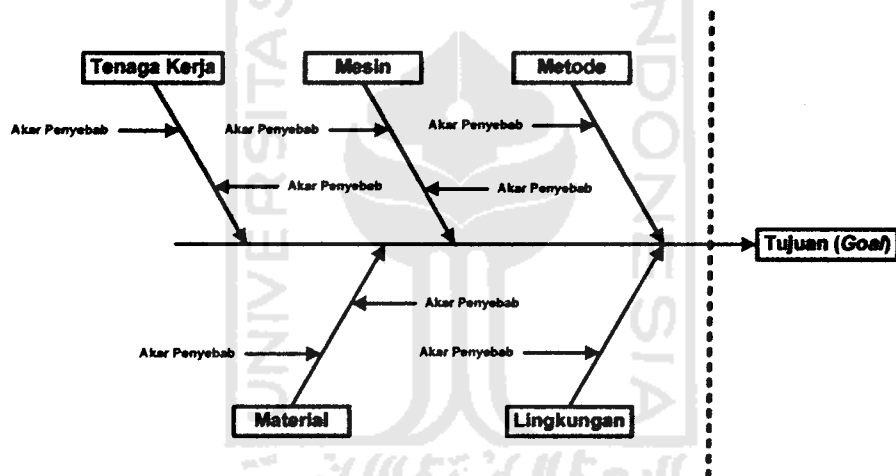
RPN digunakan untuk memberikan prioritas pada item mana yang membutuhkan tindakan perbaikan segera.

5. *Fishbone* Diagram

Fishbone Diagram atau diagram tulang ikan sering juga disebut juga sebagai diagram Ishikawa, sebab yang pertama kali menemukan metode ini adalah Kaoru

Ishikawa. Diagram tulang ikan adalah sebuah diagram yang menggambarkan hubungan antara karakteristik kualitas dengan berbagai faktor. *Tool* ini merupakan salah satu *tools* yang paling sering digunakan dalam perbaikan kualitas. *Tool* ini digunakan hanya untuk mengidentifikasi sebab dan akibat yang ditimbulkan.

Untuk mengilustrasikan pada sebuah diagram hubungan antara sebab dan akibat, kita ingin mengetahui sebab dan akibat dalam bentuk yang nyata. Oleh karenanya, akibat = karakteristik mutu dan sebab = faktor. Gambar 2.5 "disebut gambar sebab-akibat". Dalam praktek umum, faktor harus ditulis lebih rinci untuk membuat diagram menjadi bermanfaat.



Gambar 2.5 Contoh Diagram Tulang Ikan

Faktor yang termasuk dalam permasalahan mutu pada pabrik kita hampir tidak terhitung. Diagram sebab-akibat berguna untuk membantu kita dalam memilih penyebab penyebaran dan mengorganisasikan hubungannya.

6. PICA (*Problem Identification and Corrective Action*)

PICA adalah *tools* yang digunakan dalam metode *Six Sigma* yaitu pada tahap perbaikan (*Improve*). PICA berbentuk tabel yang isinya diperoleh dari hasil FMEA yang memiliki RPN tertinggi atau yang berwarna merah. Hasil yang akan diperoleh oleh PICA adalah berupa usulan-usulan kegiatan perbaikan terhadap

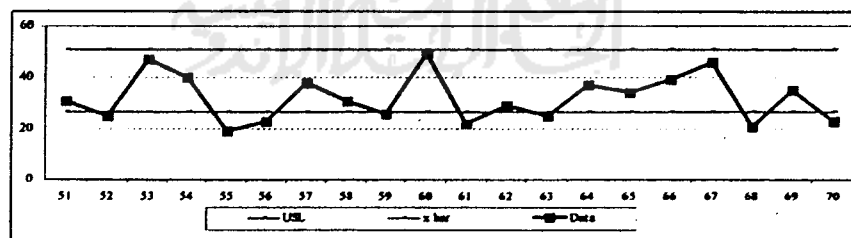
sebab-sebab masalah yang sudah diidentifikasi sebelumnya. Contoh tabel PICA ditunjukkan pada tabel 2.3.

Tabel 2.3 Contoh Tabel PICA

No.	Masalah	Perbaikan	Mengapa	Bagaimana	Kapan	Biaya	Dimana	PIC

7. Control Chart (Grafik Kontrol)

Grafik kontrol adalah grafik yang secara khusus memberi kita informasi dalam dua dimensi, distribusi proses (rata-rata dan varian) dan kecenderungan proses. *Control Chart* dapat digunakan untuk memonitor dan mendeteksi perubahan-perubahan yang terjadi pada proses yang diakibatkan oleh variasi sebab khusus (*Special Cause Variation*). *Control Chart* terjadi dari beberapa jenis, seperti control chart p, np, Xbar-R dan lain-lain. Jenis *control chart* yang dipakai harus disesuaikan dengan jenis data yang ada.



Gambar 2.6 Contoh *Control Chart*

Grafik Kontrol (*Control Chart*) dapat dibedakan menjadi 2 :

1. Grafik kontrol untuk data atribut (Diskrit).
2. Grafik kontrol untuk data variabel (*Continuous*).

2.2.2.1 Data Atribut

Data atribut merupakan data karakteristik yang menunjukkan suatu sifat produk. Banyak karakteristik kualitas tidak dapat dengan mudah dinyatakan secara numerik. Biasanya yang diperiksa kita klasifikasi sebagai sesuai dengan spesifikasi pada karakteristik kualitas itu atau tidak sesuai dengan spesifikasi. Istilah “cacat” atau “tidak cacat” kadang-kadang digunakan untuk mengidentifikasi kedua klasifikasi produk ini. Karakteristik seperti ini dinamakan sifat (atribut). Termasuk di dalamnya peta kendali np.

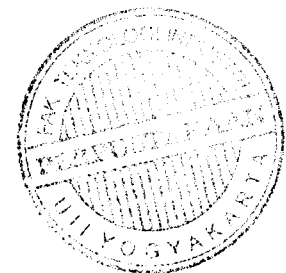
Peta Kendali $n\bar{p}$

Bila ukuran sampel beragam, peta kendali harus memperlihatkan bagian yang ditolak dan bukan jumlah aktual yang ditolak. Jika jumlah yang ditolak ditebar, garis pusat pada peta tersebut akan perlu diubah untuk setiap perubahan dalam ukuran sampel. Akan tetapi jika ukuran sampel konstan, peta untuk jumlah aktual yang ditolak dapat digunakan. Peta seperti itu disebut peta np atau pn. Maka perhitungan batas kendalinya :

$$CL \ n\bar{p} = \frac{\text{totalcacat}}{\text{banyaknyapengambilansampel}}$$

$$UCL = n\bar{p} + 3\sqrt{n\bar{p}(1 - \bar{p})}$$

$$LCL = n\bar{p} - 3\sqrt{n\bar{p}(1 - \bar{p})}$$



2.2.2.2 Data Variabel

Data variabel merupakan data karakteristik yang memiliki banyak variasi. Misalkan ukuran tinggi manusia. Data variabel dapat menunjukkan seberapa jauh penyimpangan dari standar proses. Dan digunakan untuk menggambarkan variasi atau penyimpangan yang terjadi pada kecendrungan memusat dan

penyebaran observasi. Pada peta kontrol variabel data diperluas harus dapat diukur dan karakteristik kualitas akan ditentukan oleh besar kecilnya penyimpangan terhadap ukuran yang distandarkan untuk hasil proses kerja yang berlangsung. Termasuk di dalamnya peta kendali \bar{x} dan R.

1. Peta kendali \bar{x}

Langkah-langkah dalam pembuatan peta \bar{x} adalah :

1. Pengumpulan data

Pengumpulan data biasanya dilakukan > 100 sampel, semuanya diambil dari proses yang sama dengan data yang diambil berurut.

2. Mengelompokkan data ke dalam sampel.

Ukuran sampel = n

Jumlah sampel = k

3. Mencatat data dalam lembar data.

Perhitungan \bar{x} dan R untuk setiap sampel pada lembar data.

4. Menghitung nilai rata-rata data.

Menjumlahkan seluruh sampel yang diambil kemudian dibagi dengan ukuran sampel.

5. Menghitung rentang.

Rumus yang digunakan untuk setiap sampel adalah :

$$R = X_{(terbesar)} - X_{(terkecil)}$$

6. Menghitung rata-rata keseluruhan ($\bar{\bar{X}}$)

Rata-rata keseluruhan merupakan total rata-rata setiap sampel yang dibagi dengan jumlah sampel.

7. Menghitung rata-rata rentangan (\bar{R}).

Seluruh nilai R dalam setiap sampel dijumlahkan lalu dibagi dengan jumlah sampel.

8. Menentukan garis batas kendali

Dalam peta kendali variabel \bar{x} ada tiga macam keadaan perhitungan garis batas pengendalian yaitu :

a. Apabila nilai standar diketahui (μ dan σ diketahui)

maka dari rumus awal : $\mu \pm Z_{\alpha/2}\sigma = \mu \pm 3\frac{\sigma}{\sqrt{n}}$ dan misal $\frac{3}{\sqrt{n}}=A$ maka

perhitungan peta kendalinya adalah :

$$CL = \mu$$

$$UCL_x = \mu + A\sigma$$

$$LCL_x = \mu - A\sigma$$

b. Apabila nilai standar tidak diketahui (μ dan σ tidak diketahui)

maka dari rumus awal : $\mu \pm Z_{\alpha/2}\sigma = \mu \pm 3\frac{\sigma}{\sqrt{n}}$ dan digunakan \bar{X}

sebagai penaksir μ dan $\frac{\bar{E}}{d_2}$ untuk penaksir σ . Maka peta kendali x

adalah :

$$CL = \bar{X}$$

$$UCL_x = \bar{X} + \frac{3}{d_2\sqrt{n}}\bar{R}$$

$$LCL_x = \bar{X} - \frac{3}{d_2\sqrt{n}}\bar{R}$$

dengan memisalkan $A_2 = \frac{3}{d_2\sqrt{n}}$ yang tergantung pada nilai n , maka

peta kendali \bar{x} menjadi sebagai berikut :

$$CL = \bar{X}$$

$$UCL_x = \bar{X} + A_2\bar{R}$$

$$LCL_x = \bar{X} - A_2\bar{R}$$

c. Apabila nilai standar tidak diketahui, penaksirannya melalui s .

maka kita gunakan \bar{x} sebagai penaksir μ dan $\frac{s}{c_4}$ digunakan untuk

menaksir σ . dengan memisalkan $A_3 = \frac{3}{c_4\sqrt{n}}$

$$CL = \bar{X}$$

$$UCL_x = \bar{X} + A_3\bar{R}$$

$$LCL_x = \bar{X} - A_3\bar{R}$$

2. Peta Kendali R

Untuk peta kendali R langkah-langkahnya hampir sama dengan langkah-langkah pada peta kendali \bar{x} , sedangkan untuk nilai batas kendalinya sebagai berikut :

1. Apabila nilai standar diberikan.

Untuk membentuk peta R dengan nilai standar σ , kita ingat bahwa

$\sigma = \frac{\bar{R}}{d_2}$, dengan d_2 adalah mean distribusi rentang relatif. Lagi pula

deviasi standar R adalah $\sigma_{R=d_3\sigma}$, dengan d_3 deviasi standar distribusi

rentang relatif. Maka parameter grafik pengendali itu adalah :

$$CL = d_{2\sigma}$$

$$UCL = D_2\sigma$$

$$LCL = D_1\sigma$$

dimana nilai $D_2 = d_2 + 3d_2$

$$D_1 = d_2 - 3d_2$$

2. Apabila nilai standar tidak diberikan.

Karena σ tidak diketahui, kita dapat menaksir σ_R dengan $\sigma_R = d_3 \frac{\bar{R}}{d_2}$

Dengan demikian, maka kita dapatkan parameter peta R dengan batas pengendali sebagai berikut :

$$CL = \bar{R}$$

$$UCL = \bar{R}D_4$$

$$LCL = \bar{R}D_3$$

dimana nilai : $D_3 = 1 - 3\frac{d_3}{d_2}$ dan $D_4 = 1 + 3\frac{d_3}{d_2}$

2.2.3 Analisis DPMO Dan Tingkat Sigma

2.2.3.1 Analisis DPMO dan Tingkat Sigma Untuk Data Atribut

Rumus perhitungan DPMO (Gaspersz, 2007) :

$$= \left\{ \frac{\sum Output_cacat}{\sum Output_diperiksa \times CTQ_Potensial} \right\} \times 1.000.000$$

Adapun rumus perhitungan tingkat *sigma* untuk data atribut yang digunakan dalam program *Microsoft Excel* adalah seperti berikut (Gaspersz, 2007) :

$$\text{Nilai } \sigma = \text{normsinv} ((1000000 - \text{DPMO}) / 1000000) + 1,5$$

2.2.3.2 Analisis DPMO dan Tingkat *Sigma* Untuk Data Variabel

Menentukan nilai DPMO dan tingkat *sigma* untuk data atribut dan data variabel.

- a. Kemungkinan cacat yang berada diatas nilai USL dengan rumus :

$$P \left[Z \geq \left(\frac{USL - X - \bar{bar}}{S} \right) \right] \times 1.000.000$$

- b. Kemungkinan cacat yang berada dibawah nilai LSL dengan rumus :

$$P \left[Z \geq \left(\frac{LSL - X - \bar{bar}}{S} \right) \right] \times 1.000.000$$

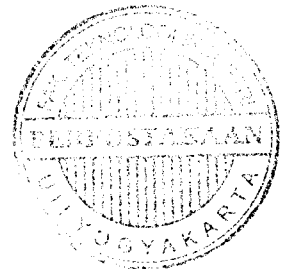
Sehingga DPMO diperoleh dengan $P(z > BPA) \times 1.000.000 + P(z < BPB) \times 1.000.000$ yang kemudian hasilnya dikonversikan kedalam nilai *sigma* dengan bantuan tabel.

Namun jika ingin mengetahui tingkat kegagalan per sejuta kesempatan (DPMO), gunakan formula berikut dalam program *Microsoft Excel* :

$$= 1000000 - \text{normsdist}(-1.5 + \text{NILAISIGMA}) * 1000000$$

untuk NILAI *SIGMA* = 2.5, maka gunakan formula berikut :

$$= 1000000 - \text{normsdist}(-1.5 + 2.5) * 1000000$$



2.2.4 Kapabilitas Proses

Indeks Kapabilitas Proses (Cpm) digunakan untuk mengukur pada tingkat mana output proses pada nilai spesifikasi target kualitas (T) yang diinginkan oleh pelanggan. Semakin tinggi nilai Cpm menunjukkan bahwa output proses itu semakin mendekati nilai spesifikasi target kualitas (T) yang diinginkan oleh pelanggan, yang berarti pula bahwa tingkat kegagalan dari proses semakin berkurang menuju target tingkat

$$\text{kegagalan nol (zero defect oriented) : } Cpm = \frac{(USL - LSL)}{6\sqrt{(Xbar - T)^2 + S^2}}$$

Bersamaan dengan penggunaan indeks Cpm, juga digunakan indeks Cpmk yang mengukur tingkat pada mana output proses itu berada dalam batas-batas toleransi yang diinginkan pelanggan dengan menggunakan formula :

$$Cpk = \text{minimum} \left[\frac{(X - \bar{x} - LSL)}{3S}; \frac{(USL - X - \bar{x})}{3S} \right]$$

$$Cpmk = \frac{Cpk}{1 + \left\{ \frac{(\bar{X} - T)}{S} \right\}^2}$$

Dan dalam program peningkatan kualitas *Six Sigma*, biasanya dipergunakan kriteria (*rule of thumb*) sebagai berikut untuk Cpm atau Cpmk.

1. Jika $Cpm \geq 2$, maka proses dianggap mampu dan kompetitif (perusahaan berkelas dunia).
2. Jika $1,00 < Cpm < 1,99$; maka proses dianggap cukup mampu namun perlu upaya-upaya peningkatan kualitas menuju target perusahaan berkelas dunia yang memiliki tingkat kegagalan sangat kecil menuju nol (*zero defect oriented*).
3. Jika $Cpm < 1$, maka proses dianggap tidak mampu dan tidak kompetitif untuk bersaing di pasar global.

Penentuan kapabilitas proses dapat juga dihitung menggunakan kalkulator *six sigma*, yang dapat diperoleh secara gratis dengan *mendownload website* www.spcwizard.com. Berikut ini adalah cara menghitung nilai kapabilitas proses dengan menggunakan kalkulator *six sigma* :

Pilih ● *variables*

USL = (masukkan nilai USL)

Average = (masukkan nilai rata-rata CTQ dari proses)

LSL = (masukkan nilai LSL)

Standard deviation = (masukkan nilai standar deviasi CTQ dari proses)

Pilih *calculate*

Process Sigma = (dihitung sendiri oleh kalkulator)

2.3 Pengertian *Lean*

Lean adalah suatu upaya terus-menerus untuk menghilangkan pemborosan (*waste*) dan meningkatkan nilai tambah (*value added*) produk agar memberikan nilai kepada pelanggan (*customer value*). Tujuan *Lean* adalah meningkatkan terus-menerus *customer value* melalui peningkatan terus-menerus rasio antara nilai tambah terhadap waste (*the value-to-waste ratio*).

APICS Dictionary (2005) mendefinisikan *Lean* sebagai suatu filosofi bisnis yang berlandaskan pada minimisasi penggunaan sumber-sumber daya (termasuk waktu) dalam berbagai aktivitas perusahaan. *Lean* berfokus pada identifikasi dan eliminasi aktivitas-aktivitas tidak bernilai tambah (*non-value-adding-activities*) dalam desain, produksi (untuk bidang manufaktur) atau operasi (untuk bidang jasa), dan *supply chain management*, yang berkaitan langsung dengan pelanggan.

Lean dapat didefinisikan sebagai suatu pendekatan sistemik dan sistematis untuk mengidentifikasi dan menghilangkan pemborosan atau aktivitas-aktivitas yang tidak bernilai tambah melalui peningkatan terus-menerus secara radikal dengan cara mengalirkan produk dan informasi menggunakan sistem tarik dari pelanggan internal dan eksternal untuk mengejar keunggulan dan kesempurnaan.

Berikut ini adalah lima prinsip dasar *Lean* yaitu :

1. Mengidentifikasi nilai produk (barang dan/atau jasa) berdasarkan perspektif pelanggan, dimana pelanggan menginginkan produk berkualitas superior, dengan harga yang kompetitif dan penyerahan tepat waktu.

2. Mengidentifikasi *value stream process mapping* (pemetaan proses pada *value stream*) untuk setiap produk.
3. Menghilangkan pemborosan yang tidak bernilai tambah dari semua aktivitas sepanjang proses *value stream* itu.
4. Mengorganisasikan agar material, informasi, dan produk itu mengalir secara lancar dan efisien sepanjang proses *value stream* menggunakan sistem tarik.
5. Terus-menerus mencari berbagai teknik dan alat peningkatan untuk mencapai keunggulan dan peningkatan terus menerus.

2.3.1 Pengertian Pemborosan (*Waste*)

Waste dapat didefinisikan sebagai segala aktivitas kerja yang tidak memberikan nilai tambah dalam proses transformasi input menjadi output sepanjang *value stream*. Berdasarkan perspektif *Lean*, semua jenis pemborosan yang terdapat sepanjang proses *value stream*, yang mentransformasikan input menjadi output, harus dihilangkan guna meningkatkan nilai produk dan selanjutnya meningkatkan *customer value*. Pada dasarnya dikenal dua kategori utama pemborosan yaitu *Type One Waste* dan *Type Two Waste*.

Type One Waste adalah aktivitas kerja yang tidak menciptakan nilai tambah dalam proses transformasi input menjadi output sepanjang *value stream*, namun aktivitas itu pada saat sekarang tidak dapat dihindarkan karena berbagai alasan. *Type One Waste* ini sering disebut sebagai *Incidental Activity* atau *Incidental Work* yang termasuk ke dalam aktivitas tidak bernilai tambah (*non-value-adding work or activity*).

2.3.2 Identifikasi Pemborosan

Secara umum kita mengenal “seven plus one” type of waste, yaitu type waste menurut *root causes* atau cara mengidentifikasi bentuk-bentuk pemborosan yang dapat dilihat dari akar penyebabnya, seperti ditunjukkan dalam Tabel 2.4.

Tabel 2.4 “Seven Plus One” Type of Waste

Type	Pemborosan (<i>waste</i>)	Akar penyebab (<i>Root Causes</i>)
1.	Overproduction : memproduksi lebih daripada kebutuhan pelanggan internal dan eksternal, atau memproduksi lebih cepat atau lebih awal daripada waktu kebutuhan pelanggan internal dan eksternal.	Ketiadaan komunikasi, sistem balas jasa dan penghargaan yang tidak tepat, hanya berfokus pada kesibukan kerja, bukan untuk memenuhi kebutuhan pelanggan internal dan eksternal.
2.	Delays (waiting time) : keterlambatan yang tampak melalui orang-orang yang sedang menunggu mesin, peralatan, bahan baku, <i>supplies</i> , perawatan/pemeliharaan (<i>maintenance</i>), dll; atau mesin-mesin yang sedang menunggu perawatan, orang-orang, bahan baku, peralatan, dll.	Inkonsistensi metode kerja, waktu penggantian produk yang panjang (<i>long changeover times</i>), dll.
3.	Transportations : memindahkan material atau orang dalam jarak yang sangat jauh dari satu proses ke proses berikut yang dapat mengakibatkan waktu penanganan material bertambah.	Tata letak yang jelek, ketiadaan koordinasi dalam proses, <i>poor housekeeping</i> , organisasi tempat kerja yang jelek, lokasi penyimpanan material yang banyak dan saling berjauhan.
4.	Processes : mencakup proses-proses tambahan atau aktivitas kerja yang tidak perlu atau tidak efisien.	Ketidaktepatan penggunaan peralatan, pemeliharaan peralatan yang jelek, gagal mengombinasi operasi-operasi kerja, proses kerja dibuat serial padahal proses-proses itu tidak saling tergantung satu sama lain, yang seharusnya dapat dibuat paralel.

Lanjutan Tabel 2.4

5.	Inventories : pada dasarnya <i>inventories</i> menyembunyikan masalah dan menimbulkan aktivitas penanganan tambahan yang seharusnya tidak diperlukan. <i>Inventories</i> juga mengakibatkan <i>extra paperwork</i> , <i>extra space</i> , dan <i>extra cost</i> .	Peralatan yang tidak andal, aliran kerja yang tidak seimbang, pemasok yang tidak kapabel, peramalan kebutuhan yang tidak akurat, ukuran <i>batch</i> yang besar, <i>long changeover times</i> .
6.	Motions : setiap pergerakan dari orang atau mesin yang tidak menambah nilai kepada barang dan jasa yang akan diserahkan kepada pelanggan, tetapi hanya menambah biaya dan waktu saja.	Organisasi tempat kerja yang jelek, tata letak yang jelek, metode kerja yang tidak konsisten, <i>poor machine design</i> .
7.	Defective Products : <i>scrap</i> , <i>rework</i> , <i>customer returns</i> , <i>customer dissatisfaction</i> .	<i>Incapable processes</i> , <i>insufficient training</i> , ketiadaan prosedur-prosedur operasi standar.
7+1	Defective Design : desain yang tidak memenuhi kebutuhan pelanggan, penambahan fitur yang tidak perlu.	<i>Lack of customer input in design</i> , <i>over-design</i> .

Dalam prespektif lain jenis pemborosan dikelompokkan dalam empat kategori utama, yaitu orang, kuantitas, kualitas, dan informasi seperti ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Pendekatan Untuk Reduksi Pemborosan.

Kategori pemborosan	Jenis pemborosan	Pendekatan reduksi pemborosan	Contoh metode lean sigma	Focus peningkatan
Orang (<i>people</i>)	<i>Processing</i> , <i>motion</i> , <i>waiting</i>	Manajemen tempat kerja	Penetapan standar-standar kerja, pengorganisasian tempat kerja, kaizen, 5S	Tata letak, pemasangan label, <i>tools/parts arrangement</i> , <i>work instructions</i> , efisiensi, <i>take time</i> , <i>skills training</i> , <i>shift meetings</i> , <i>cell/area teams</i> , <i>visual displays</i>

Lanjutan Tabel 2.5

Kuantitas (Quantity)	<i>Inventory, moving things, making too much</i>	<i>Just-in-time (JIT)</i>	<i>Leveling, kanban, Quick setup, preventive maintenance</i>	<i>Work balance, work-in-process (WIP) location/amount, kanban location, kanban types, lot sizes, changeover analysis, preventive maintenance analysis</i>
Kualitas (Quality)	<i>Fixing Defect</i>	<i>Error (mistake), proofing, autonomation</i>	<i>Detection, warning, prediction, prevention, jikoda</i>	<i>Fixture modifications, successive checks, limit switches, check sheets, appropriate automated assistance, templates.</i>
Informasi (Information)	<i>Planning scheduling, execution</i>	Teknologi informasi berfokus proses	<i>Plan, schedule, track, anticipate, optimize</i>	<i>Queue analysis, dynamic scheduling of order/job status by process element, timing/completion.</i>

2.4 Pengertian *Lean Sigma*

Metode *Lean Sigma* adalah salah satu aplikasi ilmu teknik untuk meningkatkan laju perusahaan, di mana kombinasinya dengan *Six Sigma* ditujukan untuk meningkatkan efisiensi dan difokuskan pada persoalan pelanggan selain itu dapat meminimalisasi waktu menunggu proses. Jika *Six Sigma* terfokus pada mengurangi variasi dalam suatu proses, sehingga proses/produk semaksimal mungkin berada dalam batas kontrol, maka *lean process* lebih menitikberatkan pada kecepatan proses.

Tool yang digunakan dalam *Lean Production System* adalah *Value Stream*.



Metrik yang digunakan dalam metode *Lean Production System* adalah sebagai berikut :

a. Efisiensi dari siklus proses (*Process Cycles Efficiency*)

Efisiensi dari siklus proses adalah suatu metrik atau ukuran untuk melihat sejauh mana efisiensi waktu dari proses terhadap waktu siklus proses secara keseluruhan.

$$\text{Efisiensi dari siklus proses} = \frac{\text{Value - Added Time}}{\text{Total Lead Time}}$$

b. Kecepatan proses (*Velocity Process*)

Kecepatan proses adalah seberapa tahapan yang ada di dalam proses dapat dilakukan dalam setiap satuan waktu.

$$\text{Process lead Time} = \frac{\text{Jumlah produk di dalam proses}}{\text{Penyelesaian dalam satuan waktu}}$$

$$\text{Kecepatan Proses} = \frac{\text{Jumlah aktivitas yang terdapat didalam proses}}{\text{process lead time}}$$

Lean sigma juga dapat didefinisikan sebagai suatu pendekatan sistemik dan sistematis untuk mengidentifikasi dan menghilangkan pemborosan atau aktivitas-aktivitas yang tidak bernilai tambah (*non-value-adding activities*) melalui peningkatan terus-menerus secara radikal untuk mencapai tingkat kinerja *six sigma*, dengan cara mengalirkan produk dan informasi menggunakan sistem tarik dari pelanggan internal dan eksternal untuk mengejar keunggulan dan kesempurnaan berupa hanya memproduksi 3,4 cacat untuk setiap satu juta kesempatan atau operasi.

Integrasi *Lean* dan *Six Sigma* akan meningkatkan kinerja bisnis dan industri melalui peningkatan kecepatan (*shorter cycle time*) dan akurasi (*zero defects*). Pendekatan *lean* akan menyingkap *Non-Value-Added* (NVA) dan *Value Added* (VA)

serta membuat *value added* mengalir secara lancar sepanjang *value stream processes*, sedangkan *Six Sigma* akan mereduksi variasi *value added* itu.

2.4.1 Tindakan Untuk Melaksanakan Peningkatan Kualitas *Lean Sigma*

Proses peningkatan kualitas (proses perbaikan kualitas) merupakan komitmen untuk perbaikan yang melibatkan secara seimbang antara aspek manusia (motivasi) dan aspek teknologi (teknik). *Kaizen* adalah suatu istilah dalam bahasa Jepang yang dapat diartikan sebagai perbaikan secara terus-menerus (*continous improvement*). *Kaizen* pada dasarnya merupakan suatu kesatuan pandangan yang komprehensif dan terintegrasi yang bertujuan untuk melaksanakan perbaikan secara terus-menerus.

Dalam melaksanakan *kaizen* kita dapat menggunakan panduan bertanya 5W+2H sebagai berikut :

1. *What?* (apa)?

Apa masalah yang menyebabkan permasalahan kualitas? Untuk mengidentifikasi permasalahan kualitas yang terjadi di tiap proses.

2. *Where?* (dimana)?

Dimana terjadinya permasalahan kualitas tersebut? Untuk mengetahui di bagian proses manakah terjadi masalah kualitas agar lebih mudah dilakukan tindakan perbaikan.

3. *Who?* (siapa)?

Siapa penanggung jawab pada proses tersebut? Untuk mengetahui penanggung jawab pada proses tersebut dan kemudian diberikan pendidikan untuk melakukan perbaikan kualitas.

4. *When?* (kapan)?

Kapan permasalahan kualitas tersebut terjadi? Untuk mengidentifikasi kapan permasalahan kualitas tersebut terjadi sehingga dapat dilakukan tindakan antisipasi.

5. *Why?* (mengapa)?

Kenapa terjadi permasalahan kualitas pada tingkat proses? Untuk mengidentifikasi akar penyebab permasalahan kualitas di tingkat proses.

6. *How?* (bagaimana)?

Bagaimana peningkatan kualitas tersebut akan dilaksanakan? Dalam hal ini akan diberikan usulan-usulan perbaikan kualitas pada perusahaan untuk mengatasi permasalahan kualitas tersebut.

7. *How Much?* (berapa biayanya)?

Dalam hal ini akan dilakukan pengukuran biaya produksi dan biaya kualitasnya sehingga diketahui akibat dari kerugian yang ditimbulkan dari permasalahan kualitas tersebut.

2.5 Pemodelan Sistem

2.5.1 Definisi Sistem

Sistem merupakan kesatuan dari elemen-elemen yang terhubung melalui sebuah mekanisme tertentu dan terikat dalam hubungan interdependensi yang mempunyai tujuan bersama. Sistem juga memiliki hubungan yang bersifat umpan balik yang menyebabkan sistem senantiasa bersifat dinamis. Sedangkan lingkungan sistem adalah segala sesuatu yang tidak merupakan bagian dari sistem, tetapi keberadaannya dapat mempengaruhi dan atau dipengaruhi sistem.

2.5.2 Definisi Model

Model merupakan suatu representasi atau formalisasi dalam bahasa tertentu dari suatu sistem nyata yang telah disepakati. Sehingga model dapat dikatakan sebagai sebuah kesatuan yang menggambarkan karakteristik suatu sistem. Model dibuat dengan cara simplifikasi dari sistem yang ada sehingga untuk mempelajari sebuah sistem dapat dilakukan dengan pengamatan pada model sistem tersebut. Walaupun model merupakan bentuk sederhana dari sebuah sistem, tapi dalam pembentukannya harus tetap memperhatikan kompetensi dari karakteristik sistem yang diamati.

2.5.3 Karakteristik Model

Karakteristik model yang baik sebagai ukuran tujuan pemodelan yaitu :

1. Tingkat generalisasi yang tinggi. Semakin tinggi tingkat generalisasi model, maka model tersebut akan dapat memecahkan masalah yang semakin besar.
2. Mekanisme transparansi sehingga model dapat menjelaskan dinamika sistem secara rinci.
3. Potensial untuk dikembangkan sehingga membangkitkan minat peneliti lain untuk menyelidikinya lebih lanjut.
4. Peka terhadap perubahan asumsi. Hal ini menunjukkan bahwa proses pemodelan tidak pernah selesai (peka terhadap perubahan lingkungan).

2.5.4 Prinsip-Prinsip Pemodelan Sistem

1. Elaborasi. Artinya pengembangan model dilakukan secara bertahap dimulai dari model sederhana hingga diperoleh model yang lebih representatif.
2. Sinektik. Artinya pengembangan model yang dilakukan secara analogis (kesamaan-kesamaan).

3. Iteratif. Artinya pengembangan model yang dilakukan secara berulang-ulang dan peninjauan kembali.

2.6 Pengertian Simulasi Sistem

Simulasi adalah suatu solusi analitis dari sebuah sistem yang digunakan untuk memecahkan berbagai masalah yang digunakan untuk memecahkan atau menguraikan persoalan-persoalan dalam kehidupan nyata yang penuh dengan ketidakpastian ketika solusi matematis tidak memadai, dengan menggunakan model atau metode tertentu untuk melihat sejauh mana input mempengaruhi pengukuran output atas performansi sistem dan lebih ditekankan pada pemakaian komputer untuk mendapatkan solusinya. Menurut Banks (1999), simulasi adalah imitasi dari proses operasi di dunia nyata. Simulasi juga di definisikan sebagai sebuah imitasi sederhana dari sistem operasi di dunia nyata yang bertujuan untuk mengevaluasi sistem tersebut (Goldsmann, 2007).

Simulasi mempunyai keunggulan menangkap perubahan dinamis dari proses yang terjadi sehingga dapat mewakili kondisi sebenarnya dari sebuah sistem serta kemampuan menjalankan *what-if scenario* untuk dapat mengetahui pengaruhnya terhadap sistem. Simulasi juga berusaha untuk merepresentasikan sistem amatan dengan presisi yang lebih mudah untuk diamati dibandingkan dengan jenis model yang lain. Dengan simulasi maka dimungkinkan untuk dapat mengamati bagaimana sistem yang direpresentasikan dapat berperilaku, sehingga model simulasi yang baik adalah model yang mampu menyelesaikan karakteristik dan perubahan sistem dari waktu ke waktu. Semakin mampu model simulasi menirukan proses dari sistem, maka semakin baik model tersebut.

2.6.1 Bagian-Bagian Model Simulasi

Beberapa bagian model simulasi yang berupa istilah-istilah asing perlu dipahami oleh pemodel karena bagian-bagian ini sangat penting dalam menyusun suatu model simulasi :

a. Entitas (*entity*)

Kebanyakan simulasi melibatkan ‘pemain’ yang disebut entitas yang bergerak, merubah status, mempengaruhi dan dipengaruhi oleh entitas yang lain serta mempengaruhi hasil pengukuran kinerja sistem. Entitas merupakan obyek yang dinamis dalam simulasi.

b. Atribut (*attribute*)

Setiap entitas memiliki ciri-ciri tertentu yang membedakan antara satu dengan yang lainnya. Karakteristik yang dimiliki oleh setiap entitas disebut dengan atribut. Satu hal yang perlu diingat bahwa nilai atribut mengikat entitas tertentu. Sebuah *part* (entitas) memiliki atribut (*arrival time, due date, priority, dan color*) yang berbeda dengan part yang lain.

c. Variabel (*variable*)

Variabel merupakan potongan informasi yang mencerminkan karakteristik suatu sistem. Variabel berbeda dengan atribut karena dia tidak mengikat suatu entitas melainkan sistem secara keseluruhan sehingga semua entitas dapat mengandung variabel yang sama. Misalnya, panjang antrian, *batch size*, dan sebagainya.

d. Sumber daya (*resource*)

Entitas-entitas seringkali saling bersaing untuk mendapat pelayanan dari resource yang ditunjukkan oleh operator, peralatan, atau ruangan penyimpanan yang terbatas. Suatu *resource* dapat berupa grup atau pelayanan individu.

e. Antrian (*queue*)

Ketika entitas tidak bergerak (diam) hal ini dimungkinkan karena *resource* menahan (*seize*) suatu entitas sehingga membuat entitas yang lain untuk menunggu. Jika *resource* telah kosong (melepas satu entitas) maka entitas yang lain bergerak kembali dan seterusnya demikian.

f. Kejadian (*event*)

Kejadian adalah sesuatu yang terjadi pada waktu tertentu yang kemungkinan menyebabkan perubahan terhadap atribut atau variabel. Ada tiga kejadian umum dalam simulasi, yaitu *arrival* (kedatangan), *departure* (entitas meninggalkan sistem) dan *the end* (simulasi berhenti).

g. *Simulation Clock*

Nilai sekarang dari waktu dalam simulasi yang dipengaruhi oleh variabel disebut sebagai *Simulation Clock*. Ketika simulasi berjalan dan pada kejadian tertentu waktu dihentikan untuk melihat nilai saat itu maka nilai tersebut adalah nilai simulasi pada saat tersebut.

h. Replikasi

Replikasi mempunyai pengertian bahwa setiap menjalankan dan menghentikan simulasi dengan cara yang sama dan menggunakan set parameter input yang sama pula (*'identical' part*), tapi menggunakan masukan bilangan random yang terpisah (*'independent' part*) untuk membangkitkan waktu antar kedatangan dan pelayanan (hasil-hasil simulasi). Sedangkan panjang waktu simulasi yang diinginkan untuk setiap replikasi disebut *length of replication*.



2.6.2 Langkah-langkah Simulasi

Dalam melakukan suatu penelitian atau kajian tidak dapat dilakukan secara sembarangan. Artinya diperlukan suatu langkah-langkah atau metodologi yang terstruktur dan terkendali sehingga kesimpulan yang dihasilkan dapat dipertanggungjawabkan. Begitu pula dalam melakukan studi simulasi terdapat metodologi umum yang digunakan yaitu sebagai berikut :

1. Formulasi masalah

Setiap studi selalu dimulai dengan suatu pernyataan yang jelas tentang tujuan yang hendak dicapai. Secara keseluruhan harus direncanakan pula variabel-variabel yang terdapat dalam sistem obyek.

2. Pengumpulan data

Informasi dan data sebaiknya dikumpulkan secara terpusat dan digunakan untuk melakukan spesifikasi prosedur operasi dan distribusi probabilitas untuk variabel random yang terdapat dalam model. Data yang dikumpulkan meliputi :

- a. Data waktu proses.
- b. Data penjadwalan bahan baku dan penjadwalan tenaga kerja.
- c. Data lain yang berhubungan dengan sistem nyata.

3. Pembuatan program komputer dan verifikasi

Pemodel harus menentukan program apakah yang akan digunakan untuk menguji dan menjalankan model. Selama melakukan translasi model ke dalam program yang dipilih, dilakukan verifikasi model terhadap sistem nyata apakah bentuk fisik model sudah seperti sistem nyatanya.

4. Jalankan program

Dengan bantuan *software* simulasi model yang telah dibuat dijalankan (*run*) untuk melihat hasilnya.

5. Validasi

Program yang dijalankan dapat digunakan untuk menguji sensitivitas hasil dari model terhadap perubahan kecil pada parameter masukan. Jika hasilnya berubah secara ekstrim maka suatu estimasi yang baik harus diambil. Jika sistem nampak sama dengan yang ada saat ini, data hasil dari program simulasi dapat dibandingkan dengan sistem nyatanya. Jika hasilnya baik maka program simulasi dinyatakan *valid* dan model dianggap representasi dari sistem nyata.

6. Mendesain model eksperimen

Jika program simulasi sudah dinyatakan *valid* maka pemodel dapat melakukan berbagai eksperimen terhadap program/model tersebut sesuai dengan penelitiannya.

7. Menjalankan model eksperimen

Model skenario yang telah dibuat dijalankan untuk dilihat performa sistem yang dihasilkan.

8. Analisa data output

Teknik-teknik statistik digunakan untuk melakukan analisa data yang dihasilkan. Dengan mengukur selang kepercayaan dan performansi yang berbeda-beda untuk setiap desain, maka dapat diketahui mana model simulasi terbaik sesuai tujuan yang hendak dicapai.

9. Implementasi

2.6.3 Keuntungan Simulasi

1. Fleksibel.
2. Menghemat waktu (*compress time*). Kemampuan dari menghemat waktu ini dapat dilihat dari pekerjaan yang bila dikerjakan akan memakan waktu tahunan tetapi

kemudian dapat disimulasikan hanya dalam beberapa menit, bahkan dalam beberapa kasus hanya dalam hitungan detik.

3. Dapat melebar-luaskan waktu (*expand time*). Hal ini terlihat terutama dalam dunia statistik dimana hasilnya diinginkan tersaji dengan cepat. Simulasi dapat digunakan untuk menunjukkan perubahan struktur dari suatu sistem nyata (*Real System*) yang sebenarnya tidak dapat diteliti pada waktu yang seharusnya (*Real Time*). Dengan demikian simulasi dapat membantu memprediksi *response* dari *Real System* hanya dengan mengubah data parameter sistem.
4. Dapat mengawasi sumber-sumber yang bervariasi (*control sources of variation*). Kemampuan pengawasan dalam simulasi ini tampak terutama apabila analisis statistik digunakan untuk meninjau hubungan antara variabel bebas (*independent*) dengan variabel terkait (*dependent*) yang merupakan faktor-faktor yang akan dibentuk dalam percobaan.
5. Mengkoreksi kesalahan-kesalahan penghitungan (*error in measurement correction*). Dalam prakteknya, pada suatu kegiatan ataupun percobaan dapat saja muncul ketidak-benaran dalam mencatat hasil-hasilnya. Sebaliknya dalam simulasi komputer jarang ditemukan kesalahan perhitungan terutama bila angka-angka diambil dari komputer secara teratur dan bebas. Komputer mempunyai kemampuan untuk melakukan penghitungan dengan akurat.
6. Dapat dihentikan dan dijalankan kembali (*stop simulation and restart*). Simulasi komputer dapat dihentikan untuk kepentingan peninjauan ataupun pencatatan semua keadaan yang relevan tanpa berakibat buruk terhadap program simulasi tersebut. Dalam dunia nyata, percobaan tidak dapat dihentikan begitu saja. Dalam simulasi komputer, setelah dilakukan penghentian maka kemudian dapat dengan cepat dijalankan kembali (*restart*).

7. Mudah diperbanyak (*easy to replicate*). Dengan simulasi komputer percobaan dapat dilakukan setiap saat dan dapat diulang-ulang. Pengulangan dilakukan terutama untuk mengubah berbagai komponen dan variabelnya, seperti dengan perubahan pada parameternya, perubahan pada kondisi operasinya, ataupun dengan memperbanyak output.
8. Tidak bertentangan dengan sistem nyata.
9. Dapat solusi analitis yang menjawab pertanyaan *what-if*.

2.6.4 Kerugian Simulasi

- a. Memerlukan masukan managerial yang baik.
- b. Tidak menghasilkan langsung solusi yang optimal.
- c. Tidak *immune* terhadap GIGO (*Garbage In Garbage Out*). Artinya apabila kita memasukkan data yang salah, maka kita akan mendapatkan output simulasi yang salah juga. Sehingga dapat disimpulkan bahwa hasil simulasi tergantung dari input yang kita masukkan.

2.6.5 Distribusi Data

Suatu cara yang cepat untuk memeriksa apakah sekumpulan data tertentu sesuai dengan distribusi teoritis tertentu adalah membandingkan secara grafis empiris kumulatif dengan fungsi kepadatan kumulatif yang bersesuaian dari distribusi teoritis yang bersangkutan. Jika kedua fungsi tersebut tidak memperlihatkan deviasi yang berlebihan, kemungkinan besar distribusi teoritis itu sesuai dengan sekumpulan data tersebut.

Gagasan untuk membandingkan distribusi empiris dan distribusi teoritis di atas merupakan dasar untuk uji *goodness-of-fit Kolmogorov Smirnov*. Uji ini dapat

diterapkan untuk variabel random *kontinyu*, jika tes ini digunakan untuk data yang *diskrit test*, maka uji ini masih dapat dipakai hanya saja hasilnya lebih konservatif, kemungkinan kesalahan tipe II akan sedikit lebih besar.

Uji lain yang berlaku baik untuk variabel random *diskrit* maupun *kontinyu* adalah uji *chi-square*. Berbeda dengan uji *Kolmogorof-Smirnov* yang didasari oleh perbandingan fungsi kepadatan probabilitas. Jika suatu ruang sampel mengandung titik yang terhingga banyaknya atau suatu deretan anggota banyaknya sama dengan banyaknya bilangan bulat, maka ruang sampel tersebut disebut ruang sampel *diskrit*, dan variabel random yang didefinisikan pada ruang sampel tersebut adalah variabel random *diskrit*. Distribusi dari variabel random *diskrit* adalah sebuah grafik, tabel atau rumus yang menyatakan suatu probabilitas yang berhubungan dengan tiap nilai yang mungkin dari variabel random *diskrit*.

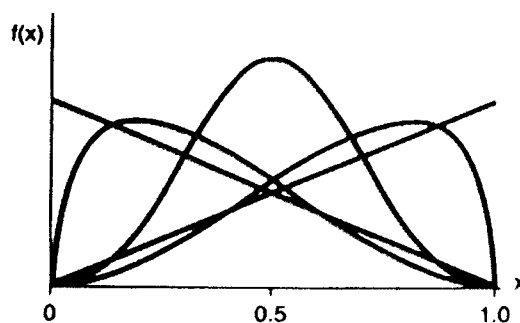
Fungsi $f(x)$ adalah suatu fungsi probabilitas atau distribusi probabilitas suatu variabel random *diskrit* X bila, untuk setiap hasil x yang mungkin memenuhi :

$$\sum f(x) = 1$$

$$F(x) \geq 0$$

Probabilitas $f(x)$ dinyatakan oleh : $F(x) = P(X=x) = 1$

1. Distribusi *Beta*



Gambar 2.7 Distribusi *Beta*

$$f(x) = \frac{x^{\beta-1}(1-x)^{\alpha-1}}{B(\beta, \alpha)} \quad \text{for } 0 < x < 1$$

Dimana :

B adalah keseluruhan fungsi *beta* diberikan oleh :

$$B(\beta, \alpha) = \int_0^1 t^{\beta-1}(1-t)^{\alpha-1} dt$$

Parameter = Bentuk parameter *beta* (β) dan *alpha* (α) spesifikasinya dengan bilangan bulat positif

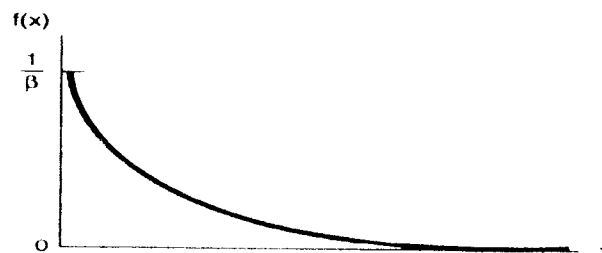
Range = (0,1) dapat juga ditransformasikan sampai [a,b]

$$\text{Rata-rata} = \frac{\beta}{\beta + \alpha}$$

$$\text{Varians} = \frac{\beta\alpha}{(\beta + \alpha)^2((\beta + \alpha) + 1)}$$

Aplikasi = Oleh karena kemampuannya untuk menerima suatu bentuk yang luas, distribusi ini sering digunakan untuk membuat suatu konsep dasar model ketidakhadiran dari data. Karena cakupan dari distribusi *beta* adalah dari 0 sampai 1, contoh x dapat diubah ke y yang berskala *beta* dengan cakupan a sampai b dengan menggunakan persamaan $Y = a + (b-a)X$. *Beta* sering digunakan untuk menghadirkan bilangan random, seperti proporsi dari materi cacat *brown bundle*.

2. Distribusi Eksponensial



Gambar 2.8 Distribusi Eksponensial

$$f(x) = \frac{1}{\beta} e^{-x/\beta} \quad \text{for } x > 0$$

Parameter = rata-rata (β) spesifikasinya dengan bilangan bulat positif

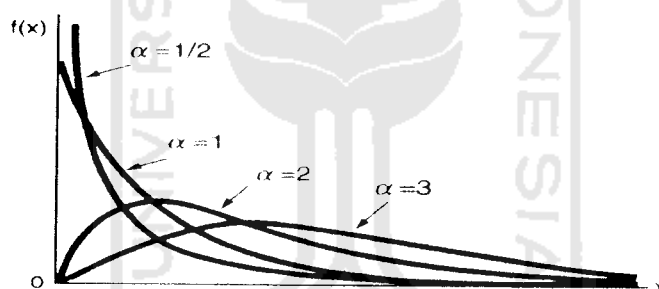
Range = $(0, +\infty)$

Rata-rata = β

Varians = β^2

Aplikasi = Distribusi ini sering digunakan untuk model waktu *interevent* dalam proses gangguan dan kedatangan yang random, tetapi secara umum hal tersebut tidak sesuai untuk memperagakan waktu proses penundaan.

3. Distribusi Gamma



Gambar 2.9 Distribusi Gamma

$$f(x) = \frac{\beta^{-\alpha} x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha)} \quad \text{for } x > 0$$

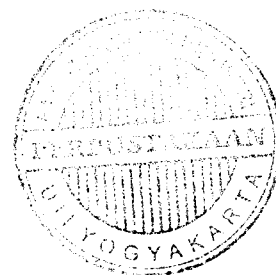
Dimana:

Γ adalah keseluruhan fungsi *gamma* diberikan oleh:

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt$$

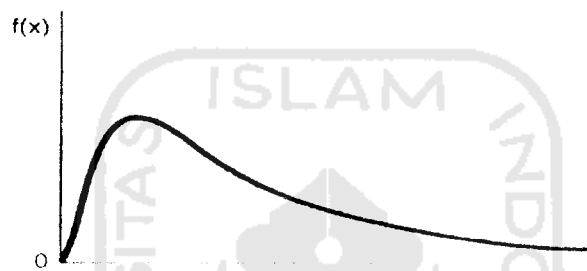
Parameter = Nilai (α) dan ukuran skala (β) spesifikasinya dengan bilangan bulat positif

Range = $(0, +\infty)$



- Rata-rata = $\alpha\beta$
- Varians = $\alpha\beta^2$
- Aplikasi = Untuk bilangan bulat bentuk parameter, *gamma* adalah sama dengan distribusi *erlang*. *Gamma* sering digunakan untuk menghadirkan waktu yang diperlukan untuk menyelesaikan tugas/kasus (contoh: waktu kerja mesin atau waktu perbaikan mesin).

4. Distribusi Log Normal



Gambar 2.10 Distribusi Log Normal

$$f(x) = \frac{1}{\sigma x \sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} \quad \text{for } x > 0$$

- Parameter = skala parameter (μ) spesifikasinya bilangan bulat dan parameter (σ) spesifikasinya dengan bilangan bulat positif

Range = $(0, +\infty)$

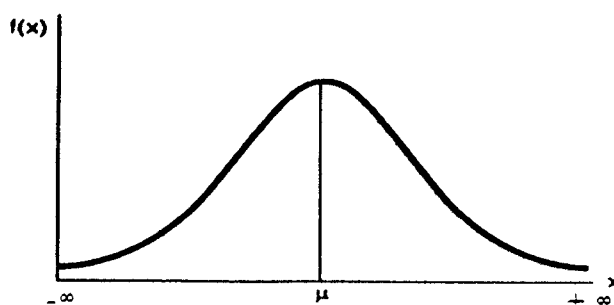
Rata-rata = $e^{\mu+\sigma^2/2}$

Varians = $e^{\mu+\sigma^2/2}(e^{\sigma^2}-1)$

- Aplikasi = Distribusi Log Normal digunakan dalam situasi dimana bilangan suatu produk dalam jumlah besar pada kuantitas random. Distribusi ini juga sering digunakan untuk menghadirkan waktu tugas yang mempunyai distribusi *skewed* di sebelah kanan. Distribusi ini dihubungkan dengan distribusi normal; Jika X mempunyai distribusi Log Normal (μ, σ), maka $\ln X$ mempunyai distribusi Normal (μ, σ).

Dengan catatan bahwa μ dan σ bukanlah rata-rata dan standart deviasi dari Ln X.

5. Distribusi Normal



Gambar 2.11 Distribusi Normal

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

untuk semua nilai x

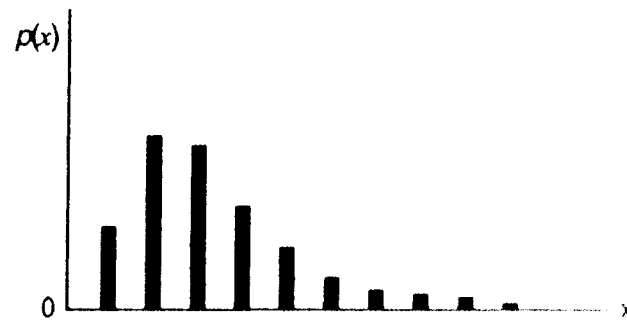
Parameter = rata-rata (μ) spesifikasinya bilangan bulat dan parameter standar deviasi (σ) spesifikasinya dengan bilangan bulat positif

Range = $(-\infty, +\infty)$

Rata-rata = μ

Varians = σ^2

Aplikasi = Distribusi Normal digunakan dalam situasi dimana menggunakan teorima pusat limit, kuantitas yang menjumlahkan dengan jumlah kuantitas yang lain. Distribusi ini juga digunakan sebagai pengalaman untuk beberapa proses yang muncul agar mempunyai distribusi simetrik. Sebab cakupan teoritisnya adalah dari $-\infty$ sampai $+\infty$, distrisbusi ini hanya digunakan untuk bilangan yang positif seperti waktu proses rata-rata yaitu sedikitnya 3 atau 4 standar deviasinya di atas 0.

6. Distribusi *Poisson*Gambar 2.12 Distribusi *Poisson*

$$P(x) = \frac{\lambda^x \cdot e^{-\lambda}}{x!} \text{ untuk } x \in (0, 1, \dots)$$

Parameter = rata-rata (λ) spesifikasinya dengan bilangan bulat positif

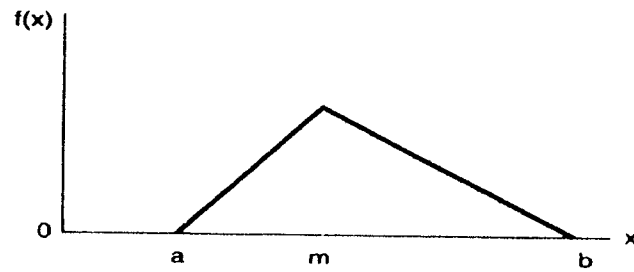
Range = (0, 1, ...)

Rata-rata = λ

Varians = λ

Aplikasi = Distribusi *Poisson* adalah suatu distribusi diskrit yang sering digunakan untuk beberapa model pada bilangan random yang terjadi di interval waktu yang tetap, perbaikan waktu interval. Jika waktu antara peristiwa yang berurutan adalah bersifat distribusi eksponensial maka banyaknya peristiwa yang terjadi pada waktu interval yang tetap mempunyai suatu distribusi *Poisson*. Distribusi *Poisson* juga digunakan untuk ukuran *batch* model yang random.

7. Distribusi *Triangular*



Gambar 2.13 Distribusi *Triangular*

$$f(x) = \frac{2(x-a)}{(m-a)(b-a)} \text{ for } a \leq x \leq m$$

$$f(x) = \frac{2(b-x)}{(b-m)(b-a)} \text{ for } m \leq x \leq b$$

Parameter = minimum (a) mode (m) dan maximum (b) nilai untuk distribusi spesifikasinya dengan bilangan bulat positif untuk $a < m < b$

Range = (a, b)

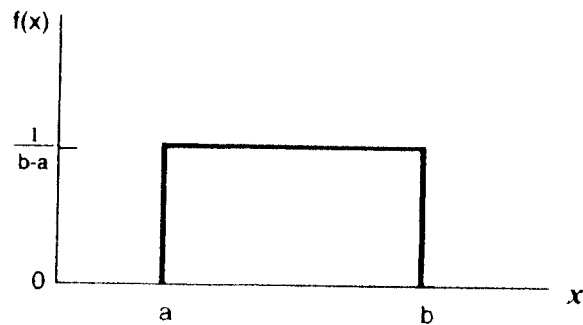
Rata-rata = $(a+m+b)/3$

Varians = $(a^2+m^2-ma-ab-mb)/18$

Aplikasi = Distribusi *triangular* biasa digunakan dalam situasi dimana formula yang tepat dari distribusi tidak diketahui, tetapi menaksir (terkaan) untuk nilai-nilai minimum, maksimum, dan nilai-nilai yang bisa dipastikan sudah tersedia. Distribusi *triangular* adalah lebih mudah untuk digunakan dan dijelaskan dibanding dengan distribusi yang lain, yang mungkin digunakan dalam situasi ini (distribusi *beta*).



8. Distribusi Seragam



Gambar 2.14 Distribusi Seragam

$$f(x) = \frac{1}{b-a} \text{ for } a \leq x \leq b$$

Parameter = minimum (a) mode (m) dan maximum (b) nilai untuk distribusi spesifikasinya dengan bilangan bulat positif untuk $a < b$

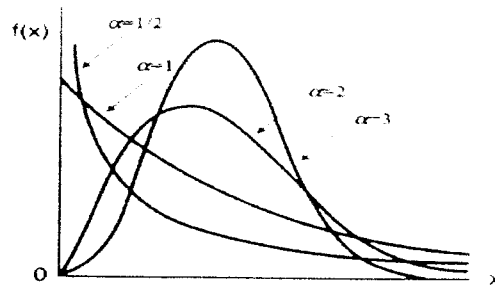
Range = (a, b)

Rata-rata = $(a+b)/2$

Varians = $(b-a)^2/12$

Aplikasi = Distribusi seragam digunakan ketika semua nilai-nilai di atas suatu cakupan yang terbatas dianggap sama. Distribusi ini kadang-kadang digunakan ketika tidak ada informasi selain dari cakupan yang tersedia. Distribusi seragam mempunyai suatu perbedaan yang lebih besar dibanding distribusi lain yang digunakan ketika kekurangan informasi (seperti distribusi *triangular*).

9. Distribusi *Weibull*



Gambar 2.15 Distribusi *Weibull*

$$f(x) = \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} \quad \text{for } x > 0$$

Parameter = parameter dari (α) dan skala parameter (β) spesifikasinya dengan bilangan bulat positif.

Range = $(0, +\infty)$

Rata-rata = $\frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right)$ dimana Γ adalah keseluruhan fungsi gamma (lihat distribusi gamma)

Varians = $\frac{\beta^2}{\alpha} \left\{ 2\Gamma\left(\frac{2}{\alpha}\right) - \frac{1}{\alpha} \left[\Gamma\left(\frac{1}{\alpha}\right) \right]^2 \right\}$

Aplikasi = Distribusi *Weibull* secara luas digunakan dalam menangani masalah seperti keandalan dan uji umur. Jika suatu sistem terdiri dari banyaknya bagian yang cacat dan jika sistem gagal dimana suatu bagian cacat, maka waktu antara kegagalan yang berurutan dapat didekati oleh distribusi *Weibull*. Distribusi ini juga digunakan untuk menghadirkan waktu tugas non-negatif yang *skewed* disebelah kiri.

2.6.6 Validasi Model

Validasi merupakan langkah untuk meyakinkan bahwa model berkelakuan/bersifat seperti sistem nyatanya. Dan suatu pendekatan paling nyata dalam suatu validasi adalah membandingkan model dengan output dari sistem nyatanya.

Dua tujuan umum dalam validasi :

1. Menghasilkan suatu model yang representatif terhadap perilaku sistem nyatanya sedekat mungkin untuk dapat digunakan sebagai substitusi dari sistem nyata dalam melakukan eksperimen tanpa mengganggu jalannya sistem.
2. Meningkatkan kredibilitas model, sehingga model dapat digunakan oleh para manajer dan para pengambil keputusan lainnya.

Tipe validasi model :

1. Validasi asumsi

Model asumsi ini dibagi kedalam dua kelas, yaitu asumsi struktural dan asumsi data.

- a. Asumsi struktural meliputi pertanyaan-pertanyaan bagaimana sistem beroperasi dan asumsi ini juga melibatkan penyederhanaan dan penggambaran kenyataan dari sistem. Sebagaimana penulis memisahkan asumsi ini kedalam validasi proses.
- b. Asumsi data harus didasarkan pada pengumpulan data yang reliabel/data terpercaya dan analisa statistik yang tepat dari suatu data.

2. Validasi Output

Cara yang paling mudah untuk melakukan validasi ini adalah dengan pendekatan visual. Beberapa orang ahli mengamati dan membandingkan antara output model terhadap sistem riil. Metode lain yang sering digunakan adalah dengan pendekatan

statistik.

2.6.6.1 Teknik Validasi

Untuk melakukan validasi model apakah sesuai dengan sistem nyatanya dapat dilakukan dengan :

1. Uji kesamaan dua variansi

Uji kesamaan dua variansi adalah pengujian apakah kedua data mempunyai variansi yang sama. Rumus yang dipakai adalah:

$$F = \frac{\text{Variansi Terbesar}}{\text{Variansi Terkecil}}$$

Dengan hipotesis uji

H_0 : Variansi output sistem sama besar dengan variansi output model

H_1 : Variansi output sistem tidak sama besar dengan variansi output model

Level of significance = α

Daerah krisis : H_0 diterima jika $F_{\text{tabel}} > F_{\text{hitung}}$

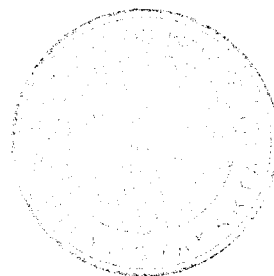
2. Uji kesamaan dua rata-rata

Uji kesamaan dua rata-rata adalah menguji apakah kedua data mempunyai rata-rata yang sama. Rumus yang dipakai untuk menguji hipotesis kesamaan dua rata-rata adalah:

$$T_{\text{hit}} = \frac{(\bar{X}_1 - \bar{X}_2)}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

dengan

$$S_p = \frac{(n_1 - 1)v_1^2 + (n_2 - 1)v_2^2}{n_1 + n_2 - 2}$$



Dimana : X_1 = rata-rata output sistem

X_2 = rata-rata output model

v_1^2 = variansi output sistem

v_2^2 = variansi output model

n_1 = jumlah output sistem

n_2 = jumlah output model

Hipotesis ujinya :

H_0 : Rata-rata output sistem sama besar dengan rata-rata output model

H_1 : Rata-rata output sistem tidak sama besar dengan rata-rata output model

Daerah kritisnya : H_0 diterima jika $-t_{\alpha/2} < t < t_{\alpha/2}$

2.6.7 Analisa Output Hasil Simulasi

Model simulasi kejadian *diskrit* memiliki karakteristik yang berbeda dari sebagian besar jenis model yang ada. Hal itu dikarenakan model simulasi kejadian diskrit terdiri dari banyak variabel random yang muncul bersamaan dalam suatu *state* yang membentuk karakteristik suatu mekanisme perubahan sistem yang diamati. Variabel random yang ada pada simulasi sistem kejadian diskrit tidak hanya pada probabilitas input yang ada, bahkan hasil output simulasinya pun merupakan variabel random, karena memiliki probabilitas dan tidak dapat diestimasi sebagai sesuatu yang pasti (definitif).

Sebuah pilihan pendekatan, untuk menentukan metode analisis yang tepat dari suatu model simulasi adalah dengan menilai tipe simulasi yang ada. Berkenaan dengan metode analisis, maka simulasi dibedakan menjadi dua jenis yaitu *terminating*

simulation dan *non-terminating simulation*. Perbedaan antara kedua jenis tipe tersebut adalah ketergantungannya pada kejelasan untuk menghentikan proses simulasi.

2.6.7.1. Terminating Simulation

Simulasi *terminating* adalah simulasi yang mempresentasikan sebuah mekanisme kejadian yang memiliki "*initial condition*", dimana simulasi ini dijalankan pada durasi waktu yang tetap (ditentukan). Kondisi inisial dapat difahami sebagai sebuah kondisi dimana keadaan sistem akan di *setup* seperti keadaan semula setiap akan melakukan simulasi. Sebagai contoh adalah sebuah sistem yang disimulasikan dimulai pada kondisi awal yang telah ditentukan, dan dihentikan setelah durasi waktu tertentu. Satu simulasi yang dapat dijadikan contoh adalah simulasi pada suatu bank dengan kondisi awal yang selalu 0 pelanggan dan memiliki durasi waktu kerja yang sama tiap harinya.

2.6.7.2. Non Terminating Simulation

Pada simulasi jenis *non terminating simulation* berbeda dengan sistem produksi sebuah perusahaan manufaktur. Misalnya diketahui sebuah perusahaan manufaktur yang memiliki kegiatan produksi untuk membuat suatu produk yang dibagi-bagi kedalam beberapa stasiun kerja yang berurutan sampai selesainya produk tersebut. Meskipun perusahaan tersebut menetapkan bahwa setiap hari memiliki waktu kerja 10 jam dan 5 hari kerja dalam seminggu, akan tetapi sistem diatas termasuk dalam sistem *non-terminating simulation*.

Pada kondisi *non terminating* penghentian simulasi tidak didasarkan pada jam kerja sebagai mana pada sistem antrian, akan tetapi karena sistem pada dasarnya

berjalan sepanjang waktu hanya dipotong oleh waktu istirahat tanpa ada inisialisasi baru.

2.7 Simulasi *Software ProModel*

ProModel merupakan suatu alat bantu simulasi dan analisis untuk sebuah tipe dan jenis sistem produksi yang berbasis *window*. ProModel memiliki kombinasi sempurna antara kemudahan dalam penggunaan, fleksibilitas yang lengkap, kemudahan memodelkan untuk setiap keadaan dan kemampuan membuat animasi yang realistis, sehingga semakin menjadi nyata.

ProModel membantu para pelaku industri untuk mencoba ide-ide baru yang mereka miliki dalam merancang dan meningkatkan sistem sebelum menentukan waktu dan sumber daya yang diperlukan untuk membangun atau mengubah sistem nyata. ProModel menfokuskan pada masalah-masalah seperti utilisasi sumber daya, kapasitas produksi, tingkat produktivitas, dan tingkat persediaan. Dengan memodelkan unsur penting dari sistem produksi, kita dapat melakukan percobaan dengan strategi dan rancangan operasi yang berbeda untuk mencapai hasil terbaik.

Elemen-elemen dalam ProModel adalah sebagai berikut :

1. *Location*

Merupakan suatu tempat dalam sistem yang tidak bergerak dimana entity akan menjalani proses, sebagai tempat penyimpanan atau tempat aktivitas lainnya.

2. *Entities*

Adalah segala sesuatu yang menjadi objek dari suatu proses. Termasuk di dalamnya orang, dokumen, bahan baku, dll. Masing-masing entitas mempunyai nama dan dapat direpresentasikan dengan satu atau lebih grafik.



3. *Arrivals*

Menyatakan kedatangan *entity* dari luar ke dalam sistem yang diamati untuk pertama kalinya.

4. *Processing*

Merupakan segala proses yang terjadi di dalam sistem dan dilakukan pada lokasi dan antar lokasi. Proses merupakan kegiatan pengolahan input yang dilakukan oleh setiap mesin sehingga menghasilkan output tertentu.

5. *Resources*

Adalah orang atau pekerja, peralatan dan alat pemindah material yang lain.

6. *Path Network*

Merupakan lintasan dimana *resource* bergerak di sepanjang lintasan yang dibuat. Arah lintasan bisa satu arah atau dua arah dan bisa dibuat berdasarkan faktor jarak atau waktu.

