

BAB II

TINJAUAN PUSTAKA

2.1. *Harbour Portal Crane (HPC)*

Harbour Portal Crane (HPC) memiliki kelebihan dapat berpindah – pindah tempat, flexible, berat kosong yang relatif ringan dengan *Safety working load (SWL)* yang besar dan memiliki kecepatan bongkar muat yang relatif cepat sehingga alat ini banyak digunakan di pelabuhan khususnya di Pelabuhan Tanjung Perak Surabaya.

Pada *Harbour Portal Crane (HPC)* yang diobservasi memiliki spesifikasi sebagai berikut:

Merk	: Italgro
Model	: IHMC 2120 E
Serial Number	: C1574
Tahun Pembuatan	: 2010
<i>Safety Working Load (SWL)</i>	: 125 Ton pada radius 51 m
<i>Working System</i>	: Semi electric Hidrolik
<i>Power System</i>	: <i>Diesel engine and medium</i> <i>Voltage source</i>



Gambar 2.1. *Harbour Portal Crane (HPC)*

2.2 Pemeliharaan pada *Harbour Portal Crane (HPC)*

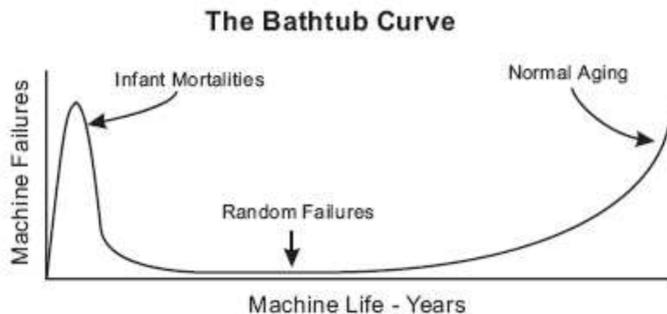
Praktik pemeliharaan *Harbour Portal Crane (HPC)* yang ada di PT Pelindo III (Persero) Cabang Tanjung Perak saat ini mengikuti buku petunjuk pemeliharaan. Rekomendasi mereka berasumsi aplikasi dan pengoperasian peralatan sesuai dengan desain dari produsen. Dalam prakteknya sendiri pemeliharaan menggunakan metode tersebut diatas kurang efektif dikarenakan alat *Harbour Portal Crane (HPC)* masih sering mengalami *down time*, sehingga perlu dilakukan pemeliharaan khusus untuk meningkatkan *availability* alat tersebut.

2.3. *Preventive Maintenance (PM)*

Preventive Maintenance (PM) adalah metode perawatan berbasis waktu dimana kegiatan pemeliharaan direncanakan dan dijadwalkan berdasarkan interval kerja yang telah ditentukan

untuk mencegah terjadinya kerusakan dan kegagalan (Cley, 2008). Dalam buku '*Applied Reliability Centered Maintenance*' (Jim Agustus, 1999) mendefinisikan PM sebagai tugas pencegahan terjadwal yang dimaksudkan untuk mengurangi probabilitas kegagalan peralatan. PM adalah pendekatan untuk mencegah masalah yang terkait dengan *Corrective Maintenance* (CM) sehingga mengurangi kegagalan yang tidak perlu terjadi dan akhirnya memperpanjang usia komponen. PM dilakukan untuk menghindari kegagalan, untuk mendeteksi gejala awal kegagalan, atau untuk menentukan kegagalan tersembunyi (Smith, 1993).

Tujuan PM adalah untuk meningkatkan kinerja peralatan dan kehandalan dengan mencegah kegagalan peralatan. PM umumnya digunakan pada rentang usia kegagalan komponen terkait atau di mana tingkat kegagalan komponen mengikuti apa yang disebut kurva bak mandi. (Gambar 2.2)



Gambar 2.2. Kurva Bak Mandi untuk *Preventive Maintenance* (Mobley, RK, 2002)

Tugas lain yang dilakukan dalam PM mencakup pemeriksaan, pengetesan, pengaturan, kalibrasi, perbaikan dan penggantian suku cadang.

Tujuan dan manfaat mengadopsi PM, (Clety, 2008):

- Peningkatan keandalan sistem.
- Penurunan biaya penggantian.
- Penurunan downtime sistem.
- Manajemen persediaan suku cadang yang lebih baik.

Namun untuk menjalankan suatu sistem PM yang baik, daftar alat, suku cadang dan instrumen yang diperlukan harus tersedia. Sebuah prosedur untuk mencatat pengukuran juga harus dibuat. Penekanannya harus dilakukan pada batas atau rentang parameter yang akan diukur. Prosedur keselamatan yang diperlukan seperti isolasi juga harus tersedia.

Dalam rangka untuk dapat mengatur strategi PM, rekomendasi pemeliharaan di buku petunjuk pemeliharaan dari produsen peralatan harus tersedia dan dipahami oleh orang-orang yang bekerja dalam tim.

Seperti semua sistem pemeliharaan, kelebihan dan kekurangan memang ada seperti dibahas di bawah ini. Kinerja PM memiliki banyak keuntungan termasuk peningkatan ketersediaan peralatan, waktu pelaksanaan pemeliharaan dilakukan dalam kondisi nyaman, beban kerja seimbang, peningkatan pendapatan produksi, konsistensi dalam kualitas, terhindar dari kebutuhan peralatan yang mendadak, dapat mengurangi inventarisasi suku cadang, ketersediaan suku cadang lebih terjamin, serta mudah dalam menjadwalkan sumber daya. Beberapa kelemahan dari PM adalah kemungkinan kerusakan dari suatu komponen lebih terekspos, menggunakan lebih banyak bagian, kenaikan biaya awal, kegagalan dalam bagian-bagian baru/komponen, dan tuntutan akses yang lebih sering untuk peralatan. (B.S Dhillon, 2002)

2.4 Condition Based Maintenance (CBM)

Sistem CBM atau Pemeliharaan prediktif merupakan metode perpanjangan dari pemeliharaan preventif dan telah

terbukti untuk meminimalkan biaya pemeliharaan, meningkatkan keselamatan operasional dan mengurangi frekuensi dan keparahan kegagalan suatu komponen. Teori dasar pemantauan kondisi adalah untuk mengetahui kondisi komponen mesin baik atau memburuk sebelum terjadi kerusakan.

CBM adalah seperangkat tindakan perawatan berdasarkan kebutuhan pemeliharaan yang diperoleh dari data inspeksi kondisi komponen secara *real time*, data tersebut bisa diperoleh dari sensor internal ataupun eksternal, bisa juga di dapat dari suatu pengukuran yang diambil dari peralatan portabel. (Michael VBrown, 2003). CBM juga melibatkan kegiatan membandingkan tren parameter fisik yang diukur terhadap batas rekayasa untuk tujuan mendeteksi, menganalisis dan memperbaiki masalah sebelum terjadi kegagalan.

Ada beberapa jenis komponen penting dalam *Harbour Portal Crane*. Komponen-komponen ini memerlukan pemeriksaan rutin untuk memastikan integritas mereka. Tujuan dari pemeriksaan ini adalah untuk mengidentifikasi degradasi dalam integritas sistem selama usia pakai mereka dan untuk memberikan peringatan dini agar tindakan perbaikan dapat diambil sebelum terjadi kegagalan. Penilaian kondisi ini diperlukan untuk mengoptimalkan jadwal inspeksi dan pemeliharaan, sehingga dapat membuat keputusan dan untuk menghindari kerusakan tidak terencana.

CBM adalah sistem yang berusaha untuk mengidentifikasi kesalahan sebelum mereka menjadi kritis yang memungkinkan perencanaan yang akurat dari PM. Dengan CBM, peralatan penting yang berbeda dinilai sementara dalam operasi dan dibuat keputusan apakah mereka membutuhkan perawatan atau tidak dan jika demikian, kapan itu harus dilakukan untuk mencegah kegagalan. Penilaian dapat dilakukan dari beberapa cara seperti inspeksi secara visual, pengujian kinerja komponen, menganalisa hasil lab baik itu oli ataupun coolant, pengawasan, mendiagnosa dan memprediksi kegagalan.

2.5 Corrective Maintenance (CM)

CM juga dikenal sebagai pemeliharaan terinci, dilakukan untuk membawa kembali peralatan keadaan kondisi kerja setelah kegagalan telah terjadi. Logika manajemen adalah *run-to-failure* mudah dan langsung.

Sebuah pabrik menggunakan manajemen *run-to-failure* tidak menghabiskan uang pada pemeliharaan sampai mesin atau sistem rusak. Namun, di hampir semua contoh beberapa pabrik yang menggunakan filosofi manajemen *run-to-failure*, tetap melakukan pencegahan kegagalan yaitu dengan mengganti pelumas dan mesin dalam jangka waktu tertentu. Membengkaknya biaya terkait dengan jenis manajemen pemeliharaan CM adalah:

- Biaya persediaan suku cadang tinggi.
- Biaya lembur tenaga kerja tinggi.
- *Downtime* mesin tinggi.
- Ketersediaan produksi rendah.

Metodologi penelitian merupakan dasar agar proses penelitian berjalan secara sistematis, terstruktur dan terarah. Dalam rangka mencapai tujuan, penelitian ini menggunakan metodologi sebagai berikut:

2.6 Failure Mode and Effects Analysis (FMEA)

Analisa Modus kegagalanan efek (FMEA) adalah analisis kualitatif terstruktur dari suatu sistem, subsistem, komponen, atau fungsi yang menggaris bawahi mode kemungkinan kegagalan, penyebabnya, dan efek dari kegagalan pada sistem operasi. FMEA juga mengevaluasi kekritisan kegagalan, yaitu, keparahan efek dari kegagalan dan kemungkinan terjadinya. Jenis analisis ini disebut sebagai analisis modus kegagalan, efek, dan kekritisan (FMEA) dan

diberikan mode prioritas kegagalan. Tujuan melakukan FMEA adalah untuk menganalisis proses dimana sistem akan dijalankan dan menilai bagaimana potensi kegagalan dalam proses yang akan mempengaruhi operasi sistem.

2.7 Keuntungan FMEA

Beberapa keuntungan dari melakukan FMEA adalah:

- a. Berfungsi sebagai alat visibilitas bagi manajer.
- b. Menyediakan pendekatan sistematis untuk mengklasifikasi kegagalan hardware.
- c. Mengidentifikasi semua mode kemungkinan kegagalan dan efeknya terhadap misi, personil dan sistem.
- d. Menghasilkan data yang berguna untuk digunakan dalam sistem keamanan dan analisis perawatan.
- e. Efektif menganalisa sistem yang kecil, besar, dan kompleks.
- f. Mudah dimengerti.
- g. Mulai dari tingkat terkecil sampai tingkat terbesar.
- h. Mendeteksi risiko terhadap kinerja sistem dan keamanan.

divisi pemeliharaan digunakan untuk membuat tabel ini dengan benar sesuai dengan fungsi *HPC*.

2.7.1 Risk Priority Number Method (RPN)

Teknik ini berdasarkan nomor prioritas risiko untuk modus kegagalan item pada tiga faktor: probabilitas kejadian, keparahan efek kegagalan, dan kemungkinan deteksi kegagalan. RPN mengevaluasi risiko yang berhubungan dengan kemungkinan kegagalan yang telah didapat. Mode kegagalan dengan RPN tinggi lebih kritis dan diberikan prioritas yang lebih tinggi daripada yang dengan RPN rendah.

$$\text{RPN} = \text{Severity(S)} * \text{Occurrence (O)} * \text{Detection (D)}$$

Mode kegagalan dengan RPN tinggi lebih kritis dan diberikan prioritas yang lebih tinggi daripada yang dengan RPN rendah. Jika skala yang digunakan berkisar dari 1 sampai 10, nilai suatu RPN antara 1 dan 1.000. Namun, skala dan kategori yang digunakan dapat bervariasi dari satu organisasi ke yang lain.

2.7.1.1 Severity (S)

Tingkat keparahan ke gagalannya dapat digambarkan sebagai sejauh mana kegagalan komponen akan mempengaruhi jalannya operasional dari mesin atau sistem.

Table 2.1 Peringkat untuk *Severity*

DeskripsiRanking	Level Severity	Ranking
Perbaikan dapat berlangsung antara 1 - 2 jam. Penggantian dan suku cadang yang dibutuhkan mudah. Tidak diperlukan penghentian <i>plant</i> .	Minor	1
Perbaikan dapat memakan waktu antara 2-4 jam dan memerlukan suku cadang penting. Tidak diperlukan penghentian <i>plant</i> .	Low	2
Perbaikan dapat memakan waktu antara 4-6 jam dan memerlukan suku cadang yang mahal. Tidak diperlukan penghentian <i>Equipment</i> .	Moderate	3
Perbaikan dapat berlangsung lebih dari 6-10 hari dapat	High	

menyebabkan kegagalan mesin.		4
Perbaikan dapat berlangsung lebih dari 10 Jam dan dapat menyebabkan <i>Equipment</i> gagal	Extremely High	5

2.7.1.2 Occurrence (O)

Terjadinya kegagalan didefinisikan sebagai bahwa probabilitas kegagalan sebagian terjadi atau jumlah relatif kegagalan selama umur pemakaian yang diharapkan. Dengan bantuan dari karyawan bagian departemen pemeliharaan, diputuskan untuk mengambil umur hidup 3 tahun.

Tabel 2.2 Ranking untuk terjadinya kegagalan (O)

	Definisi	Rangking
Remote	Jumlah kegagalan suku cadang antara 0-2	1
Extremely Low	Jumlah kegagalan suku cadang antara 2-4	2
Very Low	Jumlah kegagalan suku cadang antara 4-6	3
Low	Jumlah kegagalan suku cadang antara 6-8	4
Reasonably Low	Jumlah kegagalan suku cadang antara 8-10	5
Moderate	Jumlah kegagalan suku cadang antara 10-12	6
Reasonably High	Jumlah kegagalan suku cadang antara 12-14	7
High	Jumlah kegagalan suku cadang antara 14-16	8

	cadang antara 14-16	8
Very High	Jumlah kegagalan suku cadang antara 16-18	9
Extremely High	Jumlah kegagalan suku cadang lebih dari 18	10

2.7.1.3 Deteksi (D)

Deteksi dapat didefinisikan sebagai kemampuan untuk mengidentifikasi masalah atau sumber potensial kesalahan. Semakin dini kesalahan terdeteksi, adalah semakin baik. Jika kerusakan tidak terdeteksi dari awal, hasil yang diperoleh dapat berakibat fatal bagi pabrik dalam hal biaya pemeliharaan perbaikan dan biaya kerugian produksi.

Tabel 2.3 Rangkaing dari Deteksi Kegagalan.

Deteksi yang pasti.	1
Masih dapat terdeteksi oleh indera manusia, tetapi mesin harus dihentikan.	2
Deteksi moderat dan membutuhkan sejumlah waktu dan pengalaman.	3
Deteksi sulit dan perangkat eksternal harus digunakan.	4
Tidak ada sensor yang tersedia dan pengujian perangkat yang sangat canggih harus digunakan	5

2.8 Penentuan Strategi Pemeliharaan

Setelah indeks kekritisitas telah dilakukan, strategi perawatan yang paling tepat harus disarankan. Sejalan dengan hal ini, manual prosedur dari pembuat peralatan dikonsultasikan dan dilakukan wawancara dengan karyawan didevisi pemeliharaan untuk memilih strategi yang sesuai untuk setiap komponen peralatan. Sehingga dapat ditentukan strategi pemeliharaan yang tepat. Dengan nilai RPN masing-masing peralatan juga ditambahkan sebagai salah satu bahan pertimbangan.

Strategi pemeliharaan yang disarankan akan membantu perusahaan untuk meramalkan kegagalan dimasa akan datang, sehingga perusahaan akan dapat mengambil tindakan korektif untuk mengatasinya sehingga dapat mengurangi biaya *downtime*. Dengan demikian, akan ada pemantauan yang lebih baik dari sistem.

Tabel 2.4 Tugas Pemeliharaan

Strategi Pemeliharaan	Catatan
Breakdown	<ul style="list-style-type: none">• Pemeriksaan rutin harus dilakukan.• Terjadinya Rendah.• Harus memastikan ketersediaan suku cadang.

Preventive	<ul style="list-style-type: none"> • Periksa alignment. • Mengoles (<i>Greasing</i>) • Memeriksa secara harian atau periodik. • Pembersihan secara harian atau periodik. • Periksa secara visual untuk pakaian, korosi, retak, atau kebocoran pelumas.
Predictive	<ul style="list-style-type: none"> • Pemeriksaan berkala. • Analisis data dan sampel yang diperoleh secara periodik.

Data Kerusakan & Pola Maintenance

- Desain dan perilaku pemakaian
- Perilaku Maintenance

Data Awal

1. Data kerusakan
2. Data Maintenance
3. Data Pola Maintenance

2.9 FTA (*Failure Tree Analysis*)

Teknik untuk mengidentifikasi kegagalan (*failure*) dari suatu sistem dengan memakai FT (*fault tree*) diperkenalkan pertama kali pada tahun 1962 oleh *Bell Telephone Laboratories* dalam kaitannya dengan studi tentang evaluasi keselamatan sistem peluncuran *minuteman missile* antar benua. *Boeing company* memperbaiki teknik yang dipakai oleh *Bell Telephone Laboratories* dan memperkenalkan program komputer untuk

melakukan analisis dengan memanfaatkan FT baik secara kualitatif maupun kuantitatif.

Fault Tree Analysis (FTA) ini biasanya digunakan untuk melihat reabilitas dari suatu produk. *Fault tree* disini bersifat *top-down* artinya analisis yang dilakukan dimulai dari kejadian umum (kerusakan secara umum) selanjutnya penyebabnya (khusus) dapat ditelusuri ke bawahnya. FTA ini merupakan *tools* yang sederhana dan *powerful* untuk melakukan pendekatan terhadap reabilitas dan keamanan (*safety*) dari suatu produk.

Sebuah *fault tree* mengilustrasikan keadaan dari komponen sistem (*basic event*) dan hubungan antara *basic event* dan *top event*. Simbol grafis yang digunakan untuk menyatakan hubungan disebut gerbang logika (*logic gate*). Output dari sebuah gerbang logika ditentukan oleh *event* yang masuk ke gerbang tersebut.

FTA menggunakan langkah-langkah terstruktur dalam melakukan analisis pada sistem. Adapun langkah-langkah FTA yaitu:

1. Mengidentifikasi kejadian/peristiwa terpenting dalam sistem (*top level event*).

Langkah pertama dalam FTA ini merupakan langkah penting karena akan mempengaruhi hasil analisis sistem. Pada tahap ini, dibutuhkan pemahaman tentang sistem dan pengetahuan tentang jenis-jenis kerusakan (*undesired event*) untuk mengidentifikasi akar permasalahan sistem. Pemahaman tentang sistem dilakukan dengan mempelajari semua informasi tentang sistem dan ruang lingkungannya.

2. Membuat pohon kesalahan.

Setelah permasalahan terpenting teridentifikasi, langkah berikutnya adalah menyusun urutan sebab akibat pohon kesalahan. Pada tahap ini, *cause and effect diagram (Ishikawa)* dapat digunakan untuk menganalisis kesalahan dan mengeksplorasi keberadaan kerusakan-kerusakan yang tersembunyi. Pembuatan pohon kesalahan dilakukan dengan menggunakan simbol-simbol *Boolean*. Standardisasi simbol-simbol tersebut diperlukan untuk komunikasi dan konsistensi pohon kesalahan.

3. Menganalisis pohon kesalahan.

Analisis pohon kesalahan diperlukan untuk memperoleh informasi yang jelas dari suatu sistem dan perbaikan-perbaikan apa yang harus dilakukan pada sistem. Tahap-tahap analisis pohon kesalahan dapat dibedakan menjadi 3, yaitu:

a. Menyederhanakan pohon kesalahan.

Tahap pertama analisis pohon kesalahan adalah menyederhanakan pohon kesalahan dengan menghilangkan cabang-cabang yang memiliki kemiripan karakteristik. Tujuan penyederhanaan ini adalah untuk mempermudah dalam melakukan analisis sistem lebih lanjut.

b. Menentukan peluang munculnya kejadian atau peristiwa terpenting dalam sistem (*top level event*).

Setelah pohon kesalahan disederhanakan, tahap berikutnya adalah menentukan peluang kejadian paling penting dalam sistem. Pada langkah ini, peluang semua input dan logika hubungan digunakan sebagai pertimbangan penentuan peluang.

c. *Me-review* hasil analisis.

Review hasil analisis dilakukan untuk mengetahui kemungkinan perbaikan yang dapat dilakukan pada sistem.

Output yang diperoleh setelah melakukan FTA adalah peluang munculnya kejadian terpenting dalam sistem dan memperoleh akar permasalahan sebabnya. Akar permasalahan tersebut kemudian digunakan untuk memperoleh prioritas perbaikan permasalahan yang tepat pada sistem.

FTA digambarkan dalam bentuk hirarki. Pada bagian atas terdapat top event. Top event ini merupakan suatu kejadian yang tidak diinginkan. Selanjutnya setelah top event, di bawahnya akan ada fault event yang lain. Fault event ini ada beberapa jenis, diantaranya:

1. *Primary fault* adalah kesalahan yang terjadi akibat kerusakan pada komponen itu sendiri yang rusak.
2. *Secondary fault* adalah kesalahan yang terjadi akibat komponen berada pada kondisi yang tidak tepat, namun komponen tersebut tidak rusak.
3. *Command fault* adalah kesalahan yang terjadi akibat komponen berada pada waktu dan tempat yang tidak tepat.

Selanjutnya setiap *fault* ini akan saling terhubung secara horizontal dengan hubungan “*and*” atau “*or*”. Jika hubungan yang terjadi antara dua kejadian adalah “*and*” berarti kejadian di atasnya baru dapat terjadi jika kedua kejadian di bawahnya terjadi, namun jika penghubungnya adalah “*or*” maka kejadian di atasnya dapat terjadi jika salah satu kejadian di bawahnya terjadi.

FTA ini dapat melihat risiko yang terjadi akibat kejadian yang simultan. Dengan kata lain, hal ini akan membuat identifikasi risiko semakin dipercaya dan valid. FMEA dan FTA ini saling melengkapi satu sama lain. Dengan kata lain, FMEA untuk menganalisis suatu risiko, sedangkan FTA menganalisis kemungkinan sumber-sumber risiko sebelum timbulnya kerugian.

Pada FTA terdapat simbol – simbol yang menunjukkan hubungan dari suatu kejadian yang diperlukan untuk terjadinya kejadian puncak. Smbol – symbol yang mendeskripsikan hubungan yaitu:

-  Gerbang AND, menunjukkan suatu kondisi dimana semua kejadian di bawah gerbang (kejadian input) ini harus memberikan andil untuk kejadian di atas gerbang (kejadian out put). Dengan kata lain, kejadian output akan terjadi jika semua kejadian input terjadi secara serentak.
-  Gerbang OR, menunjukkan suatu kondisi dimana satu atau beberapa kejadian di bawah gerbang (kejadian input) menuju kejadian di atas gerbang (kejadian out put). Dengan kata lain, kejadian output akan terjadi jika hanya satu atau beberapa kejadian input terjadi.

Sedangkan simbol yang menunjukkan suatu kejadian yaitu :

1.  Persegi panjang, merupakan bangunan blok yang utama untuk *tree* analitik. Menunjukkan kejadian negatif dan lokasinya berada di atas *tree* dan dapat juga ditempatkan sepanjang *tree* untuk mengindikasikan kejadian lain yang dapat dikembangkan lagi. Ini merupakan simbol yang mempunyai gerbang logika dan kejadian input di bawahnya.
2.  Lingkaran, menunjukkan kejadian dasar pada *tree*. Terletak di bawah deretan bertingkat dari *tree* dan tidak perlu pengembangan atau perincian lagi. Tidak ada gerbang atau kejadian di bawahnya.
3.  Belah ketupat, mengidentifikasi bagian kejadian yang tidak berkembang. Kejadian yang tidak sepenuhnya dibangun karena kekurangan informasi.
4.  Elips, menunjukkan suatu kejadian khusus yang hanya dapat terjadi jika kejadian-kejadian tertentu terjadi.

Gerbang OR adalah gerbang yang menggambarkan gabungan dari kejadian – kejadian. Gerbang OR ekuivalen pada symbol Boolean “ + .” untuk kejadian- kejadian. Masukan yang digambarkan pada gerbang OR ekuivalen dengan ekspresi Boolean $T = A_1 + A_2 + \dots + A_n$ untuk T adalah kejadian Output dan A_1, A_2, \dots, A_n merupakan kejadian – kejadian masukan (Vesely dkk.2009).

Gerbang AND adalah gerbang yang menggambarkan irisan dari kejadian – kejadian. Gerbang AND ekuivalen dengan symbol Boolean “ *.” Untuk n kejadian – kejadian masukan pada gerbang AND ekuivalen dengan ekspresi Boolean $T = A_1 * A_2 \dots * A_n$ untuk T merupakan kejadian output dan $A_1, A_2 \dots A_n$ merupakan kejadian – kejadian masukan.

Pada Fault Tree “0” diartikan sebagai kejadian gagal yang tidak terjadi dan “1” diartikan sebagai kejadian gagal terjadi.

2.10 Aljabar Boolean

Misalkan B adalah himpunan yang didefinisikan pada dua operasi biner $+$ dan $*$ dan sebuah operasi unary yang dinotasikan $'$; misalkan 0 dan 1 menyatakan dua elemen yang berbeda dari B maka $(B, +, *, ', 0, 1)$ disebut aljabar Boolean jika aksioma-aksioma berikut berlaku untuk setiap elemen a, b, c dari himpunan B (Lipschutz, 2001).

1. Hukum Komutatif
 - a. $a + b = b + a$
 - b. $a * b = b * a$
2. Hukum distributif
 - a. $a + (b * c) = (a + b) * (a + c)$
 - b. $a * (b + c) = (a * b) + (a * c)$
3. Hukum identitas
 - a. $a + 0 = a$
 - b. $a * 1 = a$
4. Hukum komplemen
 - a. $a + a' = 1$
 - b. $a * a' = 0$

Ekspresi Boolean pada symbol x_1, \dots, x_n didefinisikan secara berulang sebagai $0, 1, x_n$ adalah ekspresi Boolean. Jika X_1 dan X_2 adalah ekspresi Boolean maka $(X_1), X_1', X_1 + X_2$ dan $X_1 * X_2$ adalah ekspresi Boolean jika X adalah ekspresi Boolean pada symbol x_1, \dots, x_n , sehingga ditulis $X = X(x_1, \dots, x_n)$.

Misalkan $B = (S, +, *, ', 0, 1)$ adalah suatu aljabar Boolean dan $a, b, c \in S$, Menenuhi sifat sebagai berikut:

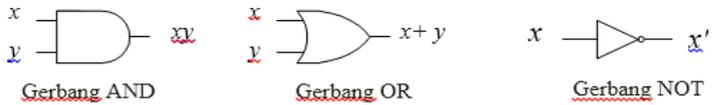
1. *Hukum Idempoten (Johnsonbaugh, 2001)*
 - a. $a + a = a$
 - b. $a * a = a$

2. *Assosiatif (Lipschutz, 2001)*
 - a. $(a + b) + c = a + (b + c)$
 - b. $(a * b) * c = a * (b * c)$
3. *Hukum Boundedness (Johsonbaugh, 2001)*
 - a. $a + 1 = 1$
 - b. $a * 0 = 0$
4. *Hukum Absorspsi (Johsonbaugh, 2001)*
 - a. $a + (a * b) = a$
 - b. $a * (a + b) = a$
5. *Hukum De Morgan's untuk aljabar Boolean (Johsonbaugh, 2001)*
 - a. $(a + b)' = a' * b'$
 - b. $(a * b)' = a' + b'$
6. *Hukum involusi $(a')' = a$ (Johsonbaugh, 2001)*
7. *Hukum 0 dan 1 (Johsonbaugh, 2001)*
 - a. $0' = 1$
 - b. $1' = 0$
8. a. $a + (d * b) = a + b$
b. $a' * (a + b) = (a + b)'$ (Vesely dkk, 2009)

2.11 Gerbang Logika

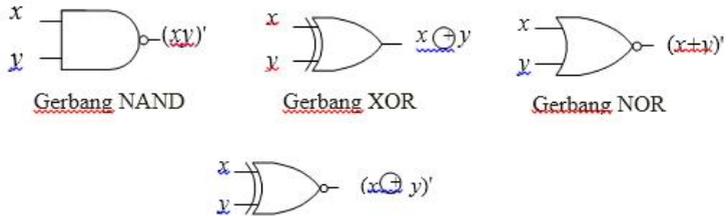
Gerbang logi digambarkan sebagai mesin yang memuat satu input atau lebih dan tepat satu output. Ada dua jenis gerbang logika pada aljabar Boolean yaitu gerbang logika dasar yang terdiri dari AND, OR dan NOT serta gerbang logika kombinasi yang merupakan turunan dari gerbang logika dasar terdiri dari NAND, NOR, XOR dan XNOR.

Gerbang Logika Dasar:



Gambar 2.3 Gerbang Logika Dasar

Kombinasi Gerbang Logika



Gambar 2.4 Kombinasi Gerbang Logika

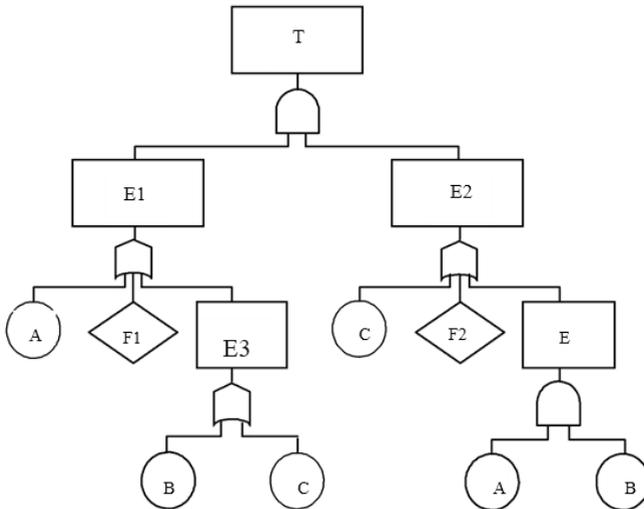
Tabel 2.5 Tabel kebenaran Gerbang Logika Dasar dan kombinasi gerbang Logika

Input		OR	AND	XOR	NOR	NAND	XNOR	NOT
X	Y	$X+Y$	$X * Y$	$X \oplus Y$	$(X+Y)'$	$(X * Y)'$	$(X \oplus Y)'$	X'
0	0	0	0	0	1	1	1	1
0	1	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	0
1	1	1	1	0	0	0	1	0

2.12 Minimal cut set dan Minimal Path Set

Suatu Fault Tree menyediakan informasi yang bernilai tentang kemungkinan kombinasi dari kejadian gagal yang dapat menghasilkan kegagalan pada kejadian puncak. Merepresentasikan fault tree ke dalam persamaan Boolean bertujuan untuk menentukan minimal cut set dan minimal path set, minimal cut set menunjukkan kejadian gagal sedangkan minimal path set menunjukkan kejadian gagal yang tidak terjadi.

Minimal cut set merupakan kombinasi terkecil dari kejadian dasar yang menyebabkan kejadian puncak terjadi. Penentuan minimal cut set dari fault tree menggunakan aturan Boolean dapat dilihat pada pengonstruksian minimal cut set dari fault tree berikut:



Gambar 2.5 Fault Tree

$$T = E1 * E2 \tag{1}$$

$$E1 = A + F1 + E3 \tag{2}$$

$$E3 = B + C \tag{3}$$

$$E2 = C + F2 + E4 \tag{4}$$

$$E4 = A * B \tag{5}$$

Persamaan kejadian puncak dapat diperoleh dengan substitusi terlebih dahulu sehingga diperoleh persamaan

$$T = (A + F1 + B + C) * (C + F2 + (A * B))$$

$$\Leftrightarrow T = (A * C) + (A * F2) + (A * (A * B)) + (F1 * C) + (F1 * F2) + (F1 * (A * B)) + (B * C) + (B * F2) + (B * (A * B)) + (C * C) + (C * F2) + (C * (A * B)) \tag{6}$$

Menggunakan teorema 1b, definisi 1b, teorema 4a maka diperoleh

$$T = (A * F2) + (A * B) + (F1 * F2) + (B * F2) + C \tag{7}$$

sehingga diperoleh lima minimal cut set, yaitu :

- | | | |
|------------------|-------------------|------------------|
| a. $M1 = A * F2$ | c. $M3 = C$ | e. $M5 = B * F2$ |
| b. $M2 = A * B$ | d. $M4 = F1 * F2$ | |

Pada M 1, M 4 dan M 5 terdapat komponen F yang bukan merupakan komponen utama penyebab kegagalan. Jika yang diperhatikan dalam analisa kegagalan hanya penyebab utamanya maka persamaan (7) dapat disederhanakan dengan mengasumsikan bahwa minimal cut set yang digunakan hanya minimal cut set yang komponennya memuat kejadian dasar (basic event) sehingga minimal cut set yang memuat komponen F dapat dihilangkan, persamaannya dapat dituliskan menjadi:

$$T = (A * B) + C \dots \dots \dots \tag{8}$$

Konstruksi fault tree untuk persamaan (8) dapat dilihat pada gambar 2.6. Setiap fault tree mempunyai minimal cut set yang berhingga dengan pola umum

$$T = M_1 + M_2 + M_3 + \dots + M_n$$

Dimana T adalah kejadian puncak dan $M_{i; i; 1, 2, \dots, n}$ adalah minimal cut set dan setiap minimal cut set merupakan hasil perkalian satu atau beberapa kejadian dasar (basic event). Untuk minimal cut set memuat satu komponen menunjukkan kejadian dasar tunggal yang menyebabkan kejadian puncak terjadi, sedangkan minimal cut set yang memuat dua komponen menunjukkan hasil perkalian dari dua kejadian dasar dan minimal cut set yang memuat k komponen menunjukkan hasil perkalian dari k kejadian dasar. Sehingga dengan minimal cut set dapat dilihat kombinasi kejadian dasar apa saja atau kejadian dasar apa saja yang bisa menyebabkan kegagalan terjadi.

Persamaan (8) didapat dua minimal cut set yaitu $(A * B)$ dan C . $(A * B)$ adalah minimal cut set yang terdiri dari dua komponen dan C adalah minimal cut set yang mempunyai komponen tunggal.

Penggunaan metode bottom-up pada dasarnya sama saja dengan top-down perbedaannya terletak pada awal memulainya. Substitusi top-down dimulai dari persamaan (1) sedangkan bottom up dimulai dari persamaan (3) dan (5) untuk disubstitusi ke persamaan (2) dan (4).

Pengonstruksian minimal path set dapat dilakukan setelah minimal cut set diperoleh. Minimal path set merupakan dari minimal cut set. Minimal cut set menunjukkan kegagalan yang terjadi maka minimal path set menunjukkan kegagalan yang tidak terjadi. Dilihat dari persamaan (8) telah didapatkan minimal cut set selajutnya mengomplemenkan T maka didapat:

$$T' = ((A * B) + C) \dots \dots \dots (9)$$

Berdasarkan teorema 5b untuk suku ke-1 dari persamaan (9) maka diperoleh :

$$T' = (A'+B') * C' \dots \dots \dots (10)$$

Berdasarkan definisi 2b untuk persamaan (10) maka didapat

$$T' = (A' * C') + (B' * C') \dots \dots \dots (11)$$

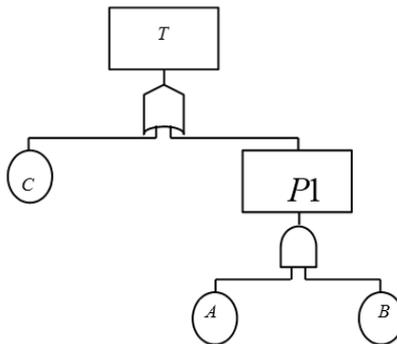
Konstruksi fault tree untuk persamaan (11) dapat dilihat pada gambar 2.7 Pada persamaan (11) diperoleh dua minimal path set yaitu :

- a. $P1 = A' * C'$
- b. $P2 = B' * C'$

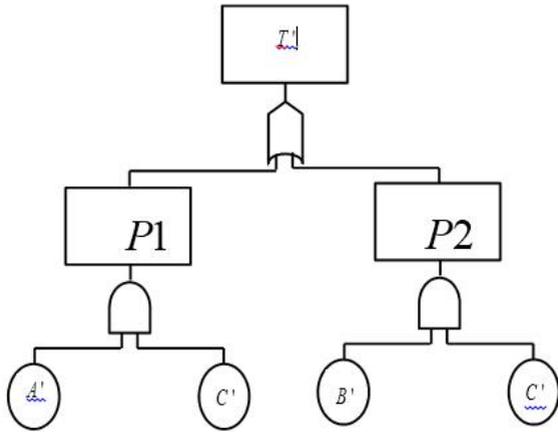
Bentuk umum dari minimal path set untuk suatu kejadian puncak adalah

$$T' = P1 + P2 + P3 + \dots \dots + Pm$$

Dimana T' merupakan kejadian puncak (kegagalan) yang tidak terjadi sedangkan $P1$, $P2$, $P3$,....., Pm merupakan minimal path set dan setiap minimal path set merupakan hasil perkalian dari satu atau beberapa kejadian dasar yang tidak gagal. Sehingga dengan minimal path set dapat dilihat kombinasi kejadian dasar tidak gagal apa saja yang membuat kegagalan tidak terjadi.



Gambar 2.6 Fault Tree dari persamaan (8)



Gambar 2.7 Fault Tree Persamaan (11)