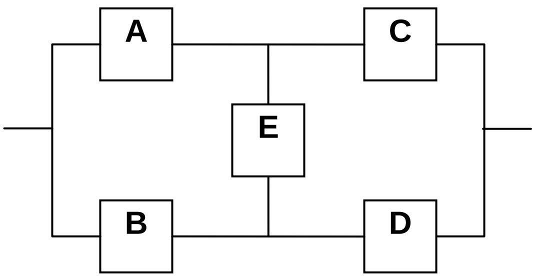
BAB 1

MODEL JARINGAN UNTUK SISTEM KOMPLEKS

1.1 Konsep Model Jaringan

Pada bab sebelumnya telah diuraikan teknik dalam melakukan pemodelan jaringan untuk sistem sederhana. Beberapa pola hubungan komponen didalam sistem telah dibahas secara grafis dan matematis. Pola hubungan tersebut meliputi susunan seri, paralel, kombinasi seri-paralel, dan standby. Namun demikian, tidak semua sistem memiliki konfigurasi yang sederhana seperti itu. Ada kalanya kita dihadapkan pada konfigurasi yang tidak mungkin disederhanakan dengan pemodelan seperti yang telah dibahas pada bab sebelumnya.

Sebagai salah satu contoh, lihatlah model jaringan *bridge type* yang terlihat pada gambar 4.1. Pada gambar tersebut terlihat lima buah komponen A, B, C, D dan E yang terhubung satu sama lain. Susunan tersebut tidak dapat secara serta merta disederhanakan menjadi susunan seri, paralel ataupun kombinasi diantara keduanya. Sistem dengan susunan seperti ini membutuhkan pendekatan yang lebih kompleks dalam proses evaluasi keandalannya.



Gambar 1.1-1 Bridge type network

Pada bab ini akan diberikan beberapa teknik untuk mengevaluasi keandalan rangkaian komponen dalam sisten yang kompleks. Metode-metode yang akan diberikan disini meliputi: metode peluang bersyarat (*conditional probability*), *cut set, tie set, event tree, dan fault tree*.

Semua metode yang disebutkan diatas pada dasarnya didasarkan atas proses penyederhanaan sistem menjadi susunan seri dan paralel sederhana, dan selanjutnya proses evaluasi diselesaikan dengan cara khusus untuk mendapatkan peluang sistem gagal ataupun suskses.

1.2 Metode Peluang Bersyarat (*conditional probability*)

Dalam melakukan evaluasi sistem yang kompleks dengan metode peluang bersyarat, sistem akan diuraikan menjadi susunan seri dan paralel sederhana, selanjutnya hasil penyederhanaan tersebut digabungkan kembali dengan meggunakan konsep peluang bersayarat.

P(sistem sukses/gagal)=P(sistem sukses/gagal jika komponen "X" selalu sukses) x P(komponen "X" sukses) + P(sistem sukses/gagal jika komponen "X" selalu gagal) x P(komponen "X" gagal)

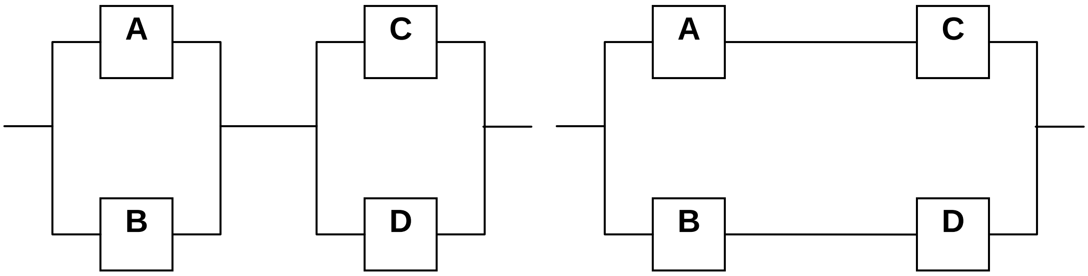
Untuk melihat aplikasi teknik ini, susunan bridge type pada gambar 4.1 akan kita bahas kembali.

Langkah pertama dalam teknik peluang bersyarat ini adalah dengan menentukan satu komponen didalam sistem yang kalau kita asumsikan selalu sukses atau selalu gagal akan dapat menyederhanakan sistem menjadi susunan seri dan paralel sederhana. Komponen ini kita sebut dengan komponen "X". Tidak ada ketentuan khusus dalam menentukan komponen "X" tersebut, kecuali pertimbangan kesederhanaan sistem setelah diasumsikan bahwa komponen tersebut selalu sukses atau selalu gagal.

Contoh 4.1:

Pada gambar 4.1, jika indeks keandalan masing-masing komponen adalah 0.99, maka berapakah indeks keandalan sistem?

Pada gambar 4.1 terlihat bahwa komponen E adalah komponen yang paling tepat mewakili komponen ”X”. Sistem akan menjadi lebih sederhana seperti terlihat pada gambar 4.2. Gambar (a) mengasumsikan komponen E selalu sukses dan (b) mengasumsikan komponen E selalu gagal.



(a) E selalu sukses (b) E selalu gagal

Gambar 1.2-2 Penyederhanaan dengan teknik peluang bersyarat

Jika komponen E selalu sukses, maka akan selalu ada maka komponen A dan B akan pada susunan paralel, demikian juga antara komponen C dan D dan kedua susunan parelel tersebut terhubung seri satu sama lain. Jika E selalu gagal, maka komponen A dan C akan pada susunan seri demikian juga dengan komponen B dan D dan kedua susunan seri tersebut akan berada pada susunan paralel satu sama lain. Kedua susunan E selalu sukses dan E selalu gagal ini adalah *mutually exclusive* karena kedua-duanya tidak dapat muncul bersama-sama dan karena itu bisa digabungkan dengan metode peluang bersyarat.

Pada kasus yang lebih kompolek mungkin dibutuhkan penyederhanaan lebih lanjut. Hal ini tidak menjadi masalah, sebab penyederhanaan lebih lanjut akan diikuti dengan penggabungan memakai metode peluang bersyarat.

Setelah penyederhanaan seperti pada gambar 4.2 tersebut dilakukan maka kita bisa rumuskan bahwa:

$R_s = R_s$(jika E selalu sukses). R_E + R_s(jika E selalu gagal). Q_E

Pada kondisi E selalu sukses maka:

$R_s = (1 – Q_AQ_B)(1-Q_CQ_D)$

Pada kondisi E selalu gagal, maka:

$R_s = 1 – (1 – R_AR_C)(1 – R_BR_D)$

Jika kedua persamaan diatas digabungkan ke persamaan awal, maka diperoleh:

$$R_s = (1 – Q_AQ_B)(1-Q_CQ_D).\ R_E + 1 – (1 – R_AR_C)(1 – R_BR_D)\ .\ Q_E$$

$$= R_AR_C + R_BR_D + R_AR_DR_E + R_BR_CR_E – R_AR_BR_CR_D – R_AR_CR_DR_E –$$

$$R_AR_BR_CR_E – R_BR_CR_DR_E – R_AR_BR_DR_E + 2\ R_AR_BR_CR_DR_E$$

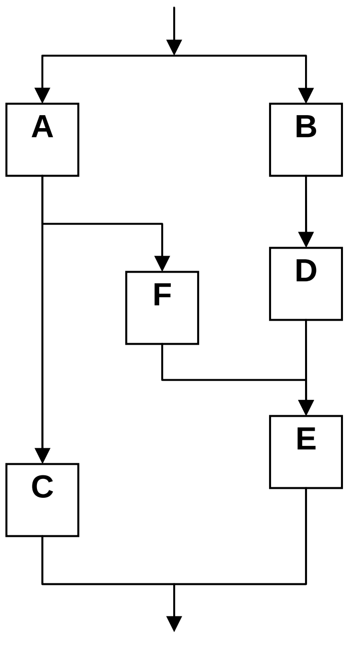
Jika $R_A = R_B = R_C = R_D = R_E = R$, maka

$$R_s = 2R^2 + 2R^3 – 5R^4 + 2R^5$$

Jika indeks keandalan komponen adalah 0.99, maka $R_s = 0.99979805$

<u>Contoh 4.2:</u>

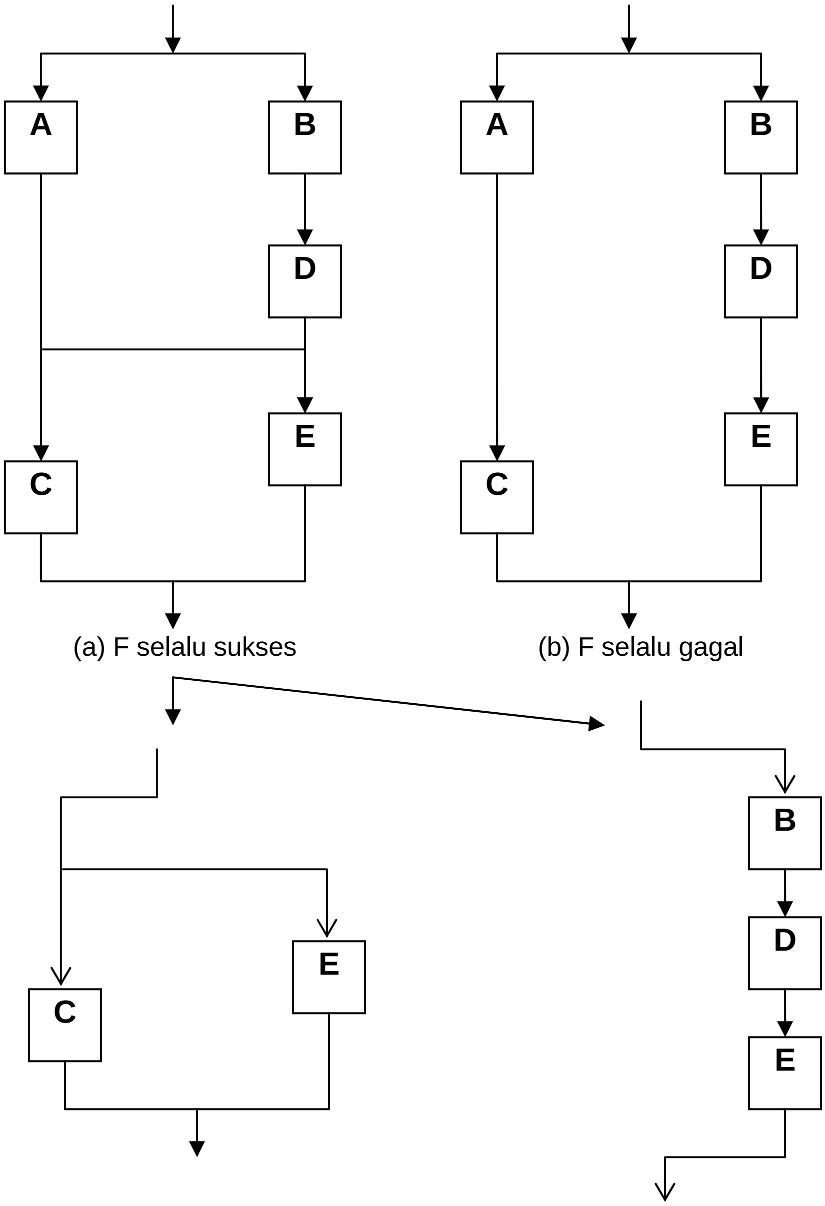
Hitunglah indeks keandalan komponen dari sistem yang terlihat pada gambar berikut jika indeks keandalan masing-masing komponen adalah 0.99?



Pada gambar 4.3, komponen F yang terlihat paling memungkinkan sebagai komponen ”X”. Maka kita bisa asumsikan bahwa komponen F selalu sukses (gambar 4.4 (a)) dan komponen F selalu gagal (gambar 4.4 (b)). Gambar 4.4 (a) selanjutnya masih dapat disederhanakan lebih lanjut dengan mengasumsikan bahwa komponen A selalu sukses (gambar 4.4 (c)) dan komponen A selalu gagal (gambar 4.4 (d)). Penggabungan dengan konsep peluang bersyarat dimulai dari 2 gambar terakhir untuk mendapatkan peluang sukses susunan (b), setelah itu susunan (b) digabungkan dengan susunan (a) untuk mendapat peluang sukses sistem secara keseluruhan.

Gambar 1.2-3 kompleks sistem

Setelah menurunkan gambar penyederhanaan, maka selanjutnya indeks keandalan sistem dapat dihitung dengan konsep peluang bersyarat.



(a) F selalu sukses (b) F selalu gagal

(c) A selalu sukses (d) A selalu gagal

Gambar 1.2-4 Penyederhanaan kompleks sistem

$R_s = R_s$(jika F selalu sukses). R_F + R_s(jika F selalu gagal). Q_F

R_s(jika F selalu gagal) = $1 – (1 – R_B R_D R_E)(1-R_A R_C)$

R_s(jika F selalu sukses)= R_s(jika A selalu sukses). R_A + R_s(jika A selalu gagal). Q_A

R_s(jika A selalu sukses) = $1 - Q_C Q_D$

R_s(jika A selalu gagal) = $R_B R_D R_E$

Dengan substitusi kita dapatkan:

$R_s = [(1- Q_C Q_D)R_A + R_B R_D R_E Q_A] \,.\, R_F + [1-(1-R_B R_D R_E)(1- R_A R_C)\,]. \,Q_F$

Jika indeks keandalam komponen adalah 0.99, maka:

$R_s = 0.999602$ dan $Q_s = 0.000398$

1.3 Metode Cut Set

Metode cut set ini adalah salah satu metode yang umum digunakan pada evaluasi keandalan karena kemudahan menterjemahkan algoritma metode ini ke bahasa pemrograman komputer serta metode ini langsung dapat mengidentifikasi proses gagalnya sistem.

Cut set adalah satu set komponen dalam sistem yang jika gagal akan menyebabkan kegagalan sistem, atau satu set komponen yang mana komponen-komponen di dalamnya harus gagal agar sistem menjadi gagal. Minimal sub set dari cut set yang menyebabkan sistem gagal disebut dengan minimal cut set.

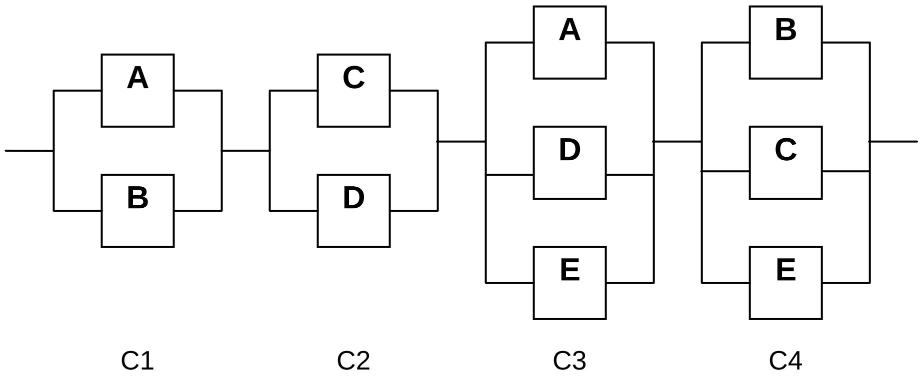
Contoh 4.3:

Dengan menggunakan pengertian diatas, maka pada gambar 4.1 bisa kita temukan beberapa cut set yaitu AB, CD, AED dan BEC. Jika AB gagal, maka sistem akan gagal, demikian juga jika CD, AED dan BEC gagal, maka sistem juga akan gagal. Keempat cut set tadi adalah minimal cut set, sebab tidak ada lagi sub set yang lebih kecil dari cut set tersebut yang akan membuat sistem gagal.

Sistem akan gagal jika cukup satu cut set saja muncul. Hal ini memiliki pola yang sama dengan sistem seri, dimana jika satu komponen gagal maka sistem pasti akan gagal. Karena itu hubungan antara cut set dalam sistem adalah seri.

Selanjutnya, cut set akan gagal jika semua komponen di dalam cut set tersebut gagal. Hal ini memiliki pola yang sama dengan sistem paralel, dimana semua komponen harus gagal untuk menjamin sistem gagal. Karena itu, hubungan antara cut set dalam sistem adalah paralel.

Dari dua pernyataan diatas maka hubungan antar cut set dan hubungan antara komponen di dalam cut set dapat digambarkan seperti dibawah ini.



Gambar 1.3-5 Minimal cut set pada contoh 4.3

Penyederhanaan seperti pada Gambar 4.5 tidak dapat langsung diselesaikan dengan konsep sistem seri dan paralel sederhana mengingat semua komponen muncul lebih dari satu kali. Karena itu hubungan antara komponen di dalam cut set dapat diselesaikan dengan konsep gabungan.

$Qs \quad = P(C_1 U C_2 U C_3 U C_4)$

$= P(C_1)+ P(C_2)+ P(C_3)+ P(C_4)-P(C_1 \cap C_2) -P(C_1 \cap C_3) -P(C_1 \cap C_4)$

$-P(C_2 \cap C_3) -P(C_2 \cap C_4) -P(C_3 \cap C_4)+P(C_1 \cap C_2 \cap C_3) +P(C_1 \cap C_2 \cap C_4)$

$+P(C_1 \cap C_3 \cap C_4) +P(C_2 \cap C_3 \cap C_4)- P(C_1 \cap C_2 \cap C_3 \cap C_4)$

Dimana:

$P(C_1) = Q_A Q_B \quad P(C_2) = Q_C Q_D \quad P(C_3) = Q_A Q_D Q_E \quad (C_4) = Q_B Q_C Q_E$

Dengan demikian

$Q_s \quad = Q_A Q_B+Q_C Q_D+Q_A Q_D Q_E+Q_B Q_C Q_E-Q_A Q_B Q_C Q_D- Q_A Q_B Q_D Q_E- Q_A Q_B Q_C Q_E-$

$Q_A Q_C Q_D Q_E-Q_B Q_C Q_D Q_E+2\, Q_A Q_B Q_C Q_D Q_E$

Jika $Q_A=Q_B=Q_C=Q_D=Q_E=Q$, maka

$Q_s \quad = 2Q^2+2Q^3-5Q^4+2Q^5$

Jika indeks keandalan komponen adalah 0.99, maka Q = 0.01

$Q_s \quad = 0.00020195 \qquad R_s = 0.99979805$

Hasil ini sama dengan hasil yang diperoleh dengan metode peluang bersyarat.

Penyelesaian seperti yang dijelaskan diatas menjadi semakin panjang jika jumlah komponen yang terlibat semakin besar. Karena itu perlu dilakukan beberapa penyederhanaan dengan syarat bahwa tingkat ketelitian dari penyederhanaan ini masih bisa ditoleransi. Salah satu cara penyederhanaan tersebut adalah dengan mengabaikan peluang kegagalan yang diperoleh dari penggabungan cut set. Sehingga peluang kegagalan sistem akan dapat dievaluasi sebagai berikut:

$Q_s \quad = P(C_1)+ P(C_2)+ P(C_3)+ P(C_4)$

$= Q_AQ_B+Q_CQ_D+Q_AQ_DQ_E+Q_BQ_CQ_E$

$= 2Q^2+2Q^3$

Dengan demikian Q_s = 0.000202, dan R_s = 0.999798

Nilai diatas hampir sama dengan nilai keandalaan yang didapatkan dengan tanpa penyederhanaan Perbedaan hanya pada digit ke 7 dan ke 8, yang secara teknis bisa diabaikan.

Penyederhanaan lebih lanjut adalah dengan menghitung cut set yang hanya terdiri dari 2 komponen saja, sehingga

$Q_s \quad = P(C_1)+ P(C_2)$

$= Q_AQ_B+Q_CQ_D$

$= 2Q^2$

Dengan demikian Q_s = 0.000200, dan R_s = 0.999800

Ingat bahwa penyederhanaan ini akan memiliki efek yang besar terhadap perubahan nilai sebenarnya jika keandalan masing-masing komponen tidak terlalu tinggi (ketidakhandalannya adalah tinggi). Namun jika keandalan komponen reltif tinggi, maka secara teknis penyederhanaan ini tidak menhasilkan perbedaan nilai keandalan atau ketidakandalan yang signifikan dibandingkan dengan hasil tanpa proses penyederhanaan.

Pada beberapa kasus sistem kompleks, penentuan cut set dapat dilakukan secara langsung melalui pengamatan terhadap block diagram sistem. Namun pada beberapa kasus lainnya (jika komponen yan terlibat semakin banyak), penentuan cut set akan menjadi susah. Karena itu dibutuhkan sebuah metode dan algoritma yang dapat dijasikan sebagai acuan dalam menurunkan jumlah dan jenis cut set yang ada di dalam sistem.

Langkah-langkah dalam penentuan cut set adalah sebagai berikut:

(1) Turunkanlah *minimal path* dari block diagram.

(2) Susun *incidence matrix* yang mengidentifikasi semua komponen di dalam sistem. Beri nilai 1 pada komponen yang ada pada path yang sesuai, dan beri nilai 0 pada komponen yang tidak muncul pada path tersebut.

(3) Jika semua elemen komlom pada *incidence matrix* adalah *non zero*, maka kolom tersebut adalah cut set orde pertama.

(4) Gabungkan 2 kolom pada *incidence matrix* secara berurutan. Jika dari penggabungan ini terdapat kolom yang semua komponennya adalah *non zero*, maka kolom tersebut adalah cut set orde kedua.

(5) Lakukan kembali langkah (4) dan temukan cut set dengan orde yang lebih tinggi hingga tidak dapat lagi ditemukan adanya cut set.

Jika langkah tersebut diatas kita aplikasikan pada sistem seperti terlihat pada gambar 4.1, maka berturut-turut akan kita dapatkan:

(1) Minimal path adalah : AB, CD, AED dan BEC

(2) *Matrix incidence* adalah: (lihat dibawah)

(3) Kita lihat bahwa tidak ada kolom yang semua komponennya adalah *non zero*. Artinya tidak ada cut set orde pertama.

path	component				
	A	**B**	**C**	**D**	**E**
1	1	0	1	0	0
2	0	1	0	1	0
3	1	0	0	1	1
4	0	1	1	0	1

(4) Dengan menggabungkan dua kolom yang berturutan, maka *incidence matrix* akan menjadi seperti dibawah ini. Kita lihat bahwa AB dan CD adalah kolom dengan komponen *non zero* semua. Karena itu AB dan CD adalah cut set orde kedua (karena terdiri dari 2 komponen).

path	component									
	AB	**AC**	**AD**	**AE**	**BC**	**BD**	**BE**	**CD**	**CE**	**DE**
1	1	2	1	1	1	0	0	1	1	0
2	1	0	1	0	1	2	1	1	0	1
3	1	1	2	2	0	1	1	1	1	2
4	1	1	0	1	2	1	2	1	2	1

(5) Dengan menggabungkan tiga kolom yang berturutan, maka *incidence matrix* akan menjadi seperti dibawah ini. Kita lihat bahwa ABC, ABD, ABE, ACD, ADE, BCD, BCE, dan CDE adalah kolom dengan komponen *non zero* semua. Tetapi AB dan CD adalah cut set orde kedua, sehingga cut set orde ketiganya adalah cut set yang tidak memiliki komponen AB dan CD. Dari pernyataan ini kita peroleh bahwa cut set orde ketiganya adalah ADE dan BCE.

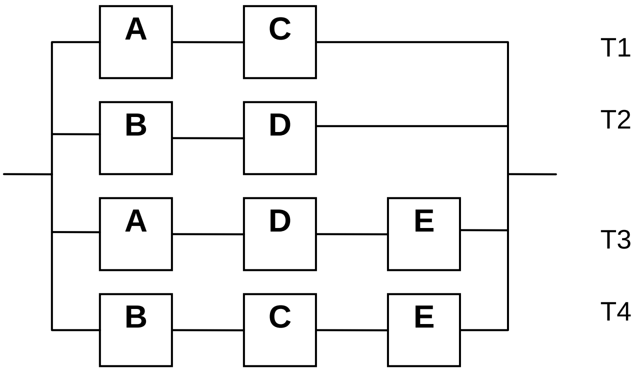
path	component								
	ABC	**ABD**	**ABE**	**ACD**	**ACE**	**ADE**	**BCD**	**BCE**	**CDE**
1	2	1	1	2	2	1	1	1	1
2	1	2	1	1	0	1	2	1	1
3	1	2	2	2	2	3	1	1	2
4	2	1	2	1	2	1	2	3	2

Berdasarkan hasil dari tahapan-tahapan diatas, maka diperoleh bahwa cut set nya adalah AB, CD, ADE dan BCE. Hasil ini sama dengan penurunan cut set dari pengamatan langsung seperti pada penjelasan sebelumnya.

1.4 Metode Tie Set

Metode tie set adalah komplemen dari metode cut set. Pada metode cut set yang kita cari adalah peluang kegagalan sistem, namun pada metode tie set yang kita cari adalah peluang suksesnya sistem. Cut set adalah satu set komponen yang jika gagal akan menjamin sistem gagal, sementara itu tie set adalah satu set komponen yang harus sukses untuk menjamin sistem sukses. Sistem akan sukses jika paling tidak ada satu tie set yang sukses, artinya hubungan antara tie set didalam sistem serupa dengan susunan paralel. Tie set akan sukses jika semua komponen di dalam tie set juga sukses. Karena itu hubungan antara komponen di dalam satu tie set serupa dengan susunan seri.

Dengan penjelasan diatas, maka hubungan komponen di dalam tie set diagram adalah seperti pada gambar 4.6 . semua komponen muncul 2 kali pada tie set diagram. Karena itu, penyederhanaan dengan susunan seri atau paralel tidak dapat dilakukan disini. Untuk mencari berapa peluang sukses dari sistem kita kembali harus menggunakan konsep gabungan seperti pada metoda cut set.



Gambar 1.4-6 Minimal tie set pada contoh 4.3

Rs $= P(T_1 U T_2 U T_3 U T_4)$

$= P(T_1)+ P(T_2)+ P(T_3)+ P(T_4)-P(T_1\cap T_2) -P(T_1\cap T_3) -P(T_1\cap T_4)$

$-P(T_2\cap T_3) -P(T_2\cap T_4) -P(T_3\cap T_4)+P(T_1\cap T_2\cap T_3) +P(T_1\cap T_2\cap T_4)$

$+P(T_1\cap T_3\cap T_4) +P(T_2\cap T_3\cap T_4)- P(T_1\cap T_2\cap T_3\cap T_4)$

Dimana:

$P(T_1) = R_A R_C$ $P(T_2) = R_B R_D$ $P(T_3) = R_A R_D R_E$ $(T_4) = R_B R_C R_E$

Dengan demikian

$R_s = R_AR_C+R_BR_D+R_AR_DR_E+R_BR_CR_E-R_AR_BR_CR_D- R_AR_BR_DR_E- R_AR_BR_CR_E-$

$R_AR_CR_DR_E-R_BR_CR_DR_E+2\ R_AR_BR_CR_DR_E$

Jika $R_A=R_B=R_C=R_D=R_E=R$, maka

$R_s \quad = 2R^2+2R^3-5R^4+2R^5$

Jika indeks keandalan komponen adalah 0.99, maka:

$R_s = 0.99979805 \qquad Q_s \quad = 0.00020195$

Hasil ini sama dengan hasil yang diperoleh dengan metode peluang bersyarat maupun metode cut set.

1.5 Metode Event Trees

Event trees adalah diagram yang menunjukkan semua kejadian yang mungkin terjadi di dalam sistem. Diagram ini diperoleh dengan mengidentifikasi semua komponen di dalam sistem, dan selanjutnya secara berturut-turut dianalisa setiap cabang dari diagram dengan memasukkan peluang sukses dan gagalnya masing-masing komponen.

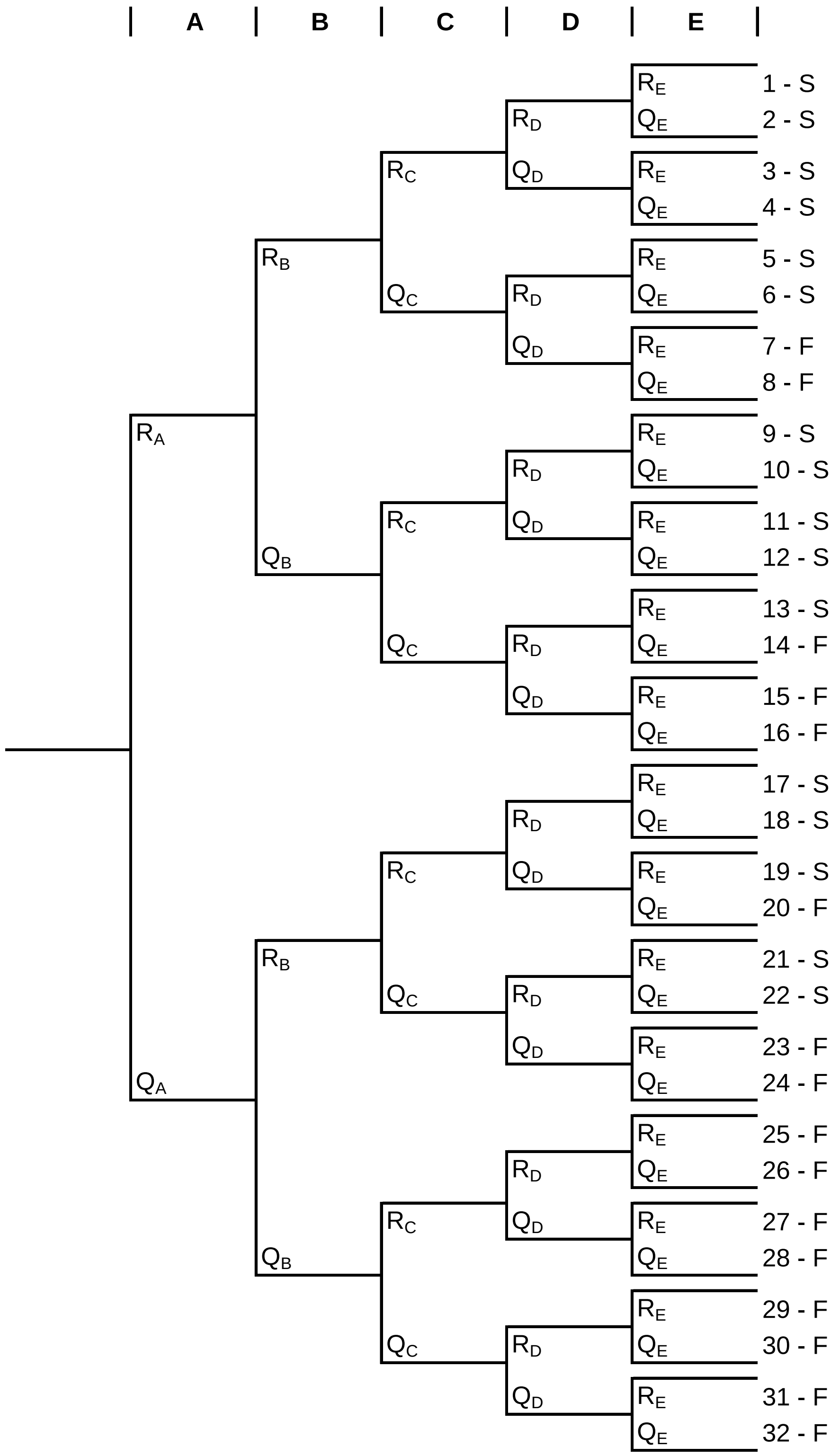
Metode event trees ini dapat dipergunakan untuk melakukan analisa keandalan pada sistem dimana semua komponen beroperasi kontinyu, maupun pada sistem yang memiliki beberapa komponen yang berada pada posisi standby (mengandung *logic switching*). Sistem yang mengadung komponen standby ini sering dipergunakan pada *safety oriented systems*.

Pada kasus yang pertama dimana semua komponen beroperasi kontinyu, maka penyusunan diagram event trees-nya bisa dimulai dari komponen yang mana saja, sebab yang dicari adalah peluang kejadian yang menyebabkan sistem gagal/sukses bukan pada bagaimana sistem tersebut menjadi gagal.

Pada kasus *safety oriented system*, tahapan terjadinya kegagalan pada sistem akan sangat berpengaruh terhadap nilai peluang kegagalan/suksesnya sistem.

Pada kasus sistem dengan komponen yang beroperasi kontinyu, dengan mengambil contoh sistem seperti terlihat pada gambar 4.1, maka event trees nya adalah seperti terlihat pada gambar 4.7.

Pada gambar tersebut terlihat terdapat 32 cabang. Cabang akan berhumlah 2^n, dimana n adalah jumlah komponen di dalam sistem. Setelah menurunkan event trees tersebut maka bisa dimasukkan peluang sukses/gagalnya masing-masing komponen, sehingga peluang sukses/gagalnya masing-masing cabang akan diketahui.



Gambar 1.5-7 Event trees untuk sistem seperti pada contoh 4.3

Jika P(Ci) adalah peluang dari kejadian (sukses/gagal) di masing-masing cabang, maka diperoleh bahwa peluang sukses sistem adalah:

$$R_s = P(C_1) + P(C_2) + P(C_3) + P(C_4) + P(C_5) + P(C_6) + P(C_9) + P(C_{10}) + P(C_{11})$$
$$+ P(C_{12}) + P(C_{13}) + P(C_{17}) + P(C_{18}) + P(C_{19}) + P(C_{21}) + P(C_{22})$$

Dimana sebagai contoh:

$P(C_1) = R_A R_B R_C R_D R_E$ $\quad$ $P(C2) = R_A R_B R_C R_D Q_E$ $\quad$ dst

Demikian juga halnya, peluang gagalnya sistem adalah:

$$Q_s = P(C_7) + P(C_8) + P(C_{14}) + P(C_{15}) + P(C_{16}) + P(C_{20}) + P(C_{23}) + P(C_{24}) +$$
$$P(C_{25}) + P(C_{26}) + P(C_{27}) + P(C_{28}) + P(C_{29}) + P(C_{30}) + P(C_{31}) + P(C_{32})$$

Dimana sebagai contoh:

$P(C_7) = R_A R_B Q_C Q_D R_E$ $\quad$ $P(C_8) = R_A R_B Q_C Q_D Q_E$ $\quad$ dst

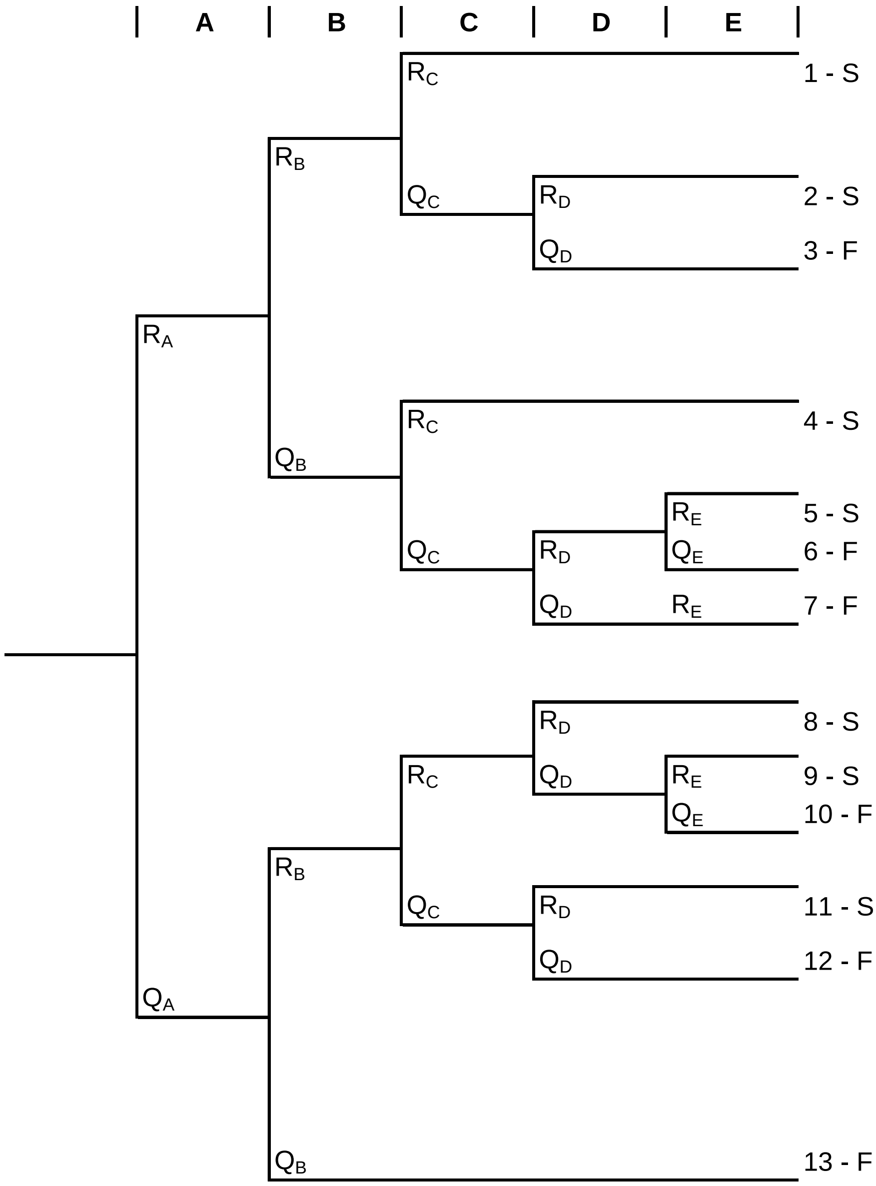
Jika indeks keandalan komponen adalah R = 0.99, dan Q = 0.01 maka:

$R_s = 0.999798$ $Q_s = 0.000202$

Hasil ini sama dengan hasil yang diperoleh dengan metode peluang bersyarat maupun metode cut set dan tie set.

Proses penyelesaian dengan metode event trees diatas membutuhkan waktu yang panjang karena kita harus menghitung semua kemungkinan kombinasi kejadian di dalam sistem. Ada beberapa cara penyederhanaan yang bisa dilakukan dalam mempercepat perhitungan dengan tidak mengurangi tingkat ketelitiannya.

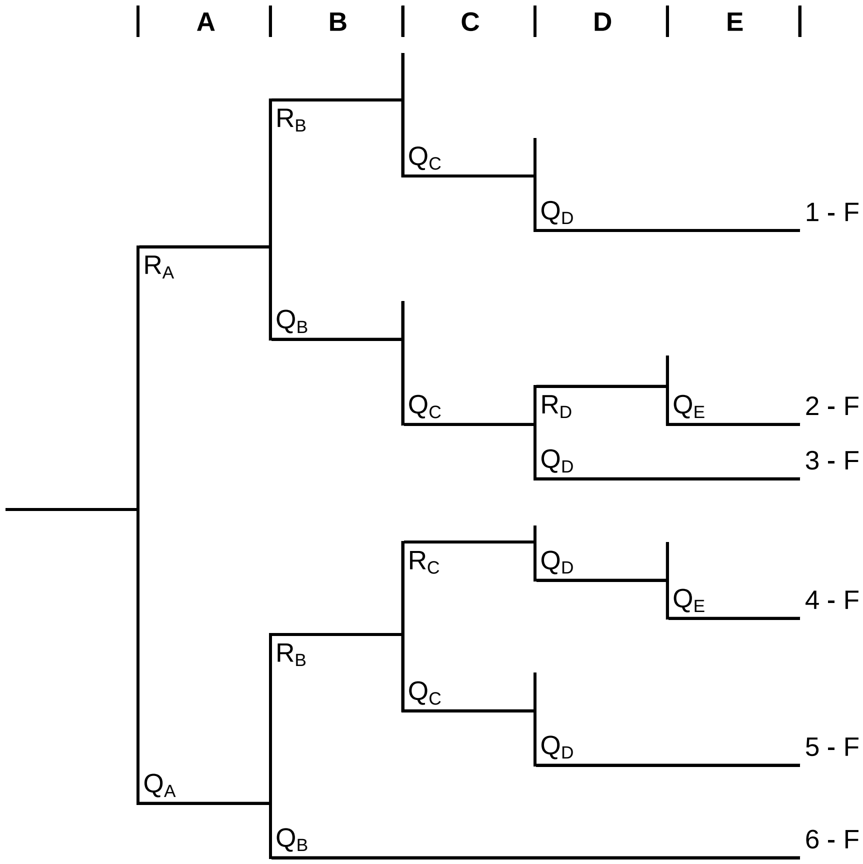
Cara pertama adalah dengan menghentikan penurunan event trees saat diketahui bahwa cabang event trees yang sedang diturunkan gagal atau sukses. Seperti terlihat pada gambar 4.8, pada cabang yang pertama, setelah diketahui bahwa jika hanya komponen A sukses, komponen B sukses dan komponen C sukses akan menjamin sistem sukses, maka penurunan cabang diselesaikan dan peluang sukses cabang pertama bisa langsung dihitung.



Gambar 1.5-8 Penyederhanaan event tress

Hal yang sama juga dilakukan untuk cabang-cabang lainnya. Setelah semua nilai peluang cabang dihitung, gabungan cabang-cabang yang menghasilkan sukses (1, 2, 4, 5, 8, 9 dan 11) dijumlahkan dan akan menghasilkan peluang suksesnya sistem (keandalan). Hal yang sama juga dilakukan untuk cabang yang menghasilkan peluang gagal. Dengan penyederhanaan ini, dibutuhkan hanya 13 cabang saja untuk bisa menentukan peluang sukses atau gagalnya sistem.

Penyederhanaan lebih lanjut bisa dilakukan dengan hanya memperhitungkan salah satu peluang saja, yaitu menghitung cabang-cabang yang menghasilkan sukses ataupun hanya cabang-cabang yang menhasilkan peluang gagal saja.



Gambar 1.5-9 Penyederhanaan event tress

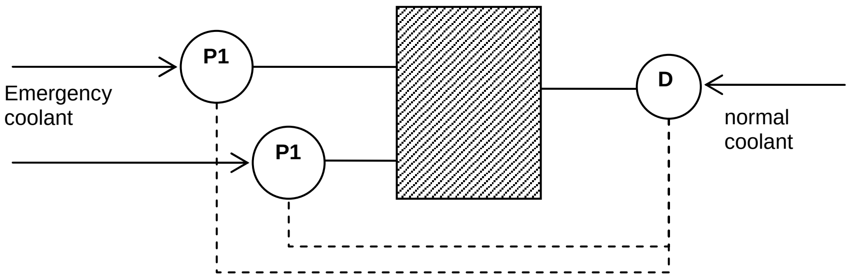
Seperti terlihat pada gambar 4.9 Hanya cabang yang menghasilkan gagal yang diturunkan dan penurunan dihentikan jika sudah diketahui bahwa cabang tersebut memberikan hasil gagal. Disini hanya ada 6 cabang saja yang perlu dievaluasi, sehingga akan menyingkat waktu evaluasi dibandingkan dengan penurunan-penurunan sebelumnya.

Seperti telah disampaikan di awal pembahasan metode event trees bahwa metode ini bisa dipergunakan untuk sistem yang komponennya beroperasi kontinyu dan untuk *safety oriented system*.

Pada kasus *safety oriented system* contoh berikut bisa memberikan gambaran lebih jelas tentang aplikasi metode event trees.

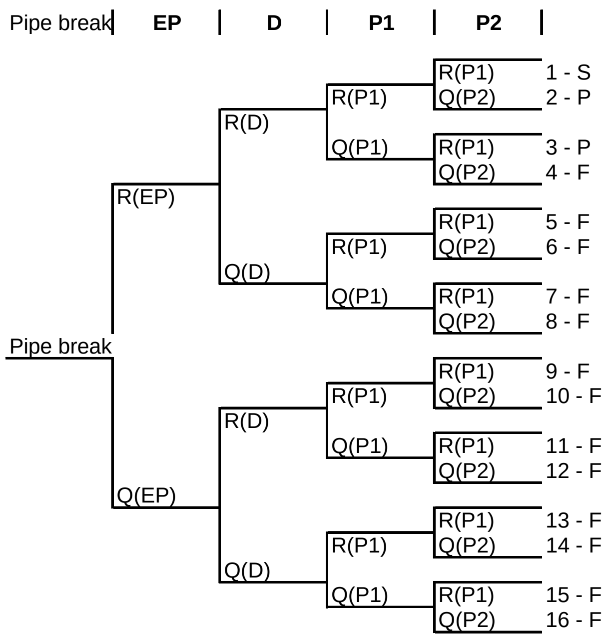
<u>Contoh 4.4:</u>

Gambar 4.10 menunjukkan sistem pendingin yang terdiri dari tangki air pendingin, detektor aliran (D) dan dua buah pompa emergency (P1 dan P2) yang dipergunaakan saat suplai air pendingin utama terganggu. Proses kegagalan sistem dimulai dari pecahnya pipa suplai air pendingin utama. Kejadian ini akan menyebabkan detektor aliran memberi sinyal ke motor listrik untuk menggerakkan P1 dan P2 untuk mensuplai air pendingin ke tangki pendingin. Jika kedua pompa beroperasi penuh akan menjamin sistem 100% sukses, 1 pompa beroperasi akan menjamin sistem 50% sukses dan kedua pompa gagal akan menyebabkan sisem gagal, maka berapakah peluang masing-masing kejadian tersebut jika masing-masing komponen memiliki indeks keandalan 0.99?



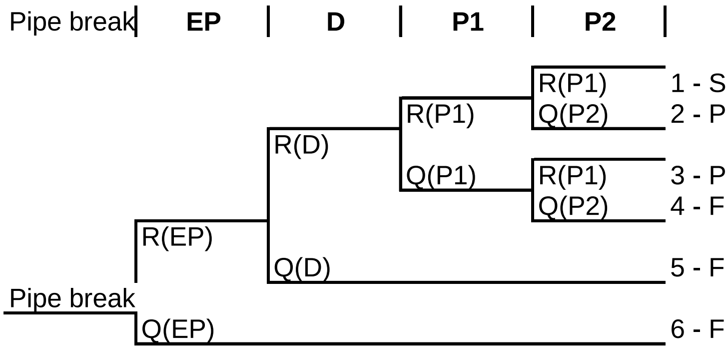
Gambar 1.5-10 Flow diagram sistem suplai air pendingin

Dari penjelasan diatas terlihat bahwa urutan proses akan menjadi: pipa pecah → listrik mengalirkan sinyal ke detektor → detektor memerintahkan pompa beroperasi → pompa 1 beroperasi → pompa 2 beroperasi. Urutan ini akan menghasilkan siagram event trees sebagai berikut:



Gambar 1.5-11 Event Trees contoh 4.4

Karena yang akan kita cari cukup peluang gagal atau sukses, maka event tree pada gambar 4.11 dapat disederhanakan menjadi:



Gambar 1.5-11 Pentederhanaan event Trees contoh 4.4

Dari gambar diatas terlihat bahwa:

R_s = R(EP) . R(D) . R(P1) . R(P2)

= 0.99^4 = 0.960596

$R_{partial}$ = R(EP) . R(D) . R(P1) . Q(P2) + R(EP) . R(D) . Q(P1) . R(P2)

= 2 x 0.99^3 x 0.01 = 0.019406

Q_s = R(EP) . R(D) . Q(P1) . Q(P2) +R(EP) . Q(D) + Q(EP)

= 0.99^2 X 0.01^2 + 0.99 X 0.01 + 0.01 = 0.019998

1.6 Fault Trees

Metode fault trees adalah salah satu metode evaluasi keandalan sistem yang umum digunakan, khususnya pada sistem keselamatan atau safety oriented system. Metode ini pertama kali dikembangkan sebagai salah satu cara untuk mengevaluasi proses kegagalan sistem secara kualitatif. Perkembangan berikutnya, dengan algoritma tertentu, metode ini dapat dipergunakan untuk melakukan evaluasi keandalan secara kuantitatif.

Fault trees menggunakan menggunakan beberapa *logical gates* untuk menghubungan antara satu kejadian (*event*) pada sistem dengan kejadian yang lainnya. Kondisi kegagalan yang sering disebut sebagai *top event* secara bertahap diturunkan menjadi kejadian-kejadian dibawahnya secara bertahap dengan bantuan logical gates hingga penyebab dasar kegagalan (*basic event*) ditemukan. Karena itu metode ini dikatagorikan sebagai *top-down approach*.

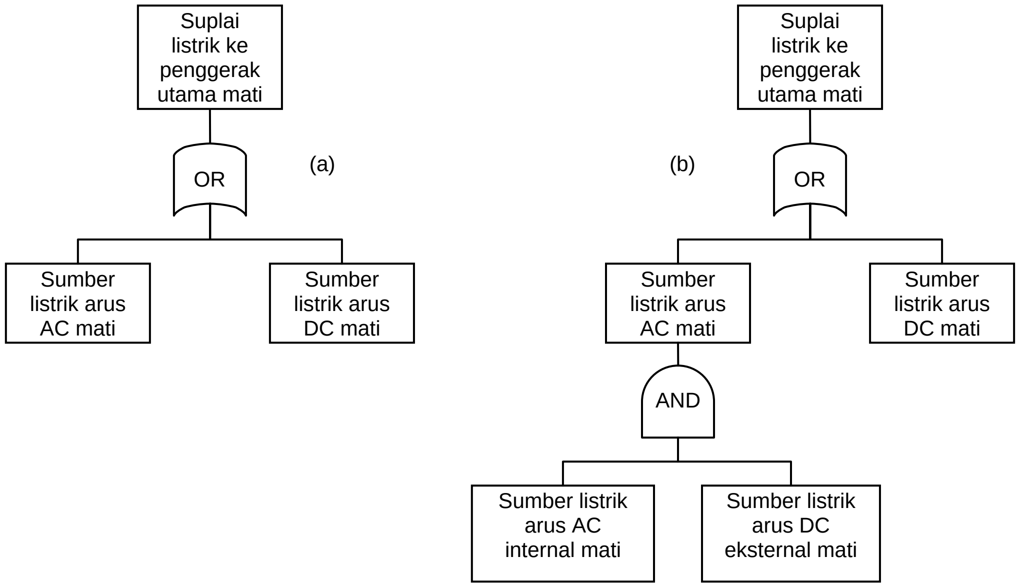
Dengan pendekatan kualitatif, maka tahapan proses kegagalan secara terperinci bisa diturunkan sehingga metode ini dapat mengidentifikasi bagaimana proses kegagalan sistem. Dengan mengetahui proses kegagalan sistem ini, maka perbaikan, pengaturan dan modifikasi pada sistem dapat dilakukan agar kejadian kegagalan yang sama bisa dicegah.

Pendekatan kuantitatif dilakukan dengan memberikan nilai peluang terhadap munculnya masing masing *basic event*, dan selanjutnya dengan bantuan *logical gates* dan algoritma tertentu akan ditemukan besarnya peluang sistem menjadi gagal atau peluang munculnya *top event* bisa didapatkan.

Materi yang disajikan pada buku ini hanya bertujuan memberikan pengantar awal bagi pemahaman umum terhadap metode event trees. Penguasaan metode ini secara komprehensif membutuhkan referensi tentang fault trees lainnya. Untuk lebih memahami konsep dasar metode fault trees ini, kita bisa amati contoh sederhana berikut.

Contoh 4.5:

Perhatikannlah konsep sistem suplai listrik seperti terlihat pada gambar 4.12.

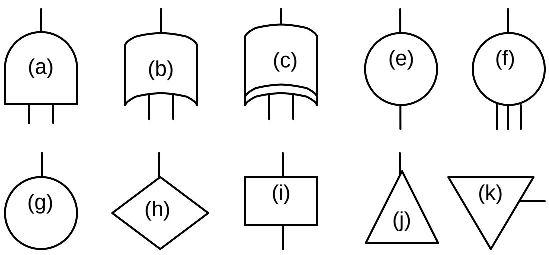


Gambar 1.6-12 Fault tree sistem suplai listrik

Pada bagian (a) terlihat bahwa hilangnya suplai listrik disebabkan karena dua kejadian yakni hilangnya suplai listrik AC atau hilangnya suplai listrik DC. Arus AC akan memberikan tenaga untuk penggerak utama dan arus DC akan memberikan tenaga listrik kepada peralatan elektronik pengatur operasi penggerak utama. Kedua kejadian ini harus sukses untuk menjamin sistem menjadi sukses. Karena itu, kedua kejadian ini dihubungkan oleh sebuah logical gate yang disebut dengan OR gate yang memiliki arti bahwa *top event* (kegagalan suplai listrik) akan terjadi paling tidak jika salah satu dari kejadian dibawahnya muncul atau satu kejadian di level dua sudah menjamin *top event* akan terjadi.

Selanjutnya pada bagian (b) terlihat bahwa jika kejadian pada level kedua masih bisa dicari penyebabnya, maka kejadian tersebut bisa diturunkan menjadi beberapa kejadian penyebab dan dihubungkan dengan *logical gate* lainnya. Terlihat bahwa kegagalan dalam mensuplai arus AC disebabkan karena sumber arus AC internal dan eksternal terputus. Jika salah satu kejadian ini tidak terjadi, maka suplai arus AC akan tetap terjamin. Artinya, kegagalan suplai arus AC terjadi jika kedua kejadian dibawahnya terjadi. Kedua kejadian pada level 3 ini dihubungkan dengan *logical gate* AND (AND gate). Penurunan penyebab kegagalan pada level tiga ini masih bisa dilakukan jika terdapat informasi yang cukup akan hal tersebut. Dengan demikian, ketelitian dan kejelasan dalam penurunan fault tree ini akan sangat tergantung pada kemampuan orang yang melakukan analisa dalam menterjemahkan kejadian-kejadian yang berpengaryh terhadap gagalnya sistem.

Dalam mendisain diagram fault trees ini terdapat beberapa *logical gates* yang bisa digunakan untuk menghubungan antara satu kejadian dengan kejadian lainnya. Beberapa *logical gates* yang umum digunakan antara lain seperti terlihat pada gambar 4.13.



Gambar 1.6-13 Logical gates pada fault trees

a. AND Gate, kejadian output akan terjadi jika dan hanya jika semua kejadian input terjadi.

b. OR gate, kejadian output akan terjadi jika paling tidak ada satu kejadian input yang terjadi.

c. EOR gate, exclusive OR gate, kejadian output akan terjadi jika dan hanya jika satu kejadian input terjadi.

d. NOT gate, kejadian output akan terjadi hanya jika kejadian input tidak terjadi.

e. *m-out of-n* gate, kejadian output akan terjadi jika dan hanya jika m kejadian input dari n kejadian input yang ada terjadi.

f. *Basic event*, kegagalan pada hirarki terendah pada fault trees

g. *Incomplete event*, kejadian yang membutuhkan penurunan lebih lanjut sampai nanti ditemukannya *basic event*.

h. *Intermediate event*, kombinasi dari kejadian kegagalan sebagai output dari *logical gates*.

i. *Transfers* OUT, titik dimana fault tree bisa dipecah menjadi sub-fault tree.

j. *Transfer* IN, titik dimana sub-fault tree bisa dimulai sebagai kelanjutan pada *transfer* OUT.

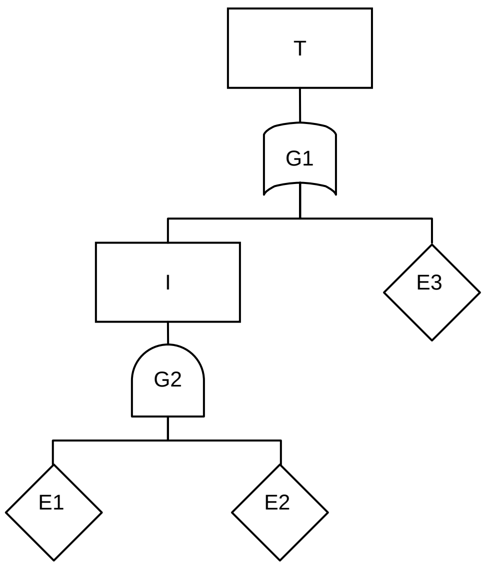
Evaluasi kuantitatif pada metode fault trees dapat dilakukan dengan dua cara. Cara pertama adalah dengan menggabungkan semua *basic event* dengan menggunakan aljabar Boolean. Beberapa aturan aljabar Boolean adalah:

Aturan dasar	aturan komutatif	aturan asosiatif	aturan distributif	aturan dengan 0 dan 1	aturan De Morgan
AA = A	AB = BA	A(BC) = (AB)C	A(B+C) = AB+AC	0A = 0	$\underline{AB} = \underline{A}+\underline{B}$
A+A = A	A+B=B+A	A+(B+C) = (A+B)+C	A+BC=(A+B)(A+C)	1A = A	$\underline{A+B} = \underline{AB}$
A(A+B) = A				0 + A = A	
A+AB = A				1 + A = 1	
AA = A					
A+A = 1					

Cara yang kedua adalah dengan pendekatan numerik yaitu menggabungkan semua peluang munculnya masing-masing kejadian dengan menggunakan aturan-aturan probabilitas.

<u>Contoh 4.6:</u>

Dengan aljabar Boolean maka permasalah seperti terlihat pada gambar 4.12 (b) akan menjadi seperti pada gambar 4.13. Pada gambar tersebut terlihat bahwa penurunan dimulai dari top event. Semua kejadian yang tertulis pada intermediate event secara langsung akan diwakili oleh kejadian-kejadian dibawahnya melalui *logical gates* yang sesuai. Jika keandalan peralatan suplai listrik internal adalah 0.925, keandalan peralatan suplai listrik eksternal adalah 0.933, dan suplai listrik DC adalah 0.995, maka peluang munculnya *top event* adalah:



pada hirarki level I

T = I + E3

Intermediate event I akan digantikan oleh:

I = E1E2

Sehingga

T = E1E2 + E3

Dengan demikian maka peluang top event akan terjadi adalah:

Gambar 1.6-14 fault trees gambar 5.12

P(T) = P(E1E2+E3) = [P(E1)P(E2)]+P(E3)- [P(E1)P(E2)]P(E3), dimana:

P(E1)=1–0.933=0.067 P(E2)=1–0.925=0.075 P(E3)=1–0.995= 0.005

Sehingga P(T) = 0.01

Contoh di atas dapat diselesaikan dengan cara yang sederhana mengingat jumlah *basic event* dan jumlah level yang kecil serta semua kejadian bebas satu sama lain. Sekalipun sederhana, namun konsep pada contoh diatas dapat dijadikan acuan untuk pengembangan evaluasi pada sistem yang lebih kompleks. Karena pada metode ini analisa dilakukan melalui level teratas hingga terbawah, maka metode ini sering disebut dengan metode top-down.

Contoh 4.7:

Dengan menggunakan metode numerik langsung kita menghitung probabilitas munculnya event mulai dari level yang terendah dan selanjutnya dengan menggabungkan event-event tersebut dengan menggunakan logical gates yang sesuai, maka nilai probabilitas intermediate events di atasnya akan dapat diketahui. Metode ini juga dikenal dengan metode bottom-up. Contoh sebelumnya jika diselesaikan dengan metode ini akan menjadi:

$P(I) \quad = P(E1)P(E2) = (1-0.933)(1-0.925) \quad = 0.005025$

$P(T) \quad = P(I) + E(3) - P(I)P(E3)$

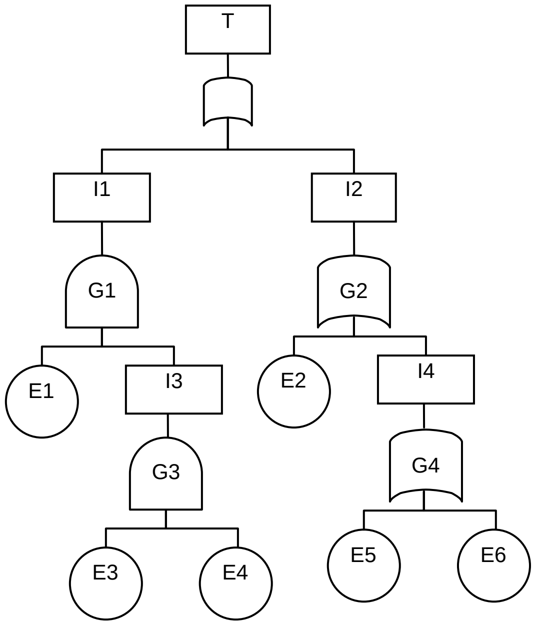
$\quad = 0.005025 + (1-0.995) - 0.005025(1-0.995) = 0.01$

Hasil ini nilainya sama dengan hasil yang diperoleh dengan penggunaan metode sebelumnya.

Pada beberapa kasus terdapat sebuah komponen memberi efek terhadap beberapa komponen atau sub-sistem. Sebagai contoh baterei sebagai sumber listrik arus DC pada suatu sistem kontrol akan memberikan efek kepada beberapa komponen sistem kontrol jika terjadi kegagalan suplai arus listrik. Dengan demikian basic event yang berhubungan dengan gagalnya baterei ini akan muncul beberapa kali pada event trees. Contoh berikut akan menggambarkan bagaimana duplikasi basic event dapat diakomodasi pada event trees.

Contoh 4.8:

Pada fault tree di samping, jika basic event E3 dan E6 adalah sama dan probabilitas masing-masing event adalah P(E1)=0.15, P(E2)=0.01, P(E3)=P(E6)=0.05, P(E4)=0.50 dan P(E3)=0.06, maka dengan aljabar Boolean didapat:



T = (I1 + I2) = (E1I1)+(E2+I4)

= (E1E3E4) + (E2+(E5+E6))

= E1E3E4+ E2+E5+E6

Dengan mensubstitusi probabilitas masing-masing kejadian dengan menggunakan aturan penggabungan AND dan OR, maka diperoleh:

P(T) = 0.119245

Gambar 1.6-15 fault trees contoh 4.8

Dengan menggunakan metode numerik, didapat:

P(I3) = P(E3)P(E4) = 0.025 P(I1) = P(E1)P(I3) = 0.00375

P(I4) = P(E5)+P(E6) - P(E5)P(E6) = 0.107

P(I2) = P(E2)+P(I4) - P(E2)+P(I4) = 0.11593

P(T) = P(I1)+P(I2) - P(I1)+P(I2) = 0.119245

Kedua metode diatas memberikan hasil yang sama.

Jika diasumsikan bahwa E3 dan E6 adalah komponen yang sama, maka metode numerik tidak lagi mampu menyelesaikan permasalahan tersebut. Namun dengan menggunakan aljabar Boolean, penyelesaian akan menjadi:

T = E1E3E4+E2+E5+E6

Dengan menggantikan E6 dengan E3 maka akan diperoleh:

T = E1E3E4+E2+E5+E3

= E3(E1E4+1) + E2 + E5

Berdasarkan aljabar Boolean, maka persamaan diatas akan menjadi

T = E3 + E2 + E5, sehingga

P(T) = 0.11593