

PENGUNAAN ALJABAR BOOLEAN DALAM MENGANALISIS KEGAGALAN PADA FAULT TREE ANALYSIS

Mia Fitria dan Faisal

Program Studi Matematika
Universitas Lambung Mangkurat
Jl. Jend. A. Yani km. 36 Kampus Unlam Banjarbaru

ABSTRAK

Sistem merupakan ruang yang dibatasi aturan dan mempunyai karakteristik masing-masing. Pelaksanaan sistem tidak selalu sukses, Ketidaksiuksesan ini disebabkan karena ada beberapa komponen yang menyebabkan kegagalan pada sistem. Meminimalkan kegagalan dapat dilakukan dengan mencari penyebab kegagalan yang nanti dikonstruksi ke fault tree. Fault tree merupakan gambaran fungsi logika yang menunjukkan hubungan antara kejadian puncak (kegagalan) dengan penyebab kegagalan. Hubungan antara kegagalan dengan penyebab kegagalan dapat dideskripsikan dengan gerbang AND dan OR. Gerbang AND digunakan jika semua kejadian input mempunyai pengaruh untuk terjadinya kejadian output dan gerbang OR digunakan jika salah satu atau semua kejadian input mempunyai pengaruh untuk terjadinya kejadian output. Untuk menganalisa fault tree, terlebih dahulu diubah ke dalam persamaan Boolean dimana gerbang AND dan OR ekuivalen dengan operasi perkalian dan penjumlahan pada aljabar.

Metode yang digunakan adalah studi literatur dengan mencari dan mengumpulkan bahan yang berhubungan dengan fault tree analysis dan aljabar Boolean serta kemudian membentuk minimal cut set dan minimal path set dengan konstruksi fault tree yang sudah ada serta mencari masalah yang nantinya akan diselesaikan dengan fault tree analysis.

Minimal cut set dari fault tree dapat diperoleh dengan menggunakan aljabar Boolean, akan tetapi tidak semua aturan dalam aljabar Boolean digunakan. Ini disebabkan kegagalan pada kejadian puncak yang berbeda mengakibatkan konstruksi fault tree berbeda untuk setiap kegagalan yang berbeda. Minimal path set didapat setelah mengomplemenkan minimal cut set yang didapat sebelumnya dan menggunakan beberapa aturan Boolean yang sesuai. Dari minimal cut set dapat dilihat apa saja yang bisa membuat kegagalan terjadi dan dari minimal path set dapat dilihat apa saja yang membuat kegagalan tidak terjadi. Banyaknya minimal cut set dan minimal path set tidak selalu sama untuk setiap fault tree.

Kata Kunci: *Fault Tree Analysis, Gerbang AND dan OR, Minimal Cut Sets dan Minimal Path Sets.*

ABSTRACT

System is the dimension which it limited by rule and have certain characteristic. Execution of system do not always success, it is caused by some component that causing fault at the system. Minimizing fault can be done by searching the cause of the fault and then its construct to fault tree. Fault tree is a

logic diagram that displays the connection between the fault event and the reason. The connection between the fault event and the reason can be described by AND gate and OR gate. AND gate is used if all of the input occurs and OR gate is used if at least one of the inputs occurs. To analyze fault tree, for the first it is changed to Boolean equation where AND gate and OR gate equivalent with product and sum operation in algebra.

The methodology of this research is study of literature with search and collect all of the materials that are related with fault tree analysis and Boolean algebra and construct minimal cut set and minimal path set by using the construction of fault tree which it is had and search the problem which it is solved by fault tree analysis.

Minimal cut set from a fault tree can be obtained by using Boolean algebra. But, not of all the rules of Boolean algebra are used. It is caused the fault at the different of top event, so that different construction of fault tree for each different fault. Minimal path set is obtained after getting the complement of minimal cut set which it get before and using some rules of Boolean algebra. From minimal cut set can be seen any kind of cause that can make the fault is happened and from minimal path set can be seen any kind of cause that can make the fault is not happened. Sum of minimal cut set and minimal path set are not always same for each fault tree.

Keywords: Fault Tree Analysis, AND Gate and OR Gate, Minimal Cut Set, Minimal Path Set.

1. PENDAHULUAN

1.1 Latar Belakang

Sistem merupakan suatu ruang yang dibatasi aturan dan mempunyai karakteristik tertentu. Dalam penerapannya, sistem tidaklah selalu sempurna. Ketidaksempurnaan sistem terjadi karena adanya kegagalan di beberapa komponen sistem yang tidak diinginkan. Kegagalan ini dapat diminimalkan dengan mencari penyebab kegagalan. Pencarian penyebab dari kegagalan (kejadian puncak) dapat dilakukan sebelum ataupun setelah kegagalan terjadi. Penyebab kegagalan dapat dicari menggunakan suatu analisis kejadian dengan mengkonstruksi setiap penyebab kegagalan ke dalam *fault tree* sehingga penyebab terjadinya kegagalan dapat diidentifikasi sampai menemukan kejadian dasar (*basic event*) yang menyebabkan kegagalan (kejadian puncak) terjadi.

Fault Tree berbeda konteks dengan *tree* pada konsep teori graf. *Tree* yang dibahas dalam teori graf yaitu graf yang tidak berarah dan tidak mengandung sirkuit. Sedangkan *fault tree* merupakan gambaran dari suatu fungsi logika yang menunjukkan hubungan antara kejadian puncak yang dalam hal ini adalah kegagalan pada sistem dengan penyebab-penyebab kegagalan. Hubungan antara kegagalan dan penyebabnya dapat dideskripsikan dengan gerbang AND dan OR. Gerbang AND digunakan jika semua input memberikan andil untuk terjadinya output dan gerbang OR digunakan jika satu atau semua input memberikan andil untuk terjadinya output.

Fault tree akan diolah ke dalam persamaan Boolean dengan gerbang AND dan OR dari *fault tree* yang ekuivalen dengan operasi perkalian dan penjumlahan

pada aljabar. *Fault tree* yang diubah dalam persamaan Boolean disederhanakan dengan menggunakan aljabar Boolean sehingga terbentuklah *minimal cut sets* dan *minimal path sets*.

Minimal cut sets merupakan sub set minimal dari *cut set* dimana *cut set* merupakan himpunan kejadian dasar (*basic event*) dalam sistem yang menyebabkan kegagalan sistem terjadi. Sedangkan *minimal path set* merupakan *minimal sub set* dari *path set*, dimana *path set* merupakan himpunan kejadian dasar (*basic event*) tidak gagal yang menyebabkan kegagalan pada sistem tidak terjadi (sistem sukses).

Tujuan dari penelitian ini adalah menentukan *minimal cut sets* dan *minimal path sets* dengan menggunakan Aljabar Boolean dan mengetahui penerapan *Fault Tree Analysis* dalam suatu kasus. Selanjutnya penelitian ini dibatasi hanya pada penggunaan gerbang AND dan OR pada Aljabar Boolean untuk *fault tree analysis*.

2. TINJAUAN PUSTAKA

2.1 Aljabar Boolean

Definisi 2.1.1 (Lipschutz, 2001)

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 $\langle B, +, *, ', 0, 1 \rangle$ disebut aljabar Boolean jika aksioma-aksioma berikut berlaku untuk setiap elemen a, b, c dari himpunan B .

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. $a * a' = 0$

Definisi 2.1.2 (Johsonbaugh, 2001)

Ekspresi Boolean pada simbol x_1, \dots, x_n didefinisikan secara berulang sebagai $0, 1, x_1, \dots, x_n$ 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 simbol x_1, \dots, x_n , seringkali ditulis $X = X(x_1, \dots, x_n)$.

Teorema 2.1.1

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

1. *Hukum Idempoten* (Johsonbaugh,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. $a + (a' * b) = a + b$
 - b. $a' * (a + b') = (a + b)'$ (Vesely dkk, 2009)

2.2 Prinsip Dualitas

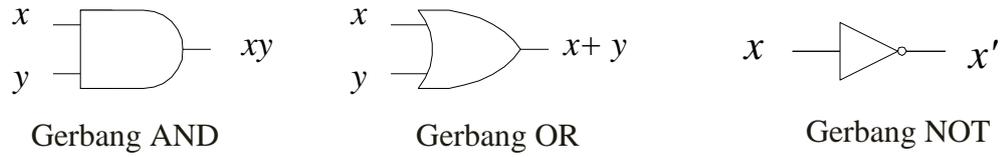
Definisi 2.2.1

Misalkan B adalah suatu ekspresi boolean, suatu ekspresi boolean A dikatakan sebagai dual dari B jika pernyataan A diperoleh dengan cara mengganti $*$ dengan $+$, $+$ dengan $*$ 0 dengan 1 dan 1 dengan 0 dan membiarkan operator komplemen tetap apa adanya. Jika A disebut sebagai dual dari B maka dapat ditulis $A = B^d$.

2.3 Gerbang Logika

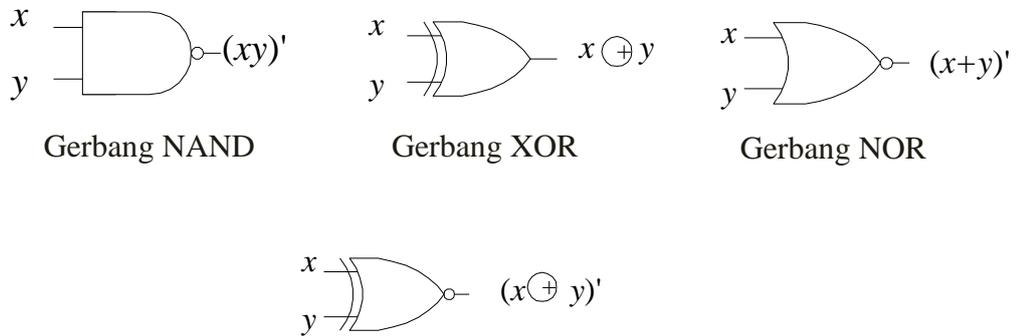
Gerbang logika digambarkan sebagai mesin yang memuat satu input atau lebih dan tepat satu output. Ada dua jenis gerbang logika pada aljabar Boolean yaitu gerbang logika dasar yang terdiri dari AND, OR, dan NOT serta gerbang logika kombinasi yang merupakan turunan dari gerbang logika dasar yang terdiri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

Gerbang Logika Dasar



Gambar 2.3.1 Gerbang Logika Dasar

Kombinasi Gerbang Logika



Gambar 2.3.2 Kombinasi Gerbang Logika

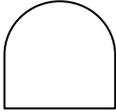
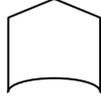
Tabel 1. Tabel Kebenaran Gerbang Logika Dasar dan Kombinasi Gerbang Logika

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

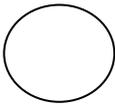
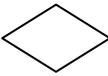
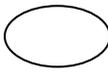
2.4 Fault Tree Analysis

Fault tree menunjukkan hubungan antara kegagalan pada sistem dan penyebab kegagalan pada sistem. Suatu *fault tree* disesuaikan dengan kejadian puncak yang merupakan kegagalan dari suatu sistem. Kejadian puncak ini membentuk cabang berupa kejadian-kejadian yang memberikan kontribusi untuk terjadinya kejadian puncak. Cabang ini akan berhenti jika semua kejadian-kejadian yang menuju kejadian puncak sudah lengkap.

Pada *fault tree* terdapat simbol-simbol yang menunjukkan hubungan dari suatu kejadian yang diperlukan untuk terjadinya kejadian puncak. Simbol-simbol yang mendeskripsikan hubungan, yaitu :

1.  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.
2.  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 simbol Boolean “ + .” Untuk n 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).

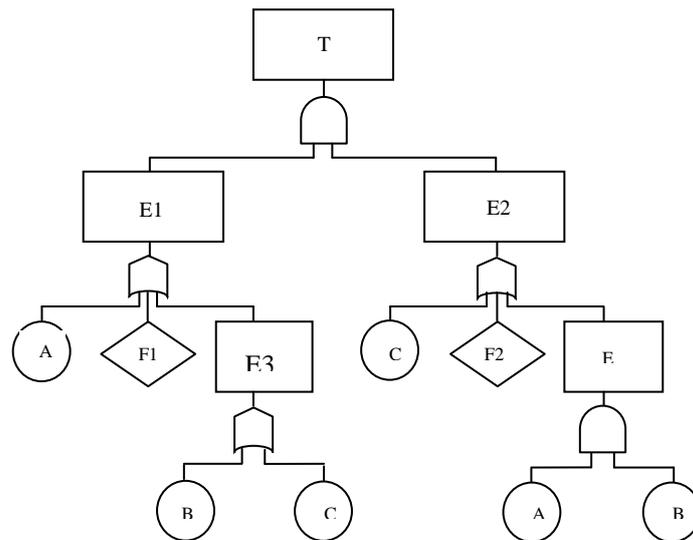
Gerbang AND adalah gerbang yang menggambarkan irisan dari kejadian-kejadian. Gerbang AND ekuivalen dengan simbol Boolean “ * .” Untuk n kejadian-kejadian masukan pada gerbang AND ekuivalen dengan ekspresi Boolean $T = A_1 * A_2 * \dots * A_n$ untuk T merupakan kejadian output dan A_1, A_2, \dots, A_n merupakan kejadian-kejadian masukan. Pada *fault tree* “0” diartikan sebagai kejadian gagal yang tidak terjadi dan “1” diartikan sebagai kejadian gagal yang terjadi.

3. HASIL DAN PEMBAHASAN

3.1 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* gambar 3.1.1



Gambar 3.1.1 Fault Tree (1)

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

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

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

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

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

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

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

$$\Leftrightarrow T = (A * C) + (A * F2) + (A * (A * B)) + (F1 * C) + (F1 * F2) + (F1 * (A * B)) + (B * C) + (B * F2) + (B * (A * B)) + (C * C) + (C * F2) + (C * (A * B)) \quad (6)$$

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

$$T = (A * F2) + (A * B) + (F1 * F2) + (B * F2) + C \quad (7)$$

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 $M1, M4$ dan $M5$ 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

$$T = (A * B) + C \quad (8)$$

Konstruksi *fault tree* untuk persamaan (8) dapat dilihat pada gambar 3.1.2.

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* selanjutnya mengomplemenkan T maka didapat

$$T' = ((A * B) + C)'$$

$$\Leftrightarrow T' = (A * B)' * C' \quad (9)$$

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

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

Berdasarkan definisi 2b untuk persamaan (10) maka didapat

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

Konstruksi fault tree untuk persamaan (11) dapat dilihat pada gambar 3.1.3

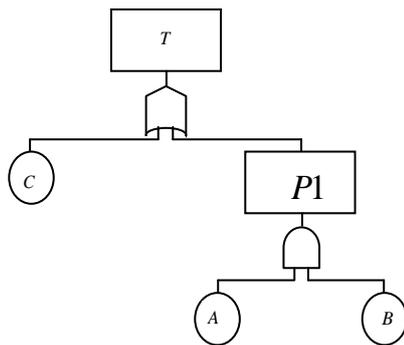
Pada persamaan (11) diperoleh dua *minimal path set* yaitu :

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

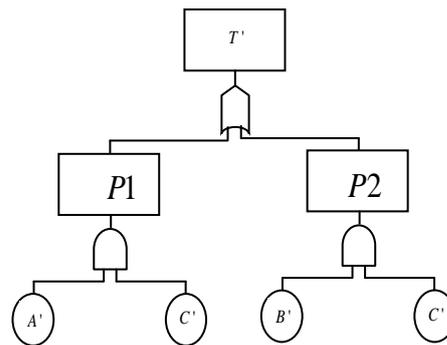
Bentuk umum dari minimal path set untuk suatu kejadian puncak adalah

$$T' = P_1 + P_2 + P_3 + \dots + P_m$$

Dimana T' merupakan kejadian puncak (kegagalan) yang tidak terjadi sedangkan $P_1, P_2, P_3, \dots, P_m$ 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 3.1.2 Fault Tree dari persamaan (8)



Gambar 3.1.3 Fault Tree dari persamaan (11)

3.2 Aplikasi Fault Tree pada suatu contoh kasus.

Pada suatu perjalanan Banjarmasin-Bukittinggi menggunakan kapal dan bis, waktu tibanya terkadang tidak bisa diprediksi. Jika tibanya malam hari maka masalah yang akan dihadapi adalah susah menemukan angkutan umum akibatnya jalan kaki ke rumah. Sehingga harapannya, ketika sampai di Bukittinggi tidak saat malam hari. Untuk membuat perencanaan supaya sampai ke Bukittinggi tidak malam hari maka dapat dilakukan dengan menganalisa apa saja yang bisa menghambat atau memperlambat dan bisa memperlancar dalam perjalanan.

Tahap I. Menentukan Kejadian Puncak.

Kejadian puncak pada masalah ini tiba di Bukittinggi malam hari.

Tahap II. Menentukan Penyebab Kejadian.

Faktor-faktor yang bisa memperlambat perjalanan Banjarmasin-Bukittinggi adalah:

- | | |
|--|--|
| a. Kapal terlambat sampai di pelabuhan Trisakti (E3). | m. Terlambat sampai di Bakauheuni (E7). |
| b. Antrian penumpang yang panjang di Trisakti (C). | n. Macet di beberapa kota di Sumatera (E8). |
| c. Kapal kesusahan melewati muara barito (B). | o. Persinggahan bis yang lama (E5). |
| d. Kapal terlambat sampai di pelabuhan Semarang (E1). | p. Melewati pasar (D). |
| e. Terlambat sampai di Terminal Pulo Gadung (E6). | q. Melewati pusat kota (E). |
| f. Terlambat mendapatkan bis malam tujuan Pulo Gadung (J). | r. Ban kebocoran (F). |
| g. Turun dari kapal memerlukan waktu yang lama (K). | s. Menunggu penumpang pada saat pemberhentian (G). |
| h. Terlambat mendapatkan tiket bisa Lampung Jaya/Gumerang Jaya untuk jam keberangkatan sebelum jam 12 (L). | t. Pergantian bis yang lama di hampir akhir perjalanan (E9). |
| i. Kemacetan sepanjang perjalanan Jakarta-Merak (M). | u. Bis yang ditumpangi tidak hanya tujuan bukittinggi tapi juga kota lain di Sumatera Barat (H). |
| j. Terlambat sampai di pelabuhan penyeberangan di Merak (E10). | v. Negosiasi supir bis sebelum pergantian bis (I). |
| k. Antrian penyeberangan di Merak (N). | w. Kapal terlambat dari pelabuhan sebelum Banjarmasin (A1) |
| l. Lamanya waktu penyeberangan merak-bakauheuni (O). | x. Bis terlambat sampai Bukittinggi (E2). |
| | y. Kapal terlambat berangkat dari Trisakti (E4). |

Tahap III. Menentukan level setiap kejadian.

Pada tahap ini, perjalanan dipisah menjadi dua tahap perjalanan. Pertama menggunakan kapal dan kedua menggunakan bis.

Tahap IV. Menentukan simbol hubungan kejadian yang mau dipakai.

Tahap V. Mengonstruksi *Fault Tree*

Konstruksi *fault tree* dapat dilihat pada gambar 3.2.1.

Tahap VI. Evaluasi *Fault Tree*

$T = E1 + E2$	$E4 = C + B$	$E8 = D * E$
$E1 = E3 + E4$	$E5 = E8 + E9 + F + G$	$E9 = H * I$
$E2 = E5 + E6 + E7$	$E6 = J + K$	$E10 = L + M$
$E3 = A1 + B$	$E7 = E10 * O * N$	

Mensubtitusi $T, E1, E2, \dots, E10$ dan dengan aturan Boolean maka didapat persamaan kejadian puncak sebagai berikut

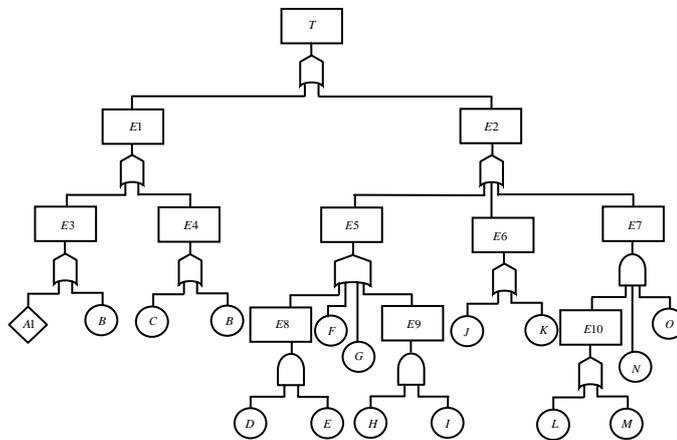
$$T = A1 + B + C + (D * E) + (H * I) + F + G + J + K + (O * N * L) + (O * N * M) \tag{12}$$

Sehingga didapat sebelas *minimal cut set* sebagai berikut:

- | | | |
|------------------|----------------------|----------------------|
| a. $M_1 = A$ | e. $M_5 = H * I$ | i. $M_9 = O * N * M$ |
| b. $M_2 = B$ | f. $M_6 = F$ | j. $M_{10} = J$ |
| c. $M_3 = C$ | g. $M_7 = G$ | k. $M_{11} = K$ |
| d. $M_4 = D * E$ | h. $M_8 = O * N * L$ | |

Jika yang diperhatikan dalam analisa ini hanya penyebab utamanya maka persamaan (12) 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 A yaitu M_1 dapat dihilangkan, persamaannya dapat dituliskan menjadi $T = B + C + (D * E) + (H * I) + F + G + J + K + (O * N * L) + (O * N * M)$ (13)

Konstruksi fault tree dari persamaan di atas dapat dilihat pada gambar 3.2.2

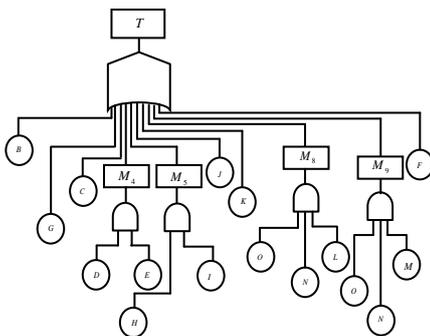


Gambar 3.2.1 Fault Tree (2)

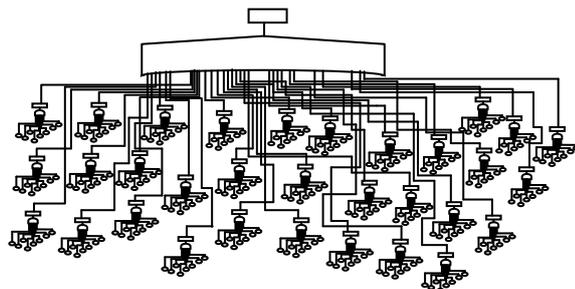
Pengonstruksian *minimal path set* didapat dengan mengomplemenkan persamaan (13) dan *fault tree* dari *minimal path set* ini dapat dilihat pada gambar 3.2.3

Tahap VII Memberikan Alternatif dan Rekomendasi Terhadap Permasalahan

Alternatif dan rekomendasi solusi dapat diberikan dengan mengacu pada *minimal cut set* atau *minimal path set*.



Gambar 3.2.2 Fault tree dari persamaan (13)



Gambar 3.2.3 Fault tree dari komplemen persamaan (13)

4. DAFTAR PUSTAKA

- [1] Johnsonbaugh, Richard.2001.*Discrete Mathematics Fifth Edition*. Prentice Hall. New Jersey.
- [2] Lipschutz, Seymour & Marc Lars Lipson. 2001. *Seri Penyelesaian Soal Schaum Jilid 1 Matematika Diskrit*.Salemba Teknika.Jakarta.
- [3] Musa.2008.*Gerbang-gerbang Sistem Digital*.
http://p_musa.staff.gunadarma.ac.id/Downloads/files/4305/02+-+Gerbang-gerbang+sistem+digital.pdf.
Diakses tanggal 11 Juni 2008.
- [4] The Texas Departmen of Insurance .2008.*Fault Tree Analysis*.
<http://www.tdi.state.tx.us/pubs/videoresource/stpfaulttree.pdf>.
Diakses tanggal 21 November 2008.
- [5] University of Iowa.2009. *Chapter 12: Boolean Algebra And Combinatorial Circuits Discrete Mathematical Structures: Theory And Applications*.
http://www.cs.uiowa.edu/~icurto/discret-ppt/Chapter_12.ppt.
Diakses tanggal 20 Februari 2009.
- [6] Vesely dkk. 2009. *Fault Tree Handbook*. <http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/sr0492.pdf>.
Diakses tanggal 29 Januari 2009.