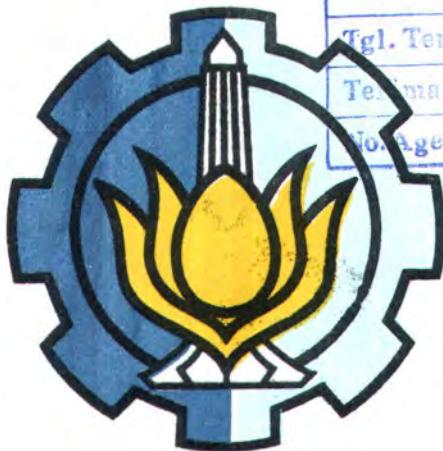


13460/H/01

TUGAS AKHIR
(OE 1701)

**PENGEMBANGAN PERANGKAT LUNAK ANALISA
KEANDALAN SISTEM DENGAN MENGGUNAKAN
METODE SIMULASI MONTE CARLO**

RSKe
005.1
Waw
P-1
1999



PERPUSTAKAAN ITS	
Tgl. Terima	6 - 7 - 2000
Tema Z... I	17
No. Agenda Pkp.	21.796

Oleh :

Wasis Prabowo Wawan
4392.100.032

**JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1999**



PENGEMBANGAN PERANGKAT LUNAK ANALISA
KEANDALAN SISTEM DENGAN MENGGUNAKAN
METODE SIMULASI MONTE CARLO

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Untuk Menyelesaikan Studi Program Sarjana
Pada
Jurusan Teknik Kelautan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya

Surabaya, Februari 1999

Mengetahui/Menyetujui :

Dosen Pembimbing I

210399



Dosen Pembimbing II

Ir. Daniel M. Rosyid PhD JURUSAN TEKNIK KELAUTAN Ir. Eko Budi Djatmiko, Msc. PhD
NIP. : 131 782 038 NIP. : 131 407 592



ABSTRAK

Tugas Akhir ini membahas tentang pengembangan perangkat lunak analisa keandalan sistem dengan menggunakan metode *Simulasi Monte Carlo*. Bahasa pemrograman yang digunakan adalah *Borland Delphi 2.0* yang berbasis pada bahasa pemrograman *Pascal*.

Sebagai contoh penerapan metode Simulasi Monte Carlo diambil 2 kasus sistem struktur, yaitu kasus *tubular joint (Y joint)* dan kasus pembebanan pada sebuah balok. Peninjauan contoh-contoh kasus tersebut di atas sekaligus sebagai validasi program.

Untuk meninjau sampai sejauh mana akurasi hasil yang didapat dengan metode *Simulasi Monte Carlo* ini, maka dilakukan perbandingan dengan metode *Advance First Order Second Moment* (AFOSM). Dari perbandingan 2 (dua) metode tersebut didapatkan perbedaan hasil yang sangat kecil.

KATA PENGANTAR

Puji syukur Alhamdulillah kepada Yang Maha Tunggal, Allah SWT, atas segala rahmat, hidayat dan kekuatan yang telah diberikan kepada penulis. Hanya atas izin dan kehendak-Nyalah Tugas Akhir ini dapat terselesaikan meskipun masih jauh dari kesempurnaan.

Tugas Akhir berjudul Pengembangan Perangkat Lunak Analisa Keandalan Sistem Dengan Metode Simulasi Monte Carlo penulis selesaikan sebagai salah satu syarat untuk menyelesaikan studi di Jurusan Teknik Kelautan, FTK - ITS. Selain itu diharapkan Tugas Akhir ini mampu memberikan manfaat bagi pembaca.

Akhirnya penulis menyadari sebagai manusia biasa yang tak lepas dari kesalahan, mengharapkan kritik dan saran yang membangun demi kebaikan bersama. Semoga Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi pembaca serta mencapai tujuan yang diharapkan.

Surabaya, Februari 1999

Penulis

UCAPAN TERIMA-KASIH

Penulis ingin mengucapkan segala rasa terima-kasih yang sedalam-dalamnya dan penghargaan yang setulus-tulusnya kepada berbagai pihak yang telah memberikan dukungan, dorongan, bantuan, dan bimbingan atas selesainya penggerjaan Tugas Akhir ini, terutama kepada :

1. Bapak, Ibuk, mas Yosi dan mbak Utik, atas cinta kasih yang tulus kepada penulis, serta doa-doa yang selalu mengalir.
2. Bapak Ir. Daniel Muhammad Rosyid PhD dan Ir. Eko Budi Djatmiko MSc, PhD selaku dosen pembimbing yang dengan segala kesabaran, ketulusan dan keikhlasannya membimbing penulis selama penyelesaian Tugas Akhir ini.
3. Bapak Ir. Eko Budi Djatmiko MSc, PhD selaku Ketua Jurusan Teknik Kelautan beserta staf pengajar dan staf administrasi Teknik Kelautan.
4. Oom Yo, Desi Adek, Dwi ngeDhot, Rofiq King of Delphi, Erry Gonzales Blanco, Ash-Sidiqi dan Edi Djoko, atas sumbangan ilmu dan sarannya.
5. Teman-teman L'92 Adi pak 'De, Agus Bali, Bhima, Thole, Dhottil, India, Ranu, Yuk ... Kopi, dan Menjes, atas kebersamaannya.
6. Teman-teman SMA cak 'Ing, Dodok dan Agam, atas kesetiakawanannya.
7. Teman-teman L'93 Buto Adek, Giri, dan lain-lain, atas kesediaannya menemani bermain kartu semasa kuliah.
8. Siapa saja yang tidak dapat penulis sebutkan di sini tanpa mengurangi rasa hormat.

Atas segala kebaikan dan bantuannya penulis panjatkan doa, semoga Allah berkenan membalas segala kebaikan yang penulis terima dengan kebaikan yang berlipat ganda dan semoga bantuannya mendapat nilai di sisi-Nya.

DAFTAR ISI

Lembar Pengesahan	i
Abstrak	ii
Kata Pengantar	iii
Ucapan Terima Kasih	iv
Daftar Isi	v
Daftar Notasi	vii
Daftar Gambar	viii

BAB I PENDAHULUAN

1 . 1 Latar Belakang Masalah	I - 1
1 . 2 Perumusan Masalah	I - 3
1 . 3 Tujuan	I - 3
1 . 4 Manfaat	I - 4
1 . 5 Metodologi	I - 4
1 . 6 Sistematika Penulisan	I - 4

BAB II DASAR TEORI

2 . 1 Fungsi Kerapatan Peluang	II - 1
2 . 2 Konsep Keandalan	II - 9
2 . 3 Fungsi Kinerja Sistem	II - 12
2 . 4 Simulasi Monte Carlo	II - 12

BAB III PROGRAM SIMULASI MONTE CARLO

3 . 1 Program Simulasi Monte Carlo	III - 1
3 . 2 Langkah Penggunaan Program	III - 5

BAB IV ANALISA KEANDALAN SISTEM DENGAN SIMULASI MONTE CARLO

4 . 1 Implementasi Simulasi Monte Carlo	IV - 1
-----------------------------------------	--------

4 . 2	Validasi Program	IV - 4
4 . 3	Diskusi	IV - 8

BAB V KESIMPULAN

Daftar Pustaka

Lampiran A

Lampiran B

Lampiran C

DAFTAR NOTASI

a_p = bilangan random

dx = interval x

FK = fungsi kinerja sistem

$f(x)$ = fungsi kerapatan peluang

K = keandalan

\bar{L} = rata-rata besaran kekuatan

n = jumlah kegagalan

N = jumlah perulangan

Pg = peluang kegagalan

\bar{S} = rata-rata besaran kekuatan

x = variabel yang ditinjau

x_p = hasil transformasi bilangan random

β = shape parameter

γ = location parameter

η = scale parameter

λ = scale parameter

μ_x = rata-rata

$\mu_{x'}$ = rata-rata dari logaritma natural x

σ_x = standart deviasi

$\sigma_{x'}$ = standart deviasi dari logaritma natural x

DAFTAR GAMBAR

Gambar 2.1	Histogram : Kelompok Kejadian terhadap Jumlah Kejadian	II - 1
Gambar 2.2	Fungsi Kerapatan Peluang : Kejadian terhadap Peluang Kejadian	II - 2
Gambar 2.3	Distribusi Normal	II - 3
Gambar 2.4	Distribusi Rayleigh	II - 4
Gambar 2.5	Distribusi Exponential	II - 5
Gambar 2.6	Distribusi Weibull	II - 6
Gambar 2.7	Distribusi Log-Normal	II - 7
Gambar 2.8	Distribusi Extreme Value Minima	II - 8
Gambar 2.9	Distribusi Extreme Value Maxima	II - 9
Gambar 2.10	Diagram Interferensi : kedua fkp terpisah sama sekali	II - 10
Gambar 2.11	Diagram Interferensi : kedua fkp mempunyai interferensi yang cukup berarti	II - 10
Gambar 2.12	Flowchart Transformasi $x_p \leftarrow a_p$	II - 14
Gambar 2.13	Flowchart Simulasi Monte Carlo	II - 15
Gambar 3.1	Flowchart Program Simulasi Monte Carlo	III - 4
Gambar 3.2	Flowchart penggunaan Program Simulasi Monte Carlo	III - 6
Gambar 4.1	Tubular Joint : Y Joint	IV - 1
Gambar 4.2	Pembebanan pada balok	IV - 3

BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

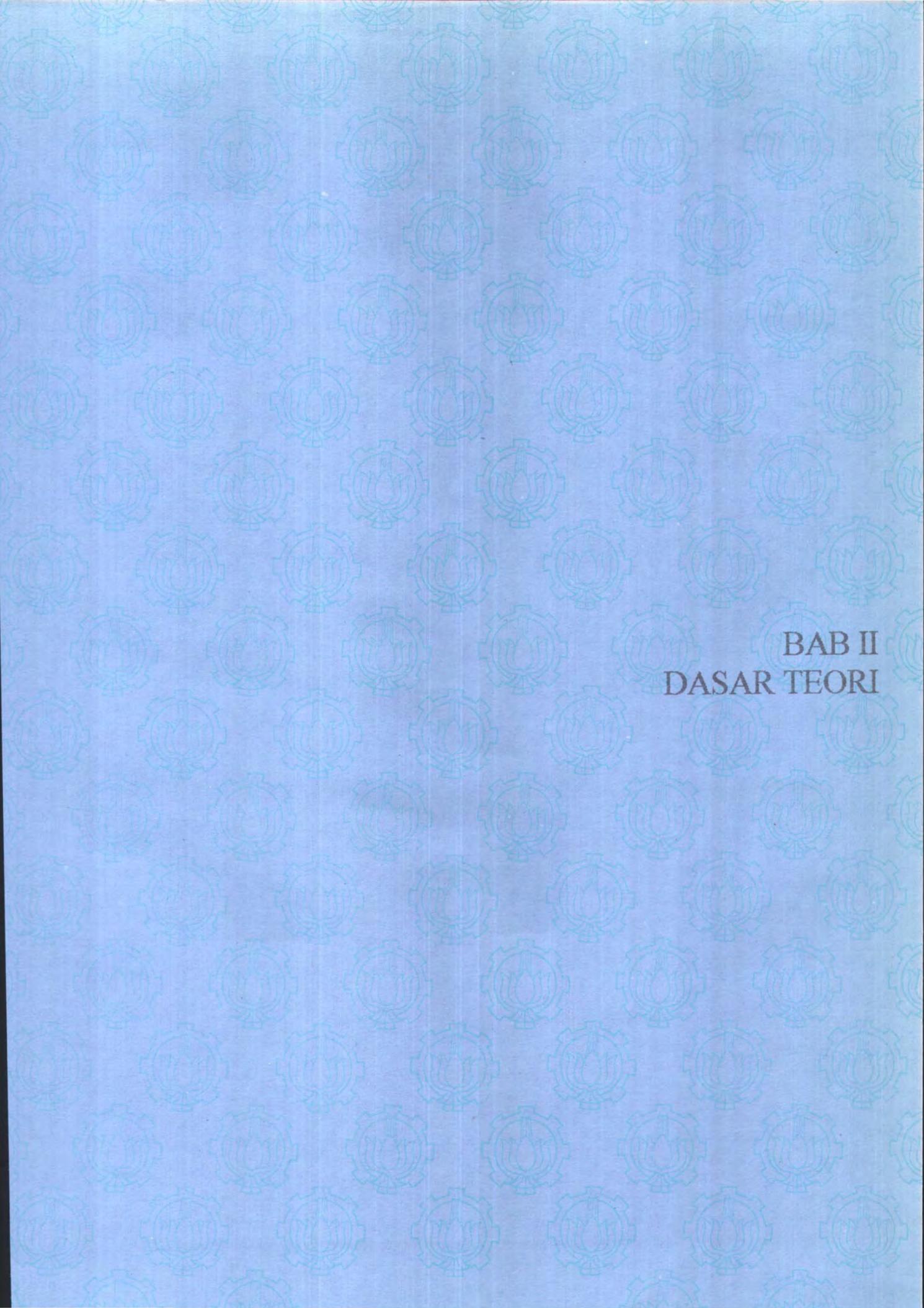
Sekalipun banyak gejala alam dalam bidang rekayasa dapat dijelaskan dengan keyakinan atau kepastian yang tinggi, dalam praktek ternyata sering dijumpai adanya variabilitas pada besaran-besaran yang mengatur kejadian tersebut dengan berbagai tingkat, sehingga muncul ketidakpastian [Rosyid, D. M (1996)]. Ketidakpastian tersebut dapat dikelompokkan menjadi 3 (tiga) jenis, yaitu :

1. Ketidakpastian fisik

Ketidakpastian fisik disebabkan oleh variabilitas pada besaran-besaran fisik suatu sistem. Misalnya, dalam bidang rekayasa, besaran beban, besaran kekuatan, besaran dimensi seperti lebar, tebal, diameter, dan sebagainya. Karakteristik ketidakpastian fisik dapat dikurangi dengan penerapan *Quality Control*. *Quality Control* adalah suatu teknik yang diterapkan untuk mengendalikan mutu suatu produk.

2. Ketidakpastian statistik

Ketidakpastian statistik disebabkan oleh perbedaan jumlah sample yang diambil dalam pengkajian sistem rekayasa. Misalnya, dalam bidang rekayasa, dilakukan pengujian kuat tarik suatu jenis logam dengan sample sebanyak 5 buah. Hasil yang didapat tentunya berbeda dengan apabila sample yang diambil sebanyak



BAB II

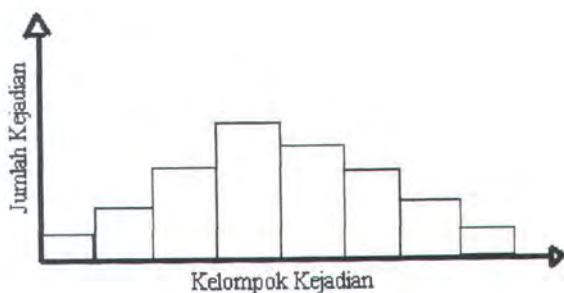
DASAR TEORI

BAB II

DASAR TEORI

2.1 FUNGSI KERAPATAN PELUANG

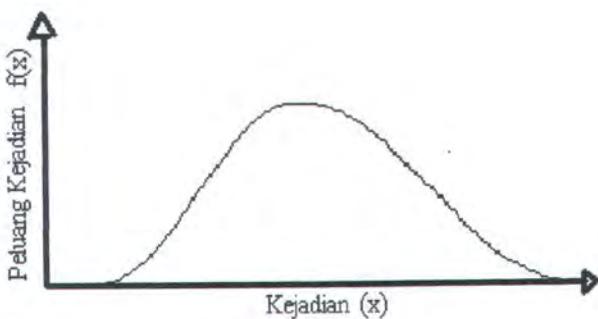
Konsep peluang adalah suatu konsep yang sangat penting dalam rekayasa keandalan. Peluang, secara mudah dapat diartikan sebagai kesempatan. Makna khusus peluang adalah bahwa sebuah kejadian tidak dapat diramalkan dengan pasti [Rosyid, D. M (1996)]. Peluang sebuah kejadian dapat dianggap sebagai besarnya kesempatan kejadian tersebut timbul. Dalam kerangka Teori Himpunan, seluruh luasan dalam semesta S mencerminkan peluang semua kejadian-kejadian yang mungkin terjadi dalam konteks tertentu. Peluang sebuah kejadian tunggal atau majemuk dalam keseluruhan sample space di S , dicerminkan oleh luasan himpunan yang beranggotakan kejadian-kejadian yang dimaksud dalam semesta S . Hal ini dapat digambarkan secara jelas dengan menggunakan sebuah *Histogram*.



Gambar 2.1- Histogram : Kelompok Kejadian terhadap Jumlah Kejadian

Sumbu absis pada histogram tersebut menunjukkan kelompok kejadian tertentu, dan sumbu ordinat menunjukkan jumlah terjadinya kejadian yang

dimaksud pada sumbu absis. Selanjutnya histogram jumlah kejadian dapat diubah menjadi histogram frekuensi relatif dengan cara membagi jumlah kejadian pada tiap-tiap kelompok kejadian dengan jumlah kejadian total. Jika interval kelompok kejadian pada sumbu absis diperkecil hingga menjadi sangat kecil, maka akan didapat sebuah kurva yang halus. Kurva inilah yang biasa disebut sebagai *Fungsi Kerapatan Peluang (f_{kp})* atau *Probability Density Function (pdf)*, seperti yang terlihat pada gambar berikut :



Gambar 2.2- Fungsi Kerapatan Peluang : Kejadian terhadap Peluang Kejadian

Pada pengamatan di lapangan, terdapat banyak fkp yang mengikuti bentuk suatu jenis distribusi. Beberapa jenis distribusi yang umum dijumpai di lapangan antara lain :

- Distribusi Normal

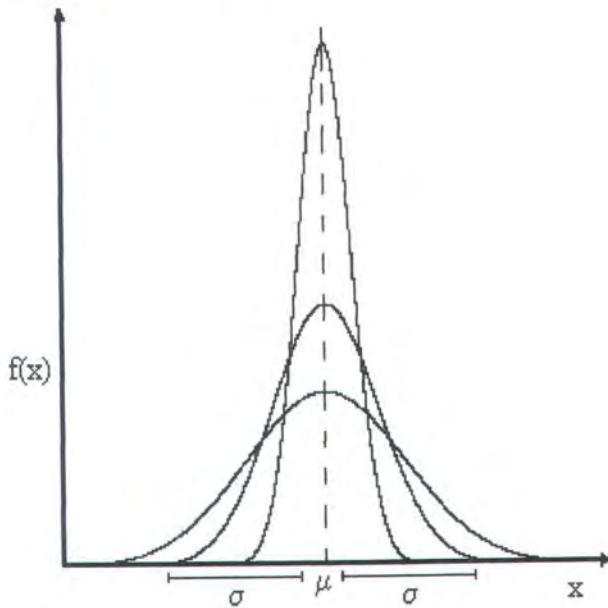
Dalam bidang rekayasa bentuk distribusi Normal merupakan bentuk distribusi yang paling umum dijumpai. Bila x adalah variabel yang ditinjau, maka dalam mengidentifikasi perilaku distribusi variabel suatu sistem dapat ditunjukkan dengan dengan persamaan :

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma_x}\right)^2} \quad (2.1)$$

dimana : $f(x) \geq 0, -\infty < x < \infty, -\infty < \mu < \infty, \sigma_x > 0,$

μ = rata-rata x ,

σ_x = standart deviasi x .



Gambar 2.3- Distribusi Normal

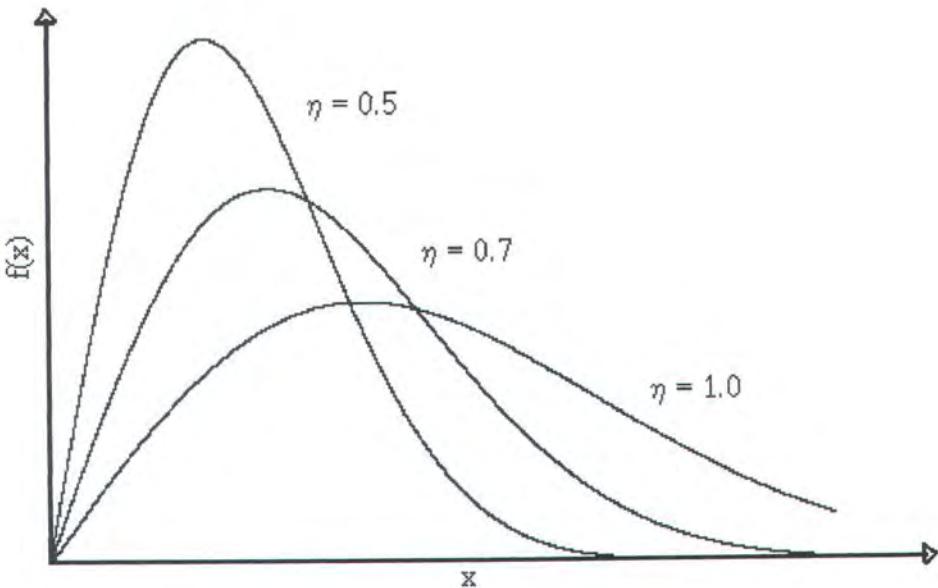
- Distribusi Rayleigh

Distribusi Rayleigh merupakan suatu kasus khusus dari distribusi Weibull, yaitu apabila $\beta = 2$ dan $\eta' = \sqrt{2\eta}$, dimana η' adalah scale parameter distribusi Weibull dan η adalah parameter distribusi Rayleigh. Distribusi ini mempunyai bentuk persamaan :

$$f(x) = \frac{x}{\eta^2} e^{-\frac{1}{2}\left(\frac{x}{\eta}\right)^2} \quad (2.2)$$

dimana : $x \geq 0, \eta \geq 0,$

η = scale parameter.



Gambar 2.4- Distribusi Rayleigh

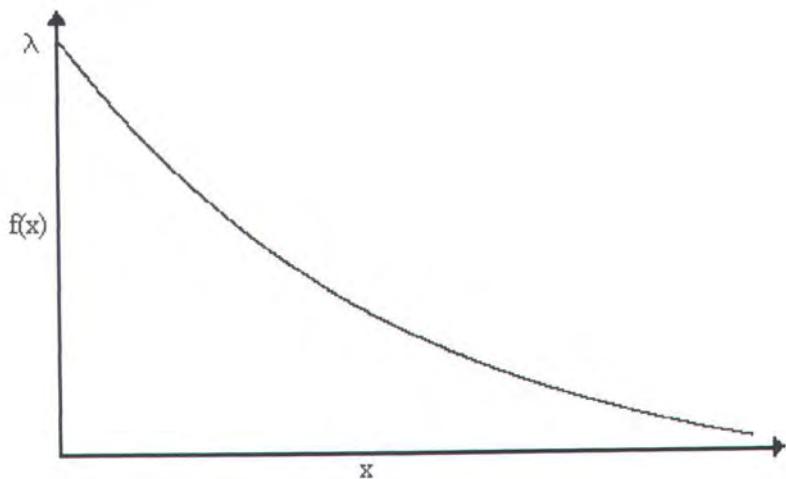
- Distribusi Exponential

Distribusi Exponential sering digunakan untuk menggambarkan laju kegagalan suatu sistem. Distribusi ini menggambarkan bahwa semakin lama waktu yang ditinjau, semakin sedikit pula sistem yang mengalami kegagalan. Distribusi ini mempunyai bentuk persamaan :

$$f(x) = \lambda e^{-\lambda(x-\gamma)} \quad (2.3)$$

dimana : λ = scale parameter,

γ = location parameter.



Gambar 2.5- Distribusi Exponential

- Distribusi Weibull

Distribusi Weibull juga merupakan salah satu distribusi yang sering dijumpai dalam bidang rekayasa. Distribusi ini mempunyai bentuk persamaan :

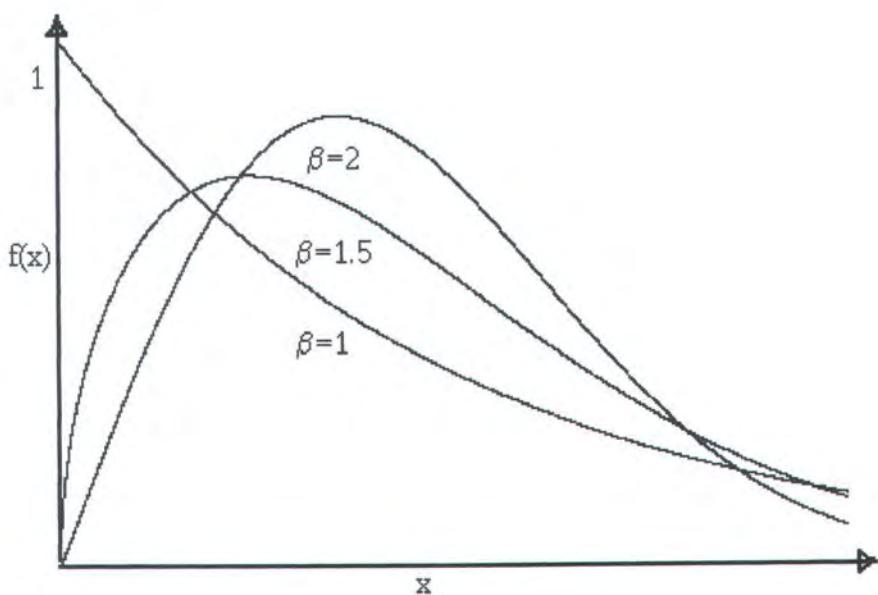
$$f(x) = \frac{\beta}{\eta} \left(\frac{x-\gamma}{\eta} \right)^{\beta-1} e^{-\left(\frac{x-\gamma}{\eta} \right)^\beta} \quad (2.4)$$

dimana : $f(x) \geq 0, x \geq \gamma, \beta \geq 0, \eta \geq 0, -\infty < \gamma < \infty$,

β = shape parameter,

η = scale parameter,

γ = location parameter.



Gambar 2.6- Distribusi Weibull

- Distribusi Log-Normal

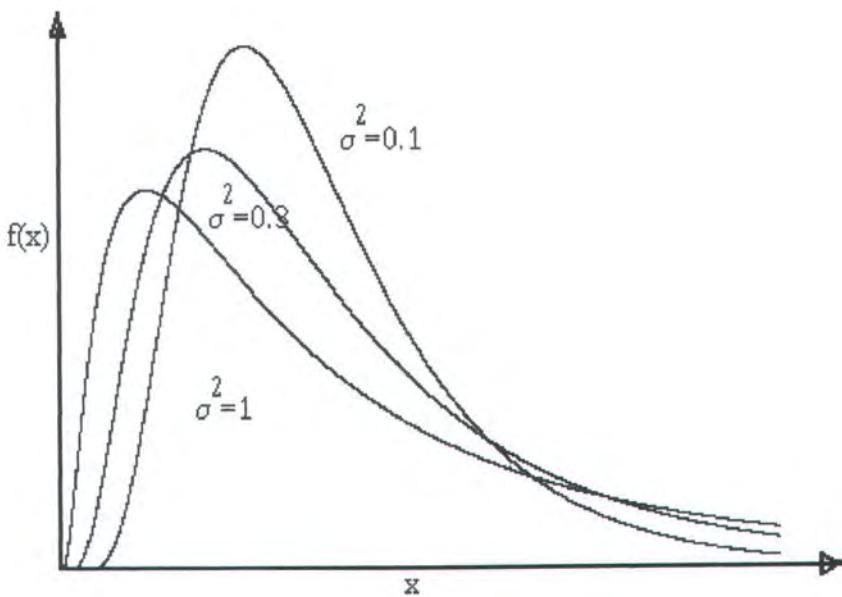
Pada hal-hal tertentu distribusi Log-Normal juga dapat dijumpai. Distribusi ini mempunyai bentuk persamaan :

$$f(x) = \frac{1}{x\sigma_{x'}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x' - \mu_{x'}}{\sigma_{x'}}\right)^2} \quad (2.5)$$

dimana : $f(t) \geq 0, t \geq 0, -\infty < \mu_{x'} < \infty, \sigma_{x'} > 0, x' = \log_e x,$

$\mu_{x'} = \text{rata-rata dari logaritma natural } x,$

$\sigma_{x'} = \text{standart deviasi dari logaritma natural } x.$



Gambar 2.7- Distribusi Log-Normal

- Distribusi Extreme Value Minima

Pada hal-hal tertentu distribusi Extreme Value Minima juga dapat dijumpai.

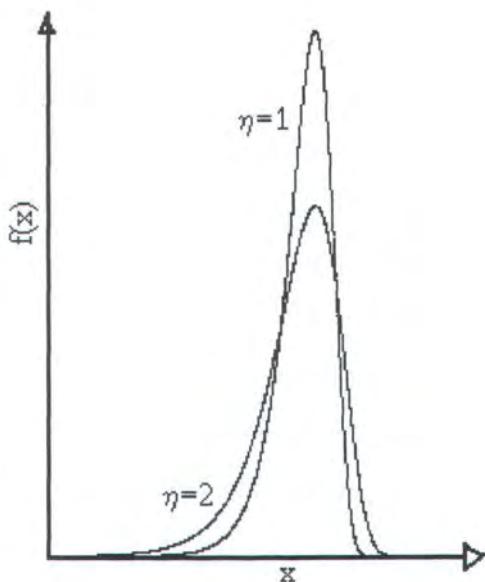
Distribusi ini mempunyai bentuk persamaan :

$$f_{\min}(x) = \frac{1}{\eta_{\min}} e^{\left(\frac{x-\gamma_{\min}}{\eta_{\min}}\right)} - e^{\left(\frac{x-\gamma_{\min}}{\eta_{\min}}\right)} \quad (2.6)$$

dimana : $-\infty < x < \infty, -\infty < \gamma_{\min} < \infty, \eta_{\min} > 0,$

γ_{\min} = location parameter,

η_{\min} = scale parameter.



Gambar 2.8- Distribusi Extreme Value Minima

- Distribusi Extreme Value Maxima

Pada hal-hal tertentu distribusi Extreme Value Maxima juga dapat dijumpai.

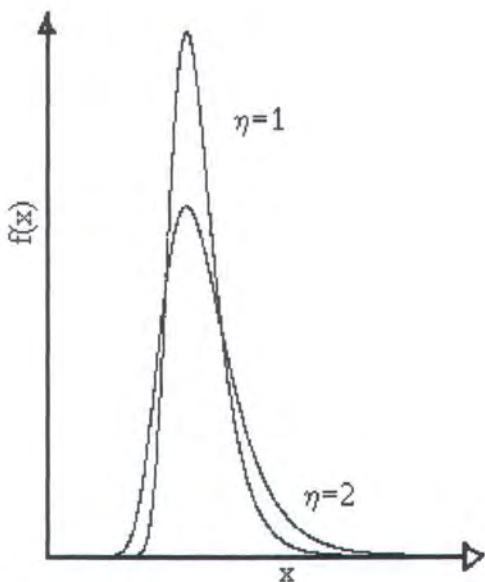
Distribusi ini mempunyai bentuk persamaan :

$$f_{\max}(x) = \frac{1}{\eta_{\max}} e^{-\left(\frac{x-\gamma_{\max}}{\eta_{\max}}\right)} - e^{-\left(\frac{x-\gamma_{\max}}{\eta_{\max}}\right)} \quad (2.7)$$

dimana : $-\infty < x < \infty$, $-\infty < \gamma_{\max} < \infty$, $\eta_{\max} > 0$,

γ_{\max} = location parameter,

η_{\max} = scale parameter.

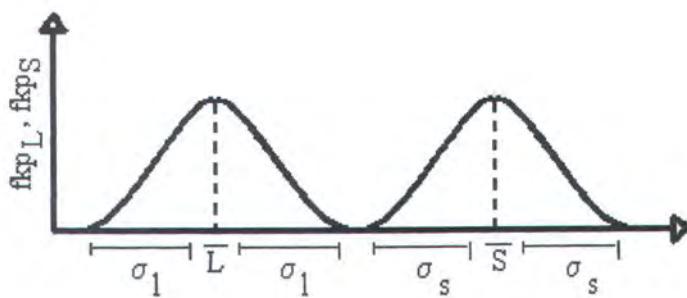


Gambar 2.9- Distribusi Extreme Value Maxima

2.2 KONSEP KEANDALAN

Definisi keandalan sebuah sistem yaitu peluang sistem tersebut untuk memenuhi tugas yang telah ditetapkan tanpa mengalami kegagalan selama kurun waktu tertentu apabila dioperasikan dengan benar dalam lingkungan tertentu [Rosyid, D. M (1996)]. Untuk mengembangkan pemahaman atas persoalan keandalan, perlu dibahas lebih dahulu tentang konsep Diagram Interferensi.

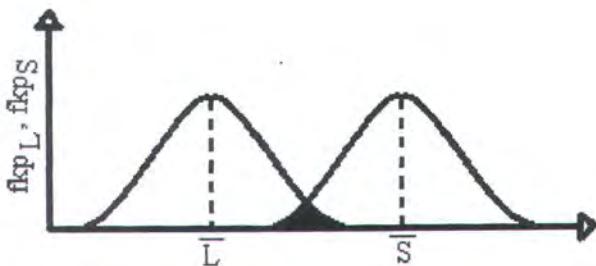
Apabila diasumsikan bahwa sebuah komponen memiliki kekuatan yang terdistribusi mengikuti suatu fungsi kerapatan peluang f_{kpS} , dan efek pembebahan yang juga terdistribusi menurut f_{kpL} , maka mudah dipahami bahwa apabila f_{kpS} terpisah sama sekali dengan f_{kpL} , secara teoritis komponen tersebut memiliki keandalan $K = 1$, dengan kata lain komponen tersebut tidak akan pernah gagal. Hal ini dapat dilihat pada gambar berikut :



Gambar 2.10- Diagram Interferensi : kedua fkp terpisah sama sekali

Pada gambar di atas, \bar{S} adalah harga rata-rata kekuatan komponen S, \bar{L} adalah harga rata-rata efek beban L, σ_s adalah simpangan baku kekuatan komponen S, σ_L adalah simpangan baku efek beban L. Simpangan baku adalah suatu ukuran penyebaran yang sederhana dari sebuah besaran dari harga ratanya.

Apabila kedua fkp memiliki bagian yang berhimpitan, seperti yang ditunjukkan pada gambar berikut, maka peluang terjadinya kegagalan pada komponen tidak lagi nol. Ini ditunjukkan pada daerah yang diarsir, sebagai suatu indikasi bahwa suatu ketika akan terjadi sebuah kejadian di mana efek beban L melebihi kekuatan komponen S, sehingga terjadi kegagalan.



Gambar 2.11- Diagram Interferensi : kedua fkp mempunyai interferensi

yang cukup berarti

Dalam konsep Diagram Interferensi seperti di atas terdapat sebuah parameter baru yang disebut *Margin Keselamatan (Safety Margin)* MK. Margin Keselamatan yaitu suatu batas antara kondisi dimana suatu sistem rekayasa dinyatakan masih dapat beroperasi dan gagal beroperasi. ini dinyatakan sebagai berikut :

$$MK = \frac{S - L}{\sqrt{\sigma_s^2 + \sigma_L^2}} \quad (2.8)$$

Dalam persamaan (2.8) di atas untuk menentukan MK harus memperhitungkan variasi kekuatan sekaligus beban. Variasi pada kekuatan komponen dapat berasal dari gabungan berbagai variasi proses-proses metalurgis dan fabrikasi yang terjadi pada saat pembuatan komponen tersebut, sementara variasi pada efek beban disebabkan karena variasi pada tugas dan lingkungan kerja komponen tersebut.

Asumsi dasar yang dipakai pada Diagram Interferensi adalah bahwa untuk masalah pembebanan berulang fkp yang terbaik adalah tidak berubah. Asumsi ini sahih untuk sistem atau komponen elektronik, sedangkan untuk sistem atau komponen mekanik, di mana kekuatannya menurun akibat pengausan, korosi, dan kelelahan, asumsi ini tidak sepenuhnya sahih. Dengan demikian, pada kondisi sebenarnya fkps untuk komponen mekanik akan bergerak ke kiri dalam Diagram Interferensi, sehingga akan menurunkan keandalannya.

2.3 FUNGSI KINERJA SISTEM

Fungsi Kinerja Sistem (FK) adalah sebuah fungsi yang menyatakan Moda Kegagalan yang akan diperkirakan peluang terjadinya dalam perubah-perubah (x) yang mengatur Moda Kegagalan tersebut [Rosyid, D. M (1996)]. Dari sekian perubah ini, sebagian tidak dapat disebut sebagai perubah acak. Hanya beberapa saja dari perubah-perubah tersebut yang memiliki ketidakpastian yang cukup besar sehingga wajar jika diperlakukan sebagai perubah acak dengan fkp tertentu. Perubah-perubah ini selanjutnya disebut *perubah dasar (basic variable)* dalam fungsi kinerja sistem. Secara matematis, fungsi kinerja yang mempunyai sejumlah *basic variable* dapat dituliskan :

$$FK(x) = FK (x_1, x_2, x_3, \dots, x_n) \quad (2.9)$$

Bentuk yang paling sederhana yaitu apabila $FK(x)$ merupakan fungsi linier (x), yaitu :

$$FK(x) = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_n x_n \quad (2.10)$$

di mana a_i , $i = 1, 2, 3, \dots, n$ adalah konstanta, dan x_i , $i = 1, 2, 3, \dots, n$ adalah perubah-perubah dasar dengan fkp yang diketahui.

2.4 SIMULASI MONTE CARLO

Gagasan dasar metode Simulasi Monte Carlo yaitu *sampling numerik* dengan bantuan *Random Number Generator*. Diambil beberapa sample untuk setiap x_i dari sebuah populasi X_i , kemudian dimasukkan ke dalam persamaan fungsi kinerja $FK(x)$, dan harga $FK(x)$ kemudian dihitung. Setiap kali $FK(x) < 0$, dianggap sistem / komponen yang ditinjau gagal [Thoft & Murotsu (1986)]. Misalnya untuk

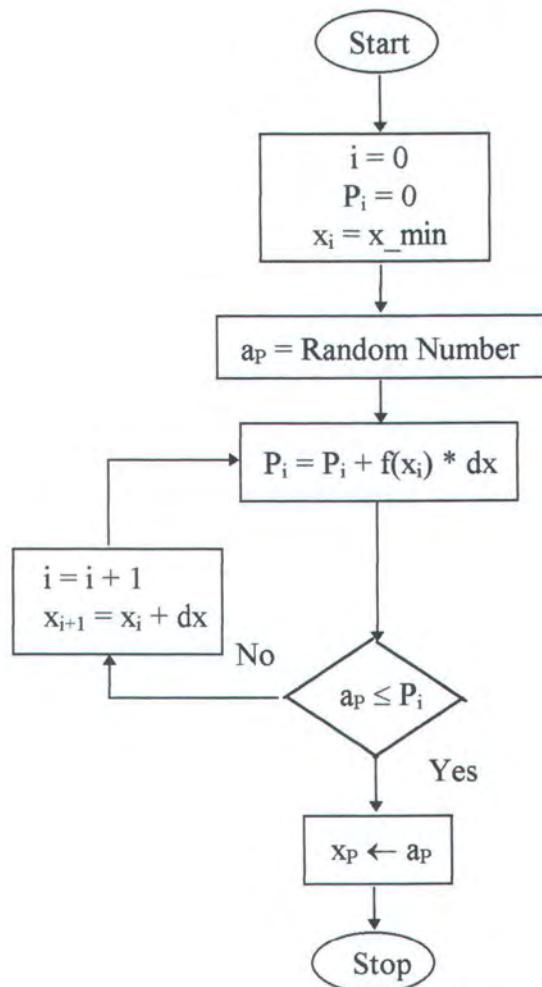
sample sebesar N , didapat hasil $FK(x) < 0$ sebanyak n kali, maka peluang kegagalan sistem / komponen yang ditinjau adalah $P_g = n / N$.

Persoalan utama yaitu bagaimana men-*transformasi*-kan angka acak yang dikeluarkan oleh *random number generator (RNG)* menjadi besaran fisis yang sesuai dengan fkp perubah dasar tersebut. Ini disebabkan karena angka acak yang dikeluarkan oleh RNG memiliki fkp uniform, sedangkan perubah dasar dalam $FK(x)$ seringkali tidak demikian (misalkan terdistribusi secara Normal, Weibull, Rayleigh, dan sebagainya).

Transformasi a_p menjadi x_p dilakukan secara *numerik* dengan prosedur sebagai berikut :

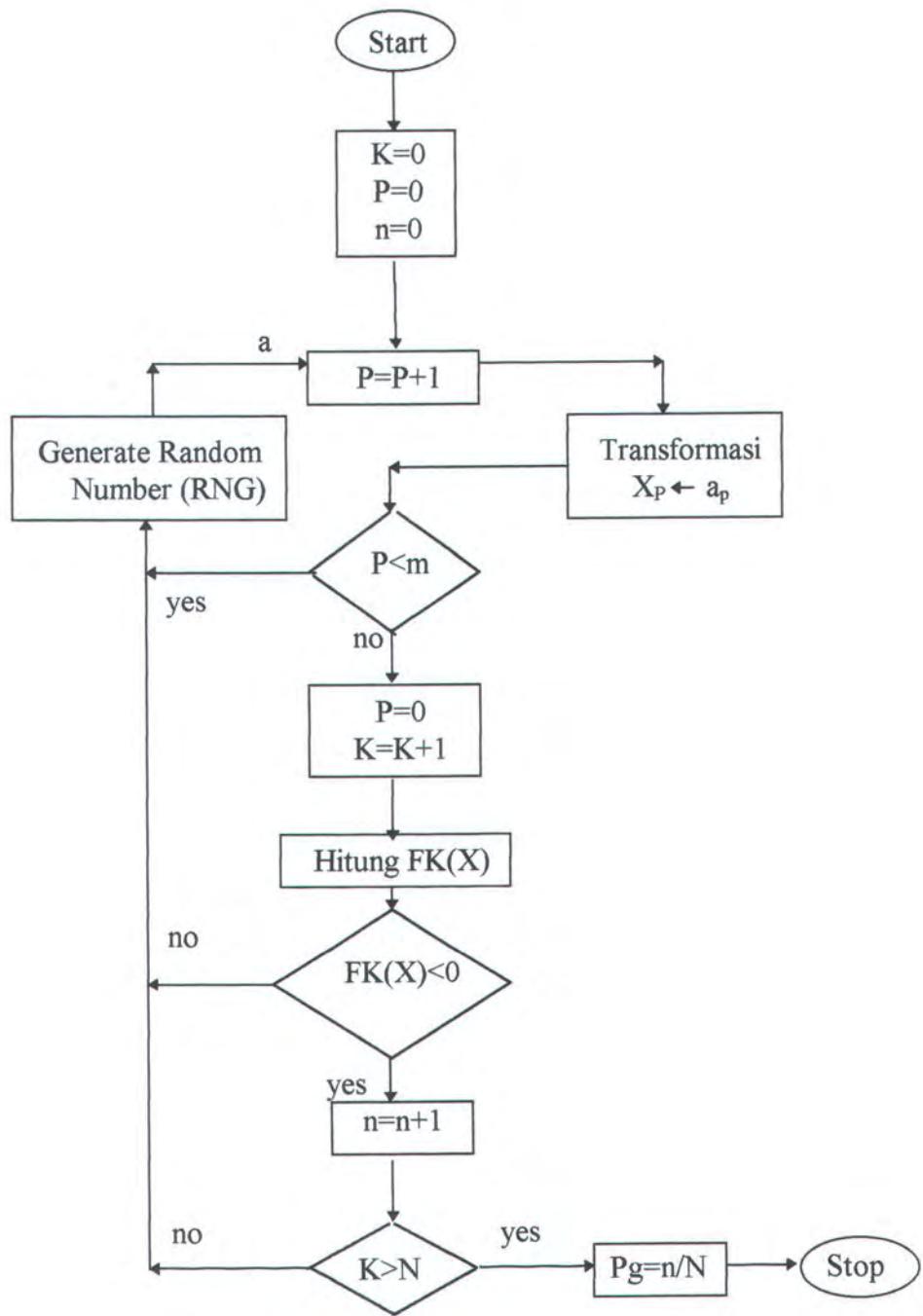
1. Untuk x_p dengan fkp yang diketahui, bagi rentang x_p menjadi i interval yang sama sepanjang dx .
2. Hitung luas tiap pias (ini akan menghasilkan peluang x_p memiliki harga dalam interval I_i , yaitu sebesar P_i) dengan mengalikan interval dx dengan tinggi fkp pada x_i . Untuk setiap a_p yang keluar dari RNG, maka a_p di-*transformasi*-kan sebagai x_p .

Untuk lebih jelasnya, transformasi a_p menjadi x_p dapat dilihat pada *flowchart* di berikut ini :



Gambar 2.12- Flowchart Transformasi $x_p \leftarrow a_p$

Flowchart Simulasi Monte Carlo dapat dilihat pada gambar di bawah ini :



Gambar 2.13- Flowchart Simulasi Monte Carlo

K dalam *flowchart* tersebut adalah *counter* jumlah eksperimen atau sampling, M adalah jumlah perubah dasar dalam $FK(x)$, dan P adalah indeks untuk perubah dasar ke x_p , sedang n adalah penghitung untuk $FK(x) < 0$, dan N adalah jumlah eksperimen maksimum yang akan dilakukan. RNG yang digunakan pada program ini adalah RNG yang telah disediakan oleh Delphi, di mana RNG tersebut akan menghasilkan suatu angka acak a_p antara 0 dan 1 dengan fkp yang mendekati uniform. Pada uji statistik yang telah dilakukan, munculnya angka acak yang sama terjadi pada perulangan ke 10^{13} sampai dengan 10^{14} , sehingga dapat dikatakan bahwa angka acak yang dihasilkan oleh RNG Delphi sudah mendekati uniform.

Angka acak a_p ini selanjutnya ditransformasikan menjadi x_p , dengan $p = 1, 2, 3, \dots, m$. $FK(x)$ kemudian dihitung, dan penghitungan nomor eksperimen dimasukkan dalam K. Apabila $FK(x) \geq 0$, maka eksperimen dilanjutkan, sedang apabila $FK(x) < 0$, maka ini dicatat dan disimpan dalam n. Eksperimen ke K dilanjutkan sampai $K = N$, sesudah itu peluang kegagalan sistem / komponen dihitung sebagai $\frac{n}{N}$.

BAB III

PROGRAM SIMULASI

BAB III

PROGRAM SIMULASI MONTE CARLO

3.1 PROGRAM SIMULASI MONTE CARLO

Program Simulasi Monte Carlo ini berjalan baik dengan konfigurasi *PC-Pentium* berbasis *Windows 95*, *RAM 8 Mb*, *free-space hard-disk 25 Mb (Recommended)*, dan dibuat dengan bahasa *Borland Delphi 2.0 (Desktop Version)*. Program ini mampu menerima input semua bentuk persamaan fungsi kinerja sistem yang telah disesuaikan dengan *format Delphi*, dan menyediakan 7 pilihan jenis fkp / distribusi, yaitu Normal, Rayleigh, Exponential, Weibull, Log-Normal, Extreme Value Minima, dan Extreme Value Maxima, beserta semua parameternya. Apabila pemakai telah memasukkan persamaan fungsi kinerja sistem beserta semua variabelnya secara benar, maka program ini akan memulai proses perhitungan.

Secara garis besar tampilan program dapat dibagi menjadi 3 bagian, antara lain :

1. *form Editor* : menyediakan fasilitas editor bagi Pemakai yang akan meng-*input*-kan persamaan fungsi kinerja sistem. Yang perlu diperhatikan adalah bentuk persamaan tersebut harus disesuaikan dahulu dengan *format Delphi*.
2. *form Probability Density Function* : menyediakan pilihan jenis-jenis distribusi (beserta semua parameternya) yang akan diterapkan Pemakai pada variabel-variabel di dalam persamaan fungsi kinerja sistem. Yang harus diperhatikan

adalah setiap jenis distribusi mempunyai parameter dengan persyaratan tertentu, misalnya distribusi Rayleigh, mempunyai syarat $\eta \geq 0$. Untuk pengecekan, pada form ini juga disediakan fasilitas untuk menampilkan *grafik / visualisasi* distribusi tersebut.

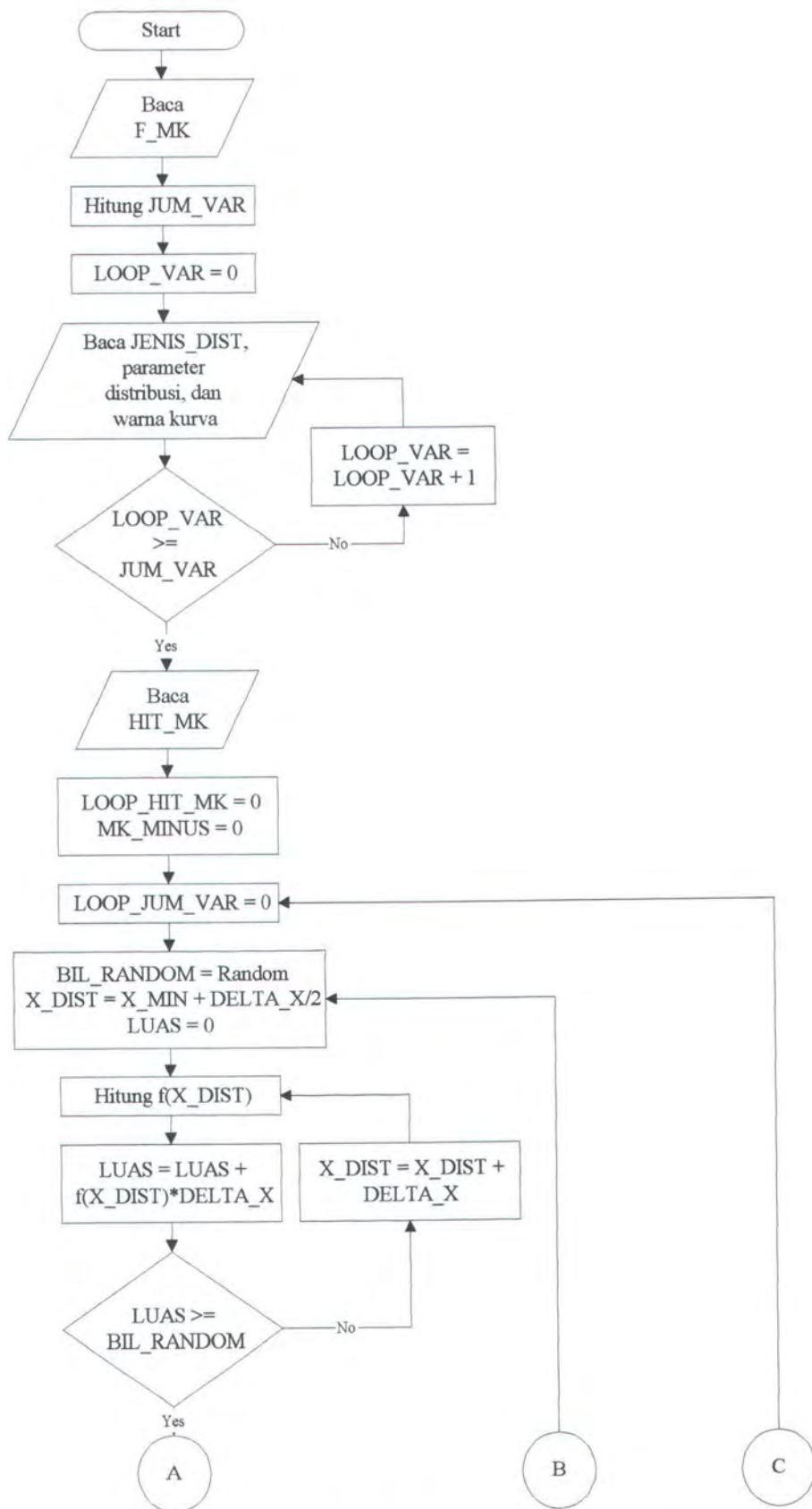
3. *form Interferensi Diagram* : menyediakan fasilitas pengecekan akhir terhadap semua parameter distribusi, menampilkan diagram interferensi, dan memberikan hasil akhir perhitungan berupa peluang kegagalan sistem.

Pada saat proses perhitungan dimulai, program akan mengecek jumlah variabel (JUM_VAR) yang digunakan, yang hasilnya akan digunakan sebagai pedoman perulangan (LOOP_VAR) pada saat pengisian variabel pada persamaan fungsi kinerja sistem (FK(x)). Kemudian program memanggil RNG yang akan menghasilkan sebuah angka acak (BIL_RANDOM) antara 0 dan 1, yang selanjutnya ditransformasikan (sesuai dengan bentuk fkp variabel) menjadi konstanta pengisi variabel (X[LOOP_VAR]) dalam persamaan FK(x). Langkah ini akan terus diulangi sampai semua variabel (X[LOOP_VAR]) dalam persamaan FK(x) terisi seluruhnya.

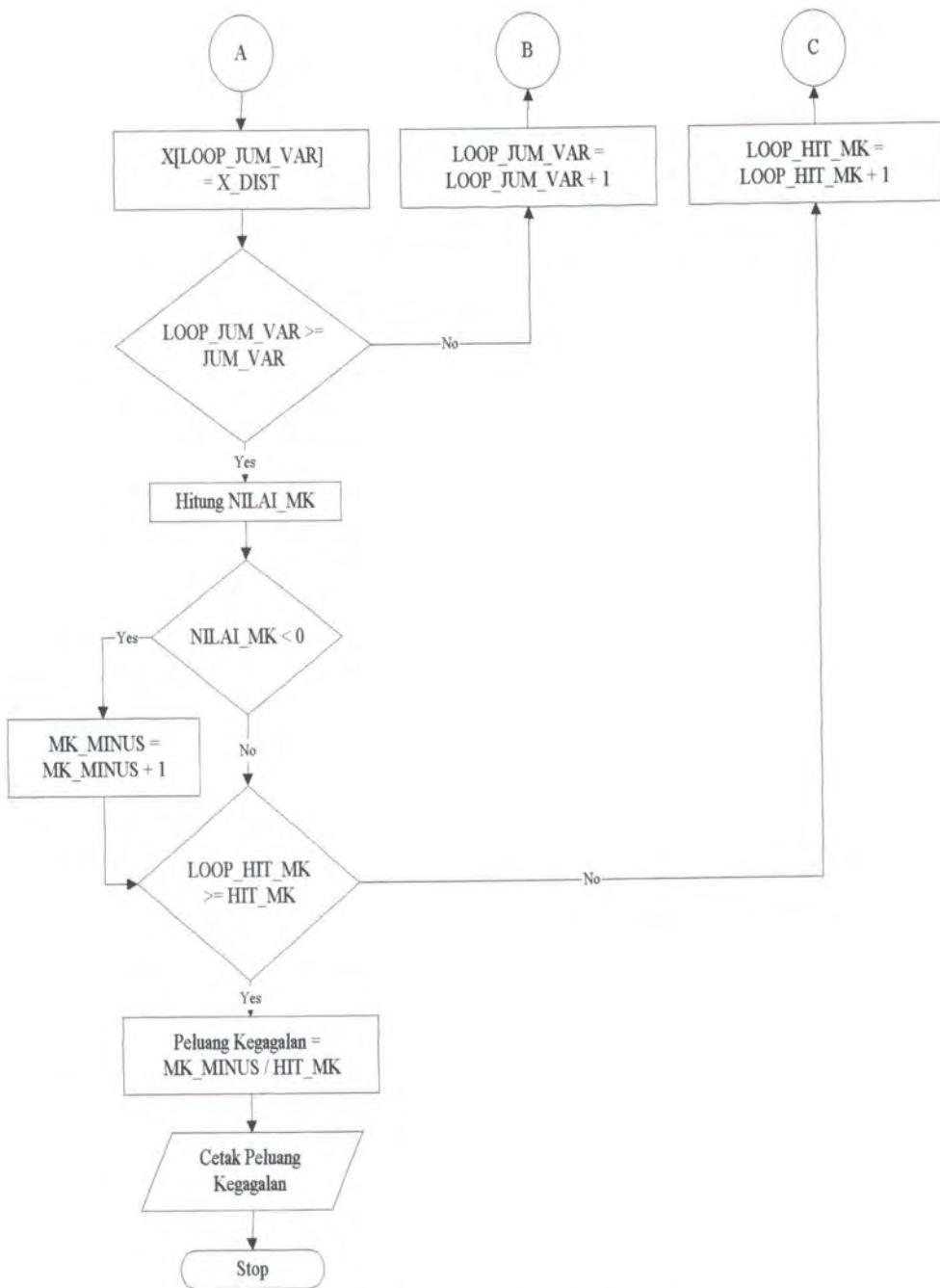
Yang perlu diperhatikan bahwa setiap variabel (X[LOOP_VAR]) di dalam persamaan FK(x) kemungkinan besar mempunyai karakteristik yang berlainan. Karakteristik ini ditentukan oleh bentuk dari fkp yang pada umumnya mengikuti bentuk suatu jenis distribusi. Jenis-jenis distribusi yang disediakan program ini antara lain distribusi Normal, Rayleigh, Exponential, Weibull, Log-Normal, Extreme Value Minima dan Extreme Value Maxima. Apabila semua variabel (X[LOOP_VAR]) di dalam persamaan FK(x) telah terisi, maka perhitungan harga

$FK(x)$ dapat dilakukan. Apabila harga $FK(x) < 0$, sistem dinyatakan gagal. Langkah ini akan diulangi sebanyak jumlah perulangan yang ditentukan oleh pemakai (HIT_MK), yang diikuti dengan penghitungan jumlah kegagalan (MK_MINUS). Selanjutnya peluang kegagalan sistem (P_g) dapat dihitung dengan cara membagi jumlah kegagalan (MK_MINUS) dengan jumlah perulangan (HIT_MK).

Flowchart program Simulasi Monte Carlo dapat dilihat pada gambar berikut :



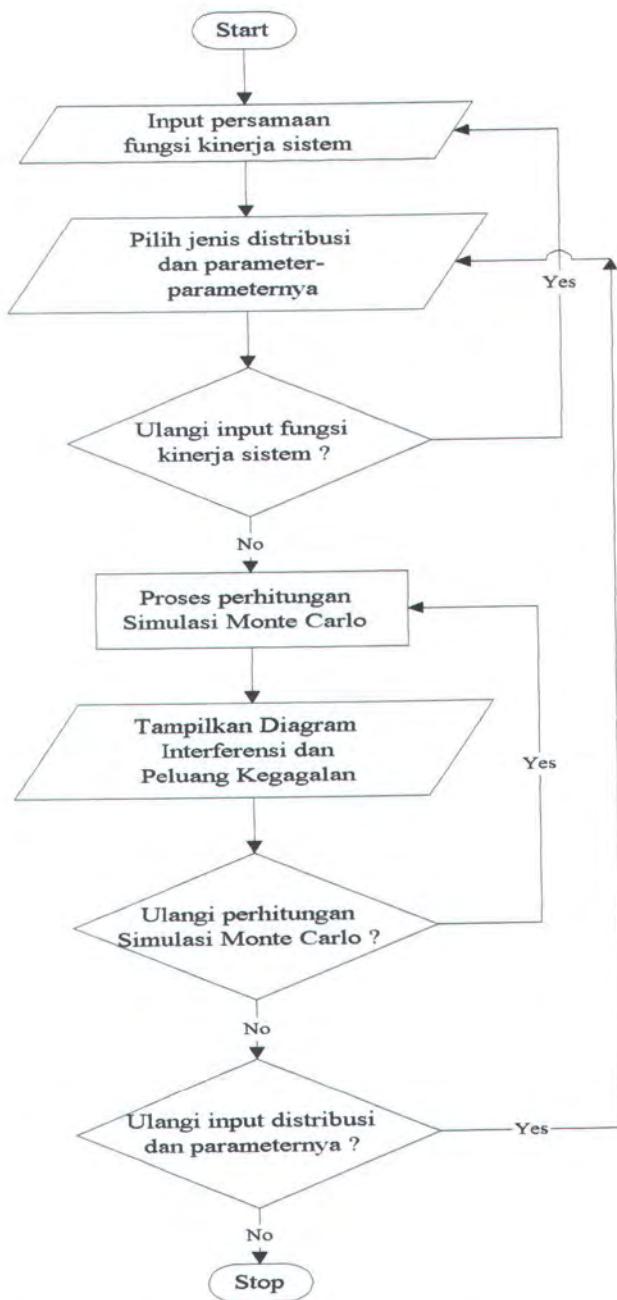
Gambar 3.1- Flowchart Program Simulasi Monte Carlo



Gambar 3.1 (lanjutan)- Flowchart Program Simulasi Monte Carlo

3.2 LANGKAH PENGGUNAAN PROGRAM

Langkah-langkah yang harus dilaksanakan apabila Pemakai menggunakan program ini dapat dilihat pada *flowchart* berikut :



Gambar 3.2- Flowchart penggunaan program Simulasi Monte Carlo

Penjelasan *flowchart* :

1. *Input* persamaan fungsi kinerja sistem.

Input persamaan fungsi kinerja sistem dapat dilakukan Pemakai pada *form Editor*. Persamaan yang ditulis haruslah persamaan yang sudah disesuaikan dengan *format Delphi*. Input persamaan ini bisa dengan cara membuat file baru ataupun membuka file lama. Apabila persamaan yang dituliskan sudah benar, maka peng-*compile*-an akan dilakukan dan pindah ke tahap selanjutnya.

2. Pemilihan jenis distribusi dan parameter-parameternya.

Pemilihan jenis distribusi dilakukan oleh Pemakai dan harus disesuaikan dengan karakteristik pdf dari variabel yang bersangkutan. Pengisian parameter distribusi juga harus memperhatikan persyaratan yang berlaku. Pada tahap ini Pemakai diminta untuk mengisi karakteristik / jenis distribusi semua variabel yang digunakan. Untuk pengecekan disediakan fasilitas yang dapat menampilkan bentuk distribusi yang bersangkutan. Jika semua itu sudah benar maka proses akan pindah ke tahap selanjutnya.

3. Proses perhitungan Simulasi Monte Carlo.

Pada tahap ini kendali dipegang sepenuhnya oleh komputer, Pemakai hanya bisa menunggu proses perhitungan selesai dilakukan. Jumlah perulangan perhitungan sesuai dengan yang di-*input*-kan oleh Pemakai.

4. *Visualisasi* Diagram Interferensi dan Peluang Kegagalan.

Apabila perhitungan telah selesai dilakukan, maka komputer akan menampilkan Diagram Interferensi sistem dan hasil perhitungan Peluang Kegagalan sistem.

BAB IV
ANALISA KEANDALAN SISTEM
DENGAN SISTEM MONTE CARLO

BAB IV

ANALISA KEANDALAN SISTEM

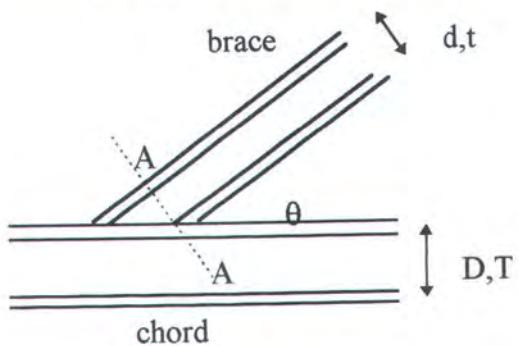
DENGAN SIMULASI MONTE CARLO

4.1 IMPLEMENTASI SIMULASI MONTE CARLO

Sebagai contoh penerapan metode Simulasi Monte Carlo, ditinjau 2 (dua) kasus, yaitu sebagai berikut:

- 1) Kasus *tubular joint (Y Joint)*.

Pada kasus *tubular joint (Y joint)*, joint mengalami gaya *axial (N)*, *in-plane bending moment (M_I)*, dan *out-plane bending moment (M_O)*, dengan gaya *axial ultimate (N_U)*, *in-plane bending moment ultimate (M_{IU})*, dan *out-plane bending moment ultimate (M_{OU})*, seperti yang diperlihatkan gambar berikut :



Gambar 4.1- Tubular joint : Y joint

Safety Margin tubular joint tersebut adalah (Hoadley and Yura [1983]) :

$$M = 1 - \frac{N}{N_U} - \left(\frac{|M_I|}{M_{IU}} \right)^{1,2} - \left(\frac{|M_O|}{M_{OU}} \right)^{2,1} \quad (4.1)$$

dimana : $N_U = 5 \cdot 10^4 \text{ N}$ $M_{IU} = 5 \cdot 10^3 \text{ Nm}$ $M_{OU} = 5 \cdot 10^4 \text{ Nm}$

$$\mu_N = 10^4 \text{ N} \quad \sigma_N = 0,2 \cdot 10^4 \text{ N}$$

$$\mu_{MI} = 2 \cdot 10^3 \text{ Nm} \quad \sigma_{MI} = 0,5 \cdot 10^3 \text{ Nm}$$

$$\mu_{MO} = 10^4 \text{ Nm} \quad \sigma_{MO} = 0,1 \cdot 10^4 \text{ Nm}$$

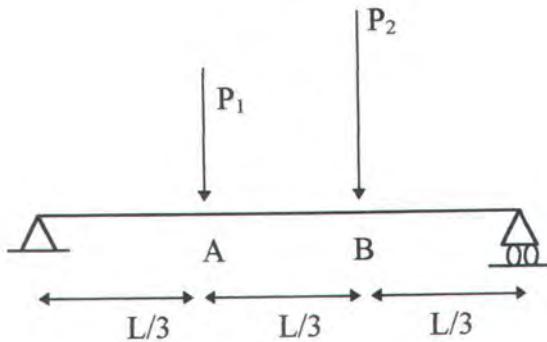
Sehingga bentuk *Safety Margin* diatas menjadi :

$$M = 1 - 2 \cdot 10^{-5} \cdot N - \left(2 \cdot 10^{-4} \cdot |M_I| \right)^{1,2} - \left(2 \cdot 10^{-5} \cdot |M_O| \right)^{2,1} \quad (4.2)$$

Persamaan inilah yang dituliskan di dalam *form Editor* sebagai fungsi kinerja sistem. Yang harus diperhatikan adalah persamaan tersebut harus dirubah lebih dulu sedemikian hingga sesuai dengan *format* persamaan yang dikenali *Delphi*. Langkah berikutnya Pemakai diminta untuk memasukkan parameter-parameter yang diperlukan tiap variabel pada persamaan tersebut. Dalam kasus ini, variabel N_U , M_{IU} dan M_{OU} berdistribusi Normal dengan μ (rata-rata) dan σ (*standar deviasi*) yang telah diketahui. Untuk pengecekan, sebaiknya Pemakai mem-*preview* distribusi tersebut dengan fasilitas *preview* yang disediakan oleh program. Sebelum memulai perhitungan, Pemakai diminta untuk memasukkan jumlah perulangan yang akan dilakukan. Selanjutnya kendali perhitungan dipegang penuh oleh komputer. Perhitungan akan berakhir apabila *counter* jumlah perulangan telah terpenuhi. Pada akhirnya, komputer akan menampilkan Diagram Interferensi sistem dan Peluang Kegagalan sebagai hasil akhir perhitungan. Pada contoh kasus *tubular joint* ini Peluang Kegagalan yang didapat adalah 0,00008.

2) Kasus pembebanan pada suatu balok.

Pada kasus ini, suatu balok dengan panjang L diberi 2 (dua) beban vertikal masing-masing P_1 dan P_2 , seperti yang terlihat pada gambar (4.2) berikut ini :



Gambar 4.2- Pembebanan pada balok

Antara beban P_1 dan P_2 terdapat korelasi, sedemikian hingga $\text{Cov}[P_1, P_2] = 0,14 \text{ kN}^2$. Balok tersebut diasumsikan mengalami kegagalan pada saat $|m| \geq m_F$, dimana m adalah momen maximum dan m_F adalah momen kritis balok. Dengan demikian *Safety Margin* (M) sistem / struktur tersebut adalah (Thoft-Christensen and Murotsu [1986]):

$$M = M_F - \frac{1}{9}L(P_1 + 2P_2) = M_F - \frac{1}{9}LP \quad (4.3)$$

dimana : $\mu_{P1} = 10 \text{ kN}$ $\sigma_{P1} = 0,283 \text{ kN}$

$\mu_{P2} = 17 \text{ kN}$ $\sigma_{P2} = 0,500 \text{ kN}$

$\mu_L = 9 \text{ m}$ $\sigma_L = 0,5 \text{ m}$

$\mu_{MF} = 60 \text{ kNm}$ $\sigma_{MF} = 6 \text{ kNm}$

karena P_1 dan P_2 berkorelasi, maka :

$$\mu_p = 10 + 2.17 = 14 \text{ kN}$$

$$\sigma_p = (0.08 + 4.0, 25 + 4.0, 14)^{1/2} = 1,28 \text{ kN}$$

Pada kasus ini, persamaan yang diinputkan pada *form editor* adalah persamaan (4.3). Persamaan ini mempunyai 3 (tiga) variabel yang masing-masing variabel tersebut mempunyai variabilitas berdistribusi normal dengan μ (rata-rata) dan σ (standart deviasi) yang telah diketahui. Peluang Kegagalan yang didapat dari *running program* yaitu 0.0085.

4.2 VALIDASI PROGRAM

Untuk pengecekan ketepatan hasil program, maka dilakukan perbandingan hasil akhir dengan metode *Advance First Order Second Moment* (AFOSM). Dengan parameter-parameter yang telah diketahui sebelumnya, maka dapat dilakukan perhitungan sebagai berikut :

- 1) Kasus *tubular joint (Y Joint)*

Persamaan *Safety Margin* :

$$M = 1 - 2.10^{-5} \cdot N - (2.10^{-4} \cdot |M_I|)^{1,2} - (2.10^{-5} \cdot |M_O|)^{2,1} \quad (4.4)$$

Variabel-variabel beban *axial* (N), *in-plane bending moment* (M_I) dan *out-plane bending moment* (M_O) di-normalize menjadi :

$$Z_1 = \frac{N - \mu_N}{\sigma_N} \rightarrow N = Z_1 \cdot \sigma_N + \mu_N = Z_1 \cdot 0,2 \cdot 10^4 + 10^4 \quad (4.5)$$

$$Z_2 = \frac{M_I - \mu_{M_I}}{\sigma_{M_I}} \rightarrow M_I = Z_2 \cdot \sigma_{M_I} + \mu_{M_I} = Z_2 \cdot 0,5 \cdot 10^3 + 2 \cdot 10^3 \quad (4.6)$$

$$Z_3 = \frac{M_O - \mu_{M_O}}{\sigma_{M_O}} \rightarrow M_O = Z_3 \cdot \sigma_{M_O} + \mu_{M_O} = Z_3 \cdot 0,110^4 + 10^4 \quad (4.7)$$

Persamaan (4.5), (4.6) dan (4.7) disubstitusikan ke persamaan (4.4), sehingga persamaan *Safety Margin* berubah menjadi :

$$1 - 2 \cdot 10^{-5} \cdot (Z_1 \cdot 0,2 \cdot 10^4 + 10^4) - (2 \cdot 10^{-4} \cdot |Z_2 \cdot 0,5 \cdot 10^3 + 2 \cdot 10^3|)^{1,2} - \\ (2 \cdot 10^{-5} \cdot |Z_3 \cdot 0,110^4 + 10^4|)^{2,1} = 0 \quad (4.8)$$

Sesuai prinsip :

$$f(\beta\alpha_1, \dots, \beta\alpha_n) = 0 \quad (4.9)$$

$$\alpha_i = -\frac{1}{k} \frac{\partial f}{\partial Z_i} (\bar{\beta\alpha}), i = 1, \dots, n \quad (4.10)$$

$$k = \left[\sum_{j=1}^n \left(\frac{\partial f}{\partial Z_j} (\bar{\beta\alpha}) \right)^2 \right]^{\frac{1}{2}} \quad (4.11)$$

Maka :

$$f(\bar{\beta\alpha}) = 1 - (Z_1 \cdot 0,04 + 0,2) - |Z_2 \cdot 0,1 + 0,4|^{1,2} - |Z_3 \cdot 0,02 + 0,2|^{2,1} \quad (4.12)$$

$$\frac{\partial f}{\partial Z_1} (\bar{\beta\alpha}) = -0,04$$

$$\frac{\partial f}{\partial Z_2} (\bar{\beta\alpha}) = -1,2 \cdot |Z_2 \cdot 0,1 + 0,4|^{0,2} \cdot 0,1 = -0,12 \cdot |Z_2 \cdot 0,1 + 0,4|^{0,2}$$

$$\frac{\partial f}{\partial Z_3} (\bar{\beta\alpha}) = -2,1 \cdot |Z_3 \cdot 0,02 + 0,2|^{1,1} \cdot 0,02 = -0,042 \cdot |Z_3 \cdot 0,02 + 0,2|^{1,1}$$

$$k = \left(-0,042 \cdot |\beta \cdot \alpha_3 \cdot 0,02 + 0,2|^{1,1} \right)^{\frac{1}{2}}$$

$$\alpha_1 = \frac{1}{k} \cdot 0,04 \quad (4.13)$$

$$\alpha_2 = \frac{1}{k} \cdot 0,12 \cdot |\beta \cdot \alpha_2 \cdot 0,1 + 0,4|^{0,2} \quad (4.14)$$

$$\alpha_3 = \frac{1}{k} \cdot 0,042 \cdot |\beta \cdot \alpha_3 \cdot 0,02 + 0,2|^{1,1} \quad (4.15)$$

$$\beta = \frac{1}{\left(\alpha_1 \cdot 0,04 + \frac{0,2}{\beta} \right) + |\beta|^{0,2} \cdot \left| \alpha_2 \cdot 0,1 + \frac{0,4}{\beta} \right|^{1,2} + |\beta|^{1,1} \cdot \left| \alpha_3 \cdot 0,02 + \frac{0,2}{\beta} \right|^{2,1}} \quad (4.16)$$

Untuk mencari harga α_1 , α_2 , α_3 dan β dilakukan proses *iterasi* terhadap persamaan (4.13), (4.14), (4.15) dan (4.16) dengan memasukkan harga awal pada α_1 , α_2 , α_3 dan β . Harga yang dimasukkan sebaiknya berada dalam range -1 sampai dengan 1. Dengan *iterasi* α_1 , α_2 , α_3 dan β didapat hasil :

$$\alpha_1 = 0,332$$

$$\alpha_2 = 0,941$$

$$\alpha_3 = 0,061$$

$$\beta = 3,78$$

Untuk menghitung Peluang Kegagalan, dipakai tabel distribusi Normal standard, yaitu :

$$Pg = \Phi(-\beta) \quad (4.17)$$

Dengan demikian, Peluang Kegagalan dapat ditentukan, yaitu :

$$\beta = 3,78 \rightarrow Pg = 0,000078$$

2) Kasus pembebanan pada balok

Persamaan *Safety Margin* :

$$M = M_F - \frac{1}{9}LP \quad (4.18)$$

Variabel-variabel P, L, dan M_F dinormalisasi menjadi :

$$Z_1 = \frac{M_F - \mu_{MF}}{\sigma_{MF}}, \quad Z_2 = \frac{L - \mu_L}{\sigma_L}, \quad Z_3 = \frac{P - \mu_P}{\sigma_P} \quad (4.19)$$

Persamaan-persamaan (4.19) disubstitusikan ke persamaan (4.18), sehingga menjadi :

$$144 + 54.Z_1 - 22.Z_2 - 11,52.Z_3 - 0,64.Z_2.Z_3 = 0$$

Sesuai prinsip seperti pada persamaan (4.9), (4.10) dan (4.11), maka didapat persamaan :

$$\alpha_1 = -\frac{1}{k}54 \quad (4.20)$$

$$\alpha_2 = \frac{1}{k}(22 + 0,64.\beta.\alpha_3) \quad (4.21)$$

$$\alpha_3 = \frac{1}{k}(11,52 + 0,64.\beta.\alpha_2) \quad (4.22)$$

$$\beta = \frac{144}{-54.\alpha_1 + 22.\alpha_2 + 11,52.\alpha_3 + 0,64.\beta.\alpha_2.\alpha_3} \quad (4.23)$$

Harga-harga β , α_1 , α_2 dan α_3 didapatkan dengan proses iterasi. Hasilnya adalah :

$$\alpha_1 = -0,91$$

$$\alpha_2 = 0,37$$

$$\alpha_3 = 0,20$$

$$\beta = 2,43$$

Sesuai persamaan (4.17) , Peluang Kegagalan dapat diketahui :

$$Pg = \Phi(-2,43) = 0,0075$$

4.3 DISKUSI

Dari program Simulasi Monte Carlo didapatkan *output* yang selanjutnya dibandingkan dengan hasil perhitungan dari metode AFOSM. Sejauh mana perbedaan hasil tersebut tergantung pada beberapa faktor, antara lain :

1. Tingkat keragaman bilangan acak yang dihasilkan oleh komputer.

Bilangan acak yang digunakan program ini adalah bilangan acak yang dikeluarkan oleh komputer. Dari beberapa kali pengujian, munculnya bilangan acak yang sama berkisar pada perulangan ke $10^{13} - 10^{14}$.

2. Jumlah perulangan perhitungan yang digunakan oleh Pemakai.

Pemilihan jumlah perulangan perhitungan sangat berpengaruh terhadap ketepatan hasil akhir perhitungan. Semakin besar jumlah perulangan perhitungan yang digunakan, hasilnya akan semakin mendekati harga yang sesungguhnya. Namun pemilihan jumlah perulangan yang terlalu besar akan memakan waktu yang terlalu lama. Dari beberapa kali pengujian, Pemakai disarankan untuk menggunakan jumlah perulangan minimal 10^4 kali perhitungan

**BAB V
KESIMPULAN**



BAB V

KESIMPULAN

Dari pembahasan pada bab-bab sebelumnya, dapat diambil kesimpulan bahwa metode Simulasi Monte Carlo amat cocok digunakan pada kasus-kasus yang mempunyai fungsi kinerja sistem yang sangat rumit (amat sangat non-linier). Yang perlu diperhatikan adalah tingkat akurasi metode ini sangat bergantung pada tingkat keragaman bilangan acak yang dihasilkan dan jumlah perulangan yang digunakan. Semakin beragam bilangan acak yang dihasilkan, semakin tinggi tingkat akurasinya. Semakin besar jumlah perulangan yang digunakan, semakin tinggi pula tingkat akurasinya. Hal ini terlihat pada perbandingan hasil antara metode Simulasi Monte Carlo dengan metode Advanced First Order Second Moment (AFOSM) dari pembahasan sebelumnya.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- E. P. Popov, (1978), Mechanics of Materials, 2nd edition (S1 Version), Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.
- Kececioglu, Dimitri, (1991), Reliability Engineering Handbook, Vol. 1, Simon & Schuster Company, Englewood Cliffs, New Jersey.
- Lewis, E. E, (1987), Introduction to Reliability Engineering, John Wiley & Sons, Inc., New York, USA.
- McClelland, Bramlette and Reifel, Michael D, (1986), Planning and Design of Fixed Offshore Platforms, Van Nostrand Reinhold Company Inc., New York, USA.
- Pranata A., (1997), Pemrograman Borland Delphi, Andy Offset, Yogyakarta.
- Rosyid, D. M., (1996), Buku Pegangan Kuliah Analisis Keandalan dan Resiko, FTK - ITS, Surabaya.
- Santoso P. I., (1993), Dasar-dasar Pemrograman Pascal Teori dan Program Terapan, Andy Offset, Yogyakarta.
- S. K. Sinha and B. K. Kale, (1980), Life Testing and Reliability Estimation, Wiley Eastern Limited, New Delhi, India.
- Thoft-Christensen, Palle and Murotsu, Yoshisada, (1986), Application of Structural System Reliability Theory, Springer-Verlag Berlin, Heidelberg.
- Wozniewicz A. and Shammas N., (1995), Teach Yourself Borland Delphi in 21 days, Sams Publishing, Indianapolis, USA.

LAMPIRAN



DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN

Kampus ITS - Sukolilo, Surabaya 60111 Telp. 5928105, 5994251 -5 psw. 1104 - 1107
Telex 34224 Fax 5947254

TUGAS AKHIR (OE.1702)

NO. : 0146/PT.12.FTK4.H8/Q/97

MATA KULIAH : TUGAS AKHIR
NOMOR MATA KULIAH : OE.1702
NAMA MAHASWA : Wasil Prabowo W.
NOMOR POKOK : 4392100032
TGL.DIBERIKAN TUGAS : 22 Juli 1997
TGL.SELESAINYA TUGAS : 22 Januari 1998
DOSEN PEMBIMBING :
1. Dr.Ir. D.M. Rosyid
2. Dr.Ir. Eko Budi Djatmiko, M.Sc.

THEMA / URAIAN / DATA-DATA YANG DIBERIKAN / JUDUL TUGAS AKHIR :

PENGEMBANGAN PERANGKAT LUNAK ANALISA KEANDALAN SISTEM DENGAN MENGGUNAKAN METODE SIMULASI MONTE CARLO

DIBUAT RANGKAP 4 :

1. Mahasiswa ybs.
2. Dekan FTK-ITS
3. Dosen Pembimbing
4. Arsip Jurusan .



JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

NAMA MAHASISWA : Wasis Prabowo Wawan
NOMOR POKOK : 4392.100.032
NAMA DOSEN PEMBIMBING :
1. Dr. Ir. Daniel M. Rosyid
2. Dr. Ir. Eko Budi Djatmiko, M.Sc
TUGAS DIMULAI :
DISELESAIKAN :
JUDUL TUGAS AKHIR : Pengembangan Perangkat Lunak
Analisa Keandalan Sistem Dengan
Menggunakan Metode Simulasi
Monte Carlo

NO	TANGGAL	KONSULTASI MENGENAI	TANDA TANGAN DOSEN PEMBIMBING
1.	10 /2-1998	- Perumusan Masalah	Dr. Nurd
2.	20 /4-1998	- Metodologi	Dr. Nurd
3.	17 /5-1998	- Software Requirements	Dr. Nurd
4.	2 /8-1998	- Case Studies	Dr. Nurd
5.	15 /9-1998	- Penilaian Bob	Dr. Nurd
6.	10 /10-1998	- anal	Dr. Nurd
7.	6 /11-1998	- Program / flowchart dew	Dr. Nurd
8.	7 /12-1998	- Validasi	Dr. Nurd
9.	5 /2-1999	- Penilaian ahli	Dr. Nurd
10.			

CATATAN : LEMBAR PRESENSI INI HARUS DITUNJUKKAN PADA SAAT UJIAN
TUGAS AKHIR

JURUSAN TEKNIK KELAUTAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA
LEMBAR PRESENSI KONSULTASI TUGAS AKHIR

NAMA MAHASISWA : Wasis Prabowo Wawan
NOMOR POKOK : 4392.100.032
NAMA DOSEN PEMBIMBING :
1. Dr. Ir. Daniel M. Rosyid
2. Dr. Ir. Eko Budi Djatmiko, M.Sc
TUGAS DIMULAI :
DISELESAIKAN :
JUDUL TUGAS AKHIR : Pengembangan Perangkat Lunak
Analisa Keandalan Sistem Dengan
Menggunakan Metode Simulasi
Monte Carlo

NO	TANGGAL	KONSULTASI MENGENAI	TANDA TANGAN DOSEN PEMBIMBING
1.	3/2 - 1999	Bab I, II	
2.	5/2 - 1999	Lanjutkan Bab II, III, IV, V	
3.			
4.			
5.			
6.			
7.			"
8.			
9.			
10.			 ITS

CATATAN : LEMBAR PRESENSI INI HARUS DITUNJUKKAN PADA SAAT UJIAN
TUGAS AKHIR

DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK KELAUTAN

Kampus ITS - Sukolilo, Surabaya 60111 Telp. 5947254, 597274 psw. 144 Telex 34224 Fax 5947254

FORMULIR EVALUASI KEMAJUAN TUGAS AKHIR

Kami, dosen pembimbing Tugas Akhir dari mahasiswa :

Nama : Wasis Prabowo, S.
NRP : 4392100032
Judul TA : Pengembangan Perangkat Lunak
Analisa Keandalan Sistem Dengan
Menggunakan Simulasi Monte Carlo

Setelah mempertimbangkan butir-butir berikut :

- a. Keaktifan mahasiswa dalam mengadakan asistensi.
- b. Proporsi Materi TA yang telah diselesaikan sampai saat ini.
- c. Prospek penyelesaian TA dalam jangka waktu yang relevan .
- d. Masa Studi yang tersisa.

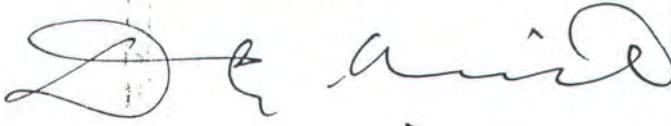
Dengan ini kami mengusulkan agar TA mahasiswa tersebut diputuskan untuk :

- Dibatalkan keseluruhannya dan mengajukan judul baru
- Diperkenankan menyelesaikan tanpa perubahan
- Diperkenankan mengikuti Ujian Tugas Akhir dengan Judul :

Pengemb. perangkat lunak Analisa Keandalan
Sistem Dengan Mengg. Simulasi Monte Carlo

Selanjutnya mahasiswa diatas diharuskan untuk dapat menyelesaikan Tugas Akhirnya dan dapat mengikuti ujian Tugas Akhir untuk Wisuda

Surabaya , 8/02/92
Dosen Pembimbing ;


Dr. Aini
D.M. Rejinal

LAMPIRAN A

LISTING PROGRAM

```
unit edit_pas;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, FileCtrl,
Buttons, Menus, ComCtrls, ShellAPI, ExtCtrls;

type
TEditor = class(TForm)
MainMenu1: TMainMenu; File1: TMenuItem; Help1: TMenuItem; New1: TMenuItem;
Open1: TMenuItem; Save1: TMenuItem; SaveAs1: TMenuItem; Exit1: TMenuItem;
HelpTopics1: TMenuItem; About1: TMenuItem; Close1: TMenuItem; RichEdit1: TRichEdit;
StatusBar1: TStatusBar; Panel1: TPanel; Open: TSpeedButton; Close: TSpeedButton;
save: TSpeedButton; New: TSpeedButton; BuildAll: TMenuItem; Compile2: TMenuItem;
Edit1: TMenuItem; Copy1: TMenuItem; Paste1: TMenuItem; Delete1: TMenuItem;
saveas: TSpeedButton; OpenDialog1: TOpenDialog; SaveDialog1: TSaveDialog;
Panel2: TPanel; Label1: TLabel; Label2: TLabel; Image1: TImage; Bevel1: TBevel;
Bevel2: TBevel; Exit: TBitBtn; Undo1: TMenuItem; Copy2: TSpeedButton;
Paste2: TSpeedButton; Delete2: TSpeedButton; Undo2: TSpeedButton; Image2: TImage;
procedure Open1Click(Sender: TObject); procedure Exit1Click(Sender: TObject);
procedure New1Click(Sender: TObject); procedure Close1Click(Sender: TObject);
procedure RichEdit1Change(Sender: TObject); procedure FormCreate(Sender: TObject);
procedure Save1Click(Sender: TObject); procedure SaveAs1Click(Sender: TObject);
procedure BuildAllClick(Sender: TObject); procedure countvar;
procedure About1Click(Sender: TObject); procedure HelpTopics1Click(Sender: TObject);
procedure Copy1Click(Sender: TObject); procedure Paste1Click(Sender: TObject);
procedure Delete1Click(Sender: TObject); procedure Undo1Click(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
end;

var
Editor: TEditor; berubah: boolean; f_text: textfile; def_dir: string [255];
cdef_dir: array [0..255] of char; x: array [0..5000] of char; x_string: string [255];
jum_var,loop_jum,loop_p,jum_kur: integer;

implementation

uses
help_me,abot,judul,progres;

{$R *.DFM}
```

```

procedure TEditor.FormCreate(Sender: TObject);
begin
  getdir(0,def_dir);
  assignfile(f_text,'dcc32.cfg');
  rewrite(f_text);
  writeln(f_text,'-u"' + def_dir + 'LIB"');
  writeln(f_text,'-aWinTypes=Windows;WinProcs=Windows;DbiProcs=BDE;DbiTypes=BDE;
  DbiErrs=BDE');
  closefile(f_text);
  berubah:= false;
  statusbar1.Panels.Items[0].Text := def_dir;
  statusbar1.Panels.Items[1].Text := 'Not Modified';
  splash.visible:= false;
end;

procedure TEditor.Exit1Click(Sender: TObject);
label 1,2;
begin
  if berubah = false then goto 1;
  if MessageDlg('File has been changed, are you sure to Exit ?',mtWarning,
    [mbYes,mbCancel],0) = mrCancel then goto 2;
1: {$I-}
  chdir(def_dir);
  assignfile(f_text,'mcs_swap.exe');
  erase(f_text);
  {$I+}
  application.terminate;
2:
end;

procedure TEditor.New1Click(Sender: TObject);
label 1,2;
begin
  if berubah = false then goto 1;
  if MessageDlg('File has been changed, are you sure to New ?',mtWarning,
    [mbYes,mbCancel],0) = mrCancel then goto 2;
1: copy2.enabled:= true; copy1.enabled:= true; paste2.enabled:= true; paste1.enabled:= true;
  delete2.enabled:= true; delete1.enabled:= true; undo2.enabled:= true; undol.enabled:= true;
  close.enabled:= true; close1.enabled:= true; richedit1.clear; richedit1.color:= clwindow;
  richedit1.enabled:= true; panel2.refresh; panel2.color:= clbtnshadow;
  chdir(def_dir);
  assignfile(f_text, 'f_mk.pas');
  rewrite(f_text);
  writeln(f_text,'unit f_mk;');
  writeln(f_text,'');
  writeln(f_text,'interface');
  writeln(f_text,'uses');
  writeln(f_text,'  math;');
  writeln(f_text,'procedure mk;');
  writeln(f_text,'const');
  writeln(f_text,'  jum_var=1;');
  writeln(f_text,'var');
  writeln(f_text,'  x: array [1..100] of real;');
  writeln(f_text,'  nilai_mk: real;');
  writeln(f_text,'');
  writeln(f_text,'implementation');
```

```

writeln(f_text,'procedure mk;');
writeln(f_text,'begin');
writeln(f_text,'  nilai_mk:=;');
writeln(f_text,'end;');
writeln(f_text,'');
writeln(f_text,'end.');
closefile(f_text);
richedit1.Lines.LoadFromFile('f_mk.pas');
opendialog1.filename:= def_dir+'\noname.mcs';
statusbar1.Panels.Items[0].Text:= opendialog1.filename;
chdir(def_dir);
image1.picture.loadfromfile('default.bmp');

2:
end;

procedure TEditor.Open1Click(Sender: TObject);
label 1,2;
begin
  editor.enabled:= false;
  if berubah = false then goto 1;
  if MessageDlg('File has been changed, are you sure to Open ?',mtWarning,
    [mbYes,mbCancel],0) = mrCancel then goto 2;
1: if opendialog1.execute then
begin
  try
    statusbar1.Panels.Items[0].Text:= opendialog1.filename;
    richedit1.Lines.LoadFromFile(opendialog1.filename);
    buildall.enabled:= true;
    berubah:= false;
    statusbar1.Panels.Items[1].Text:= 'Not Modified';
    close1.enabled:= true; close.enabled:= true; richedit1.color:= clwindow;
    richedit1.enabled:= true; copy2.enabled:= true; copy1.enabled:= true;
    paste2.enabled:= true; paste1.enabled:= true; delete2.enabled:= true;
    delete1.enabled:= true; undo2.enabled:= true; undo1.enabled:= true;
  except
    on efopenerror do messagedlg('Error - File not Open',mterror,[mbok],0);
  end;
  x_string:= opendialog1.filename;
  strpcopy(x,x_string);
  loop_p:= length(x_string);
  repeat
    loop_p:= loop_p-1;
    until x[loop_p] = '.';
    loop_p:= loop_p+2;
    delete(x_string,loop_p,length(opendialog1.filename));
    insert('bmp',x_string,length(x_string)+1);
  try
    image1.stretch:= true;
    image1.picture.loadfromfile(x_string);
  except
    on efopenerror do
      begin
        chdir(def_dir);
        image1.picture.loadfromfile('default.bmp');
      end;
  end;
end;

```

```

end;
2: editor.enabled:= true;
  image2.picture.loadfromfile(def_dir+"backgrnd.bmp");
end;

procedure TEditor.Save1Click(Sender: TObject);
begin
  {$I-}
  editor.richedit1.Lines.savetoFile(opendialog1.filename);
  statusbar1.Panels.Items[0].Text:= opendialog1.filename;
  {$I+}
  if ioresult <> 0 then messagedlg('There was an error proceesing the file',mterror,[mbok],0);
  berubah:= false;
  statusbar1.Panels.Items[1].Text:= 'Not Modified';
  save1.enabled:= false;
  save.enabled:= false;
end;

procedure TEditor.SaveAs1Click(Sender: TObject);
begin
  editor.enabled:= false;
  if savedialog1.execute then
  begin
    chdir(def_dir);
    {$I-}
    opendialog1.filename:= savedialog1.filename;
    statusbar1.Panels.Items[0].Text:= opendialog1.filename;
    richedit1.Lines.SaveToFile(opendialog1.filename);
    {$I+}
    if ioresult <> 0 then messagedlg('There was an error proceesing the file',mterror,[mbok],0);
    buildall.enabled:= true;
    berubah:= false;
    statusbar1.Panels.Items[1].Text := 'Not Modified';
    chdir(def_dir);
    image1.picture.loadfromfile('default.bmp');
  end;
  editor.enabled:= true;
end;

procedure TEditor.Close1Click(Sender: TObject);
label 1,2;
begin
  if berubah = false then goto 1;
  if MessageDlg('File has been changed, are you sure to Close ?',mtWarning,
    [mbYes,mbCancel],0) = mrCancel then goto 2;
1: richedit1.clear;richedit1.color:= clbtnshadow; richedit1.enabled:= false;
  close1.enabled:= false; close.enabled:= false; berubah:= false;
  statusbar1.Panels.Items[1].Text := 'Not Modified'; save1.enabled:= false; save.enabled:= false;
  saveas1.enabled:= false; saveas.enabled:= false; statusbar1.Panels.Items[0].Text := "";
  buildall.enabled:= false; chdir(def_dir); image1.picture.loadfromfile('default.bmp');
  copy2.enabled:= false; copy1.enabled:= false; paste2.enabled:= false; paste1.enabled:= false;
  delete2.enabled:= false; delete1.enabled:= false; undo2.enabled:= false; undo1.enabled:= false;
2:
end;

procedure TEditor.RichEdit1Change(Sender: TObject);

```

```

begin
  berubah:= true; statusbar1.Panels.Items[1].Text:= 'Modified'; save1.enabled:= true;
  save.enabled:= true; saveas1.enabled:= true; saveas.enabled:= true; buildall.enabled:= true;
end;

procedure TEditor.BuildAllClick(Sender: TObject);
begin
{$I-}
  chdir(def_dir);
  assignfile(f_text,'mcs_swap.exe');
  erase(f_text);
{$I+}
  if iorestart=0 then
    statusbar1.Panels.Items[1].Text:= 'File Erased'
  else
    statusbar1.Panels.Items[1].Text:= 'File not Found';
  countvar;
  richedit1.Lines.savetoFile('f_mk.pas');
  strpcopy(cdef_dir,def_dir);
  ShellExecute(Application.Handle,nil,'dcc32.exe','mcs_swap.dpr',cdef_dir, sw_hide);
  buildall.enabled:= false;
  progress.showmodal;
end;

procedure teditor.countvar;
begin
  strpcopy(x,richedit1.lines[15]);
  jum_kur:= 0;
  loop_p:= 0;
  repeat
    if x[loop_p] = '[' then
      begin
        jum_kur:= jum_kur+1;
        loop_p:= loop_p+1;
        x[(jum_kur-1)*2]:= x[loop_p];
        loop_p:=loop_p+1;
        if x[loop_p] <> ']' then
          x[(jum_kur-1)*2+1]:= x[loop_p]
        else
          x[(jum_kur-1)*2+1]:= '';
      end;
    loop_p:= loop_p+1;
  until loop_p = length(richedit1.lines[15]);
  loop_p:= 0;
  jum_var:= jum_kur;
  repeat
    loop_jum:= loop_p+2;
    repeat
      if (x[loop_p] = x[loop_jum]) and (x[loop_p] <> ' ') then
        begin
          if x[loop_p+1] = x[loop_jum+1] then
            begin
              jum_var:= jum_var-1;
              x[loop_jum]:= '';
            end;
        end;
    end;
end;

```

```
    loop_jum:= loop_jum+2;
    until loop_jum >= jum_kur*2;
    loop_p:= loop_p+2;
    until loop_p >= jum_kur*2;
    rchedit1.lines[7]:= '  jum_var = '+inttostr(jum_var)+':';
end;

procedure TEditor.About1Click(Sender: TObject);
begin
    about.showmodal;
end;

procedure TEditor.HelpTopics1Click(Sender: TObject);
begin
    chdir(def_dir);
    help.rchedit1.lines.loadfromfile('readme.rtf');
    help.showmodal;
end;

procedure TEditor.Copy1Click(Sender: TObject);
begin
    RichEdit1.CopyToClipboard;
end;

procedure TEditor.Paste1Click(Sender: TObject);
begin
    RichEdit1.PasteFromClipboard;
end;

procedure TEditor.Delete1Click(Sender: TObject);
begin
    RichEdit1.CutToClipboard;
end;

procedure TEditor.Undo1Click(Sender: TObject);
begin
    with RichEdit1 do
        if HandleAllocated then SendMessage(Handle, EM_UNDO, 0, 0);
end;

end.
```

```

unit monte;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls,
  Buttons, math;

type
  TDistribusi = class(TForm)
    Label8: TLabel; Edit2: TEdit; Edit3: TEdit; Label9: TLabel; ComboBox1: TComboBox;
    Label11: TLabel; Label12: TLabel; Edit5: TEdit; Label13: TLabel; Edit6: TEdit;
    Label14: TLabel; Label15: TLabel; Edit7: TEdit; Label16: TLabel; Edit8: TEdit;
    Panel1: TPanel; PaintBox1: TPaintBox; Label3: TLabel; Edit11: TEdit; Label4: TLabel;
    Label2: TLabel; ColorDialog1: TColorDialog; Back: TBitBtn; Label5: TLabel;
    Label6: TLabel; Label7: TLabel; Label10: TLabel; Panel2: TPanel; Preview: TBitBtn;
    Next: TBitBtn; Label11: TLabel; Bevel1: TBevel; Bevel2: TBevel; Bevel4: TBevel;
    Label17: TLabel; Label18: TLabel; Button1: TButton; Bevel3: TBevel; Bevel5: TBevel;
    Image1: TImage; Image2: TImage; procedure sumbu_xy; procedure f_gamma;
    procedure curve_dist; procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject); procedure param_dist;
    procedure get_param; procedure show_param; procedure BackClick(Sender: TObject);
    procedure Panel2Click(Sender: TObject); procedure NextClick(Sender: TObject);
    procedure PreviewClick(Sender: TObject); procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Distribusi: TDistribusi;
  loop_var, x_pos, y_pos, geser: integer;
  x_dist, y_dist, x_aksen, x_gamma, y_gamma, l_gamma, x_gammamax, deltax_gamma: real;
  scale_x, scale_y, numpong: variant;
  beta, gamma, nu, lamda, sigma, miu, deltax_dist, x_distmax, x_distmin,
  tinggi, lebar, n: array [1..100] of real; jenis_dist: array [1..100] of string[20];
  color_var: array [1..100] of tcolor;

implementation

uses
  f_mk, carlo;

{$R *.DFM}

{procedure Next}
procedure TDistribusi.NextClick(Sender: TObject);
begin
  get_param;
  interferensi.showmodal;
end;

{procedure Preview}
procedure TDistribusi.PreviewClick(Sender: TObject);
begin

```

```

get_param;
sumbu_xy;
curve_dist;
end;

{procedure ComboBox1Change}
procedure TDistribusi.ComboBox1Change(Sender: TObject);
begin
  param_dist;
end;

{procedure FormCreate}
procedure TDistribusi.FormCreate(Sender: TObject);
begin
  loop_var:= 1;
  edit11.text:= inttostr(jum_var);
  repeat
    jenis_dist[loop_var]:= 'Normal'; deltax_dist[loop_var]:= 1/1000; sigma[loop_var]:= 1;
    miu[loop_var]:= 1; gamma[loop_var]:= 1; lamda[loop_var]:= 1; beta[loop_var]:= 1;
    nu[loop_var]:= 1; color_var[loop_var]:= clblack; loop_var:= loop_var+1;
  until loop_var > jum_var;
end;

{procedure sumbu_xy}
procedure tDistribusi.sumbu_xy;
begin
  with distribusi.paintbox1.canvas do
  begin
    distribusi.panel1.refresh;
    geser:= numpang;
    setmapmode(handle,mm_anisotropic);
    setviewportex(handle,paintbox1.width,paintbox1.height,nil);
    setwindowextex(handle,paintbox1.width,-paintbox1.height,nil);
    setviewportorgex(handle,paintbox1.left+paintbox1.width div 2-geser,
                      paintbox1.top+paintbox1.height,nil);
    pen.color:= clwhite;
    moveto(0,0);
    lineto(0,paintbox1.height);
  end;
end;

{procedure curve_dist}
procedure tDistribusi.curve_dist;
begin
  x_dist:= x_distmin[loop_var];
  if jenis_dist[loop_var]='Normal' then
  begin
    repeat
      if x_dist-miu[loop_var]=0 then
        y_dist:= 1/(sigma[loop_var]*sqrt(2*pi)) {Normal dist}
      else
        y_dist:= 1/(sigma[loop_var]*sqrt(2*pi))*exp(-1/2*sqr((x_dist-miu[loop_var])/sigma[loop_var]));
      x_pos:= x_dist*scale_x;
      y_pos:= y_dist*scale_y;
      paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_var];
    until x_pos>=paintbox1.width;
  end;
end;

```

```

paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_var];
x_dist:= x_dist+deltax_dist[loop_var];
until x_dist > x_distmax[loop_var];
end
else
if jenis_dist[loop_var]='Extreme Val. Min' then
begin {Ext Val Min. dist}
repeat
y_dist:= 1/nu[loop_var]*exp(((x_dist-gamma[loop_var])/nu[loop_var])-  

    exp((x_dist-gamma[loop_var])/nu[loop_var]));
x_pos:= x_dist*scale_x;
y_pos:= y_dist*scale_y;
paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_var];
x_dist:= x_dist+deltax_dist[loop_var];
until x_dist > x_distmax[loop_var];
end
else
begin {Ext Val Max. dist}
repeat
y_dist:= 1/nu[loop_var]*exp(-((x_dist-gamma[loop_var])/nu[loop_var])-  

    exp(-((x_dist-gamma[loop_var])/nu[loop_var])));
x_pos:= x_dist*scale_x;
y_pos:= y_dist*scale_y;
paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_var];
x_dist:= x_dist+deltax_dist[loop_var];
until x_dist > x_distmax[loop_var];
end;
end;

{procedure f_gamma}
procedure tDistribusi.f_gamma;
begin
deltax_gamma:= 1/1000;
x_gamma:= deltax_gamma/2;
l_gamma:= 0;
x_gammamax:= n[loop_var]*10;
repeat
y_gamma:= exp(-x_gamma)*pangkat(x_gamma,n[loop_var]-1);
l_gamma:= l_gamma+y_gamma*deltax_gamma;
x_gamma:= x_gamma+deltax_gamma;
until x_gamma > x_gammamax;
end;

{get_param}
procedure tDistribusi.get_param;
begin
try
loop_var:= strtoint(button1.caption); color_var[loop_var]:= panel2.color;
jenis_dist[loop_var]:= combobox1.text; sigma[loop_var]:= strtofloat(edit2.text);
miu[loop_var]:= strtofloat(edit3.text); gamma[loop_var]:= strtofloat(edit5.text);
lamda[loop_var]:= strtofloat(edit6.text); beta[loop_var]:= strtofloat(edit7.text);
nu[loop_var]:= strtofloat(edit8.text);
if jenis_dist[loop_var]='Normal' then
begin
deltax_dist[loop_var]:= abs(5*sigma[loop_var])/5000;
x_distmin[loop_var]:= miu[loop_var]-5*sigma[loop_var]+deltax_dist[loop_var]/2;

```

```

x_distmax[loop_var]:= 2*abs(5*sigma[loop_var])+x_distmin[loop_var];
tinggi[loop_var]:= 1/(sigma[loop_var]*sqrt(2*pi));
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= miu[loop_var]*scale_x;
end
else
if jenis_dist[loop_var]='Weibull' then
begin
n[loop_var]:= 1/beta[loop_var]+1;
f_gamma;
deltax_dist[loop_var]:= abs(nu[loop_var]*l_gamma)/5000;
x_distmin[loop_var]:= gamma[loop_var]+deltax_dist[loop_var]/2;
x_distmax[loop_var]:= 2*abs(nu[loop_var]*l_gamma)+x_distmin[loop_var];
if beta[loop_var] <= 1 then
    tinggi[loop_var]:= beta[loop_var]/nu[loop_var]*exp((beta[loop_var]-1)*
        ln(deltax_dist[loop_var]/(nu[loop_var])))*exp(-1*exp(beta[loop_var]*
        ln(deltax_dist[loop_var]/(nu[loop_var]))))
else
    tinggi[loop_var]:= beta[loop_var]/nu[loop_var]*exp((1-1/beta[loop_var])*_
        ln(1-1/beta[loop_var]))*exp(-(1-1/beta[loop_var]));
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= gamma[loop_var]*scale_x+paintbox1.width/2;
end
else
if jenis_dist[loop_var]='Rayleigh' then
begin
deltax_dist[loop_var]:= exp(1/2*ln(pi*sqr(nu[loop_var])/2))/5000;
x_distmin[loop_var]:= deltax_dist[loop_var]/2;
x_distmax[loop_var]:= 2*exp(1/2*ln(pi*sqr(nu[loop_var])/2))+x_distmin[loop_var];
tinggi[loop_var]:= 1/nu[loop_var]*exp(-1/2);
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= paintbox1.width/2;
end
else
if jenis_dist[loop_var]='Exponential' then
begin
deltax_dist[loop_var]:= (1/lamda[loop_var])/5000;
x_distmin[loop_var]:= gamma[loop_var]+deltax_dist[loop_var]/2;

```

```

x_distmax[loop_var]:= 2*(1/lamda[loop_var])+x_distmin[loop_var];
tinggi[loop_var]:= lamda[loop_var];
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= gamma[loop_var]*scale_x+paintbox1.width/2;
end
else
if jenis_dist[loop_var]='Log - Normal' then
begin
deltax_dist[loop_var]:= abs(exp(miu[loop_var]+1/2*sqr(sigma[loop_var])))/ 5000;
x_distmin[loop_var]:= deltax_dist[loop_var]/2;
x_distmax[loop_var]:= 2*abs(exp(miu[loop_var]+1/2*sqr(sigma[loop_var])))+x_distmin[loop_var];
tinggi[loop_var]:= 1/(exp(miu[loop_var]-sqr(sigma[loop_var]))*sigma[loop_var]*
sqrt(2*pi))*exp(-1/2*sqr(-sigma[loop_var]));
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= paintbox1.width/2;
end
else
if jenis_dist[loop_var]='Extreme Val. Min' then
begin
x_distmin[loop_var]:= gamma[loop_var]-nu[loop_var]*20;
x_distmax[loop_var]:= gamma[loop_var]+nu[loop_var]*5;
deltax_dist[loop_var]:= (x_distmax[loop_var]-x_distmin[loop_var])/5000;
x_distmin[loop_var]:= x_distmin[loop_var]+deltax_dist[loop_var]/2;
tinggi[loop_var]:= 1/nu[loop_var]*exp(-1);
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);
label10.caption:= floattostrf(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostrf(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
numpang:= gamma[loop_var]*scale_x;
end
else
if jenis_dist[loop_var]='Extreme Val. Max' then
begin
x_distmin[loop_var]:= gamma[loop_var]-nu[loop_var]*5;
x_distmax[loop_var]:= gamma[loop_var]+nu[loop_var]*20;
deltax_dist[loop_var]:= (x_distmax[loop_var]-x_distmin[loop_var])/5000;
x_distmin[loop_var]:= x_distmin[loop_var]+deltax_dist[loop_var]/2;
tinggi[loop_var]:= 1/nu[loop_var]*exp(-1);
lebar[loop_var]:= x_distmax[loop_var]-x_distmin[loop_var];
label7.caption:= floattostrf(x_distmin[loop_var],ffExponent,6,0);

```

```

label10.caption:= floattostr(x_distmax[loop_var],ffExponent,6,0);
label6.caption:= '0';
label5.caption:= floattostr(tinggi[loop_var],ffExponent,6,0);
scale_x:= paintbox1.width/lebar[loop_var];
scale_y:= paintbox1.height/tinggi[loop_var];
num_pang:= gamma[loop_var]*scale_x;
end
else
begin
  messagedlg('Jenis distribusi tersebut tidak ada ',mterror,[mbok],0);
  combobox1.text:= 'Normal';
end;
except
  on eConvertError do messagedlg('Error on Parameter Input',mtError,[mbok],0);
  on eOverflow do messagedlg('Parameter Input is Too Large',mtError,[mbok],0);
end;
end;

{procedure show_param}
procedure tDistribusi.show_param;
begin
  loop_var:= strtoint(button1.caption); panel2.color:= color_var[loop_var];
  combobox1.text:= jenis_dist[loop_var]; edit2.text:= floattostr(sigma[loop_var]);
  edit3.text:= floattostr(miu[loop_var]); edit5.text:= floattostr(gamma[loop_var]);
  edit6.text:= floattostr(lamda[loop_var]); edit7.text:= floattostr(beta[loop_var]);
  edit8.text:= floattostr(nu[loop_var]);
end;

{procedure param_dist}
procedure tDistribusi.param_dist;
begin
  jenis_dist[loop_var]:= combobox1.text;
  image1.Stretch := True;
  if jenis_dist[loop_var]='Normal' then
    begin
      edit2.enabled:= true; edit3.enabled:= true; edit5.enabled:= false; edit6.enabled:= false;
      edit7.enabled:= false; edit8.enabled:= false; image1.Picture.LoadFromFile('norm.bmp');
    end
  else
    if jenis_dist[loop_var]='Weibull' then
      begin
        edit2.enabled:= false; edit3.enabled:= false; edit5.enabled:= true; edit6.enabled:= false;
        edit7.enabled:= true; edit8.enabled:= true; image1.Picture.LoadFromFile('weib.bmp');
      end
    else
      if jenis_dist[loop_var]='Rayleigh' then
        begin
          edit2.enabled:= false; edit3.enabled:= false; edit5.enabled:= false; edit6.enabled:= false;
          edit7.enabled:= false; edit8.enabled:= true; image1.Picture.LoadFromFile('raylgh.bmp');
        end
      else
        if jenis_dist[loop_var]='Exponential' then
          begin
            edit2.enabled:= false; edit3.enabled:= false; edit5.enabled:= true; edit6.enabled:= true;
            edit7.enabled:= false; edit8.enabled:= false; image1.Picture.LoadFromFile('exp.bmp');
          end
        end
      end
    end
  end;

```

```

else
if jenis_dist[loop_var]='Log - Normal' then
begin
  edit2.enabled:= true; edit3.enabled:= true; edit5.enabled:= false; edit6.enabled:= false;
  edit7.enabled:= false; edit8.enabled:= false; image1.Picture.LoadFromFile('lognorm.bmp');
end
else
if jenis_dist[loop_var]='Extreme Val. Min' then
begin
  edit2.enabled:= false; edit3.enabled:= false; edit5.enabled:= true; edit6.enabled:= false;
  edit7.enabled:= false; edit8.enabled:= true; image1.Picture.LoadFromFile('exvalmin.bmp');
end
else
if jenis_dist[loop_var]='Extreme Val. Max' then
begin
  edit2.enabled:= false; edit3.enabled:= false; edit5.enabled:= true; edit6.enabled:= false;
  edit7.enabled:= false; edit8.enabled:= true; image1.Picture.LoadFromFile('exvalmax.bmp');
end
else
begin
  messagedlg('Jenis Distribusi tersebut tidak ada',mterror,[mbok],0);
  combobox1.text:= 'Normal';
  image1.Picture.LoadFromFile('norm.bmp');
end;
end;

{back button}
procedure TDistribusi.BackClick(Sender: TObject);
begin
  application.terminate;
end;

{pilih color}
procedure TDistribusi.Panel2Click(Sender: TObject);
begin
  if colordialog1.execute then
    panel2.color:= colordialog1.color;
  loop_var:= strtoint(button1.caption);
  color_var[loop_var]:= colordialog1.color;
end;

procedure TDistribusi.Button1Click(Sender: TObject);
begin
  loop_var:= strtoint(button1.caption);
  get_param;
  loop_var:= loop_var+1;
  if loop_var > jum_var then loop_var:= 1;
  button1.caption:= inttostr(loop_var);
  show_param;
  param_dist;
end;

end.

```

```

unit carlo;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, StdCtrls,
  Buttons, ComCtrls, math, Gauges;

type
  TInterferensi = class(TForm)
    Panel1: TPanel; PaintBox1: TPaintBox; Label6: TLabel; Label8: TLabel; Label9: TLabel;
    Label11: TLabel; Label12: TLabel; Label13: TLabel; Label15: TLabel; Label16: TLabel;
    Label3: TLabel; Label4: TLabel; Edit1: TEdit; Edit3: TEdit; Edit4: TEdit; Edit5: TEdit;
    Edit6: TEdit; Edit7: TEdit; Edit8: TEdit; Edit9: TEdit; Edit11: TEdit; Finish: TBitBtn;
    Label5: TLabel; Back: TBitBtn; Execute: TBitBtn; Label2: TLabel; Label7: TLabel;
    Label10: TLabel; Label17: TLabel; Edit10: TEdit; Edit12: TEdit; Label1: TLabel;
    Bevel1: TBevel; Bevel2: TBevel; Label18: TLabel; Bevel3: TBevel; Label19: TLabel;
    Bevel4: TBevel; Label20: TLabel; Label21: TLabel; Bevel5: TBevel; Button1: TButton;
    Image1: TImage; CheckBox1: TCheckBox; Gauge1: TGauge; procedure sumbu_xy;
    procedure curve_dist; procedure FinishClick(Sender: TObject);
    procedure ExecuteClick(Sender: TObject); procedure BackClick(Sender: TObject);
    procedure FormActivate(Sender: TObject); procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Interferensi: TInterferensi; loop_hit_mk, hit_mk, mk_minus, loop_varc: integer;
  bil_random, x_min, x_max, ting_max, l_dist: real;

implementation

uses
  monte, f_mk;

{$R *.DFM}

{procedure execute}
procedure TInterferensi.ExecuteClick(Sender: TObject);
label 11, 12, 13, 14, 15, 16, 17, 21, 22;
begin
  execute.enabled:= false; back.enabled:= false; finish.enabled:= false;
  interferensi.panel1.refresh;
  hit_mk:= strtoint(edit11.text);
  edit12.text:= "";
  x_min:= x_distmin[1];
  x_max:= x_distmax[1];
  ting_max:= tinggi[1];
  loop_varc:= 1;
  repeat
    if x_min > x_distmin[loop_varc] then
      x_min:= x_distmin[loop_varc];
    if x_max < x_distmax[loop_varc] then

```

```

x_max:= x_distmax[loop_varc];
if ting_max < tinggi[loop_varc] then
  ting_max:= tinggi[loop_varc];
loop_varc:= loop_varc+1;
until loop_varc > jum_var;
if checkbox1.checked then
begin
  label10.caption:= floattostrf(x_min,ffExponent,6,0);
  label17.caption:= floattostrf(x_max,ffExponent,6,0);
  label7.caption:= '0';
  label2.caption:= floattostrf(ting_max,ffExponent,6,0);
  scale_x:= paintbox1.width/(x_max-x_min);
  scale_y:= paintbox1.height/ting_max;
  numpang:= ((x_max-x_min)/2+x_min)*scale_x;
  sumbu_xy;
  for loop_varc:= 1 to jum_var do curve_dist;
end;
gauge1.progress:= 0;
gauge1.maxvalue:= hit_mk;
mk_minus:= 0;
randomize;
for loop_hit_mk:= 1 to hit_mk do {loop perhitungan mk}
begin
  gauge1.progress:= gauge1.progress+1;
  for loop_varc:= 1 to jum_var do {loop jumlah variabel}
begin
  bil_random:= random; {pilih bil. random}
  x_dist:= x_distmin[loop_varc];
  l_dist:= 0;
  if jenis_dist[loop_varc]='Normal' then
begin
  11: if x_dist-miu[loop_varc]=0 then {Normal dist}
      y_dist:= 1/(sigma[loop_varc]*sqrt(2*pi))
    else
      y_dist:= 1/(sigma[loop_varc]*sqrt(2*pi))*
        exp(-1/2*sqr((x_dist-miu[loop_varc])/sigma[loop_varc]));
    goto 21;
  end
  else
  if jenis_dist[loop_varc]='Weibull' then
begin
  12: if x_dist-gamma[loop_varc]=0 then {Weibull dist}
      y_dist:= 0
    else
      y_dist:= beta[loop_varc]/nu[loop_varc]*
        pangkat(((x_dist-gamma[loop_varc])/nu[loop_varc]),
        (beta[loop_varc]-1))*exp(-pangkat(((x_dist-
        gamma[loop_varc])/nu[loop_varc]),beta[loop_varc]));
    goto 21;
  end
  else
  if jenis_dist[loop_varc]='Rayleigh' then
begin
  13: if x_dist=0 then {Rayleigh dist}
      y_dist:= 0
    else

```

```

        y_dist:= x_dist/(sqr(nu[loop_varc]))*exp(-1/2*sqr(x_dist/nu[loop_varc]));
        goto 21;
end
else
if jenis_dist[loop_varc]='Exponential' then
begin
    y_dist:= lamda[loop_varc]*exp(-lamda[loop_varc]*(x_dist-gamma[loop_varc]));
    goto 21;                                         {Exp. dist}
end
else
if jenis_dist[loop_varc]='Log - Normal' then
begin
    x_aksen:= ln (x_dist);                         {Log - Normal dist}
    if x_aksen-miu[loop_varc]=0 then
        y_dist:= 1/(x_dist*sigma[loop_varc]*sqrt(2*pi))
    else
        y_dist:= 1/(x_dist*sigma[loop_varc]*sqrt(2*pi))*  

            exp(-1/2*sqr((x_aksen-miu[loop_varc])/ sigma[loop_varc]));
    goto 21;
end
else
if jenis_dist[loop_varc]='Extreme Val. Min' then
begin
    y_dist:= 1/nu[loop_varc]*exp(((x_dist-gamma[loop_varc])/
        nu[loop_varc])-exp(-1*((x_dist-gamma[loop_varc])/ nu[loop_varc])));
    goto 21;
end
else
begin
    y_dist:= 1/nu[loop_varc]*exp(-1*((x_dist-gamma[loop_varc])/
        nu[loop_varc])-exp(-1*((x_dist-gamma[loop_varc])/ nu[loop_varc])));
    goto 21;
end;
1:   l_dist:= l_dist+y_dist*deltax_dist[loop_varc];
if l_dist >= bil_random then
    goto 22
else
    x_dist:= x_dist+deltax_dist[loop_varc];
if jenis_dist[loop_varc]='Normal' then goto 11;
if jenis_dist[loop_varc]='Weibull' then goto 12;
if jenis_dist[loop_varc]='Rayleigh' then goto 13;
if jenis_dist[loop_varc]='Exponential' then goto 14;
if jenis_dist[loop_varc]='Log - Normal' then goto 15;
if jenis_dist[loop_varc]='Extreme Val. Min' then
    goto 16
else
    goto 17;
2:   x[loop_varc]:= x_dist;                           {Transformasi}
end;
mk;
if nilai_mk < 0 then
    mk_minus:= mk_minus+1;
end;
gauge1.progress:= 0;
edit12.text:= floatstr(mk_minus/hit_mk);
execute.enabled:= true; back.enabled:= true; finish.enabled:= true;

```

```

end;

{procedure sumbu_xy}
procedure tInterferensi.sumbu_xy;
begin
  with interferensi.paintbox1.canvas do
  begin
    interferensi.panel1.refresh;
    geser:= numpang;
    setmapmode(handle,mm_anisotropic);
    setviewportex(handle,paintbox1.width,paintbox1.height,nil);
    setwindowextex(handle,paintbox1.width,-paintbox1.height,nil);
    setviewportorgex(handle,paintbox1.left+paintbox1.width div 2-geser,
      paintbox1.top+paintbox1.height,nil);
    pen.color:= clwhite;
    moveto(0,0);
    lineto(0,paintbox1.height);
  end;
end;

{procedure curve_dist}
procedure tInterferensi.curve_dist;
begin
  x_dist:= x_distmin[loop_varc];
  if jenis_dist[loop_varc]='Normal' then
  begin
    repeat
      if x_dist-miu[loop_varc]=0 then {Normal dist}
        y_dist:= 1/(sigma[loop_varc]*sqrt(2*pi))
      else
        y_dist:= 1/(sigma[loop_varc]*sqrt(2*pi))*exp(-1/2*sqr((x_dist-miu[loop_varc])/sigma[loop_varc]));
      x_pos:= x_dist*scale_x;
      y_pos:= y_dist*scale_y;
      paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
      x_dist:= x_dist+deltax_dist[loop_varc];
    until x_dist > x_max;
  end
  else
    if jenis_dist[loop_varc]='Weibull' then {Weibull dist}
    begin
      repeat
        if x_dist-gamma[loop_varc]<=0 then
          y_dist:= 0
        else
          y_dist:= beta[loop_varc]/nu[loop_varc]*pangkat(((x_dist-gamma[loop_varc])/nu[loop_varc]), (beta[loop_varc]-1))*
            exp(-pangkat(((x_dist-gamma[loop_varc])/nu[loop_varc]), beta[loop_varc]));
        x_pos:= x_dist*scale_x;
        y_pos:= y_dist*scale_y;
        paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
        x_dist:= x_dist+deltax_dist[loop_varc];
      until x_dist > x_max;
    end
    else
      if jenis_dist[loop_varc]='Rayleigh' then

```



```

begin
repeat
  if x_dist<=0 then {Rayleigh dist}
    y_dist:= 0
  else
    y_dist:= x_dist/(sqrt(nu[loop_varc]))*exp(-1/2*sqrt(x_dist/nu[loop_varc]));
    x_pos:= x_dist*scale_x;
    y_pos:= y_dist*scale_y;
    paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
    x_dist:= x_dist+deltax_dist[loop_varc];
  until x_dist > x_max;
end
else
if jenis_dist[loop_varc]='Exponential' then {Exp. dist}
begin
repeat
  if x_dist <= gamma[loop_varc] then
    y_dist:= 0
  else
    y_dist:= lamda[loop_varc]*exp(-lamda[loop_varc]*(x_dist-gamma[loop_varc]));
    x_pos:= x_dist*scale_x;
    y_pos:= y_dist*scale_y;
    paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
    x_dist:= x_dist+deltax_dist[loop_varc];
  until x_dist > x_max;
end
else
if jenis_dist[loop_varc]='Log - Normal' then {Log - Normal dist}
begin
repeat
  x_aksen:= ln(x_dist);
  if x_dist <= 0 then
    y_dist:= 0
  else
    if x_aksen-miu[loop_varc]=0 then
      y_dist:= 1/(x_dist*sigma[loop_varc]*sqrt(2*pi))
    else
      y_dist:= 1/(x_dist*sigma[loop_varc]*sqrt(2*pi))* exp(-1/2*
          sqrt((x_aksen-miu[loop_varc])/ sigma[loop_varc]));
    x_pos:= x_dist*scale_x;
    y_pos:= y_dist*scale_y;
    paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
    x_dist:= x_dist+deltax_dist[loop_varc];
  until x_dist > x_max;
end
else
if jenis_dist[loop_varc]='Extreme Val. Min' then {Ext Val Min. dist}
begin
repeat
  y_dist:= 1/nu[loop_varc]*exp(((x_dist-gamma[loop_varc])/nu[loop_varc])-*
      exp((x_dist-gamma[loop_varc])/nu[loop_varc]));
  x_pos:= x_dist*scale_x;
  y_pos:= y_dist*scale_y;
  paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
  x_dist:= x_dist+deltax_dist[loop_varc];
until x_dist > x_max;

```

```

end
else
begin
repeat
  y_dist:= 1/nu[loop_varc]*exp(-((x_dist-gamma[loop_varc])/nu[loop_varc])-  

    exp(-((x_dist-gamma[loop_varc])/nu[loop_varc])));
  x_pos:= x_dist*scale_x;
  y_pos:= y_dist*scale_y;
  paintbox1.canvas.pixels[x_pos,y_pos]:= color_var[loop_varc];
  x_dist:= x_dist+deltax_dist[loop_varc];
until x_dist > x_max;
end;
end;

{procedure interferensi finish}
procedure TInterferensi.FinishClick(Sender: TObject);
begin
  application.terminate;
end;

{procedure interferensi back}
procedure TInterferensi.BackClick(Sender: TObject);
begin
  interferensi.close;
end;

procedure TInterferensi.FormActivate(Sender: TObject);
begin
  gauge1.progress:= 0; panel1.refresh; edit1.text:= inttostr(jum_var); edit12.text:= "";
  button1.caption:= '1'; loop_varc:= stroint(button1.caption);
  edit10.color:= color_var[loop_varc]; edit3.text:= jenis_dist[loop_varc];
  edit4.text:= floattosstr(sigma[loop_varc]); edit5.text:= floattosstr(miu[loop_varc]);
  edit6.text:= floattosstr(gamma[loop_varc]); edit7.text:= floattosstr(lamda[loop_varc]);
  edit8.text:= floattosstr(beta[loop_varc]); edit9.text:= floattosstr(nu[loop_varc]);
end;

procedure TInterferensi.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  interferensi.visible:= false;
  distribusi.visible:= true;
end;

procedure TInterferensi.Button1Click(Sender: TObject);
begin
  loop_varc:= stroint(button1.caption); loop_varc:= loop_varc+1;
  if loop_varc > jum_var then loop_varc:= 1; button1.caption:= inttostr(loop_varc);
  edit10.color:= color_var[loop_varc]; edit3.text:= jenis_dist[loop_varc];
  edit4.text:= floattosstr(sigma[loop_varc]); edit5.text:= floattosstr(miu[loop_varc]);
  edit6.text:= floattosstr(gamma[loop_varc]); edit7.text:= floattosstr(lamda[loop_varc]);
  edit8.text:= floattosstr(beta[loop_varc]); edit9.text:= floattosstr(nu[loop_varc]);
end;

end.

```

LAMPIRAN B

PETUNJUK PEMAKAIAN

Program *Monte Carlo Simulation* ini dibuat dengan menggunakan bahasa pemrograman *Borland Delphi 2*, yang berbasis pada bahasa pemrograman *Pascal*. Konfigurasi minimum yang dibutuhkan antara lain *Operating Sistem Windows 95*, komputer (CPU) 386 dan *hard-disk* dengan *free-space* 15 megabytes. Pemakai sebaiknya menggunakan tampilan layar dengan resolusi 640 x 480 untuk mendapatkan hasil tampilan yang maksimal.

Pemakai hendaknya telah mempersiapkan input-input yang dibutuhkan oleh program *Monte Carlo Simulation* ini, antara lain :

1. Persamaan fungsi kinerja sistem yang akan ditinjau.
2. Jenis distribusi dari *random variable* beserta segala parameternya.

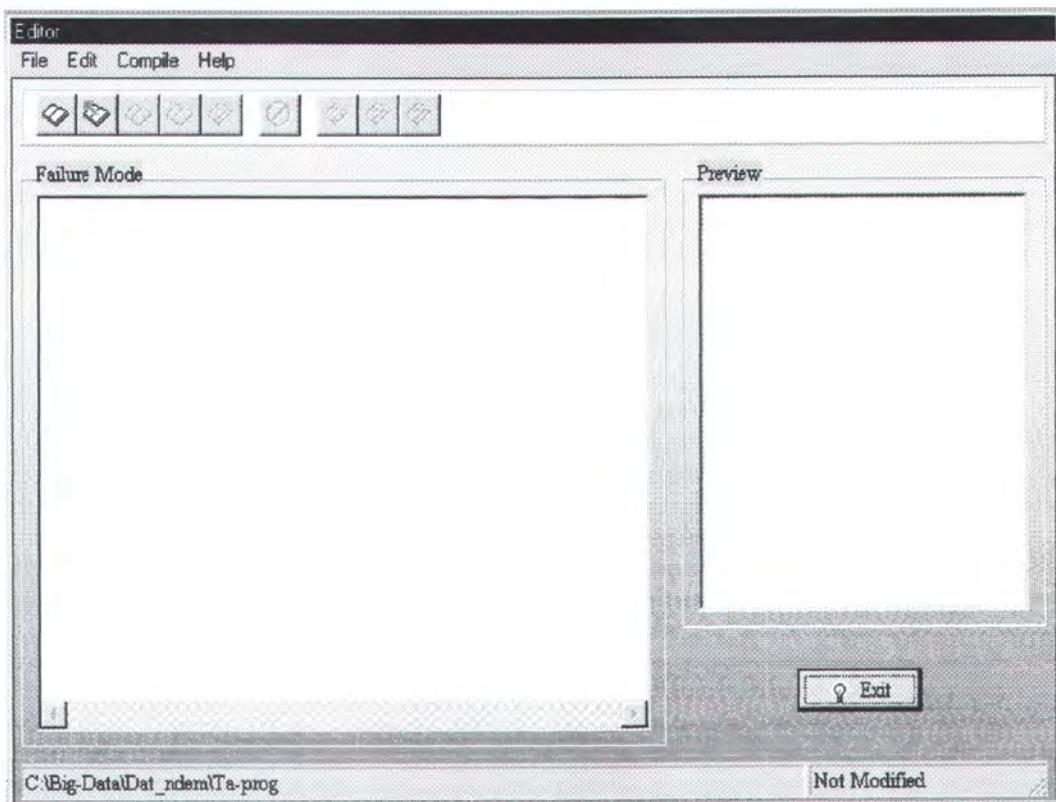
Secara garis besar, program *Monte Carlo Simulation* terdiri dari 3 (tiga) bagian, yaitu :

1. *Form Editor*.
2. *Form Probability Distribution Function*.
3. *Form Interferensi Diagram*.

Form Editor adalah tampilan yang pertama kali muncul. *Form Editor* ini berfungsi untuk menerima input berupa persamaan fungsi kinerja sistem yang dituliskan pemakai. Pada saat memberikan input persamaan fungsi kinerja, pemakai harus memperhatikan aturan-aturan berikut :

1. Variabel yang digunakan dalam fungsi kinerja sistem harus ditulis dalam $x[...]$.

2. Indeks variabel yang digunakan harus berupa urutan array / larik dan dimulai dari 1 (satu).
3. Penulisan persamaan fungsi kinerja sistem harus dalam format *Borland Delphi*.
4. Baris tempat penulisan persamaan fungsi kinerja sistem, yaitu pada baris ke 16, harus di daerah antara tanda '=' dan ';' .
5. Pemakai tidak diperbolehkan merubah posisi / baris dan fungsi-fungsi yang telah dibuat secara otomatis oleh program.



Gambar B.1- Tampilan Form Editor.

Di dalam *form editor* ini terdapat :

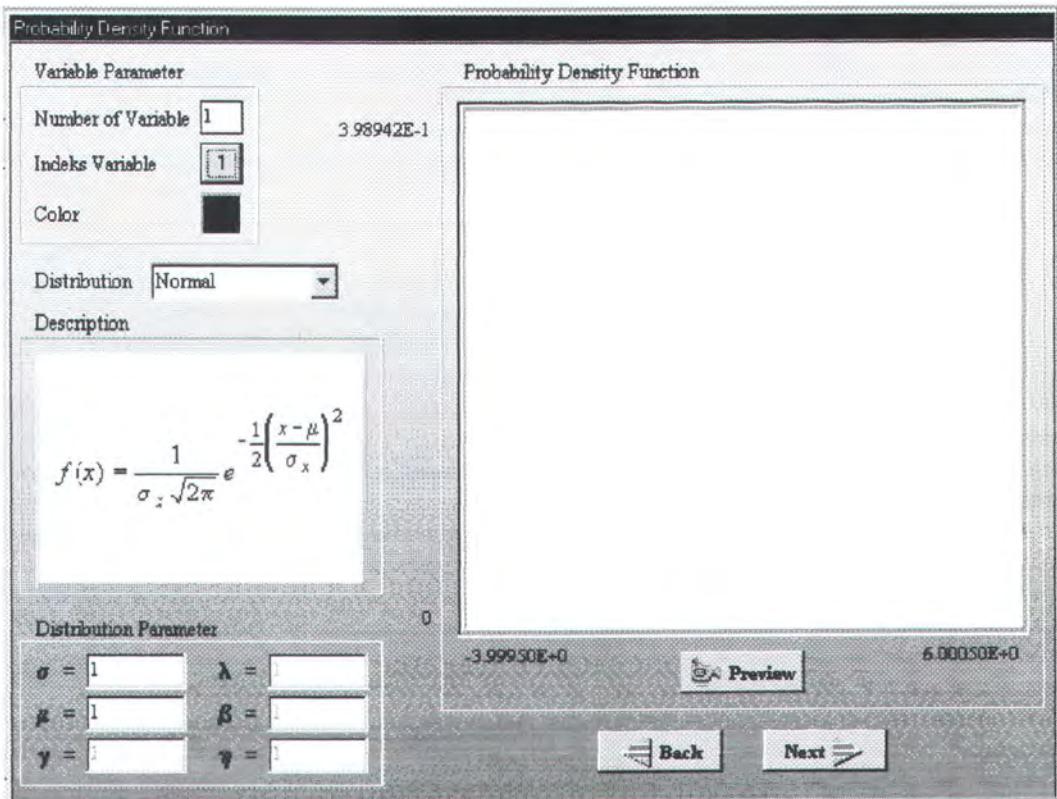
- 1) Menu Utama :

- *File* : fungsinya menangani segala operasi file (membuat, menyimpan, menutup, dan lain-lain).
 - *Edit* : fungsinya membantu pemakai saat memberikan input persamaan fungsi kinerja sistem (meng-copy, menghapus, meng-edit, dan lain-lain).
 - *Compile* : fungsinya meng-*compile* persamaan fungsi kinerja yang diinputkan oleh pemakai.
 - *Help* : fungsinya memberi petunjuk pemakaian program *Monte Carlo Simulation* (secara garis besar).
- 2) *Failure Mode Editor* : fungsinya sebagai tempat penulisan persamaan fungsi kinerja sistem.
- 3) *Preview* : fungsinya menampilkan gambar sistem (harus dalam format *bitmap*) yang dibuat oleh pemakai.

Apabila telah selesai meng-input-kan persamaan fungsi kinerja sistem, maka pemakai harus melakukan peng-*compile*-an sebelum melanjutkan ke langkah berikutnya.

Form selanjutnya yaitu *form Probability Density Function*. *Form Probability Density Function* ini berfungsi untuk menyediakan pilihan jenis distribusi dari variabel yang digunakan dalam persamaan fungsi kinerja, beserta segala macam parameter dari variabel tersebut. *Form Probability Density Function* ini juga menyediakan fasilitas untuk mem-*preview* distribusi dari variabel yang sedang diproses. Angka yang diinputkan pemakai pada masing-masing parameter

untuk setiap jenis distribusi harus memenuhi persyaratan distribusi yang bersangkutan.

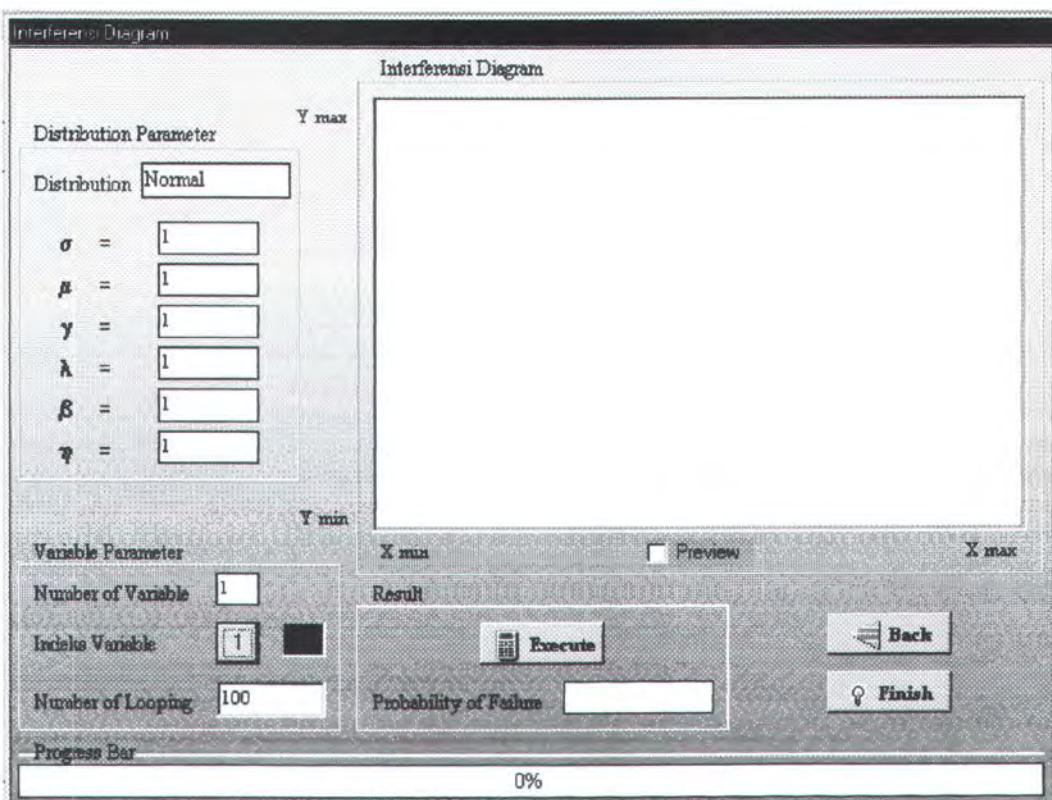


Gambar B.2- Tampilan Form Probability Distribution Function.

Apabila telah selesai meng-input-kan jenis distribusi beserta segala parameternya untuk setiap variabel, maka pemakai dapat melanjutkan ke langkah terakhir.

Form yang paling akhir adalah *form Interferensi Diagram*. *Form Interferensi Diagram* ini menyediakan fasilitas pengecekan terhadap jenis distribusi beserta parameternya untuk setiap variabel yang digunakan. *Form* ini juga menyediakan fasilitas untuk mem-preview *Interferensi Diagram* serta hasil akhir

perhitungan yang berupa Peluang Kegagalan sistem. Akurasi perhitungan Peluang Kegagalan sangat bergantung pada jumlah perhitungan yang digunakan pemakai. Semakin besar jumlah perhitungan yang digunakan, semakin akurat pula hasil yang didapat, namun waktu yang diperlukan juga semakin lama. Pemakai disarankan untuk menggunakan jumlah perhitungan minimal 10000 (sepuluh ribu) kali untuk mendapatkan hasil yang optimal.



Gambar B.3- Tampilan Form Interferensi Diagram.

Apabila telah selesai melaksanakan perhitungan, selanjutnya terserah pemakai apakah mengulangi perhitungan kembali atau keluar dari program.

LAMPIRAN C

Tabel fungsi kumulatif distribusi Normal Standart

x	cdf										
-5	1.44999E-08	-4.54	2.54034E-06	-4.08	2.22441E-05	-3.62	0.000147021	-3.16	0.000788538	-2.7	0.003466584
-4.99	2.97433E-08	-4.53	2.6768E-06	-4.07	2.32327E-05	-3.61	0.000152818	-3.15	0.000816044	-2.69	0.003572209
-4.98	4.57666E-08	-4.52	2.81959E-06	-4.06	2.42625E-05	-3.6	0.000158827	-3.14	0.000844429	-2.68	0.003680713
-4.97	6.26081E-08	-4.51	2.96897E-06	-4.05	2.53348E-05	-3.59	0.000165057	-3.13	0.000873721	-2.67	0.003792165
-4.96	8.03077E-08	-4.5	3.12525E-06	-4.04	2.64516E-05	-3.58	0.000171515	-3.12	0.000903943	-2.66	0.003906632
-4.95	9.89074E-08	-4.49	3.28873E-06	-4.03	2.76143E-05	-3.57	0.000178208	-3.11	0.000935124	-2.65	0.004024185
-4.94	1.18451E-07	-4.48	3.4597E-06	-4.02	2.88249E-05	-3.56	0.000185145	-3.1	0.000967289	-2.64	0.004144895
-4.93	1.38984E-07	-4.47	3.63852E-06	-4.01	3.00851E-05	-3.55	0.000192333	-3.09	0.001000467	-2.63	0.004268834
-4.92	1.60555E-07	-4.46	3.82551E-06	-4	3.13969E-05	-3.54	0.00019978	-3.08	0.001034686	-2.62	0.004396075
-4.91	1.83214E-07	-4.45	4.02102E-06	-3.99	3.27622E-05	-3.53	0.000207496	-3.07	0.001069976	-2.61	0.004526695
-4.9	2.07013E-07	-4.44	4.22544E-06	-3.98	3.41831E-05	-3.52	0.000215489	-3.06	0.001106366	-2.6	0.004660769
-4.89	2.32007E-07	-4.43	4.43913E-06	-3.97	3.56617E-05	-3.51	0.000223769	-3.05	0.001143886	-2.59	0.004798374
-4.88	2.58254E-07	-4.42	4.6625E-06	-3.96	3.72002E-05	-3.5	0.000232344	-3.04	0.001182569	-2.58	0.00493959
-4.87	2.85813E-07	-4.41	4.89597E-06	-3.95	3.88008E-05	-3.49	0.000241225	-3.03	0.001222445	-2.57	0.005084496
-4.86	3.14748E-07	-4.4	5.13997E-06	-3.94	4.04659E-05	-3.48	0.000250421	-3.02	0.001263549	-2.56	0.005233175
-4.85	3.45124E-07	-4.39	5.39494E-06	-3.93	4.21979E-05	-3.47	0.000259943	-3.01	0.001305912	-2.55	0.00538571
-4.84	3.77009E-07	-4.38	5.66135E-06	-3.92	4.39994E-05	-3.46	0.000269801	-3	0.001349571	-2.54	0.00542184
-4.83	4.10476E-07	-4.37	5.93969E-06	-3.91	4.58728E-05	-3.45	0.000280006	-2.99	0.001394558	-2.53	0.005702683
-4.82	4.45598E-07	-4.36	6.23046E-06	-3.9	4.7821E-05	-3.44	0.00029057	-2.98	0.001440911	-2.52	0.005867295
-4.81	4.82455E-07	-4.35	6.53419E-06	-3.89	4.98466E-05	-3.43	0.000301503	-2.97	0.001488667	-2.51	0.006036107
-4.8	5.21128E-07	-4.34	6.85143E-06	-3.88	5.19526E-05	-3.42	0.000312817	-2.96	0.001537862	-2.5	0.006209211
-4.79	5.61703E-07	-4.33	7.18274E-06	-3.87	5.4142E-05	-3.41	0.000324525	-2.95	0.001588534	-2.49	0.006386696
-4.78	6.04268E-07	-4.32	7.5287E-06	-3.86	5.64177E-05	-3.4	0.00033664	-2.94	0.001640724	-2.48	0.006568657
-4.77	6.48918E-07	-4.31	7.88994E-06	-3.85	5.87829E-05	-3.39	0.000349173	-2.93	0.001694471	-2.47	0.006755186
-4.76	6.95749E-07	-4.3	8.26709E-06	-3.84	6.1241E-05	-3.38	0.000362138	-2.92	0.001749816	-2.46	0.00694638
-4.75	7.44863E-07	-4.29	8.66082E-06	-3.83	6.37954E-05	-3.37	0.00037555	-2.91	0.001806802	-2.45	0.007142336
-4.74	7.96366E-07	-4.28	9.0718E-06	-3.82	6.64494E-05	-3.36	0.00038942	-2.9	0.001865469	-2.44	0.007343152
-4.73	8.5037E-07	-4.27	9.50075E-06	-3.81	6.92068E-05	-3.35	0.000403765	-2.89	0.001925863	-2.43	0.007548928
-4.72	9.06989E-07	-4.26	9.94841E-06	-3.8	7.20713E-05	-3.34	0.000418599	-2.88	0.001988028	-2.42	0.007759766
-4.71	9.66344E-07	-4.25	1.04156E-05	-3.79	7.50467E-05	-3.33	0.000433936	-2.87	0.002052009	-2.41	0.007975769
-4.7	1.02856E-06	-4.24	1.0903E-05	-3.78	7.81371E-05	-3.32	0.000449793	-2.86	0.002117853	-2.4	0.00819704
-4.69	1.09377E-06	-4.23	1.14115E-05	-3.77	8.13465E-05	-3.31	0.000466185	-2.85	0.002185608	-2.39	0.008423686
-4.68	1.16212E-06	-4.22	1.1942E-05	-3.76	8.46792E-05	-3.3	0.000483128	-2.84	0.002255321	-2.38	0.008655814
-4.67	1.23374E-06	-4.21	1.24954E-05	-3.75	8.81397E-05	-3.29	0.00050064	-2.83	0.002327042	-2.37	0.008893533
-4.66	1.30878E-06	-4.2	1.30726E-05	-3.74	9.17323E-05	-3.28	0.000518738	-2.82	0.002400822	-2.36	0.009136953
-4.65	1.3874E-06	-4.19	1.36745E-05	-3.73	9.54619E-05	-3.27	0.000537439	-2.81	0.002476713	-2.35	0.009386186
-4.64	1.46976E-06	-4.18	1.43022E-05	-3.72	9.93332E-05	-3.26	0.000556762	-2.8	0.002554766	-2.34	0.009641346
-4.63	1.55604E-06	-4.17	1.49567E-05	-3.71	0.000103351	-3.25	0.000576725	-2.79	0.002635035	-2.33	0.009902547
-4.62	1.6464E-06	-4.16	1.56391E-05	-3.7	0.000107521	-3.24	0.000597348	-2.78	0.002717576	-2.32	0.010169905
-4.61	1.74104E-06	-4.15	1.63504E-05	-3.69	0.000111848	-3.23	0.00061865	-2.77	0.002802443	-2.31	0.010443539
-4.6	1.84014E-06	-4.14	1.70919E-05	-3.68	0.000116338	-3.22	0.000640651	-2.76	0.002889694	-2.3	0.010723566
-4.59	1.94391E-06	-4.13	1.78647E-05	-3.67	0.000120996	-3.21	0.000663372	-2.75	0.002979387	-2.29	0.01101011
-4.58	2.05255E-06	-4.12	1.86701E-05	-3.66	0.000125828	-3.2	0.000686834	-2.74	0.00307158	-2.28	0.01130329
-4.57	2.16628E-06	-4.11	1.95094E-05	-3.65	0.00013084	-3.19	0.000711059	-2.73	0.003166335	-2.27	0.011603232
-4.56	2.28533E-06	-4.1	2.03839E-05	-3.64	0.000136039	-3.18	0.000736069	-2.72	0.003263712	-2.26	0.011910061
-4.55	2.40993E-06	-4.09	2.1295E-05	-3.63	0.00014143	-3.17	0.000761888	-2.71	0.003363774	-2.25	0.012223903

Tabel fungsi kumulatif distribusi Normal Standart

x	cdf								
-2.24	0.012544886	-1.78	0.037537101	-1.32	0.093416319	-0.86	0.194893261	-0.4	0.344577372
-2.23	0.012873141	-1.77	0.038362684	-1.31	0.095096722	-0.85	0.197661286	-0.39	0.348267401
-2.22	0.013208798	-1.76	0.039203009	-1.3	0.096799284	-0.84	0.20045294	-0.38	0.351971848
-2.21	0.013551989	-1.75	0.040058256	-1.29	0.098524124	-0.83	0.203268142	-0.37	0.355690399
-2.2	0.01390285	-1.74	0.0409286	-1.28	0.100271358	-0.82	0.206106808	-0.36	0.359422734
-2.19	0.014261515	-1.73	0.041814222	-1.27	0.1020411	-0.81	0.208968846	-0.35	0.363168529
-2.18	0.014628122	-1.72	0.042715297	-1.26	0.103833462	-0.8	0.211854161	-0.34	0.366927458
-2.17	0.015002808	-1.71	0.043632006	-1.25	0.10564855	-0.79	0.214762651	-0.33	0.370699189
-2.16	0.015385714	-1.7	0.044564524	-1.24	0.107486469	-0.78	0.217694209	-0.32	0.374483388
-2.15	0.015776981	-1.69	0.045513032	-1.23	0.109347321	-0.77	0.220648723	-0.31	0.378279715
-2.14	0.016176751	-1.68	0.046477705	-1.22	0.111231202	-0.76	0.223626074	-0.3	0.382087829
-2.13	0.016585168	-1.67	0.047458721	-1.21	0.113138207	-0.75	0.226626139	-0.29	0.385907384
-2.12	0.017002378	-1.66	0.048456258	-1.2	0.115068427	-0.74	0.22964879	-0.28	0.389738033
-2.11	0.017428527	-1.65	0.049470493	-1.19	0.117021949	-0.73	0.232693891	-0.27	0.393579422
-2.1	0.017863764	-1.64	0.050501601	-1.18	0.118998858	-0.72	0.235761302	-0.26	0.397431197
-2.09	0.018308237	-1.63	0.051549759	-1.17	0.120999231	-0.71	0.238850879	-0.25	0.401292999
-2.08	0.018762097	-1.62	0.052615141	-1.16	0.123023147	-0.7	0.241962469	-0.24	0.405164469
-2.07	0.019225496	-1.61	0.053697924	-1.15	0.125070677	-0.69	0.245095918	-0.23	0.40904524
-2.06	0.019698588	-1.6	0.05479828	-1.14	0.127141889	-0.68	0.248251061	-0.22	0.412934948
-2.05	0.020181527	-1.59	0.055916384	-1.13	0.129236848	-0.67	0.251427733	-0.21	0.416833223
-2.04	0.020674467	-1.58	0.057052407	-1.12	0.131355615	-0.66	0.25462576	-0.2	0.420739693
-2.03	0.021177568	-1.57	0.058206523	-1.11	0.133498245	-0.65	0.257844964	-0.19	0.424653983
-2.02	0.021690985	-1.56	0.0593789	-1.1	0.13566479	-0.64	0.261085161	-0.18	0.428575718
-2.01	0.022214879	-1.55	0.060569711	-1.09	0.1378553	-0.63	0.264346161	-0.17	0.432504518
-2	0.02274941	-1.54	0.061779123	-1.08	0.140069816	-0.62	0.267627771	-0.16	0.436440002
-1.99	0.023294739	-1.53	0.063007303	-1.07	0.142308379	-0.61	0.27092979	-0.15	0.440381789
-1.98	0.023851029	-1.52	0.06425442	-1.06	0.144571023	-0.6	0.274252013	-0.14	0.444329493
-1.97	0.024418443	-1.51	0.065520637	-1.05	0.146857779	-0.59	0.277594229	-0.13	0.448282727
-1.96	0.024997146	-1.5	0.06680612	-1.04	0.149168672	-0.58	0.280956222	-0.12	0.452241104
-1.95	0.025587303	-1.49	0.06811103	-1.03	0.151503723	-0.57	0.284337771	-0.11	0.456204234
-1.94	0.026189082	-1.48	0.069435528	-1.02	0.15386295	-0.56	0.287738651	-0.1	0.460171725
-1.93	0.026802649	-1.47	0.070779775	-1.01	0.156246365	-0.55	0.291158629	-0.09	0.464143186
-1.92	0.027428172	-1.46	0.072143929	-1	0.158653974	-0.54	0.294597468	-0.08	0.468118223
-1.91	0.028065822	-1.45	0.073528145	-0.99	0.161085779	-0.53	0.298054928	-0.07	0.472096442
-1.9	0.028715768	-1.44	0.074932579	-0.98	0.163541779	-0.52	0.30153076	-0.06	0.476077446
-1.89	0.029378181	-1.43	0.076357382	-0.97	0.166021967	-0.51	0.305024714	-0.05	0.480660839
-1.88	0.030053233	-1.42	0.077802707	-0.96	0.168526329	-0.5	0.308536533	-0.04	0.484046225
-1.87	0.030741096	-1.41	0.079268702	-0.95	0.171054849	-0.49	0.312065955	-0.03	0.488033205
-1.86	0.031441943	-1.4	0.080755514	-0.94	0.173607504	-0.48	0.315612713	-0.02	0.492021381
-1.85	0.032155947	-1.39	0.082263287	-0.93	0.176184267	-0.47	0.319176537	-0.01	0.496010355
-1.84	0.032883284	-1.38	0.083792165	-0.92	0.178785106	-0.46	0.32275715	0	0.499999728
-1.83	0.033624127	-1.37	0.085342288	-0.91	0.181409983	-0.45	0.326354272		
-1.82	0.034378653	-1.36	0.086913793	-0.9	0.184058855	-0.44	0.329967618		
-1.81	0.035147037	-1.35	0.088506817	-0.89	0.186731675	-0.43	0.333596897		
-1.8	0.035929455	-1.34	0.090121493	-0.88	0.189428389	-0.42	0.337241816		
-1.79	0.036726084	-1.33	0.091757951	-0.87	0.192148939	-0.41	0.340902075		