

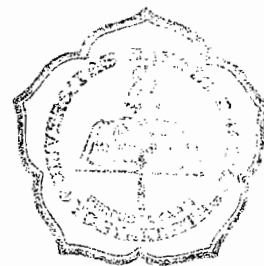
**SIMULASI OPERASI PERBEKALAN (*INVENTORY*)
DENGAN PERMINTAAN YANG BERUBAH DENGAN WAKTU**

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Jurusan Teknik Informatika



Oleh :
Ismiasih
NIM : 005314019



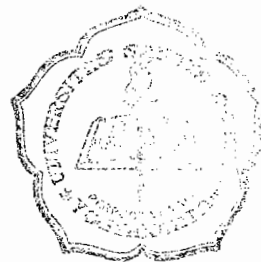
**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2005**

**SIMULATION OF INVENTORY OPERATION
WITH VOLATILE DEMAND WITH TIME**

SKRIPSI



Oleh :
Ismiasih
NIM : 005314019



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2005**

HALAMAN PERSETUJUAN

SKRIPSI

**SIMULASI OPERASI PERBEKALAN (*INVENTORY*)
DENGAN PERMINTAAN YANG BERUBAH DENGAN WAKTU**

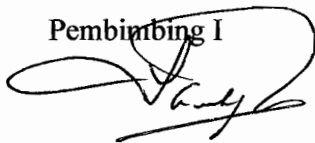
Oleh :

Ismiasih

NIM : 005314019

Telah disetujui oleh :

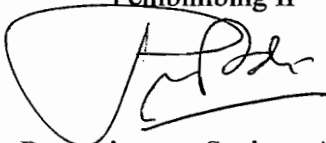
Pembimbing I



Ir. F. Soesianto, B.Sc., Ph.D.

Tanggal, 17 - 5 - 2005

Pembimbing II



Puspaningtyas Sanjoyo Adi, S.T.

Tanggal, 3 - 05 - 2005

HALAMAN PENGESAHAN

SKRIPSI

**SIMULASI OPERASI PERBEKALAN (*INVENTORY*)
DENGAN PERMINTAAN YANG BERUBAH DENGAN WAKTU**

Dipersiapkan dan ditulis oleh :

Ismiasih

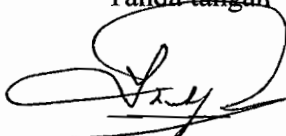
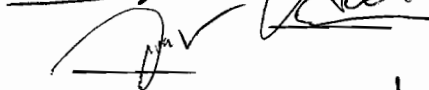

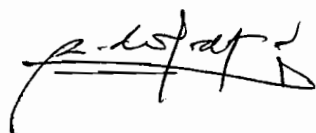
NIM: 0053141019

Telah dipertahankan di depan panitia penguji

pada tanggal 18 Mei 2005

dan dinyatakan memenuhi syarat

Susunan Panitia Penguji :

| Nama lengkap | Tanda tangan |
|--|---|
| Ketua : Ir. F.Soesianto, B.Sc., Ph.D. |  |
| Sekretaris : Puspaningtyas Sanjoyo Adi, S.T. |  |
| Anggota : JB. Budi Darmawan, S.T., M.Sc. |  |
| Anggota : Ridowati Gunawan, S.Kom., M.T. |  |

Yogyakarta, Mei 2005

Fakultas Teknik

Universitas Sanata Dharma

Dekan,




(Ir. Greg Heliarko, S.J., S.S., B.S.T., M.A., M.Sc.)

HALAMAN PERSEMBAHAN

Skripsi ini kupersembahkan untuk:

Almamaterku tercinta, Universitas Sanata Dharma

Ayah (alm) dan Ibu(Alm)

Kakak-kakakku tercinta

Keponakanku tercinta

PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, Mei 2005

Penulis

(Ismiasih)

ABSTRAK

Tujuan aplikasi ini, membuat sebuah program bantu dalam pengoptimalan perencanaan pemesanan (*order*) dari bagian *inventory* ke pemasok atas suatu barang tertentu menggunakan model *Economic Order Quantity (EOQ)*.

Permasalahan yang dibahas adalah bagaimana mendesain sebuah program bantu untuk perencanaan pemesanan (*order*) yang optimal atas suatu barang tertentu dari bagian *inventory* ke pemasok menggunakan model *Economic Order Quantity (EOQ)*.

Metodologi penelitian yang dipergunakan untuk menyelesaikan masalah ada empat yaitu *studi literature*, perencanaan program, pembuatan program dan analisis serta pengujian program.

Hasil yang dicapai, aplikasi ini dapat dijadikan tolak ukur perencanaan pemesanan (*order*) barang dari bagian *inventory* ke pemasok. Model *Economic Order Quantity (EOQ)* membantu mengoptimalkan kuantitas/jumlah barang yang dipesan dengan meminimalkan total biaya *inventory*.

Tools yang digunakan untuk pembuatan program : visual basic 6.0

ABSTRACT

The objective of this application is to make an aid program in optimizing order plan from inventory division to supplier of certain commodity by using Economic Order Quantity (EOQ) model.

The problem to be discussed is how to design an aid program for planning optimal order on certain commodity from inventory division to supplier by using Economic order Quantity (EOQ) model.

The research methodology used to solve problem are literature study, program planning, program creation and analysis, and program assessment.

The result showed that this application could be used as standardized order planning of commodity from inventory division to supplier. Economic Order Quantity (EOQ) model assisted in optimizing the quantity of ordered commodity by minimizing total inventory cost.

Program was created by using visual basic 6.0.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Kasih, berkat rahmat dan kasihNya sehingga laporan skripsi ini dapat terselesaikan dengan baik.

Skripsi dengan judul “ SIMULASI OPERASI PERBEKALAN (*INVENTORY*) DENGAN PERMINTAAN YANG BERUBAH DENGAN WAKTU” ini diharapkan dapat bermanfaat bagi bagian *inventory* sebagai tolak ukur perencanaan pemesanan barang dari bagian *inventory* ke pemasok.

Dalam proses pengerjaan akhir ini, penulis mendapatkan bantuan dan bimbingan dari berbagai pihak, maka pada kesempatan ini penulis hendak mengucapkan terima kasih kepada :

1. Romo Ir. Greg. Heliarko, S.J., S.S., B.S.T., M.A., M.Sc. selaku Dekan Fakultas Teknik.
2. Ibu AM. Polina, S.kom., M.Sc. selaku Ketua Jurusan Teknik Informatika. Terima kasih atas bantuan dan bimbingan selama ini.
3. Bapak Ir. F. Soesianto, B.Sc., Ph.D selaku Dosen Pembimbing I. Terima kasih atas kesabaran, curahan perhatian, dukungan dan bimbingan yang telah diberikan selama ini.
4. Bapak Puspaningtyas Sanjoyo Adi, S.T. selaku Dosen Pembimbing II. Terima kasih atas kesabaran, curahan perhatian, dukungan dan bimbingan yang telah diberikan selama ini.
5. Bapak DS. Bambang Soelistijanto, S.T. selaku Dosen Wali. Terima kasih untuk bimbingannya.

6. Bapak Donny Ariananda, S.T. yang pernah jadi Dosen Wali, makasih untuk nasehat-nasehatnya.
7. Staff laboratorium: Pak Bele, Mas Danang, Mbak Sri, Mas Catur.
8. Mbak Mur, Mas Timbul, Mas Kardi, Mbak Pri “Bu Titik”, Mas Supri, yang selalu memberi dukungan dan doa, trima kasih untuk kesabaran, kasih sayang dan perhatian yang selalu diberikan. Michael “kin-kin” yang selalu membuat hatiku terhibur.
9. Teman-teman TI’00, Reni “makasih banget untuk semuanya ya, aku selalu ngrepotin kamu, trims untuk dukungan dan doannya”, Rosa “trims banget untuk dukungan ,doa dan bantunnya ”., Nana “tetap semangat”, Sundari, Pina, Bowo “gajian bagi-bagi dong”, Arif, Wawan “gundul”, Daniel Adi “nggak muncul-muncul kemana?”, Agung, Melly “makasih untuk bukunya dan dukungannya”. Teman-teman TI’99 Hestu, Eko, Iin Colona, Iin, Ona.
10. Semua pihak yang telah membantu dan memberikan dukungan selama penulisan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

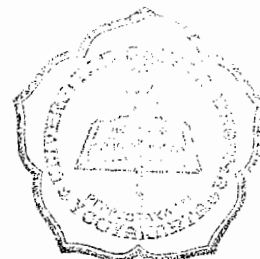
Akhirnya penulis berharap semoga laporan ini dapat bermanfaat bagi yang membacanya.

Yogyakarta,

Penulis

DAFTAR ISI

| | Hal |
|--|-------|
| HALAMAN JUDUL..... | i |
| HALAMAN PERSETUJUAN..... | ii |
| HALAMAN PENGESAHAN..... | iii |
| HALAMAN PERSEMBAHAN..... | iv |
| PERNYATAAN KEASLIAN KARYA..... | v |
| ABSTRAK..... | vi |
| ABSTRACT..... | viii |
| KATA PENGANTAR..... | viii |
| DAFTAR ISI | x |
| DAFTAR TABEL | ixiii |
| DAFTAR GAMBAR..... | xiv |
| BAB I PENDAHULUAN | |
| 1.1 Latar Belakang Masalah | 1 |
| 1.2 Tujuan | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Rumusan Masalah | 3 |
| 1.5 Metodologi Penelitian | 3 |
| 1.6 Sistematika Penulisan..... | 3 |
| BAB II LANDASAN TEORI | |
| 2.1 Tinjauan Umum <i>Inventory</i> | 5 |



| | | |
|-------|--|----|
| 2.2 | <i>EOQ (Economic Order Quantity)</i> | 6 |
| 2.2.1 | Tinjauan Umum <i>EOQ</i> | 6 |
| 2.2.2 | Rumus – Rumus yang digunakan | 8 |
| 2.3 | Diagram Arus Data (DAD)..... | 13 |
| 2.4 | Kamus Data | 15 |
| 2.5 | Bagan Alir Program | 16 |
| 2.6 | Algoritma Pembangkitan Bilangan <i>Random</i> (Acak) | |
| | Distribusi Normal | 17 |
| 2.7 | Sekilas Visual Basic 6.0 | 18 |

BAB III ANALISIS DAN PERANCANGAN

| | | |
|-------|---|----|
| 3.1 | Analisis Sistem | 20 |
| 3.1.1 | Analisis Kebutuhan Sistem | 20 |
| 3.1.2 | Pemodelan <i>Use Case (Use case Modelling)</i> . | 21 |
| 3.2 | Sistem yang diusulkan..... | 23 |
| 3.2.1 | Diagram Arus Data (DAD) | 23 |
| 3.2.2 | Kamus Data..... | 31 |
| 3.3 | Perancangan..... | 38 |
| 3.3.1 | Flowchart Proses | 38 |
| 3.3.2 | Perancangan Antar Muka | 47 |
| 3.4 | Spesifikasi Kebutuhan sumber Daya | 50 |
| 3.4.1 | Kebutuhan Perangkat Keras (<i>Hardware</i>) | 50 |
| 3.4.2 | Kebutuhan Perangkat Lunak (<i>Software</i>)..... | 50 |

BAB IV IMPLEMENTASI

| | | |
|-------|---------------------------------------|----|
| 4.1 | Implementasi program | 51 |
| 4.1.1 | Pembangkitan Bilangan Acak | 51 |
| 4.1.2 | Proses Perhitungan Rumus..... | 52 |
| 4.1.3 | Proses Perhitungan <i>Stock</i> | 53 |
| 4.2 | Implementasi Antarmuka..... | 57 |
| 4.2.1 | Implementasi Tampilan Awal..... | 57 |
| 4.2.2 | Implementasi Tampilan Utama..... | 58 |
| 4.2.3 | Implementasi Tampilan Info..... | 59 |

BAB V ANALISIS HASIL

| | | |
|-----|-------------------------|----|
| 5.1 | Studi Kasus..... | 60 |
| 5.2 | Analisis hasil..... | 67 |
| 5.3 | Kelebihan Program..... | 69 |
| 5.4 | Kekurangan Program..... | 69 |

BAB VI KESIMPULAN DAN SARAN

| | | |
|-----|-----------------|----|
| 6.1 | Kesimpulan..... | 70 |
| 6.2 | Saran..... | 70 |

DAFTAR PUSTAKA

LAMPIRAN A

Tabel hasil simulasi program

LAMPIRAN B

Listing Program

DAFTAR TABEL

| | Hal |
|--|-----|
| Tabel 2-1 Kebutuhan Keju 15 Minggu Terakhir..... | 11 |
| Tabel 3-1' Tabel Kesatuan Luar, <i>Input</i> , dan <i>Output</i> | 23 |
| Tabel 5-1 Kebutuhan Keju 15 Minggu Terakhir..... | 61 |
| Tabel 5-2 Hasil Perhitungan Kebijakan Robert..... | 63 |
| Tabel 5-3 Hasil Perhitungan <i>EOQ</i> Teoritis..... | 64 |

DAFTAR GAMBAR

| | Hal |
|---|-----|
| Gambar 2-1 <i>Cycle Inventor</i> | 9 |
| Gambar 2-2 Hubungan antara biaya pemesanan dan biaya penyimpanan..... | 10 |
| Gambar 3-1 Diagram <i>Use Case</i> | 22 |
| Gambar 3-2 Context Diagram | 23 |
| Gambar 3-3 Diagram Berjenjang | 24 |
| Gambar 3-4 DAD level 0 | 25 |
| Gambar 3-5 DAD level 1 Proses 1 | 26 |
| Gambar 3-6 DAD level 1 Proses 2 | 27 |
| Gambar 3-7 DAD level 1 Proses 3 | 28 |
| Gambar 3-8 DAD level 1 Proses 4 | 29 |
| Gambar 3-9 DAD level 1 Proses 5 | 29 |
| Gambar 3-10 DAD Keseluruhan | 30 |
| Gambar 3-11 <i>Flowchart</i> Proses 2p | 38 |
| Gambar 3-12 <i>Flowchart</i> Proses 3.1p | 39 |
| Gambar 3-13 <i>Flowchart</i> Proses 3.2p | 40 |
| Gambar 3-14 <i>Flowchart</i> Proses 3.3p | 40 |
| Gambar 3-15 <i>Flowchart</i> Proses 3.4p | 41 |
| Gambar 3-16 <i>Flowchart</i> Proses 3.5p | 41 |
| Gambar 3-17 <i>Flowchart</i> Proses 4p | 42 |
| Gambar 3-18 <i>Flowchart</i> Proses 5.1p | 45 |
| Gambar 3-19 <i>Flowchart</i> Proses 5.2p | 45 |

| | |
|--|-----|
| Gambar 3-22 <i>Flowchart</i> Proses 5.3p..... | 46 |
| Gambar 3-23 Perancangan Tampilan Awal | 47 |
| Gambar 3-24 Perancangan Tampilan Utama | 48 |
| Gambar 3-25 Perancangan Tampilan Info | 49- |
| Gambar 4-1 Form Tampilan Awal..... | 57 |
| Gambar 4-2 Form Tampilan Utama..... | 58 |
| Gambar 4-3 Form Tampilan Info..... | 59 |
| Gambar 5-1 Uji Coba Periode Permintaan Harian..... | 65 |
| Gambar 5-2 Uji Coba Periode Permintaan Mingguan..... | 66 |
| Gambar 5-3 Uji Coba Periode Permintaan Bulanan..... | 66 |

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Bagian perbekalan (*Inventory*) merupakan salah satu bagian yang penting dalam suatu perusahaan atau instansi. Perusahaan manufaktur menyediakan *stock* barang diantaranya adalah bahan mentah. Instansi seperti rumah sakit menyediakan *stock* obat-obatan, perlengkapan operasi, perlengkapan istirahat pasien dan sebagainya. Toko atau swalayan menyediakan *stock* bahan kebutuhan sehari-hari.

Inventory bagi perusahaan atau instansi perlu diperhatikan karena (www.caam.rice.edu) :

- a. Adanya perbedaan antara permintaan dan pemasok (*supplier*).
- b. Penyimpanan dalam *inventory* dapat berperan untuk mengurangi biaya produksi.
- c. *Inventory* suatu cara untuk menyimpan tenaga.
- d. *Inventory* dapat menyediakan pelayanan lebih cepat.

Stock yang cukup merupakan faktor penting bagi kelancaran proses produksi atau proses pelayanan. *Stock* barang yang terlalu tinggi mengakibatkan tingginya biaya penyimpanan, meskipun dari segi kelancaran proses, keadaan tersebut adalah positif. Jika terjadi kekurangan *stock* barang maka jalannya proses produksi menjadi tidak lancar atau pelayanan menjadi tidak cepat. Bagian *inventory* perlu menentukan kebijakan dalam penyediaan *stock* barang yang tepat

dalam arti tidak mengganggu proses produksi atau proses pelayanan selain itu biaya yang ditanggung bagian *inventory* tidak tinggi.

Perkembangan teknologi komputer yang pesat sehingga mendukung penggunaan dan pemanfaatan informasi secara luas hampir di semua bidang kehidupan. Salah satunya adalah di bagian *inventory*. Komputer sudah banyak digunakan di bagian *inventory*, untuk menyimpan data, mengolah data atau mendapatkan informasi. Hal tersebut menempatkan informasi sebagai kebutuhan penting yang harus dikelola dengan baik dan digunakan secara maksimal.

Berdasarkan uraian tersebut maka penulisan skripsi ini mengambil judul “ **Simulasi Operasi Perbekalan (*Inventory*) dengan permintaan yang Berubah dengan Waktu** “.

1.2 Tujuan

Membuat sebuah program bantu dalam pengoptimalan perencanaan pemesanan (*order*) dari bagian *inventory* ke pemasok atas suatu barang tertentu menggunakan model *Economic Order Quantity (EOQ)*.

1.3 Batasan Masalah

- a. Sistem dibatasi pada perencanaan pemesanan (*order*) dari bagian *inventory* ke pemasok atas suatu barang tertentu menggunakan model *EOQ*.
- b. Sistem yang dibuat berfungsi sebagai *simulator stock* gudang
- c. Asumsi permintaan bersifat konstan (rerata tetap).

1.4 Rumusan Masalah

Bagaimana mendesain sebuah program bantu untuk perencanaan pemesanan (*order*) yang optimal atas suatu barang tertentu dari bagian *inventory* ke pemasok menggunakan model *Economic Order Quantity (EOQ)* ?

1.5 Metodologi Penelitian

1. *Study literature* : penulis mencari dan mempelajari bahan – bahan yang mendukung dalam pembuatan program.
2. Perencanaan program : setelah mempelajari bahan – bahan yang diperlukan dalam pembuatan program, penulis mencoba untuk merancang aplikasi yang akan dibuat. Perancangan tersebut antara lain rancangan antar muka program yang akan dibuat, rancangan proses yang ada dalam program yang akan di buat serta spesifikasi kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*).
3. Pembuatan program : membuat program aplikasi berdasarkan rancangan yang telah ada.
4. Analisis dan pengujian program : menganalisis dan menguji program yang sudah jadi menghasilkan kelebihan dan kekurangan sistem.

1.6 Sistematika Penulisan

Bab I Pendahuluan

Bab ini menjelaskan latar belakang masalah, tujuan , rumusan masalah, batasan masalah, metode penelitian dan sistematika penulisan.

Bab II Landasan Teori

Bab ini berisi mengenai dasar – dasar teori yang digunakan untuk mendukung dalam pembuatan program. Dasar teori yang ditulis tidak dibahas secara mendetail.

Bab III Analisis dan Perancangan

Bab ini berisi mengenai analisis sistem , perancangan antar muka program yang akan dibuat, *flowchart* proses, diagram arus data, spesifikasi kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*)

Bab IV Implementasi

Bab ini berisi hasil perancangan ke dalam bentuk nyata yang dapat dipergunakan.

Bab V Analisis Hasil

Bab ini berisi peninjauan hasil yang telah dicapai secara optimal dan dikemukakan kelebihan dan kekurangan dari sistem yang dihasilkan.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran dari aplikasi yang dibuat.

BAB II

LANDASAN TEORI

2.1 Tinjauan Umum *Inventory*

Inventory atau perbekalan (Baroto, 2002) adalah segala sumber daya organisasi yang disimpan dalam antisipasinya terhadap pemenuhan permintaan. *Inventory* adalah komponen, material, atau produk jadi yang tersedia di tangan, menunggu untuk digunakan atau dijual. *Inventory* adalah bahan mentah, barang dalam proses (*work in process*), barang jadi, bahan pembantu, bahan pelengkap, komponen yang disimpan dalam antisipasinya terhadap pemenuhan permintaan.

Biaya-biaya *inventory* (Baroto , 2002) :

1. Biaya pemesanan adalah biaya yang harus dikeluarkan untuk melakukan pemesanan ke pemasok , yang besarnya tidak dipengaruhi oleh kuantitas atau jumlah pesanan.

Biaya ini meliputi :

- a. Biaya pemrosesan pesanan
- b. Biaya ekspedisi
- c. Upah
- d. Biaya telepon/fax
- e. Biaya dokumentasi/transaksi
- f. Biaya pengepakan
- g. Biaya pemeriksaan
- h. Biaya lainnya yang tidak tergantung kuantitas pesanan.

- 2 Biaya penyimpanan adalah biaya yang dikeluarkan dalam penanganan/penyimpanan material, semi *finished product*, *sub assembly*, atau pun produk jadi. Biaya simpan tergantung dari lama penyimpanan dan jumlah yang disimpan. Biaya simpan dinyatakan dalam biaya per unit per periode.

Biaya penyimpanan meliputi berikut ini :

- a. Biaya kesempatan : penumpukan barang berarti penumpukan modal. Padahal modal ini dapat diinvestasikan pada tabungan bank atau bisnis lain. Biaya modal merupakan *opportunity cost* yang hilang karena menyimpan persediaan.
- b. Biaya simpan. Termasuk dalam biaya simpan adalah biaya sewa gudang, biaya asuransi dan pajak, biaya administrasi dan pemindahan, serta biaya kerusakan dan penyusutan.
- c. Biaya keusangan. Barang yang disimpan dapat mengalami penurunan nilai karena perubahan teknologi (misal komputer).
- d. Biaya-biaya lain yang besarnya bersifat variabel tergantung pada kuantitas barang / *item*.

2.2 *EOQ (Economic Order Quantity)*

2.2.1 Tinjauan Umum *EOQ*

EOQ (Juneau dan Coates, 2001) adalah kuantitas pesanan (*order*) dimana secara teori meminimalkan total biaya tahunan pemesanan dan biaya tahunan penyimpanan *inventory*.

Asumsi *EOQ* menurut Juneau and Coates (2001):

- a. Biaya pemesanan dan biaya penyimpanan *inventory* diketahui.
- b. Pengisian kembali *inventory* dengan cepat (pengiriman datang dalam satu waktu)
- c. Kekurangan barang (*stock*) tidak boleh terjadi
- d. Biaya pemesanan konstan untuk tiap – tiap pemesanan.
- e. Biaya penyimpanan secara langsung proportional dengan rata- rata *inventory*.
- f. Permintaan dianggap sama dan konstan.

Asumsi *EOQ* menurut Dennis dan Dennis (1991) :

- a. Hanya satu barang yang dihitung.
- b. Permintaan diketahui dengan pasti. Contoh, permintaan tahunan.
- c. Permintaan konstan setiap waktu. Contoh : 72 unit per hari, 20 unit per minggu, 85 unit per bulan.
- d. Biaya pemesanan konstan dan tidak tergantung dari kuantitas pesanan. Biaya penyimpanan per unit konstan.
- e. Seluruh kuantitas pesanan tiba atau diterima dalam satu waktu.
- f. Tidak boleh terjadi kekurangan *inventory*. Tidak ada barang pengaman.
- g. Kuantitas pesanan adalah konstan.

Asumsi *EOQ* menurut Stevenson (1986):

- a. Perhitungan hanya untuk satu barang.
- b. Permintaan tahunan diketahui.
- c. Permintaan dianggap konstan.
- d. *Lead time* konstan.

- e. Tiap-tiap pesanan diterima dalam satu kali pengiriman.
- f. Tidak ada diskon.

2.2.2 Rumus-Rumus yang Digunakan

1. Rumus kuantitas pesanan

EOQ sering dilambangkan dengan *Q*.

$$Q = \sqrt{\frac{2DC_o}{C_s}} \text{ (satuan : unit, kilogram, dll)}$$

2. Rumus frekuensi pemesanan

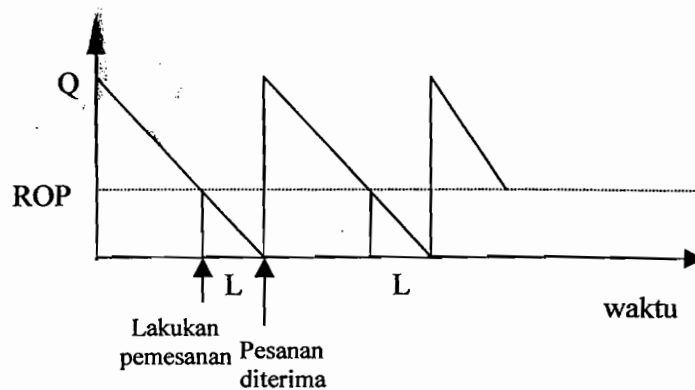
Frekuensi pemesanan dalam satu tahun (periode) adalah permintaan tahunan dibagi dengan kuantitas pesanan.

$$\text{Frekuensi } order = \frac{D}{Q} \text{ kali per tahun}$$

3. Rumus *cycle time*

Cycle time dapat diartikan sebagai jarak waktu antar pesanan

$$\text{Cycle time} = \frac{Q}{D} \text{ tahun}$$



Gambar 2-1 *Cycle Inventory*

4. Rumus *Reorder Point*

Reorder point (ROP) adalah menunjukkan suatu tingkatan *inventory* di mana saat itu harus dilakukan pesanan atau waktu saat dilakukan pesanan.

$$\text{ROP} = \text{cycle time} - L.$$

5. Rumus Total biaya *inventory*

Biaya total merupakan penjumlahan dari dua fungsi yaitu fungsi biaya penyimpanan dan biaya pemesanan.

$$\text{Total biaya } \textit{inventory} = (C_h * Q/2) + (C_o * D/Q)$$

Notasi yang digunakan :

C_h : Biaya penyimpanan (per unit barang per tahun)

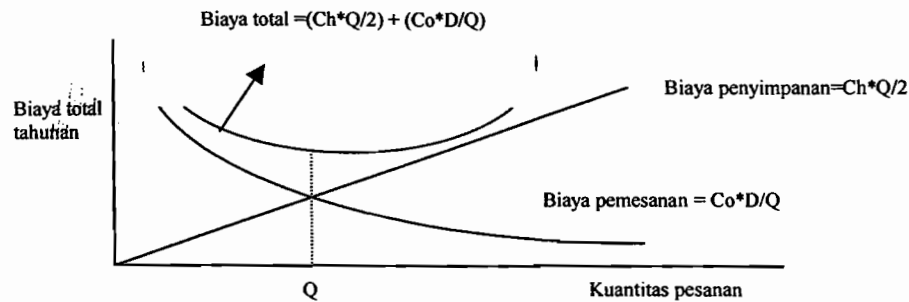
C_o : Biaya pemesanan (per 1x pesan)

D : Jumlah Permintaan / total permintaan (unit barang per tahun)

Q : Kuantitas pesanan.

ROP: *Reorder Point*

L : *Lead time* (jarak waktu antara pemesanan sampai pesanan datang/diterima)



Gambar 2-2 Hubungan antara biaya pemesanan dan biaya penyimpanan

Contoh :

Soal

Robert Pizaman memulai bisnis pizzanya 5 tahun yang lalu di Jakarta. Untuk pembuatan pizza dia memerlukan beberapa bahan. Salah satu bahan tersebut adalah bubur keju yang dia pesan dari toko milik John Cheeseman, Surabaya. Bubur keju ini merupakan bahan yang paling vital untuk pizza-nya. Untuk membuat kebijakan pemesanan (*order*), Robert Pizaman telah mengidentifikasi biaya-biaya yang terkait. Biaya-biaya tersebut adalah sebagai berikut :

- a. Harga bubur keju per kilogram = Rp 12.000,-. Harga ini merupakan suatu hasil negosiasi berupa kontrak antara Robert Pizaman dan John Cheeseman yang akan diperbaharui yang akan diperbaharui tahun depan.
- b. Biaya penyimpanan per kilogram pertahun = 25% harga beli. Jadi biaya penyimpanannya = $25\% \times \text{Rp } 12.000,- = \text{Rp } 3.000,-$ per kilogram per tahun . Persentase 25% didapat dari perhitungan *opportunity cost* (biaya kesempatan),

seandainya uang yang tertanam dipersediaan ditabung di bank. Bunga bank = 25% per tahun, menurut Robert, persentase ini akurat.

c. Biaya pemesanan (berapapun jumlah yang dipesan) = Rp 500.000,- per
1xpesan

Lead time (jarak waktu antara pemesanan sampai pesanan datang) = 4 hari.

Faktor berikutnya yang diperhitungkan Robert adalah permintaan pizza yang berfluktuasi setiap minggunya. Data permintaan pizza yang telah dikonversikan ke dalam kebutuhan bubuk keju selama 15 minggu terakhir seperti terlihat pada tabel 2-1. diduga permintaan yang akan terjadi akan berpola mirip seperti pola permintaan tersebut.

| Minggu ke- | Kilogram | Minggu ke- | Kilogram |
|------------|----------|------------|----------|
| 1 | 1.560 | 9 | 1.590 |
| 2 | 1.623 | 10 | 1.600 |
| 3 | 1.580 | 11 | 1.605 |
| 4 | 1.578 | 12 | 1.585 |
| 5 | 1.610 | 13 | 1.590 |
| 6 | 1.615 | 14 | 1.600 |
| 7 | 1.590 | 15 | 1.605 |
| 8 | 1.605 | | |
| | | Total | 23.936 |
| | | Rata-rata | 1.595 |

Tabel 2-1 Kebutuhan Keju 15 Minggu Terakhir

Tahun depan Robert memperkirakan pertumbuhan pasar pizza-nya pesat. Melihat data permintaan keju nya 15 minggu terakhir tersebut, dia memperkirakan tahun depan permintaan keju nya menjadi 1600 kilogram per minggu. Dalam setahun berarti 83.200 kilogram (52 minggu x 1600), dalam sehari 228 kilogram (83.200 per 365 hari). Hitung kuantitas pesanan, *cycle time*, frekuensi pemesanan, *reorder point*, dan total biaya *inventory*!

Jawaban :

$$\begin{aligned} \text{Kuantitas pesanan optimal bubuk keju} &= \sqrt{\frac{2xDxCo}{Ch}} \\ &= \sqrt{\frac{2x83.200x500.000}{3.000}} \\ &= 5266,2 \text{ kilogram} \end{aligned}$$

$$\begin{aligned} \text{Cycle time} &= (Q/D) \times 365 \text{ hari} \\ &= (5266,2/83.200) \times 365 \\ &= 23,1 = 23 \text{ hari} \end{aligned}$$

$$\begin{aligned} \text{Frekuensi pemesanan} &= D/Q \\ &= 83.200/5266,2 = 15,79 = 16 \text{ kali per tahun} \end{aligned}$$

$$\begin{aligned} \text{Reorder point} &= \text{cycle time} - \text{lead time} \\ &= 23 - 4 = 19 \text{ hari setelah pesanan diterima} \end{aligned}$$

$$\begin{aligned} \text{Total biaya } \textit{inventory} \text{ (1 tahun)} &= ((Q/2) \times Ch) + ((D/Q) \times Co) \\ &= ((5266,2/2) \times 3.000) + ((83.200/5266,2) \times 500.000) \\ &= \text{Rp } 15.798.734,13,- \\ &= \text{Rp } 15.798.800,- \end{aligned}$$

Dalam praktik, angka 5266,2 kilogram hasil penghitungan kuantitas pesanan mungkin sulit diterapkan, sedikit pembulatan untuk kemudahan pesanan dapat dilakukan. Misalnya dibulatkan menjadi 5000 atau 5270.

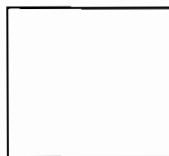
2.3 Diagram Arus Data (DAD)

Diagram arus data (DAD) digunakan untuk menggambarkan suatu sistem yang ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya file kartu, *microfiche*, *hard disk*, *tape*, disket dan lain sebagainya). DAD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured Analysis and Design*).

Beberapa simbol yang digunakan di DAD adalah :

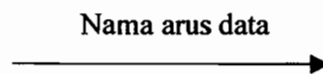
1. Kesatuan luar (*external entity*)

Kesatuan luar merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem. Kesatuan luar dapat disimbolkan sebagai berikut :



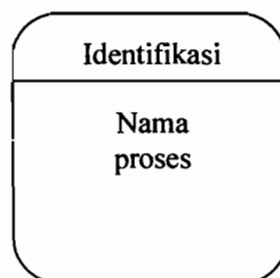
2. Arus data (*data flow*)

Arus data di DAD diberi simbol suatu anak panah. Arus data mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar. Arus data menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari sistem. Arus data diberi nama yang jelas dan mempunyai arti. Arus data dapat disimbolkan sebagai berikut :



3. Proses (*process*)

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Proses dapat disimbolkan sebagai berikut :



Pedoman menggambar DAD :

1. Identifikasi terlebih dahulu semua kesatuan luar yang terlibat di sistem.
2. Identifikasi semua *input* dan *output* yang terlibat dengan kesatuan luar.
3. Gambar terlebih dahulu suatu diagram konteks (*context diagram*). *Context diagram* merupakan penggambaran sistem pertama kali secara garis besar (*Top level*). Dari *context diagram* kemudian digambar dengan lebih rinci lagi

yang disebut dengan level 0 atau *overview diagram*. Tiap-tiap proses di level 0 digambar lebih rinci lagi dan disebut level 1. Tiap-tiap proses di level 1 digambar lebih rinci lagi dan disebut level 2 dan seterusnya sampai tiap-tiap proses tidak dapat digambar lebih rinci lagi.

4. Gambar diagram berjenjang untuk semua proses yang ada di sistem terlebih dahulu. Bagan berjenjang digunakan untuk mempersiapkan penggambaran DAD le level-level lebih bawah lagi.
5. Gambar sketsa DAD untuk level 0 berdasarkan proses di bagan berjenjang. Jika suatu proses tidak dipecah menjadi lebih rinci lagi, pada bagian identifikasi proses diberi huruf P (primitiv) setelah nomor proses.
6. Gambar DAD untuk level-level berikutnya, untuk tiap proses yang sesuai dengan bagan berjenjang.
7. Jika semua level DAD digambar, maka semua DAD dapat digabung dalam satu diagram.

2.4 Kamus Data

Kamus data adalah katalog fakta tentang data pada suatu sistem informasi. Kamus data harus memuat hal-hal berikut agar dapat mencerminkan keterangan yang jelas tentang data yang dicatat :

1. Nama arus data : harus sama dengan nama arus data yang mengalir di DAD.
2. Alias : nama lain dari nama arus data.
3. Bentuk data, dapat berupa dokumen/formulir, tampilan di layar monitor, variable, parameter.

4. Arus data, menunjukkan dari mana data mengalir dan menuju ke mana.
5. Penjelasan, berisi keterangan-keterangan tentang arus data tersebut.
6. Periode : kapan terjadinya arus data ini.
7. Volume : volume rata-rata dan volume puncak dari arus data.
8. Struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari item data.

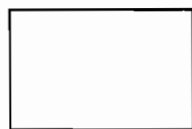
2.5 Bagan Alir Program

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dapat menggunakan simbol-simbol sebagai berikut :

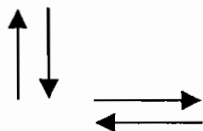
1. Simbol *input/output*, digunakan untuk mewakili data input/output.



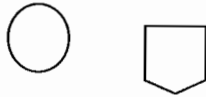
2. Simbol proses, digunakan untuk mewakili suatu proses.



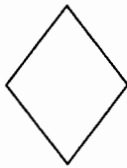
3. Symbol garis alir, digunakan untuk menunjukkan arus dari proses.



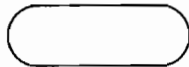
4. Simbol penghubung, digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dari halaman yang masih sama atau di halaman lainnya.



5. Simbol keputusan, digunakan untuk suatu penyelesaian kondisi di dalam program.



6. Simbol titik terminal, digunakan untuk menunjukkan awal dan akhir dari suatu proses.



2.6 Algoritma Pembangkitan Bilangan *Random* (Acak) Distribusi Normal

Pembangkitan bilangan acak dari $N(\mu, \sigma^2)$. Dengan, μ adalah rata-rata dan σ^2 adalah varians. Z adalah variabel standard normal, yang merupakan $N(0,1)$.

Algoritma pembangkitan bilangan acak menurut pendekatan Tocher (Rubinstein, 1981) :

1. Bangkitkan u_1 dan u_2 dari $U(0,1)$
2. $X \leftarrow \sqrt{\pi/8} \ln[(1+u_1)/(1-u_1)]$
3. Jika $u_2 \leq 0,5$ maka $Z = -X$
4. Selain itu $Z = X$

Rata-rata dari distribusi normal baku (standart normal) = 0 dan varians (kuadrat dari deviasi standart) sama dengan deviasi standart yaitu 1 . Tansformasi dilakukan jika diinginkan bilangan acak normal tertentu dengan rata-rata tertentu dan varians tertentu. Transformasi tersebut dengan rumus :

$$Z = (Y - \mu_Y) / \sigma_Y$$

$Y = (Z * \sigma_Y) + \mu_Y$; dengan Y adalah bilangan acak yang telah di-transformasi.

2.7 Sekilas *Visual Basic 6.0*

Visual Basic berasal dari kata *visual* yang berarti menunjukkan cara yang digunakan untuk membuat graphical user interface (GUI) dan *basic* merupakan bagian bahasa BASIC (*Beginners Allpurpose Symbolic Instruction Code*). *Visual Basic* merupakan *event-driven programming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai (*user*) berupa *event/kejadian* tertentu (tombol diklik, menu dipilih, dan lain-lain). Ketika *event* terdeteksi, kode yang berhubungan dengan *event* atau prosedur *event* akan dijalankan.

Beberapa struktur aplikasi proyek terdiri dari :

1. *Form* -- windows/jendela dimana pemakai akan membuat *user interface/tampilan*.

2. Kontrol – tampilan berbasis grafis yang dimasukkan pada *form* untuk membuat interaksi dengan pemakai (*user*) seperti *text box*, *label*, tombol *command.Form* dan kontrol adalah objek.
3. Properti – nilai/karakteristik yang dimiliki oleh sebuah objek *visual basic*.
4. Metode – serangkaian perintah yang sudah tersedia pada suatu objek yang dapat diminta untuk mengerjakan tugas tertentu.
5. Prosedur kejadian – kode yang berhubungan dengan suatu objek. Kode ini dieksekusi ketika ada respon dari pemakai berupa *event* tertentu.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Rencana dan kebijakan dalam pengoperasian *inventory* diperlukan dalam setiap periode *inventory*. Tingkat *inventory* harus tetap dijaga dengan memperhitungkan kuantitas pesanan dan waktu pemesanan dilakukan. Hal ini bertujuan untuk menjamin tersediannya *inventory* dalam jumlah/kuantitas dan waktu yang tepat, sehingga diharapkan dapat meminimalkan biaya total *inventory*.

3.1.1 Analisis Kebutuhan Sistem

Spesifikasi perangkat lunak yang akan dikembangkan adalah sebagai berikut :

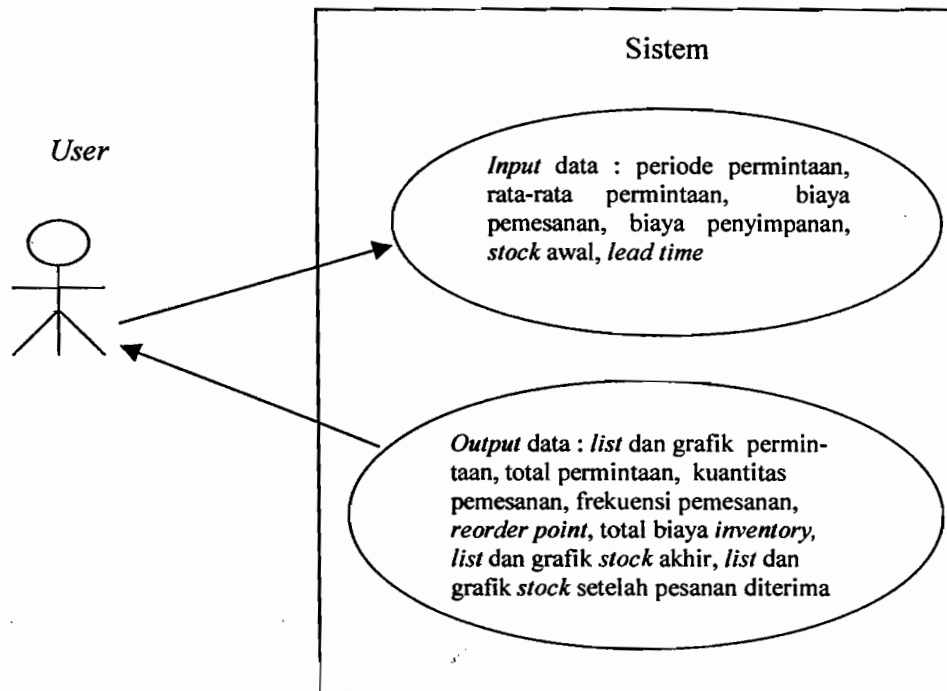
- a. *Input*-an data, terdiri dari :
 1. Periode permintaan, yang dibagi dalam tiga pilihan yaitu harian, mingguan dan bulanan .
 2. *Input*-an data lainnya adalah rata-rata permintaan, biaya pemesanan, biaya penyimpanan, *stock* awal, dan *lead time*.
- b. Jumlah data permintaan yang dipakai ditentukan oleh pilihan periode permintaannya. Periode permintaan harian, data permintaannya 365. Periode permintaan mingguan, data permintaannya 52. Periode permintaan bulanan, data permintaannya 12 .

- c. Pemesanan pertama kali diasumsikan pada hari pertama dari periode *inventory*, pemesanan selanjutnya ditentukan dengan perhitungan .
- d. Informasi(*output*) yang dihasilkan ada tiga yaitu hasil perhitungan, pencatatan dalam bentuk *list* dan *output* yang disajikan dalam bentuk grafik .
- e. User berinteraksi dengan sistem melalui *monitor*, *keyboard*, dan *mouse*.

3.1.2 Pemodelan Use Case (Use case Modelling)

Diagram *use case* adalah sebuah program yang menggambarkan interaksi antara sistem dan eksternal sistem dan *user*.

Hanya ada satu aktor yang terlibat dalam sistem yakni *user*. *User* adalah aktor yang memasukkan data-data *input* dan meminta sistem untuk mengolahnya, dan sistem akan memberikan hasilnya. Hubungan asosiasi antara aktor *user* dengan sistem berlangsung dua arah. *User* memberikan *input-an* data yaitu periode permintaan, rata-rata permintaan, biaya pemesanan, biaya penyimpanan, *stock* awal, *lead time*. Sistem akan mengolah *input* dari aktor *user*, memberikan hasilnya dalam bentuk hasil perhitungan, *list* dan grafik. Untuk hasil perhitungan yaitu total permintaan, kuantitas pesanan, frekuensi pemesanan, *cycle time*, *reorder point*, dan total biaya *inventory*. Untuk hasil grafik yaitu grafik permintaan, grafik *stock* setelah pesanan diterima dan grafik *stock* akhir. Untuk hasil *list* yaitu *list* permintaan, *list stock* akhir dan *list stock* setelah pesanan diterima. Untuk lebih jelasnya dapat dilihat gambar 3-1 Diagram *Use Case*.

Gambar 3-1 Diagram *Use case*

3.2 Sistem yang diusulkan

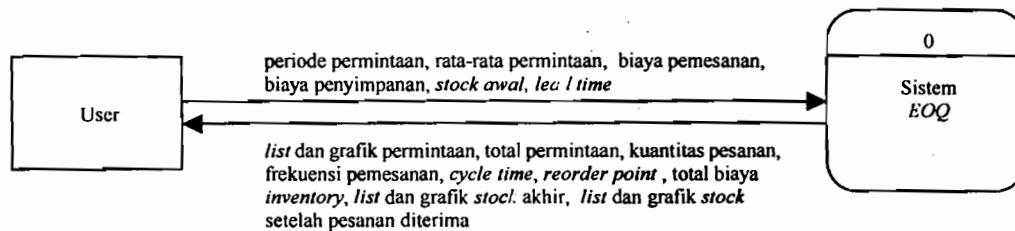
3.2.1 Diagram Arus Data (DAD)

3.2.1.1 Tabel kesatuan luar, *input* dan *output*

| Kesatuan luar | <i>Input</i> | <i>Output</i> |
|---------------|--|---|
| User | periode permintaan, rata-rata permintaan, biaya pemesanan, biaya penyimpanan, <i>stock awal</i> , <i>lead time</i> | <i>list</i> dan grafik permintaan, total permintaan, kuantitas pesanan, frekuensi pemesanan, <i>cycle time</i> , <i>reorder point</i> , total biaya <i>inventory</i> , <i>list</i> dan grafik <i>stock</i> akhir, <i>list</i> dan grafik <i>stock</i> setelah pesanan diterima. |

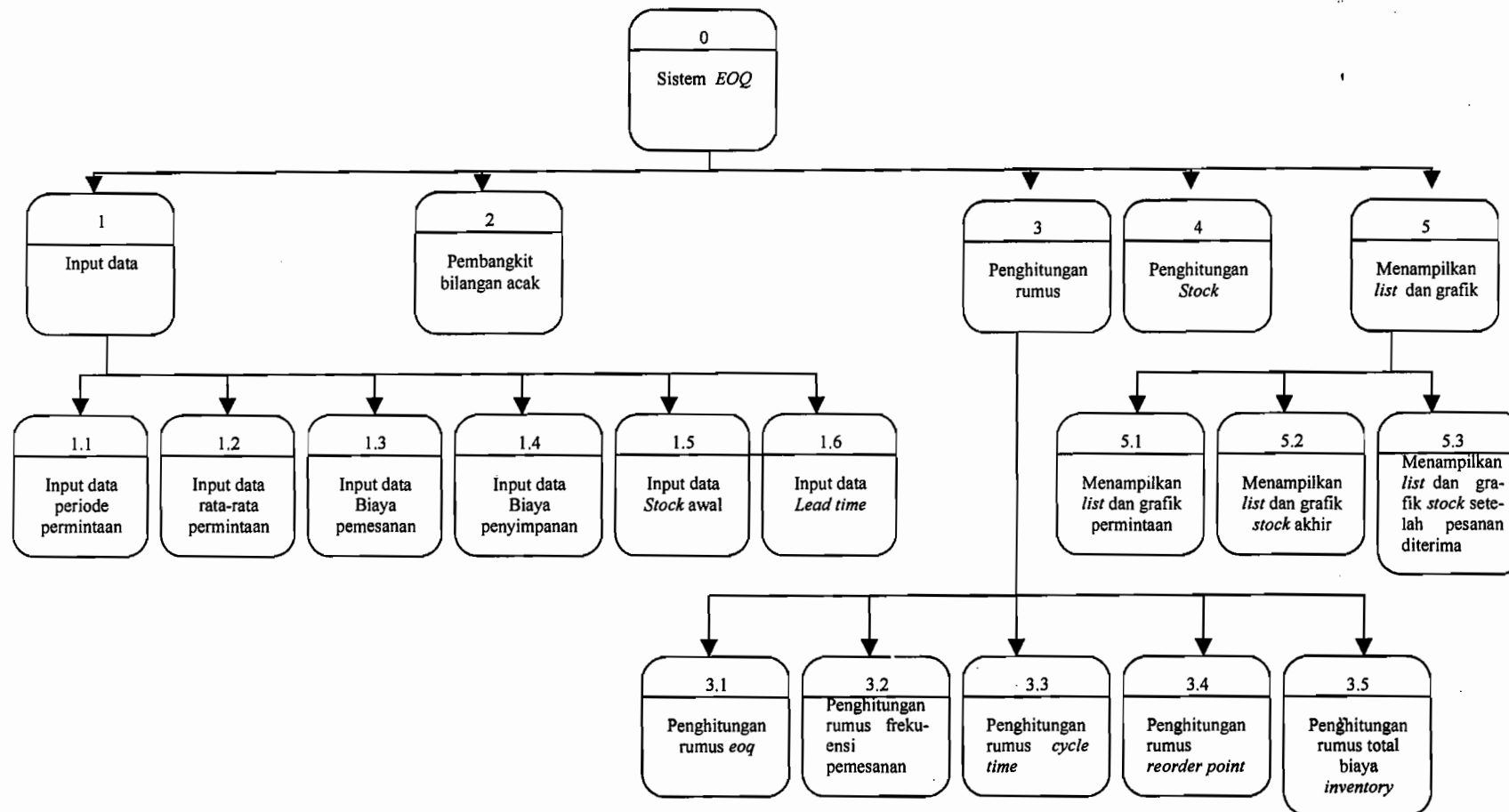
Tabel 3-1 Tabel kesatuan luar, *input* dan *output*

3.2.1.2 Context Diagram



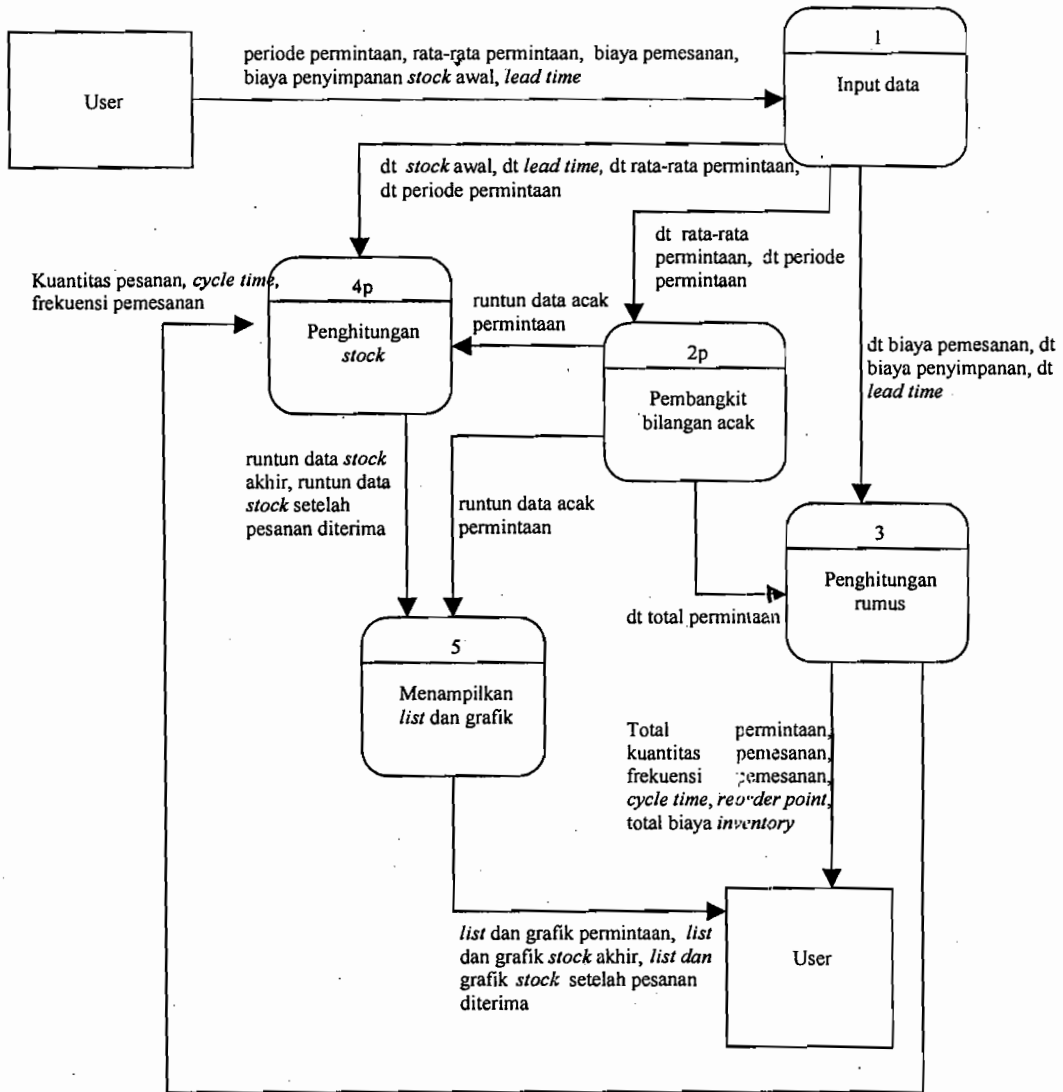
Gambar 3-2 Context Diagram

3.2.1.3 Diagram Berjenjang



Gambar 3-3 Diagram Berjenjang

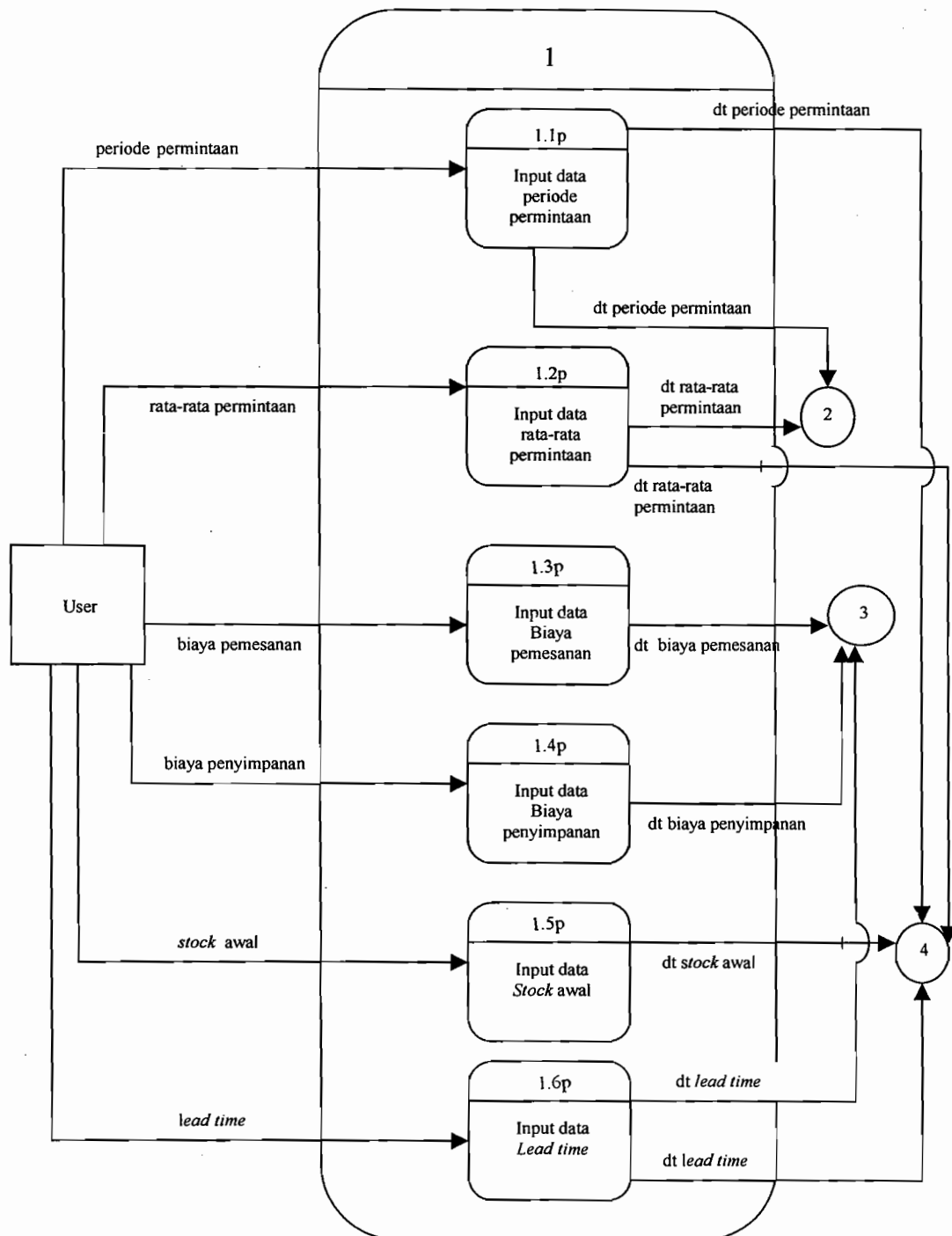
3.2.1.4 DAD level 0



Gambar 3-4 DAD Level 0

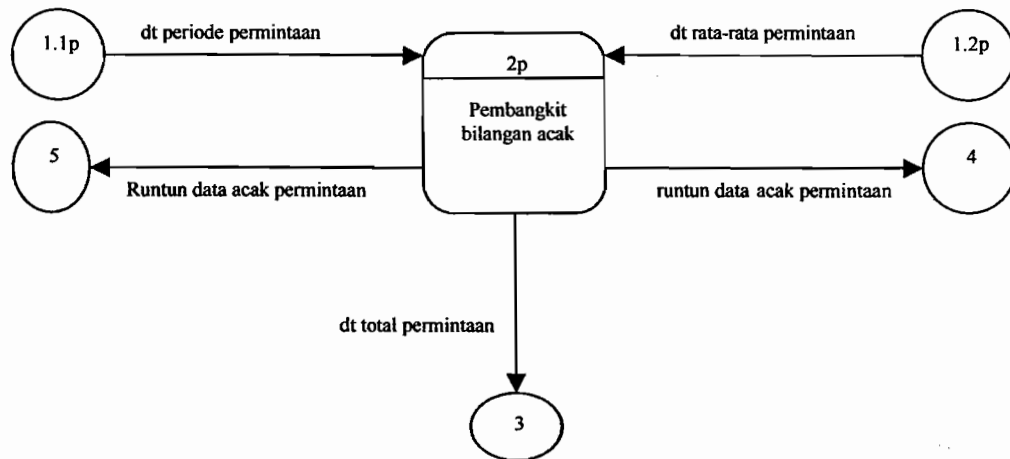


3.2.1.5 DAD level 1 Proses 1



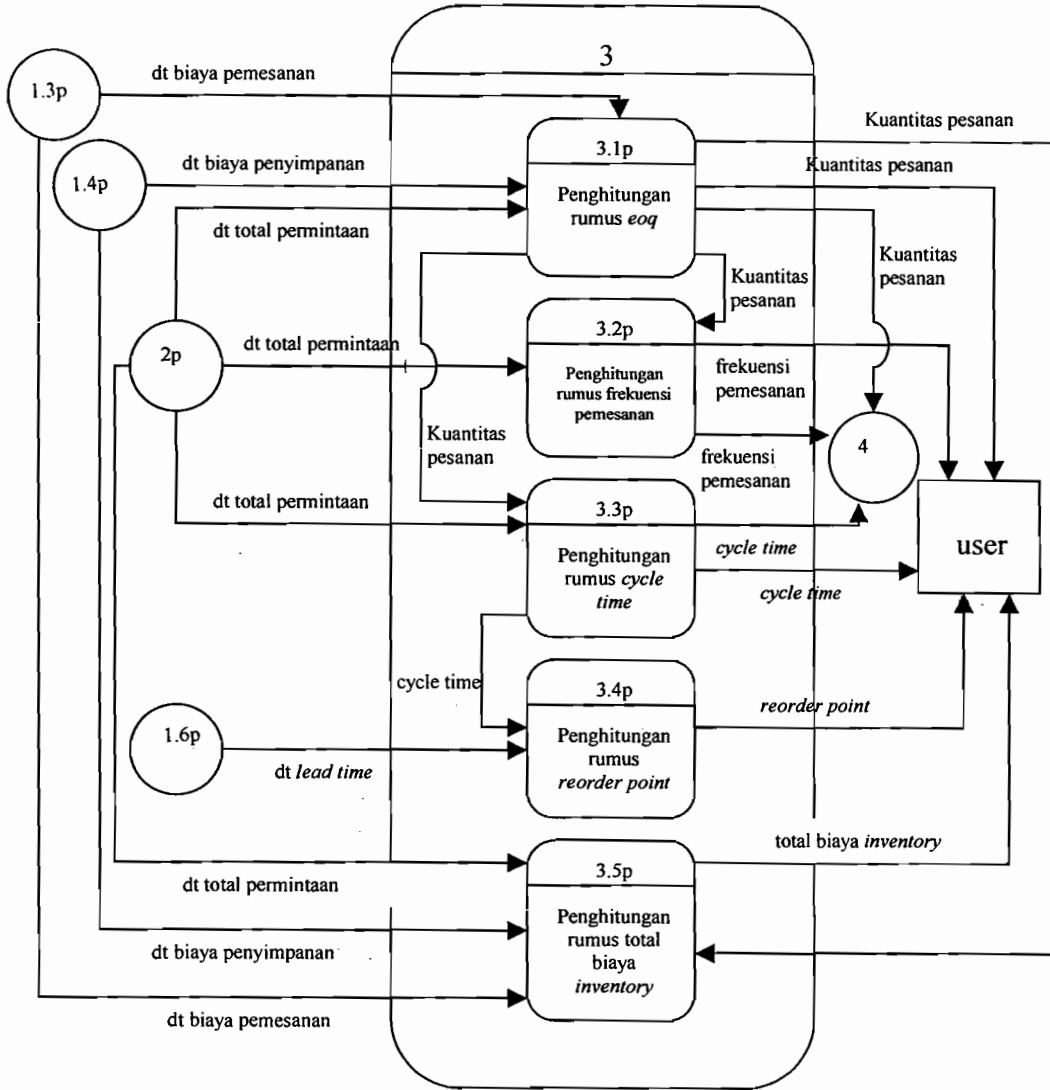
Gambar 3-5 DAD level 1 Proses 1

3.2.1.6 DAD level 1 Proses 2



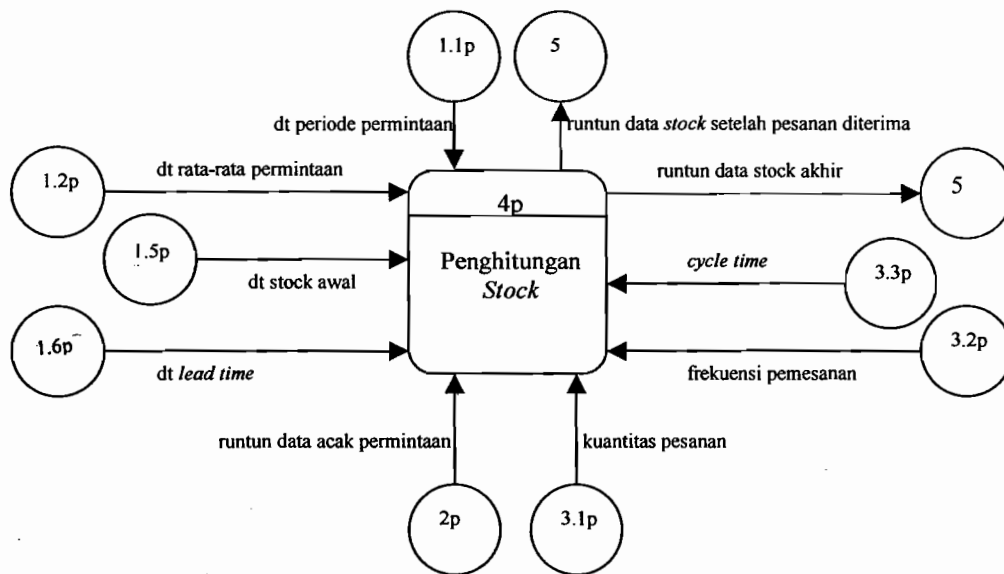
Gambar 3-6 DAD level 1 Proses 2

3.2.1.7 DAD level 1 Proses 3



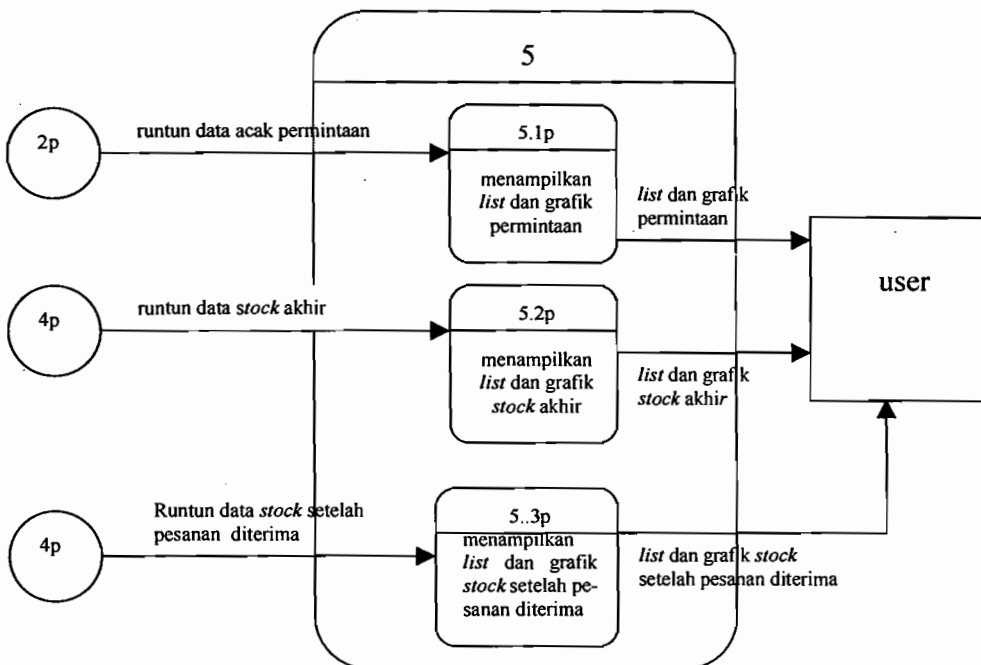
Gambar 3-7 DAD level 1 Proses 3

3.2.1.8 DAD level 1 Proses 4



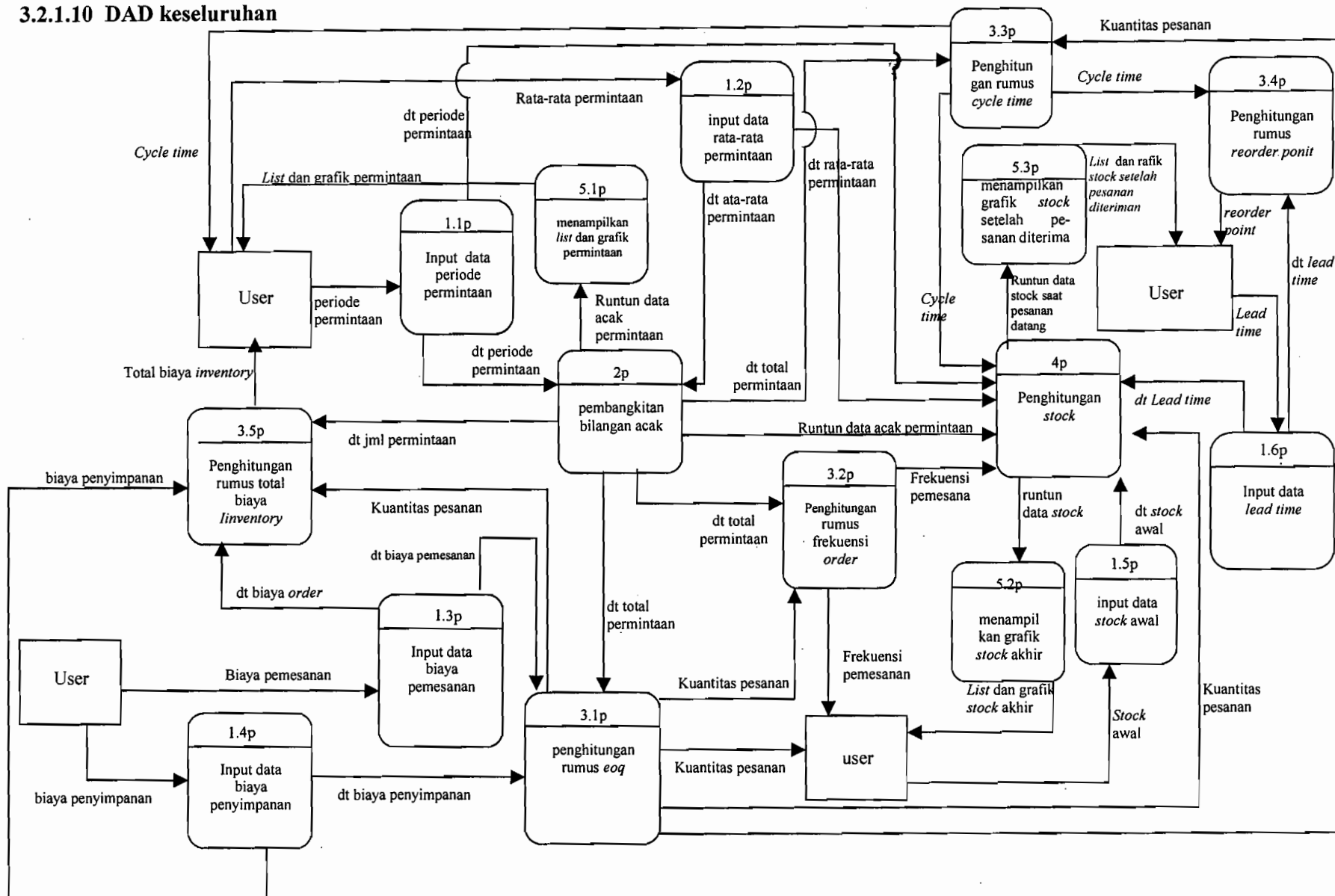
Gambar 3-8 DAD level 1 Proses 4

3.2.1.9 DAD level 1 Proses 5



Gambar 3-9 DAD level 1 Proses 5

3.2.1.10 DAD keseluruhan



Gambar 3-10 DAD Keseluruhan

3.2.2 Kamus Data

1. Nama arus data : dt periode permintaan
 - Alias : -
 - Bentuk data : variabel
 - Arus data : Proses 1.1p → Proses 2p; Proses 1.1p → Proses 4p
 - Penjelasan : dalam satu periode (satu tahun), permintaannya dalam harian, mingguan, atau bulanan
 - Periode : setiap ada *input* dari *user*
 - Volume : -
 - Struktur Data : -

2. Nama arus data : dt rata-rata permintaan
 - Alias : -
 - Bentuk data : variabel
 - Arus data : Proses 1.2p → Proses 2p, Proses 1.2p → Proses 4p
 - Penjelasan : rata – rata permintaan dalam satu periode permintaan
 - Periode : setiap ada *input* dari *user*
 - Volume : -
 - Struktur Data : -

3. Nama arus data :dt biaya pemesanan
 - Alias : -
 - Bentuk data : variabel
 - Arus data : Proses 1.3p → Proses 3.1p; Proses 1.3p → Proses 3.5p
 - Penjelasan : biaya pemesanan yang dikeluarkan untuk sekali pesan

- Periode : setiap ada *input* dari *user*
 Volume :-
 Struktur Data :-
4. Nama arus data :dt biaya penyimpanan
- Alias :-
 Bentuk data : variabel
 Arus data : Proses 1.4p → Proses 3.1p; Proses 1.4p → Proses 3.5p
 Penjelasan : biaya simpan per – barang/unit per- tahun
 Periode : setiap ada *input* dari *user*
 Volume :-
 Struktur Data :-
5. Nama arus data :dt *stock* awal
- Alias :-
 Bentuk data : variabel
 Arus data : Proses 1.5p → Proses 4p
 Penjelasan : banyaknya *stock* barang yang disediakan, sebelum operasi
inventory dijalankan
 Periode : setiap ada *input* dari *user*
 Volume :-
 Struktur Data :-
6. Nama arus data :dt *lead time*
- Alias :-
 Bentuk data : variabel

- Arus data : Proses 1.6p → Proses 3.4p; Proses 1.6p → Proses 4p
- Penjelasan : jarak waktu antara pemesanan sampai pesanan diterima
- Periode : setiap ada *input* dari *user*
- Volume : -
- Struktur Data : -
7. Nama arus data : runtun data acak permintaan
- Alias : -
- Bentuk data : runtun data
- Arus data : Proses 2p → Proses 4p; Proses 2p → Proses 5.1p
- Penjelasan : bilangan *random* terdistribusi normal
- Periode : setiap ada pembangkitan bilangan acak
- Volume : -
- Struktur Data : -
8. Nama arus data : dt total permintaan
- Alias : -
- Bentuk data : variabel
- Arus data : Proses 2p → Proses 3.1p; Proses 2p → Proses 3.3p
Proses 2p → Proses 3.5p
- Penjelasan : hasil penjumlahan dari runtun data acak permintaan
- Periode : setiap ada pembangkitan bilangan acak atau perubahan
rata-rata
- Volume : -
- Struktur Data : -

9. Nama arus data : kuantitas pesanan
- Alias : -
- Bentuk data : tampilan dilayar monitor, variabel
- Arus data : Proses 3.1p → Proses 3.2p
 Proses 3.1p → Proses3.3p
 Proses 3.1p → Proses 3.5p
 Proses 3.1p → User
- Penjelasan : kuantitas pesanan hasil perhitungan
- Periode : setiap ada proses perhitungan *EOQ*
- Volume :-
- Struktur Data :-
10. Nama arus data : frekuensi pemesanan
- Alias : -
- Bentuk data : variable, tampilan di layar monitor
- Arus data : Proses 3.2p → Proses 4p
 Proses 3.2p → User
- Penjelasan : frekuensi pemesanan dalam 1 tahun
- Periode : setiap ada proses perhitungan frekuensi pemesanan
- Volume :-
- Struktur Data :-
11. Nama arus data : *cycle time*
- Alias : -
- Bentuk data : variable, tampilan di layar monitor

- Arus data : Proses 3.3p → Proses 4p
 Proses 3.3p → Proses3.4p
 Proses 3.3p → User
- Penjelasan : jarak waktu antar pesanan
- Periode : setiap ada proses perhitungan *cycle time*
- Volume : -
- Struktur Data : -
12. Nama arus data : *reorder point*
- Alias : -
- Bentuk data : tampilan di layar monitor
- Arus data : Proses 3.4p → User
- Penjelasan : menunjukkan tingkatan *inventory* di mana saat itu harus dilakukan pesanan atau waktu saat dilakukan pesanan
- Periode : Setiap ada proses perhitungan *reorder point*
- Volume : -
- Struktur Data : -
13. Nama arus data : total biaya *inventory*
- Alias : -
- Bentuk data : tampilan dilayar monitor
- Arus data : Proses 3.5 → User
- Penjelasan : total biaya dalam satu tahun
- Periode : Setiap ada proses perhitungan total biaya *inventory*
- Volume : -

Struktur Data : -

14. Nama arus data : runtun data *stock* akhir

Alias : -

Bentuk data : runtun data

Arus data : Proses 4 p → Proses 5.2p

Penjelasan : memperlihatkan *stock* dalam kurun waktu satu tahun

Periode : Setiap ada proses penghitungan *stock*

Volume : -

Struktur Data : -

15. Nama arus data : *list* dan grafik permintaan

Alias : -

Bentuk data : tampilan di layar monitor

Arus data : Proses 5.1p → User

Penjelasan : menggambarkan permintaan dalam bentuk *list* dan grafik

Periode : Setiap ada proses pembangkitan bilangan *random*

Volume : -

Struktur Data : -

16. Nama arus data : *list* dan grafik *stock* akhir

Alias : -

Bentuk data : tampilan di layar monitor

Arus data : Proses 5.2p → User

Penjelasan : menggambarkan *stock* dalam bentuk *list* dan grafik

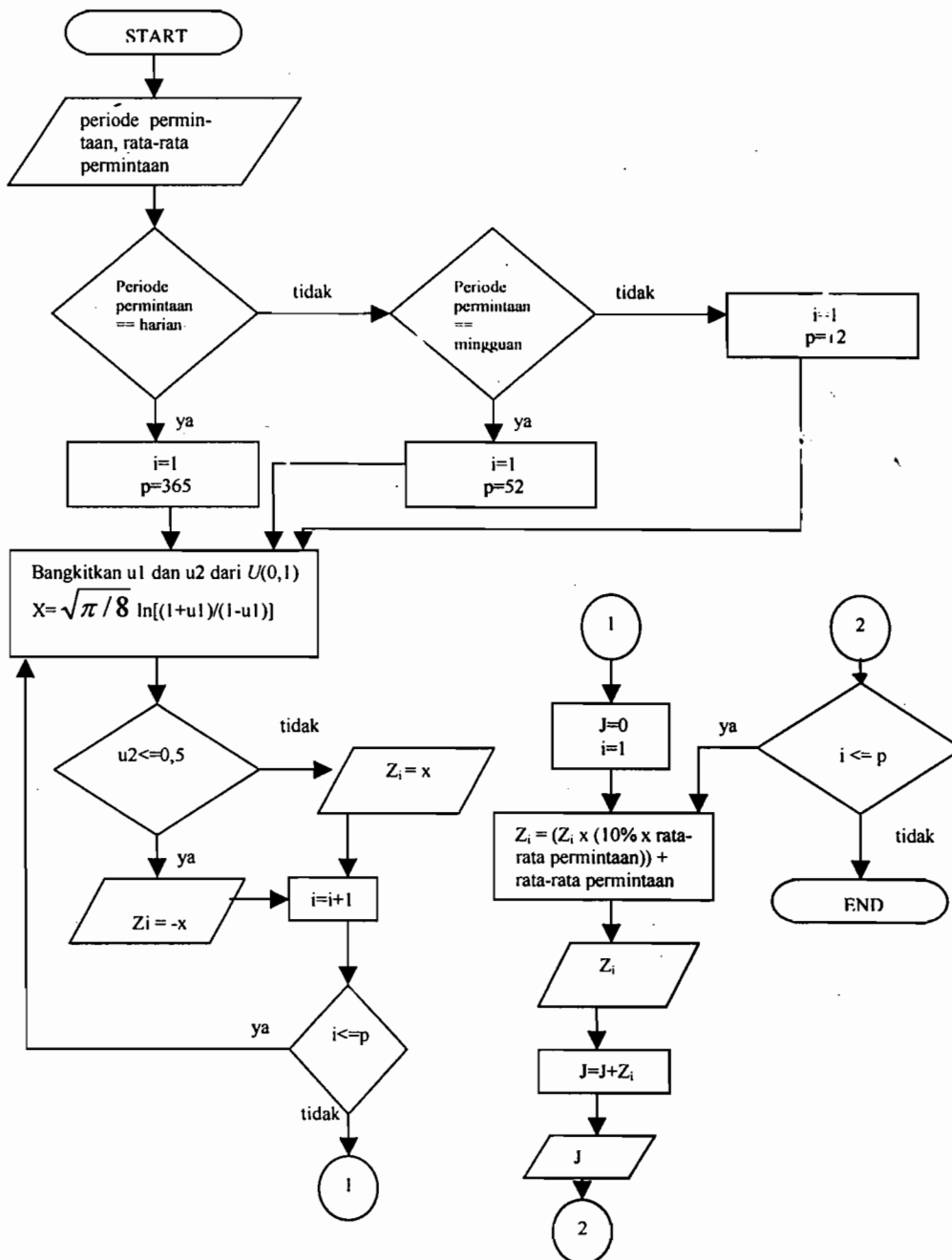
Periode : Setiap ada proses perhitungan *stock*

- Volume : -
- Struktur Data : -
17. Nama arus data : runtun data *stock* setelah pesanan diterima
- Alias : -
- Bentuk data : kumpulan data
- Arus data : Proses 4p → Proses 5.3 p
- Penjelasan : data *stock* setelah pesanan diterima
- Periode : Setiap ada *input* dari *user*
- Volume : -
- Struktur Data :-
18. Nama arus data : *list* dan grafik *stock* setelah pesanan diterima
- Alias : -
- Bentuk data : tampilan di layar monitor
- Arus data : Proses 5.3p → User
- Penjelasan : menggambarkan *stock* setelah *stock* akhir ditambah dengan barang yang diterima dan sebelum dikurangi dengan permintaan dalam bentuk *list* dan grafik
- Periode : Setiap ada pesanan yang datang/diterima.
- Volume : -
- Struktur Data : -

3.3 Perancangan

3.3.1 Flowchart Proses

3.3.1.1 Flowchart Proses 2p (Pembangkitan Bilangan Acak)



Gambar 3-11 Flowchart Proses 2p

Keterangan :

u1 & u2 : variabel

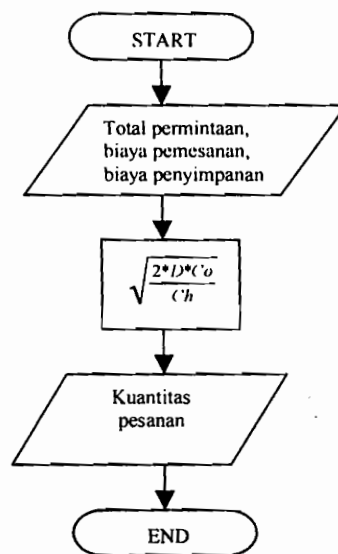
i,n: menunjukkan *counter*

p : Banyaknya data yang akan di *generate*

Z : Bilangan acak

J : Total permintaan

3.3.1.2 Flowchart Proses 3.1p (Perhitungan Kuantitas Pesanan)



Gambar 3-12 Flowchart Proses 3-1p

Keterangan :

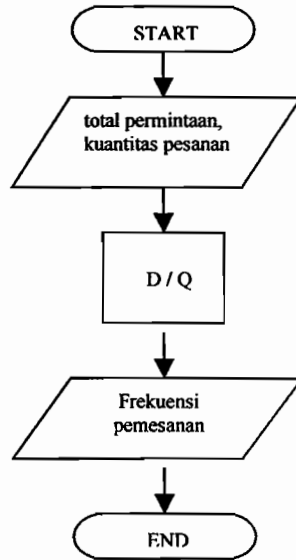
Co : biaya pemesanan

Ch : biaya penyimpanan

D : total permintaan

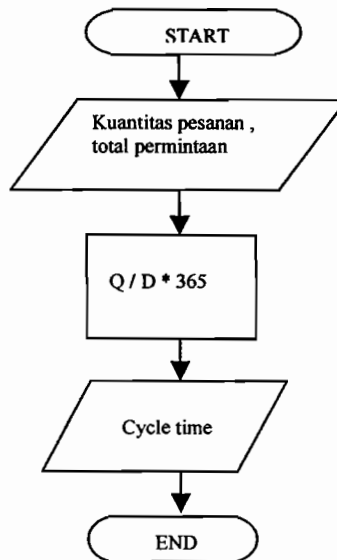
Q : kuantitas pesanan

3.3.1.3 Flowchart Proses 3.2p (Perhitungan Frekuensi Pemesanan)



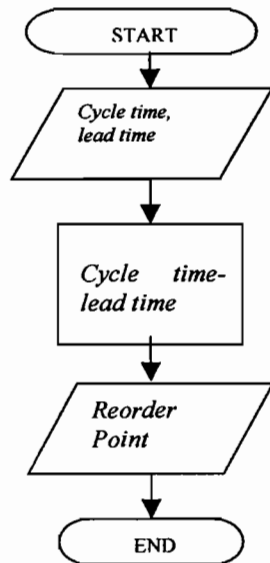
Gambar 3-13 Flowchart Proses 3.2p

3.3.1.4 Flowchart Proses 3.3p (Perhitungan Cycle Time)



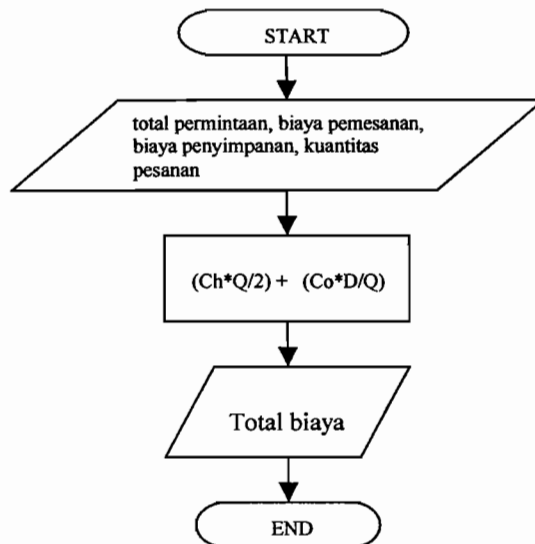
Gambar 3-14 Flowchart Proses 3.3p

3.3.1.5 Flowchart Proses 3.4p (Perhitungan Reorder Point)



Gambar 3-15 Flowchart Proses 3.4p

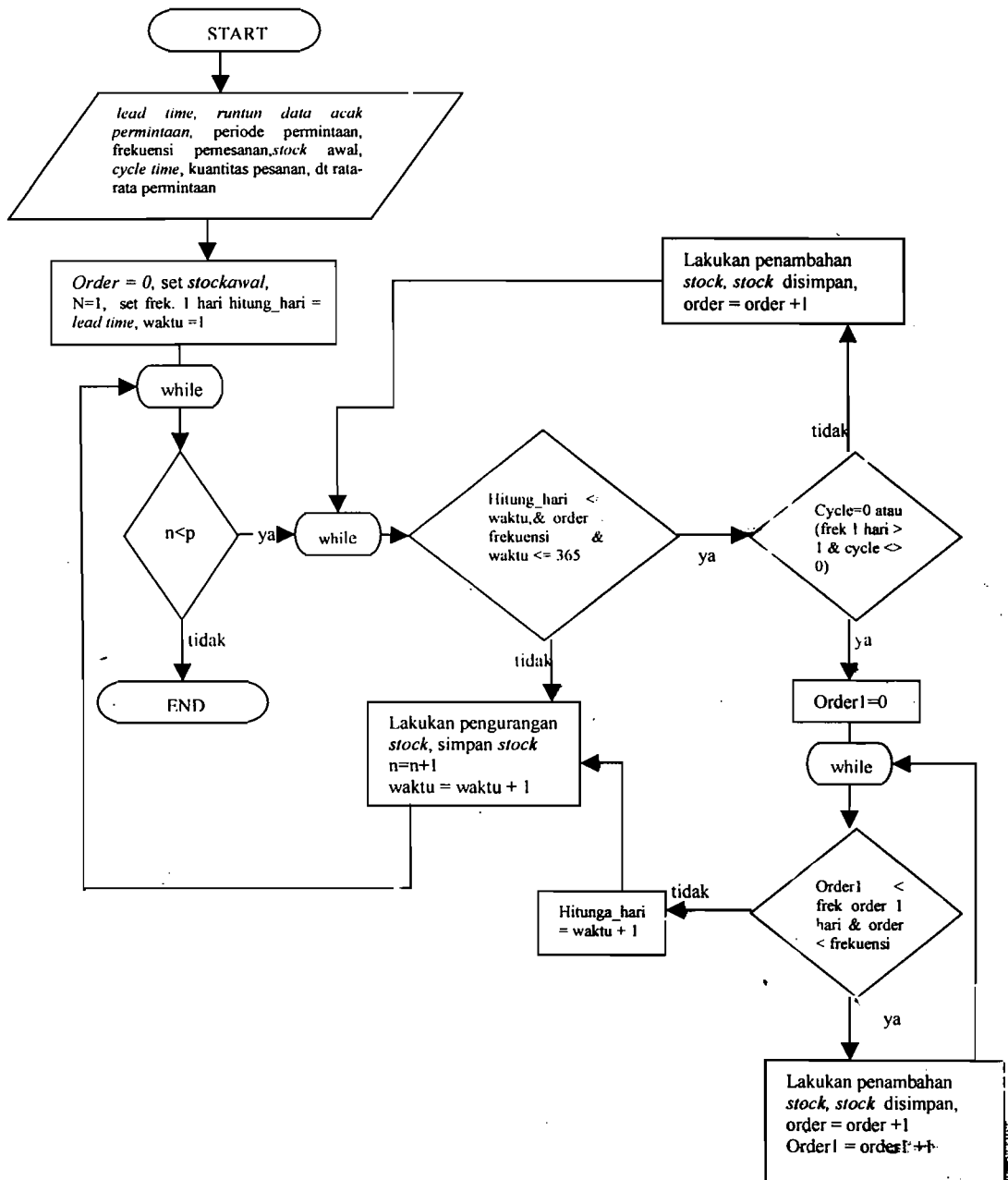
3.3.1.6 Flowchart Proses 3.5p (Perhitungan Total Biaya Inventory)



Gambar 3-16 Flowchart Proses 3.5p

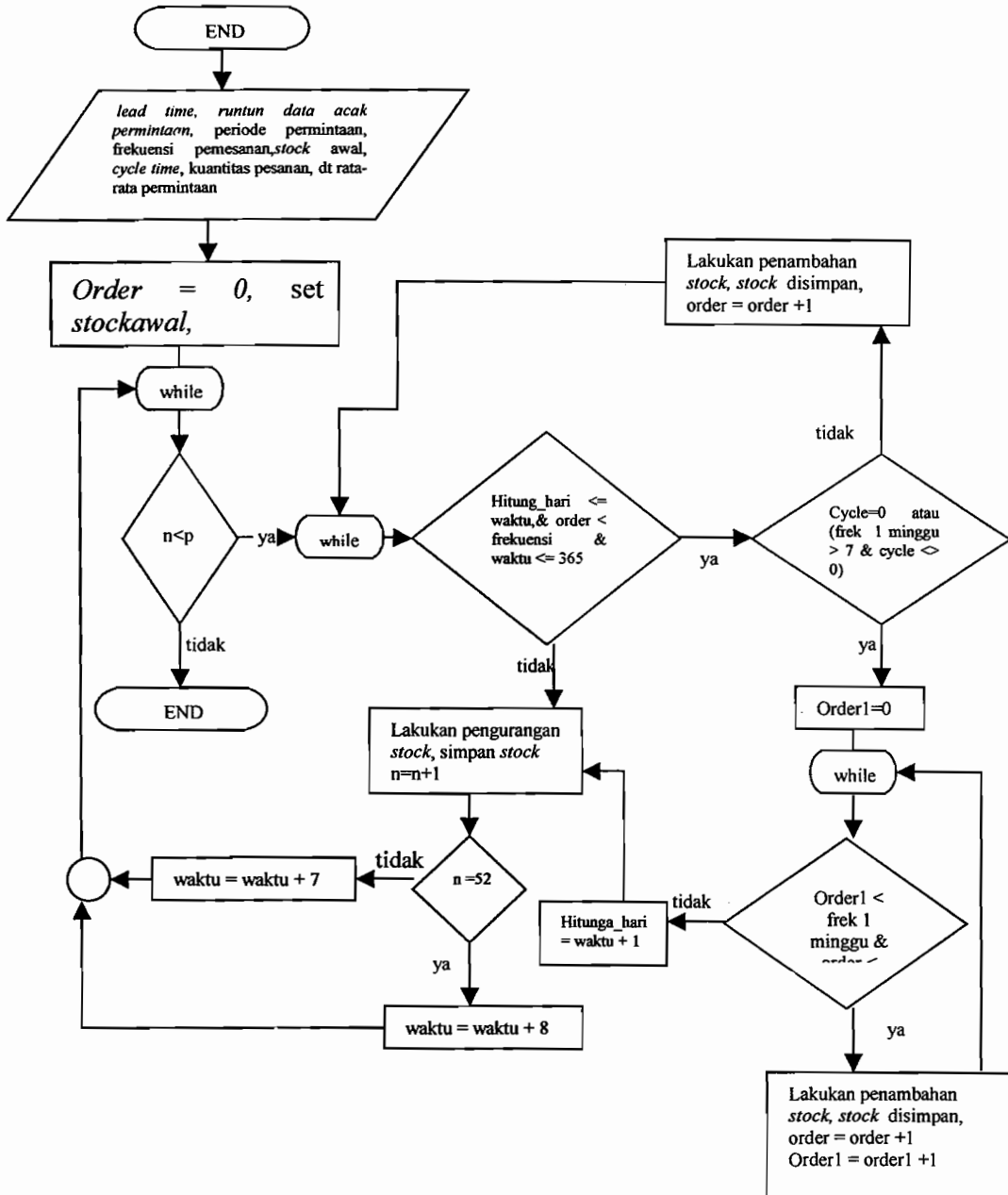
3.3.1.7 Flowchart Proses 4p (Perhitungan Stock)

a. Periode permintaan harian



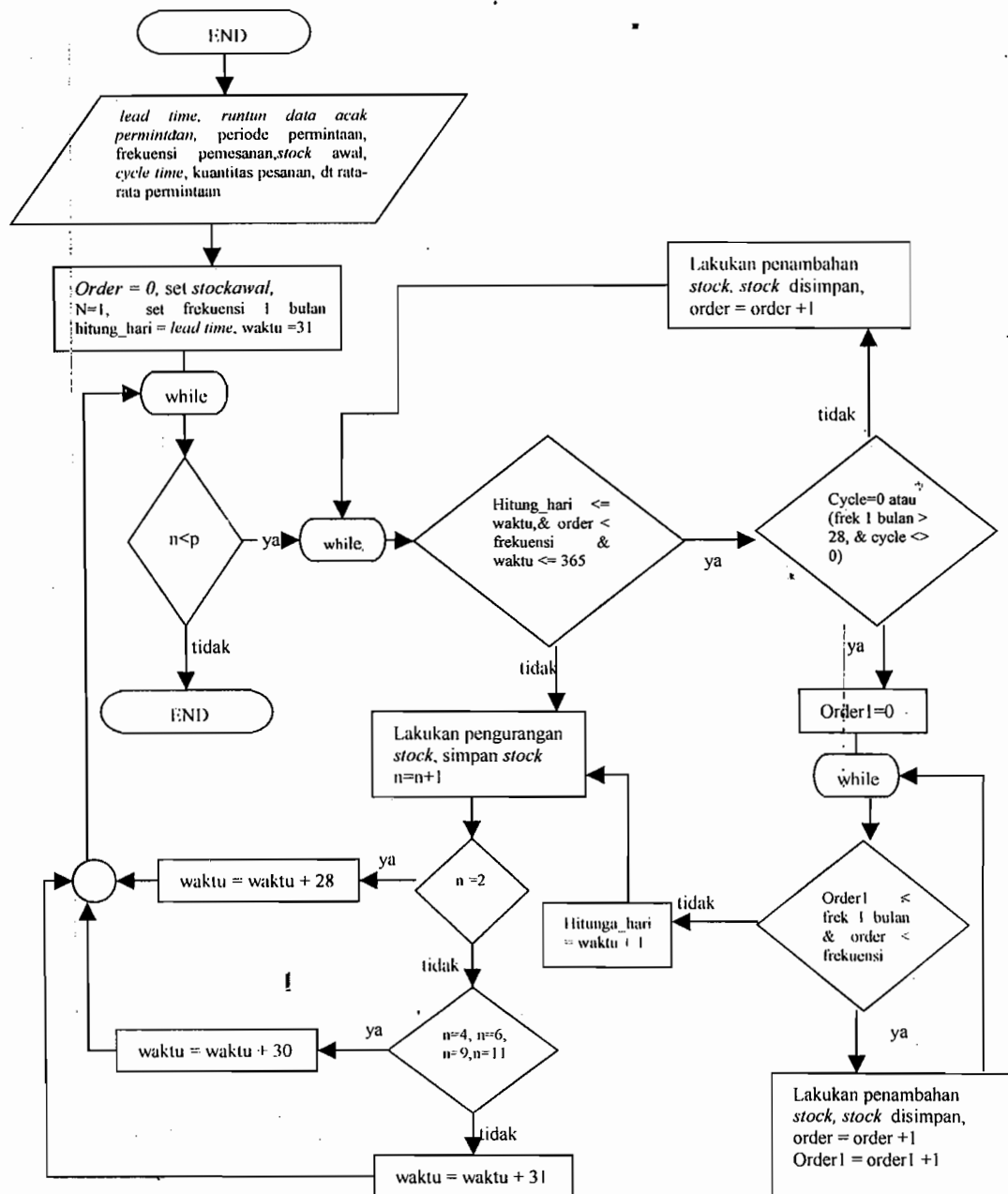
Gambar 3-17 Flowchart Proses 4p Periode Permintaan Harian

b. Periode permintaan mingguan



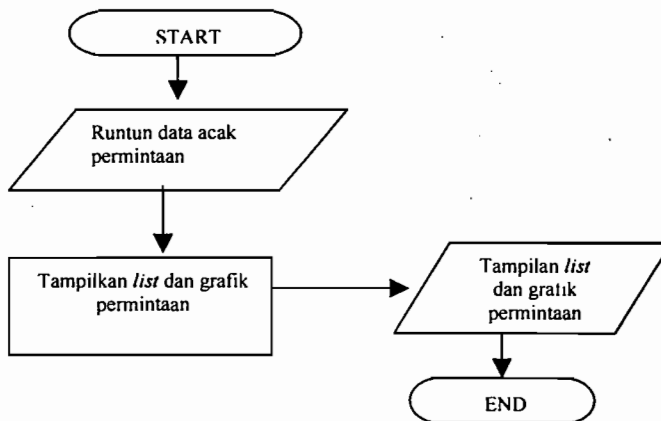
Gambar 3-18 Flowchart Proses 4p Periode Permintaan Mingguan

c. Periode permintaan bulanan



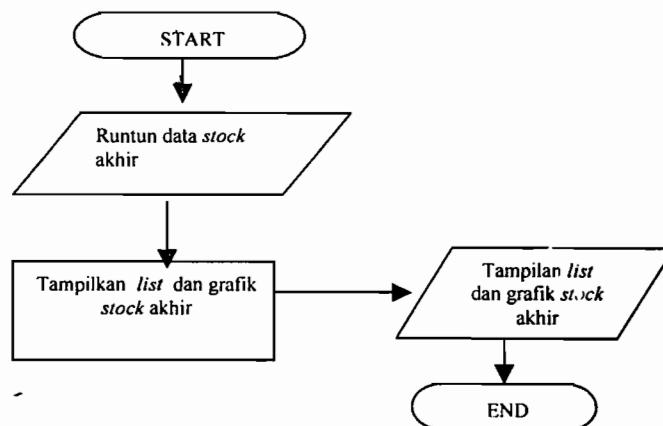
Gambar 3-19 Flowchart Proses 4p Periode Permintaan Bulanan

3.3.1.8 Flowchart Proses 5.1p.(Menampilkan *List* dan Grafik Permintaan)



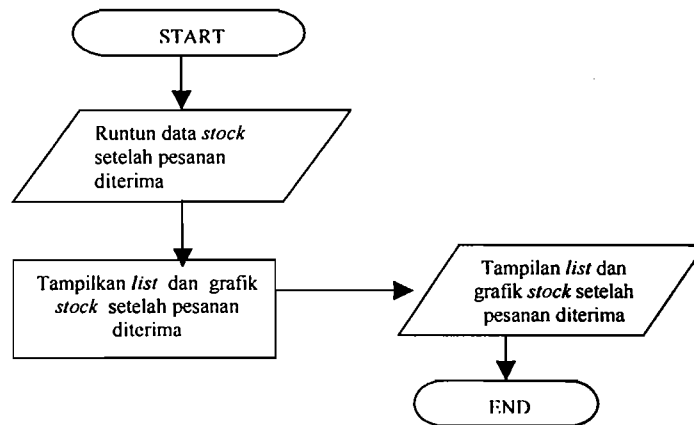
Gambar 3-20 Flowchart Proses 5.1p

3.3.1.9 Flowchart Proses 5.2p (Menampilkan *List* dan Grafik Stock Akhir)



Gambar 3-21 Flowchart Proses 5.2p

3.3.1.10 Flowchart Proses 5.3p (Menampilkan *List* dan Grafik *Stock* Setelah Pesanan Diterima)



Gambar 3-22 Flowchart Proses 5.3p

3.3.2 Perancangan Antarmuka

3.3.2.1 Perancangan Tampilan Awal

| | | |
|---|---|---|
| Form Judul | | |
| <p style="text-align: center;">SIMULASI OPERASI PERBEKALAN(INVENTORY) dengan PERMINTAAN yang BERUBAH dengan WAKTU</p> <div style="text-align: center; border: 1px solid black; width: 100px; height: 40px; margin: 0 auto 20px auto;">LOGO</div> <p style="text-align: center;">Oleh : Ismiasih (005314019)</p> <table style="width: 100%;"><tr><td style="text-align: center;">Pembimbing I : Ir.F.Soesianto,B.Sc,PhD</td><td style="text-align: center;">Pembimbing II : Puspaningtyas S Adi, S.T</td></tr></table> <p style="text-align: center;">JURUSAN TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS SANATA DHARMA YOGYAKARTA 2005</p> | Pembimbing I : Ir.F.Soesianto,B.Sc,PhD | Pembimbing II : Puspaningtyas S Adi, S.T |
| Pembimbing I : Ir.F.Soesianto,B.Sc,PhD | Pembimbing II : Puspaningtyas S Adi, S.T | |

Gambar 3-23 Perancangan Tampilan Awal

Antarmuka ini menampilkan informasi tentang judul program, logo universitas, pembuat program, pembimbing program, alamat identitas pembuat program. Antarmuka awal ini berupa form *start up* untuk masuk ke antarmuka utama. Form *start up* merupakan form *splash screen* yang secara otomatis menutup dengan kontrol waktu

3.3.2.2 Perancangan Tampilan Utama

Form Utama

| | | |
|---|-----------------------------------|--|
| <p>Stock setelah pesanan diterima</p> <p>List 3</p> | <p>Judul Grafik</p> <p>GRAFIK</p> | <p>Input</p> <p>Periode permintaan : <input type="text"/> <input type="button" value="SIMULASI"/></p> <p>Rate-rata permintaan : <input type="text"/> unit</p> <p>Biaya pemesanan (Rp) : <input type="text"/> per 1x pesan</p> <p>Biaya penyimpanan (Rp) : <input type="text"/> per unit/ tahun</p> <p>Stock awal : <input type="text"/> unit</p> <p>Lead time : <input type="text"/> hari</p> <p><input type="button" value="HITUNG"/></p> |
| <p>Permintaan</p> <p>List 1</p> | <p>Judul Grafik</p> <p>GRAFIK</p> | <p>Output</p> <p>Total permintaan : <input type="text"/> unit per tahun</p> <p>Kuantitas pesanan : <input type="text"/> unit setiap kali pesan</p> <p>Frekuensi pemesanan : <input type="text"/> kali / tahun</p> <p>Cycle time : <input type="text"/> hari</p> <p>Reorder point : <input type="text"/></p> <p>Total biaya inventory (1 tahun) : <input type="text"/></p> <p><input type="button" value="INFO"/> <input type="button" value="EXIT"/></p> |
| <p>Stock akhir</p> <p>List 2</p> | <p>Judul Grafik</p> <p>GRAFIK</p> | |

Gambar 3-24 Perancangan Tampilan Utama

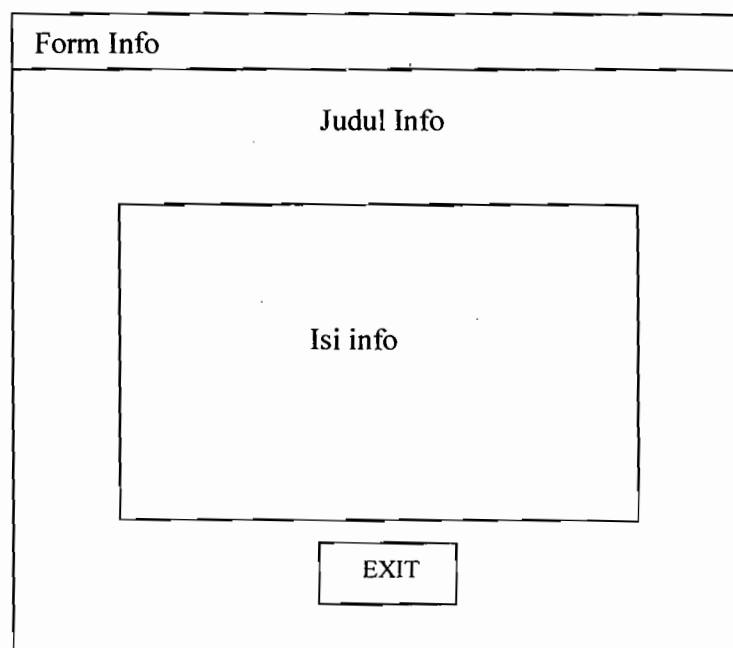
Antarmuka ini merupakan program utama yang akan dibuat. Fasilitas dalam antar muka ini adalah :

- Fasilitas *input* data, berupa *input* data periode permintaan, rata-rata permintaan, biaya pemesanan, biaya penyimpanan, *stock* awal, dan *lead time*.
- Informasi *output* data hasil perhitungan yaitu total permintaan,, kuantitas pesanan, frekuensi pemesanan, *cycle time*, *reorder point* dan biaya total *inventory*. Informasi *output* grafik yaitu grafik permintaan, grafik *stock*

setelah pesanan diterima dan grafik *stock* akhir. Informasi *list* permintaan, *list stock* setelah pesanan diterima, *list stock* akhir

- Tombol “SIMULASI” untuk melakukan proses *generate* data bilangan acak yang baru, dan kemudian melakukan proses perhitungan berdasarkan *inputan* yang ada.
- Tombol “HITUNG” untuk melakukan proses perhitungan tapi tidak *generate* data lagi.
- Tombol “INFO” untuk memanggil antar muka info.
- Tombol “EXIT” untuk keluar dari program.

3.3.2.3 Perancangan Tampilan Info



Gambar 3-25 Perancangan Tampilan Info

Antar muka info digunakan untuk menampilkan informasi sedikit tentang program yang dibuat. Pada antar muka info hanya ada satu tombol yaitu tombol "EXIT", tombol tersebut digunakan untuk keluar dari tampilan info.

3.4 Spesifikasi Kebutuhan Sumber Daya

3.4.1 Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dengan spesifikasi sebagai berikut :

- Minimal prosesor Intel Pentium II.
- Memori : 128 MB.
- Media penyimpanan : hardisk.
- Monitor
- Alat masukan : papan ketik (*keyboard*) dan *mouse*.

3.4.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan adalah :

- Sistem Operasi Windows 98
- Visual Basic 6.0



BAB IV

IMPLEMENTASI

4.1 Implementasi Program

Program aplikasi ini, mempunyai beberapa variable global.

```
Option Base 1
Dim data(), total_permintaan, q As Long
Dim frekuensi, cycle, rop As Long
Dim hasil() As Single
Dim p As Integer
```

4.1.1 Pembangkitan Bilangan Acak

Program aplikasi ini, men-*generate* data permintaan dengan algoritma pembangkitan bilangan acak menurut pendekatan Tocher. Dengan konstanta π diset 3.14.

```
Const pi = 3.14
```

Program akan men-*generate* data permintaan sebanyak p. P bernilai 365 jika periode permintaan harian. P bernilai 52 jika periode permintaan mingguan. P bernilai 12 jika periode permintaan bulanan. Kode program pembangkitan bilangan acak adalah sebagai berikut :

```
For i = 1 To p
  u1 = Rnd()
  u2 = Rnd()
  x = (Sqr(pi / 8)) * (Math.Log((1 + u1) / (1 - u1)))
  If u2 <= 0.5 Then
    hasil(i) = (-x)
  Else
    hasil(i) = (x)
  End If
Next i
```

Hasil dari pembangkitan bilangan acak tersebut adalah bilangan acak terdistribusi normal baku. Hasil tersebut ditransformasi untuk mendapatkan bilangan acak yang diinginkan, dengan rata-rata dan varians tertentu. Rata-rata disini yang dimaksud adalah rata-rata permintaan yang di-*input*-kan. *Varians* yang digunakan adalah 10% dari rata-rata permintaan yang di-*input*-kan Hasil dari transformasi dijumlahkan untuk mendapatkan total permintaan. Kode programnya sebagai berikut :

```
total_permintaan = 0
For n = 1 To p
  data(n) = CLng((hasil(n) * (10 / 100 * txtrata.Text)) + txtrata.Text)
  List1.AddItem data(n)
  total_permintaan = total_permintaan + data(n)
Next n
```

4.1.2 Proses Perhitungan Rumus

Total permintaan yang didapat dari hasil proses transformasi bilangan acak akan dipakai untuk menghitung nilai kuantitas pesanan(q). Kode programnya :

```
q = CLng(Sqr((2 * txtbiayaorder.Text * total_permintaan) / txtbiayahold.Text))
```

Hasil perhitungan kuantitas pesanan(q) dan total permintaan akan digunakan untuk menghitung frekuensi pemesanan, *cycle time*, dan biaya total *inventory*.

Frekuensi pemesanan didapat dari hasil perhitungan total permintaan tahunan dibagi dengan kuantitas pesanan. Kode program:

```
frekuensi = CLng(total_permintaan / q)
```

Cycle time didapat dari hasil perhitungan kuantitas pesanan dibagi total permintaan tahunan di kali 365. Kode program :

```
cycle = CLng((q / total_permintaan) * 365)
```

Hasil *cycle time* akan digunakan untuk perhitungan *reorder point*. *Reorder point* didapat dari hasil perhitungan *cycle time* dikurangi *lead time*. Kode program:

```
rop = (cycle - txttime.Text)
```

Biaya total *inventory* (1 tahun) didapat dari hasil penjumlahan total biaya penyimpanan dan biaya pemesanan. Kode programnya sebagai berikut :

```
biaya = ((txtbiayahold.Text * q) / 2) + ((txtbiayaorder.Text * total_permintaan) / q)
```

4.1.3. Proses Penghitungan *Stock*

Untuk memulai proses perhitungan *stock*, dibutuhkan beberapa variabel bantu dan pengesetan awal data

```
Dim hitung_hari As Integer
Dim c0, stockawal As Long
Dim order1, order, n As Integer
ReDim stockakhir(p), datang(p)
Dim waktu As Integer

order = 0
stockawal = txtstock.Text
n = 1
c0 = CLng(txtrata.Text / q)
hitung_hari = txttime.Text
```

a. Proses perhitungan *stock* dengan periode permintaan harian

Untuk memulai proses perhitungan *stock* dengan pilihan periode permintaan harian, dibutuhkan pengesetan waktu awal.

```
waktu = 1
```

Proses penambahan *stock* dicek setiap hari apakah ada pesanan yang diterima pada hari tersebut. Jika ada, tambahkan kuantitas pesanan yang diterima dengan *stock* akhir. Jika tidak ada proses akan melakukan proses pengurangan *stock*. Proses pengurangan akan dilakukan tiap hari. Proses penambahan *stock* dilakukan sebanyak frekuensi pemesanan. Proses penambahan dan pengurangan *stock* dilakukan selama 1 periode permintaan (1 tahun = 365 hari). Kode programnya sebagai berikut :

```
While n <= p
  While hitung_hari <= waktu And order < frekuensi
    If cycle = 0 Or (c0 > 1 And cycle <> 0) Then
      order1 = 0
      While order1 < c0 And order < frekuensi
        stockawal = stockawal + q
        datang(n) = stockawal
        List3.AddItem datang(n)
        order = order + 1
        order1 = order1 + 1
      Wend
      hitung_hari = waktu + 1
    Else
      stockawal = stockawal + q
      datang(n) = stockawal
      List3.AddItem datang(n)
      order = order + 1
      hitung_hari = hitung_hari + cycle
    End If
  Wend
  stockakhir(n) = stockawal - data(n)
  List2.AddItem stockakhir(n)
  stockawal = stockakhir(n)
  n = n + 1
  waktu = waktu + 1
Wend
```

b. Proses perhitungan *stock* dengan periode permintaan harian

Untuk memulai proses perhitungan *stock* dengan pilihan periode permintaan mingguan, dibutuhkan pengesetan waktu awal.

```
waktu = 7
```

Proses perhitungan *stock* dengan periode permintaan mingguan sama dengan proses perhitungan dengan periode permintaan harian. Tetapi proses pengurangannya dilakukan tiap minggu dan proses pengecekan pesanan yang diterima dilakukan perminggu. 1 minggu sama dengan 7 hari, kecuali minggu terakhir ada 8 hari, jadi 1 tahun = 52 minggu = 365 hari.

```
While n <= p
  While hitung_hari <= waktu And order < frekuensi And waktu <= 365
    If cycle = 0 Or (c0 > 7 And cycle <> 0) Then
      order1 = 0
      While order1 < c0 And order < frekuensi
        stockawal = stockawal + q
        datang(n) = stockawal
        List3.AddItem datang(n)
        order = order + 1
        order1 = order1 + 1
      Wend
      hitung_hari = waktu + 1
    Else
      stockawal = stockawal + q
      datang(n) = stockawal
      List3.AddItem datang(n)
      order = order + 1
      hitung_hari = hitung_hari + cycle
    End If
  Wend
  stockakhir(n) = stockawal - data(n)
  List2.AddItem stockakhir(n)
  stockawal = stockakhir(n)
  n = n + 1
```

```

If n = 52 Then
    waktu = waktu + 8
Else
    waktu = waktu + 7
End If
Wend

```

b. Proses perhitungan *stock* dengan periode permintaan bulanan

Untuk memulai proses perhitungan *stock* dengan pilihan periode permintaan bulanan, dibutuhkan pengesetan waktu awal.

```
waktu = 31
```

Proses perhitungan *stock* dengan periode permintaan bulanan sama dengan proses perhitungan dengan periode permintaan harian. Tetapi proses pengurangannya dilakukan tiap bulan dan proses pengecekan pesanan yang diterima dilakukan perbulan. 1 periode (1 tahun) = 12 bulan = 365 hari.

```

While n <= p
    While hitung_hari <= waktu And order < frekuensi And waktu <= 365
        If cycle = 0 Or (c0 > 28 And cycle <> 0) Then
            order1 = 0
            While order1 < c0 And order < frekuensi
                stockawal = stockawal + q
                datang(n) = stockawal
                List3.AddItem datang(n)
                order = order + 1
                order1 = order1 + 1
            Wend
            hitung_hari = waktu + 1
        Else
            stockawal = stockawal + q
            datang(n) = stockawal
            List3.AddItem datang(n)
            order = order + 1
            hitung_hari = hitung_hari + cycle
        End If
    Wend
Wend

```



```

stockakhir(n) = stockawal - data(n)
List2.AddItem stockakhir(n)
stockawal = stockakhir(n)
n = n + 1

If n = 2 Then
    waktu = waktu + 28
ElseIf n = 4 Or n = 6 Or n = 9 Or n = 11 Then
    waktu = waktu + 30
Else
    waktu = waktu + 31
End If
Wend

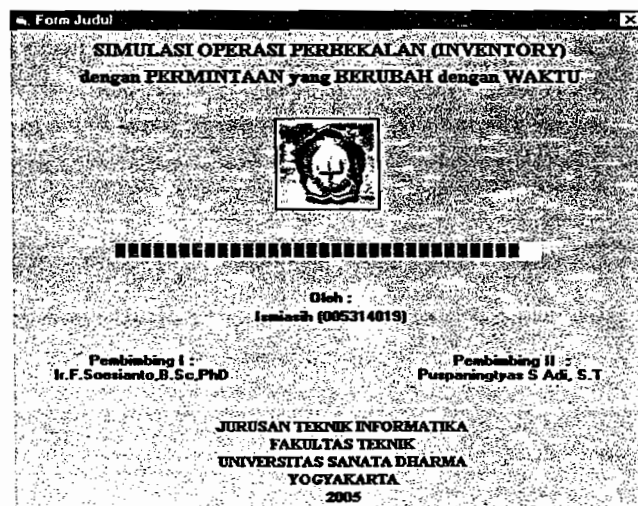
```

4.2 Implementasi Antarmuka

Antarmuka merupakan tampilan yang nantinya berinteraksi langsung dengan pengguna(*user*).

4.2.1 Implementasi Tampilan Awal

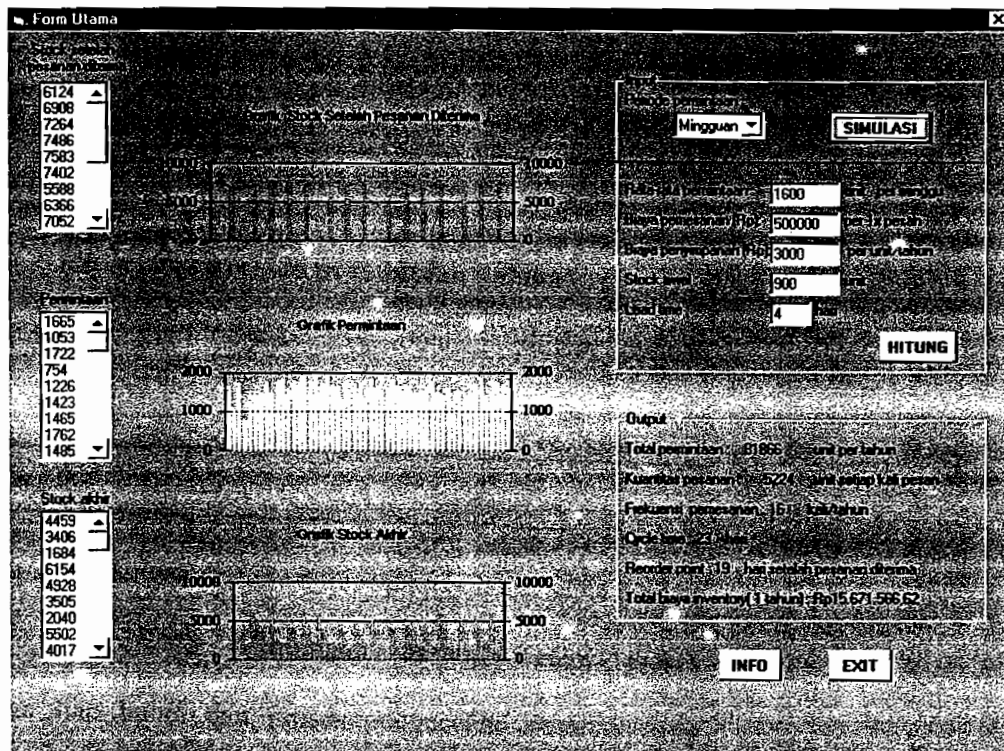
Tampilan awal adalah tampilan yang pertama kali ditampilkan pada saat pertama kali program dijalankan. Tampilan awal berupa form *splash screen* yang secara otomatis menutup dengan kontrol waktu.



Gambar 4-1 Form Tampilan Awal

4.2.2 Implementasi Tampilan Utama

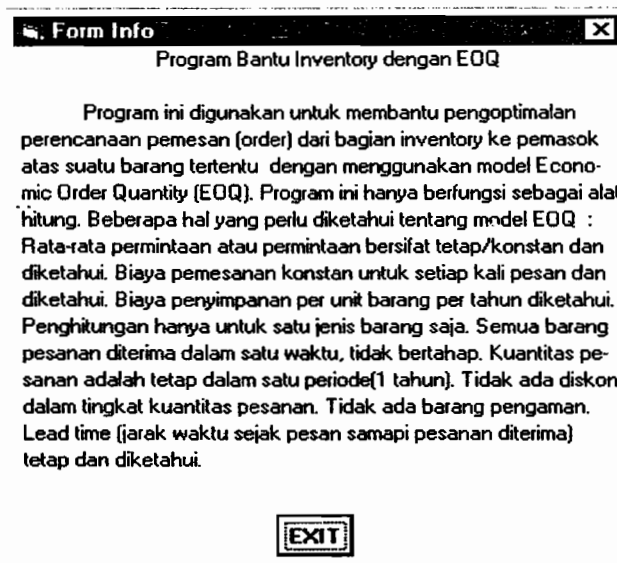
Tampilan utama merupakan inti dari program yang dijalankan.



Gambar 4-2 Form Tampilan Utama

Setelah form tampilan awal tertutup, maka tampilan utama akan langsung ditampilkan. Saat pertama kali ditampilkan form ini sudah ada nilai-nilai *input* dan *output*-nya. Pengguna bebas untuk mengganti *input*-an secara langsung pada *textbox* yang tersedia. Jika pengguna menginginkan data yang bervariasi, klik tombol simulasi, maka akan didapatkan data baru dan sekaligus hasil perhitungannya (*output*-nya). Jika pengguna tidak menginginkan data baru tekan tombol hitung.

4.2.3 Implementasi Tampilan Info



Gambar 4-3 Form Tampilan info

Jika tombol info pada tampilan awal di-klik maka akan ditampilkan tampilan form info yang mana berisi sedikit informasi tentang program dan *economic order quantity* (model yang dipakai).

BAB V

ANALISIS HASIL

5.1 Studi Kasus

Studi kasus yang digunakan adalah contoh soal pada bab II.

Soal :

Diketahui : 1 unit = 1 kilogram

Robert Pizaman memulai bisnis pizzanya 5 tahun yang lalu di Jakarta. Untuk pembuatan pizza dia memerlukan beberapa bahan. Salah satu bahan tersebut adalah bubur keju yang dia pesan dari toko milik John Cheeseman, Surabaya. Bubur keju ini merupakan bahan yang paling vital untuk pizza-nya. Untuk membuat kebijakan pemesanan (*order*), Robert Pizaman telah mengidentifikasi biaya-biaya yang terkait. Biaya-biaya tersebut adalah sebagai berikut :

- a. Harga bubur keju per kilogram = Rp 12.000,-. Harga ini merupakan suatu hasil negosiasi berupa kontrak antara Robert Pizaman dan John Cheeseman yang akan diperbaharui tahun depan.
- b. Biaya penyimpanan per kilogram pertahun = 25% harga beli. Jadi biaya penyimpanannya = $25\% \times \text{Rp } 12.000,- = \text{Rp } 3.000,-$ per kilogram per tahun . Persentase 25% didapat dari perhitungan *opportunity cost* (biaya kesempatan), seandainya uang yang tertanam dipersediaan ditabung di bank. Bunga bank = 25% per tahun, menurut Robert, persentase ini akurat.

c. Biaya pemesanan (berapapun jumlah yang dipesan) = Rp 500.000,- per 1xpesan

Lead time (jarak waktu antara pemesanan sampai pesanan datang) = 4 hari.

Faktor berikutnya yang diperhitungkan Robert adalah permintaan pizza yang berfluktuasi setiap minggunya. Data permintaan pizza yang telah dikonversikan ke dalam kebutuhan bubuk keju selama 15 minggu terakhir seperti terlihat pada tabel 5-1. Diduga permintaan yang akan terjadi akan berpola mirip seperti pola permintaan tersebut.

| Minggu ke- | Kilogram | Minggu ke- | Kilogram |
|------------|----------|------------|----------|
| 1 | 1.560 | 9 | 1.590 |
| 2 | 1.623 | 10 | 1.600 |
| 3 | 1.580 | 11 | 1.605 |
| 4 | 1.578 | 12 | 1.585 |
| 5 | 1.610 | 13 | 1.590 |
| 6 | 1.615 | 14 | 1.600 |
| 7 | 1.590 | 15 | 1.605 |
| 8 | 1.605 | | |
| | | Total | 23.936 |
| | | Rata-rata | 1.595 |

Tabel 5-1 Kebutuhan Keju 15 Minggu Terakhir

Tahun depan Robert memperkirakan pertumbuhan pasar pizza-nya pesat, dengan melihat data permintaan keju nya 15 minggu terakhir tersebut. Dia

memperkirakan tahun depan permintaan keju nya menjadi 1600 kilogram per minggu. Dalam setahun berarti 83.200 kilogram (52 minggu x 1600 kilogram), dalam sehari 228 kilogram (83.200 kilogram per 365 hari), dalam sebulan 6933 kilogram (83.200 kilogram per 12 bulan).

1. Perhitungan menurut kebijakan Robert

Robert mencoba suatu metode untuk membuat kebijakan berupa berapa dan kapan dia harus melakukan pemesanan. Dia menentukan persediaan harus mencukupi untuk kebutuhan minimal 4 hari (4 hari berikutnya pesanan akan tiba). Sehingga kuantitas (jumlah) kilogram keju yang dia pesan = 912 kilogram (4 hari x 228 kilogram/hari). Dia menganggap kuantitas ini akan membuat total biaya pemesanan dan biaya penyimpanan optimal. Alasannya, jika kuantitas lebih dari ini, maka biaya penyimpanan akan meningkat, meskipun biaya pesan menurun (kuantitas pesanan besar berarti akan melakukan sedikit pemesanan untuk semua kebutuhan). Bila kuantitas pemesanan kurang dari 912 kilogram, meskipun biaya simpan mengecil, namun pemesanan harus sering dilakukan sehingga biayanya membesar. Jadi, Robert menganggap 912 kilogram adalah kuantitas pemesanan yang optimal.

Perhitungan berikut ini menggunakan rumus-rumus dari EOQ:

Keterangan : D = total permintaan tahunan

Q = kuantitas pesanan

C_o = Biaya pemesanan

C_h = Biaya penyimpanan

Total permintaan (D) = 83.200 kilogram

Kuantitas Pesanan (Q) = 912 kilogram setiap kali pesan

Frekuensi pemesanan = D/Q

$$= 83.200/912 = 91,2 = 91 \text{ kali per tahun}$$

Total biaya *inventory* (1 tahun) = $((Q/2) \times Ch) + ((D/Q) \times Co)$

$$= ((912/2) \times (3.000)) + ((83.200/912) \times (500.000))$$

$$= 1.368.000 + 45.614.035,09$$

$$= \text{Rp } 46.982.035,09,- = \text{Rp } 46.982.000,-$$

Hasil perhitungan menurut kebijakan Robert dapat dilihat pada tabel 5-2

| | |
|---------------------------------|-----------------------------------|
| Keterangan | |
| Total permintaan | 83.200 kilogram |
| Kuantitas pesanan | 912 kilogram setiap kali pesan |
| Frekuensi pemesanan | 91 kali per tahun |
| Cycle time | 4 hari |
| Reorder Point | Saat pesanan diterima, pesan lagi |
| Total biaya Inventory (1 tahun) | Rp 46.982.000,- |

Tabel 5-2 Hasil Perhitungan kebijakan Robert

2. Perhitungan *Economic Order Quantity (EOQ)* Teoritis

$$\begin{aligned} \text{Kuantitas pesanan optimal bubuk keju} &= \sqrt{\frac{2xDxCo}{Ch}} = \sqrt{\frac{2 \times 83.200 \times 500.000}{3.000}} \\ &= 5266,2 \text{ kilogram} \end{aligned}$$

$$\begin{aligned} \text{Cycle time} &= (Q/D) \times 365 \text{ hari} \\ &= (5266,2/83.200) \times 365 \\ &= 23,1 = 23 \text{ hari} \end{aligned}$$

$$\begin{aligned} \text{Frekuensi pemesanan} &= D/Q \\ &= 83.200/5266,2 = 15,79 = 16 \text{ kali per tahun} \end{aligned}$$

$$\begin{aligned} \text{Reorder point} &= \text{cycle time} - \text{lead time} \\ &= 23 - 4 = 19 \text{ hari setelah pesanan diterima} \end{aligned}$$

$$\begin{aligned} \text{Total biaya inventory (1 tahun)} &= ((Q/2) \times Ch) + ((D/Q) \times Co) \\ &= ((5266,2/2) \times 3.000) + ((83.200/5266,2) \times 500.000) \\ &= \text{Rp } 15.798.734,13,- = \text{Rp } 15.798.800,- \end{aligned}$$

Dalam praktik, angka 5266,2 kilogram hasil penghitungan kuantitas pesanan mungkin sulit diterapkan, sedikit pembulatan untuk kemudahan pesanan dapat dilakukan. Misalnya dibulatkan menjadi 5000 atau 5270.

Hasil perhitungan dapat dilihat pada tabel 5-3.

| Keterangan | |
|---|----------------------------------|
| Total permintaan | 83.200 kilogram |
| Kuantitas pesanan | 5266 kilogram setiap kali pesan |
| Frekuensi pemesanan | 16 kali per tahun |
| Cycle time | 23 hari |
| Reorder point | 19 hari setelah pesanan diterima |
| Total biaya <i>inventory</i> (1 tahun) | Rp 15.798.734,13,- |

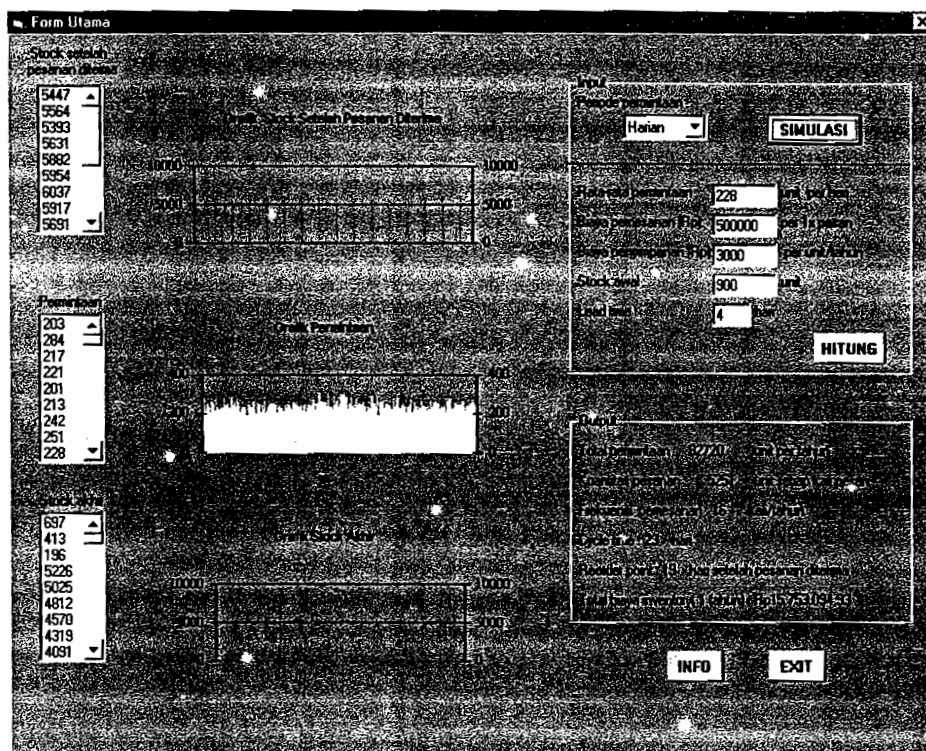
Tabel 5-3 Hasil Perhitungan *EOQ* Teoritis

3. Perhitungan Simulasi Program

Hasil perhitungan simulasi program dapat dilihat di tabel A-1, tabel A-2, tabel A-3 pada lampiran A dengan keterangan sebagai berikut : Rata-rata permintaan untuk periode harian adalah 228 kilogram, untuk periode mingguan 1.600 kilogram, untuk periode bulanan adalah 6.933, dengan biaya-biaya sama dengan pada kasus.

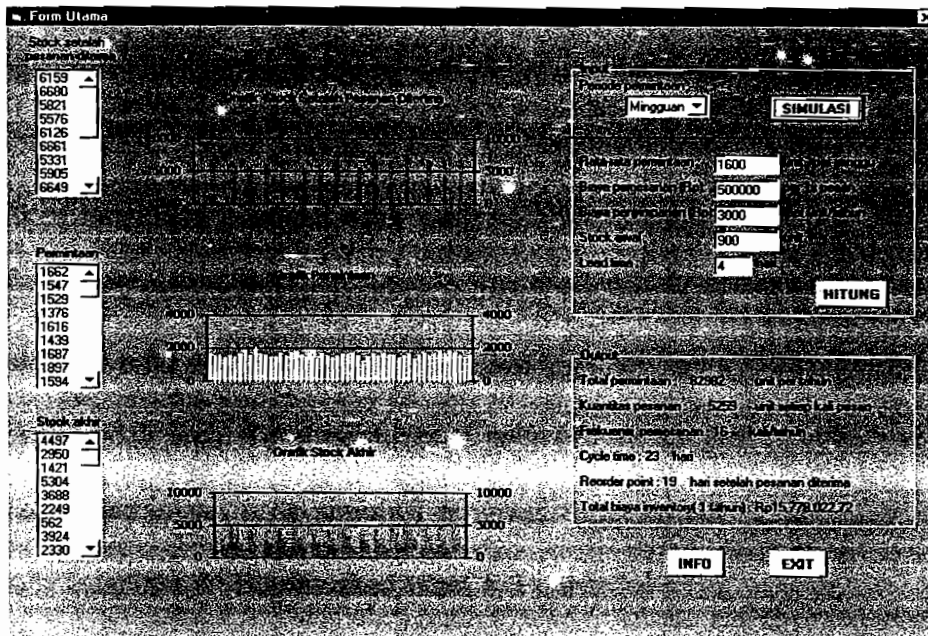
3 contoh hasil *running* program :

a. Periode permintaan harian



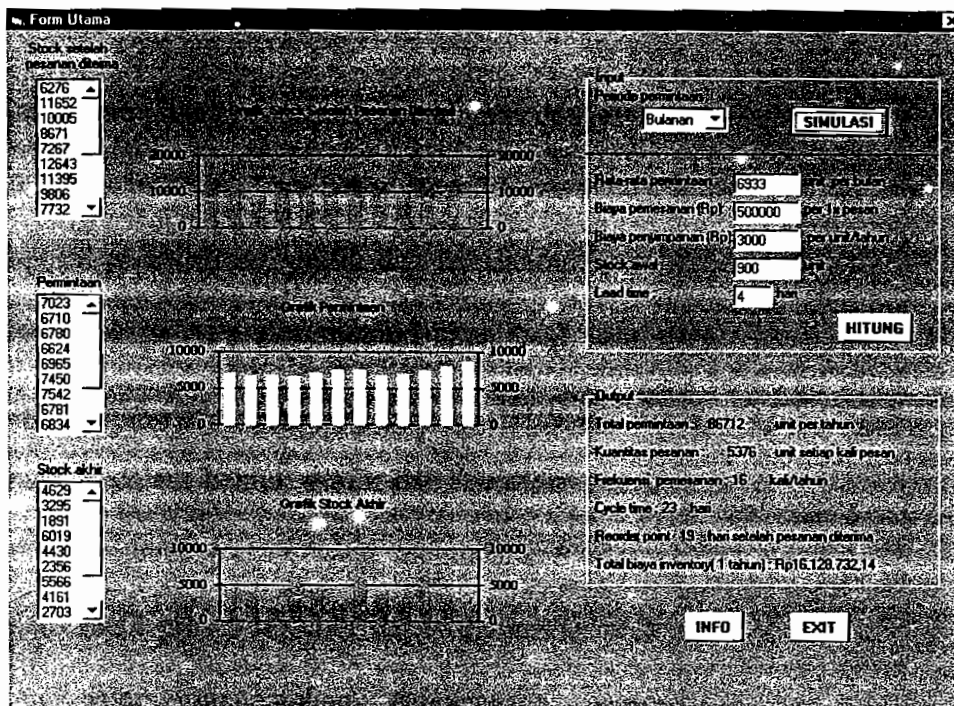
Gambar 5-1 Uji Coba Periode Permintaan Harian

b. Periode permintaan mingguan



Gambar 5-2 Uji Coba Periode Permintaan Mingguan

c. Periode permintaan bulanan



Gambar 5-3 Uji Coba Periode Permintaan Bulanan

Penjelasan untuk Perhitungan EOQ Teoritis dan Simulasi Program

Berdasarkan tabel hasil uji coba pada lampiran A, dapat dilihat bahwa total permintaannya selalu berubah. Hal ini terjadi karena untuk setiap simulasi, satu simulasi akan membangkitkan rangkaian bilangan acak yang berbeda, sehingga akan menghasilkan total permintaan yang berbeda. Total permintaan tahunan pada contoh kasus tidak sama dengan total permintaan pada simulasi program, tapi masih dalam jarak(*range*) yang tidak jauh. Total permintaan pada contoh kasus adalah 83.200 kilogram. Total permintaan pada simulasi program dapat dilihat pada tabel hasil uji coba pada lampiran A. Berdasarkan tabel 5-3 dan tabel hasil uji coba pada lampiran A dapat diambil kesimpulan bahwa proses pembangkitan bilangan acak yang digunakan benar (berhasil).

5.2 Analisis sistem

Program simulasi ini menampilkan pencatatan permintaan dengan periode permintaan harian, mingguan, dan bulanan. *User* bebas memilih pencatatan yang diinginkan. Hasil proses berupa perhitungan, grafik dan *list* ditampilkan dalam satu form(tampilan) dengan *input-an* sehingga memudahkan bagi *user* untuk melihat hasil perubahan proses jika salah satu *input* diubah. Hasil perhitungan yang dihasilkan disimulasikan kedalam bentuk grafik dan *list*(daftar pencatatan yang akan digunakan untuk membuat grafik), yang memudahkan *user* untuk melihat hasil yang diperoleh. Hasil proses tersebut dapat digunakan oleh *user* sebagai bahan masukan dalam menentukan perencanaan pemesanan perbekalan(*inventory*).

Berdasarkan hasil perhitungan kebijakan Robert pada tabel 5-2 dengan tabel hasil uji coba pada lampiran A, dapat dianalisis :

1. Perhitungan kebijakan Robert belum optimal, dengan acuan bahwa total biaya *inventory* simulasi lebih sedikit dibandingkan dengan total biaya *inventory* pada kebijakan Robert. Dengan perhitungan *EOQ*, total biaya *inventory* dapat dikurangi sekitar Rp 30.000.000,-
2. Kuantitas pesanan hasil simulasi merupakan kuantitas pesanan yang ekonomis karena dapat meminimalkan total biaya tahunan *inventory*.
3. Perhitungan *EOQ* mengurangi frekuensi pemesanan dari 91 kali per tahun menjadi 16 kali per tahun.

Dari ketiga hal tersebut dapat disimpulkan bahwa model *EOQ* dapat membantu mengoptimalkan jumlah/kuantitas barang yang akan dipesan. Perhitungan simulasi program, lihat tabel pada lampiran A tidak berbeda dengan perhitungan teoritis pada tabel 5-3, sehingga dapat disimpulkan bahwa proses pembangkitan bilangan acak yang digunakan benar (berhasil).

Simulasi hanya melakukan pencatatan hasil dari perhitungan yang dicatat dalam bentuk *list*. Pencatatan yang dilakukan proses tidak disertai keterangan waktu. Misalnya pencatatan hasil *stock* akhir tidak langsung dapat diketahui bahwa hasil *stock* akhir merupakan *stock* akhir pada hari atau minggu atau bulan ke x. Pencatatan dalam bentuk *list* disimulasikan kedalam bentuk grafik dan grafik tersebut akan secara otomatis menampilkan gambaran hasil dari *list* berdasarkan pilihan pencatatan. Contoh kasus, hasil dari *list* *stock* setelah pesanan diterima ada 16 pencatatan, namun pada grafik hanya akan ditampilkan sebanyak 12 grafik

stock setelah pesanan diterima. Grafik tidak menggambarkan secara langsung seluruh pencatatan(16 pencatatan), grafik hanya menggambarkan pencatatan pada akhir bulan. Pencatatan akhir tersebut merupakan hasil penjumlahan keseluruhan pesanan yang diterima pada bulan tersebut. Seperti telah dijelaskan didepan bahwa pencatatan dalam *list* tidak disertai dengan keterangan, hal ini menyulitkan *user* melihat secara langsung, berapa kali dalam bulan tersebut pesanan diterima dan berapa hasil akhir dari pencatatan bulan tersebut. Grafik yang dihasilkan merupakan grafik buta, sehingga menyulitkan *user* untuk membaca grafik.

5.3 Kelebihan Program

1. Program menyediakan fasilitas bagi *user* untuk memilih proses pencatatan.
2. Hasil perhitungan, grafik dan *list* ditampilkan dalam satu tampilan dengan *input-an*, sehingga memudahkan *user* untuk melihat perubahan hasil yang didapat jika *user* mengubah salah satu *input*.

5.4 Kekurangan Program

Hasil *output* grafik dan pencatatan dalam bentuk *list* sulit dibaca oleh *user*.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang diambil dari program yang telah dibuat adalah :

1. Program aplikasi ini dapat diimplementasikan pada bagian *inventory*. Program ini juga dapat dijadikan tolak ukur perencanaan pemesanan barang, dengan memperhatikan simulasi grafik dan hasil perhitungan.
2. Model *Economic Order Quantity(EOQ)* membantu mengoptimalkan kuantitas/jumlah barang yang dipesan dengan meminimalkan total biaya *inventory*.

6.2 Saran

1. Biaya penyimpanan dalam program ini diset sebesar x rupiah per unit per tahun. Jika biaya penyimpanan diketahui sebesar x persen dari harga beli per unit, seperti pada contoh kasus, maka *user* harus menghitung terlebih dulu biaya penyimpanan sebenarnya (prosentase dikalikan harga beli). Penulis menyarankan agar input biaya penyimpanan diberikan dua pilihan yaitu langsung memasukkan besarnya rupiah biaya penyimpanan per unit barang per tahun dan memasukkan harga beli serta berapa persen biaya penyimpanannya dari harga beli per unit.
2. Penulis juga menyarankan pembuatan hasil *output* grafik dan pencatatan *list* yang mudah dibaca oleh *user*.

DAFTAR PUSTAKA

- Stevenson, William J, 1986, *Production/Operatin Management*, Irwin, Illinois
- Reksohadiprodjo, Sukanto, 1984, *Management Produksi*, Yogyakarta : BPFE
- Jogianto HM, 1990, *Analisis dan Desain*, Yogyakarta : Andi
- Polina, AM, 2002, *Diktat Mata Kuliah Analisa Data*, Jurusan Teknik Informatika Universitas Sanata Dharma
- Handoko, Hani, 1984, *Manajemen Produksi dan Operasi*, Yogyakarta: BPFE
- Baroto Teguh, 2002, *Perencanaan & Pengendalian Produksi*, Jakarta, Ghalia Indonesia
- Siswanto, 1985, *Persediaan, Model dan Analisis*, Yogyakarta, Andi offset
- Juneau, Jon dan Coates, Eyler R, 2001, An Economic Order Quantity Model For Time-Varying Demand, *The International Journal Of Modern Engineering*, (<http://www.21centuryengineer.com/issues/spring2001/articles/EconomicOrder.htm>)
- <http://www.caam.rice.edu/~timred1/caam376/Lecture13/sld002.htm>
- Dennis, Terry L. dan Dennis, Laurie B., 1991, *Management science*, San Francisco : West publishing company
- Supranto, J, 1985, *Pengantar Probabilitas dan Statistika Induktif*, Jakarta : Erlangga

- Kusumo, Ario Suryo, *Buku latihan Microsoft Visual Basic 6.0*, Jakarta : Elex Media Komputindo
- Dewobroto, Wiryanto, 2003, *Aplikasi SAIN dan Teknik dengan Visual Basic 6.0*, Jakarta : Elex Media Komputindo
- Pamungkas, 2000, *Tip & Trik Microsoft Visual Basic 6.0*, Jakarta : Elex Media Komputindo

LAMPIRAN A
(TABEL HASIL SIMULASI PROGRAM)

| No | Total Permintaan (unit per tahun) | Kuantitas Pesanan (unit) setiap kali pesan | Frekuensi Pemesanan (kali) dalam 1 tahun | Cycle time (hari) | ROP (hari) setelah pesanan diterima | Total Biaya Inventory (Rupiah) dalam 1 tahun |
|----|-----------------------------------|--|--|-------------------|-------------------------------------|--|
| 1 | 82.886 | 5.256 | 16 | 23 | 19 | 15.766.990,87 |
| 2 | 83.841 | 5.286 | 16 | 23 | 19 | 15.859.476,73 |
| 3 | 82.999 | 5.260 | 16 | 23 | 19 | 15.779.638,78 |
| 4 | 83.738 | 5.283 | 16 | 23 | 19 | 15.849.731,88 |
| 5 | 83.935 | 5.289 | 16 | 23 | 19 | 15.868.364,81 |
| 6 | 84.182 | 5.297 | 16 | 23 | 19 | 15.891.695,96 |
| 7 | 83.904 | 5.288 | 16 | 23 | 19 | 15.865.434,19 |
| 8 | 83.801 | 5.285 | 16 | 23 | 19 | 15.855.693,00 |
| 9 | 84.397 | 5.304 | 16 | 23 | 19 | 15.911.976,62 |
| 10 | 83.105 | 5.263 | 16 | 23 | 19 | 15.789.711,86 |
| 11 | 83.874 | 5.288 | 16 | 23 | 19 | 15.862.597,58 |
| 12 | 83.597 | 5.279 | 16 | 23 | 19 | 15.836.382,17 |
| 13 | 83.023 | 5.261 | 16 | 23 | 19 | 15.781.920,07 |
| 14 | 83.348 | 5.271 | 16 | 23 | 19 | 15.812.779,64 |
| 15 | 84.157 | 5.296 | 16 | 23 | 19 | 15.889.336,10 |
| 16 | 83.164 | 5.265 | 16 | 23 | 19 | 15.795.315,76 |
| 17 | 83.942 | 5.290 | 16 | 23 | 19 | 15.869.026,47 |
| 18 | 83.692 | 5.282 | 16 | 23 | 19 | 15.845.377,89 |
| 19 | 83.553 | 5.277 | 16 | 23 | 19 | 15.832.214,04 |
| 20 | 83.873 | 5.288 | 16 | 23 | 19 | 15.862.503,03 |
| 21 | 83.123 | 5.264 | 16 | 23 | 19 | 15.791.421,73 |
| 22 | 84.402 | 5.304 | 16 | 23 | 19 | 15.912.447,96 |
| 23 | 84.332 | 5.302 | 16 | 23 | 19 | 15.905.847,98 |

Tabel A-1 Hasil Uji Coba dengan Periode Harian

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 24 | 84.526 | 5.308 | 16 | 23 | 19 | 15.924.132,63 |
| 25 | 83.317 | 5.270 | 16 | 23 | 19 | 15.809.838,71 |
| 26 | 83.482 | 5.275 | 16 | 23 | 19 | 15.825.485,78 |
| 27 | 83.780 | 5.285 | 16 | 23 | 19 | 15.853.706,24 |
| 28 | 83.456 | 5.274 | 16 | 23 | 19 | 15.823.021,24 |
| 29 | 83.047 | 5.261 | 16 | 23 | 19 | 15.782.680,38 |
| 30 | 84.118 | 5.295 | 16 | 23 | 19 | 15.885.653,92 |
| 31 | 83.878 | 5.288 | 16 | 23 | 19 | 15.862.975,79 |
| 32 | 84.016 | 5.292 | 16 | 23 | 19 | 15.876.019,65 |
| 33 | 82.634 | 5.248 | 16 | 23 | 19 | 15.744.903,96 |
| 34 | 84.233 | 5.299 | 16 | 23 | 19 | 15.896.509,06 |
| 35 | 84.196 | 5.298 | 16 | 23 | 19 | 15.893.017,37 |
| 36 | 83.991 | 5.291 | 16 | 23 | 19 | 15.873.657,44 |
| 37 | 83.486 | 5.275 | 16 | 23 | 19 | 15.825.864,93 |
| 38 | 84.280 | 5.300 | 16 | 23 | 19 | 15.900.943,40 |
| 39 | 83.389 | 5.272 | 16 | 23 | 19 | 15.816.668,44 |
| 40 | 84.020 | 5.292 | 16 | 23 | 19 | 15.876.397,58 |
| 41 | 83.157 | 5.265 | 16 | 23 | 19 | 15.794.651,00 |
| 42 | 82.888 | 5.256 | 16 | 23 | 19 | 15.769.083,71 |
| 43 | 84.029 | 5.292 | 16 | 23 | 19 | 15.877.247,92 |
| 44 | 83.408 | 5.273 | 16 | 23 | 19 | 15.818.470,23 |
| 45 | 83.570 | 5.278 | 16 | 23 | 19 | 15.833.824,55 |
| 46 | 83.505 | 5.276 | 16 | 23 | 19 | 15.827.665,66 |
| 47 | 83.663 | 5.281 | 16 | 23 | 19 | 15.842.632,36 |
| 48 | 82.845 | 5.255 | 16 | 23 | 19 | 15.764.992,86 |
| 49 | 84.644 | 5.312 | 16 | 23 | 19 | 15.935.243,98 |
| 50 | 83.495 | 5.276 | 16 | 23 | 19 | 15.826.717,97 |



Tabel A-1 Hasil Uji Coba dengan Periode Harian

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 51 | 83.987 | 5.291 | 16 | 23 | 19 | 15.873.279,44 |
| 52 | 84.353 | 5.303 | 16 | 23 | 19 | 15.907.828,30 |
| 53 | 83.801 | 5.285 | 16 | 23 | 19 | 15.855.693,00 |
| 54 | 83.722 | 5.283 | 16 | 23 | 19 | 15.848.217,58 |
| 55 | 83.657 | 5.287 | 16 | 23 | 19 | 15.842.064,29 |
| 56 | 83.864 | 5.287 | 16 | 23 | 19 | 15.861.651,88 |
| 57 | 83.968 | 5.290 | 16 | 23 | 19 | 15.871.483,93 |
| 58 | 83.877 | 5.288 | 16 | 23 | 19 | 15.862.881,24 |
| 59 | 83.841 | 5.286 | 16 | 23 | 19 | 15.859.476,73 |
| 60 | 83.728 | 5.283 | 16 | 23 | 19 | 15.848.785,44 |
| 61 | 82.954 | 5.258 | 16 | 23 | 19 | 15.775.360,59 |
| 62 | 83.681 | 5.281 | 16 | 23 | 19 | 15.844.336,58 |
| 63 | 83.599 | 5.279 | 16 | 23 | 19 | 15.836.571,60 |
| 64 | 83.515 | 5.276 | 16 | 23 | 19 | 15.828.613,34 |
| 65 | 83.669 | 5.281 | 16 | 23 | 19 | 15.843.200,44 |
| 66 | 83.160 | 5.265 | 16 | 23 | 19 | 15.794.935,90 |
| 67 | 83.230 | 5.267 | 16 | 23 | 19 | 15.801.582,21 |
| 68 | 83.618 | 5.279 | 16 | 23 | 19 | 15.838.371,19 |
| 69 | 83.409 | 5.273 | 16 | 23 | 19 | 15.818.565,05 |
| 70 | 83.495 | 5.276 | 16 | 23 | 19 | 15.826.717,97 |
| 71 | 83.866 | 5.287 | 16 | 23 | 19 | 15.861.841,03 |
| 72 | 83.608 | 5.279 | 16 | 23 | 19 | 15.837.424,04 |
| 73 | 82.970 | 5.259 | 16 | 23 | 19 | 15.776.881,82 |
| 74 | 83.772 | 5.284 | 16 | 23 | 19 | 15.852.949,28 |
| 75 | 83.059 | 5.262 | 16 | 23 | 19 | 15.785.341,32 |
| 76 | 82.275 | 5.237 | 16 | 23 | 19 | 15.710.665,17 |
| 77 | 84.221 | 5.298 | 16 | 23 | 19 | 15.895.376,75 |

Tabel A-1 Hasil Uji Coba dengan Periode Harian

| | | | | | | |
|------------------|--------|-------|----|----|----|---------------|
| 78 | 83.514 | 5.276 | 16 | 23 | 19 | 15.828.518,57 |
| 79 | 83.943 | 5.290 | 16 | 23 | 19 | 15.869.120,98 |
| 80 | 84.078 | 5.294 | 16 | 23 | 19 | 15.881.876,46 |
| 81 | 83.654 | 5.281 | 16 | 23 | 19 | 15.841.780,25 |
| 82 | 83.301 | 5.269 | 16 | 23 | 19 | 15.808.320,65 |
| 83 | 83.560 | 5.278 | 16 | 23 | 19 | 15.832.877,23 |
| 84 | 82.943 | 5.258 | 16 | 23 | 19 | 15.774.314,57 |
| 85 | 83.516 | 5.276 | 16 | 23 | 19 | 15.828.708,11 |
| 86 | 83.785 | 5.285 | 16 | 23 | 19 | 15.854.179,28 |
| 87 | 83.899 | 5.288 | 16 | 23 | 19 | 15.864.961,42 |
| 88 | 82.851 | 5.255 | 16 | 23 | 19 | 15.765.563,75 |
| 89 | 83.121 | 5.264 | 16 | 23 | 19 | 15.791.231,76 |
| 90 | 83.476 | 5.275 | 16 | 23 | 19 | 15.824.917,06 |
| 91 | 84.050 | 5.293 | 16 | 23 | 19 | 15.879.231,72 |
| 92 | 84.360 | 5.303 | 16 | 23 | 19 | 15.908.488,31 |
| 93 | 83.900 | 5.288 | 16 | 23 | 19 | 15.865.055,98 |
| 94 | 83.317 | 5.270 | 16 | 23 | 19 | 15.809.838,71 |
| 95 | 83.256 | 5.268 | 16 | 23 | 19 | 15.804.050,11 |
| 96 | 83.895 | 5.288 | 16 | 23 | 19 | 15.864.583,21 |
| 97 | 83.318 | 5.270 | 16 | 23 | 19 | 15.809.933,59 |
| 98 | 84.116 | 5.295 | 16 | 23 | 19 | 15.885.465,06 |
| 99 | 83.544 | 5.277 | 16 | 23 | 19 | 15.831.361,28 |
| 100 | 83.505 | 5.276 | 16 | 23 | 19 | 15.827.665,66 |
| Rerata | 83.637 | 5.280 | 16 | 23 | 19 | 15.840.713,11 |
| Deviasi standard | 450 | 14 | 0 | 0 | 0 | 42.675,60 |

Tabel A-1 Hasil Uji Coba dengan Periode Harian

| No | Total Permintaan (unit per tahun) | Kuantitas Pesanan (unit) setiap kali pesan | Frekuensi Pemesanan (kali) dalam 1 tahun | Cycle time (hari) | ROP (hari)setelah pesanan diterima | Total Biaya Inventory (Rupiah) dalam 1 tahun |
|----|-----------------------------------|--|--|-------------------|--------------------------------------|--|
| 1 | 81.547 | 5.214 | 16 | 23 | 19 | 15.641.003,84 |
| 2 | 82.965 | 5.259 | 16 | 23 | 19 | 15.776.406,45 |
| 3 | 83.088 | 5.263 | 16 | 23 | 19 | 15.788.096,81 |
| 4 | 81.681 | 5.218 | 16 | 23 | 19 | 15.653.849,37 |
| 5 | 81.770 | 5.221 | 16 | 23 | 19 | 15.662.375,31 |
| 6 | 82.772 | 5.253 | 16 | 23 | 19 | 15.758.045,59 |
| 7 | 84.570 | 5.309 | 16 | 23 | 19 | 15.928.276,79 |
| 8 | 82.786 | 5.253 | 16 | 23 | 19 | 15.759.378,16 |
| 9 | 82.340 | 5.239 | 16 | 23 | 19 | 15.716.869,92 |
| 10 | 80.895 | 5.193 | 16 | 23 | 19 | 15.578.350,38 |
| 11 | 82.006 | 5.228 | 16 | 23 | 19 | 15.684.960,98 |
| 12 | 82.827 | 5.254 | 16 | 23 | 19 | 15.763.280,17 |
| 13 | 82.718 | 5.251 | 16 | 23 | 19 | 15.752.904,49 |
| 14 | 81.042 | 5.197 | 16 | 23 | 19 | 15.592.498,27 |
| 15 | 80.833 | 5.191 | 16 | 23 | 19 | 15.572.379,41 |
| 16 | 84.140 | 5.296 | 16 | 23 | 19 | 15.887.731,12 |
| 17 | 85.469 | 5.338 | 16 | 23 | 19 | 16.012.713,12 |
| 18 | 85.437 | 5.337 | 16 | 23 | 19 | 16.009.715,85 |
| 19 | 85.057 | 5.325 | 16 | 23 | 19 | 15.974.072,77 |
| 20 | 82.869 | 5.256 | 16 | 23 | 19 | 15.767.276,26 |
| 21 | 81.605 | 5.216 | 16 | 23 | 19 | 15.646.565,18 |
| 22 | 83.945 | 5.290 | 16 | 23 | 19 | 15.869.310,02 |
| 23 | 83.520 | 5.276 | 16 | 23 | 19 | 15.829.087,19 |

Tabel A-2 Hasil Uji Coba dengan Periode Mingguan

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 24 | 84.752 | 5.315 | 16 | 23 | 19 | 15.945.406,87 |
| 25 | 82.869 | 5.256 | 16 | 23 | 19 | 15.767.276,26 |
| 26 | 82.967 | 5.256 | 16 | 23 | 19 | 15.776.596,66 |
| 27 | 84.436 | 5.305 | 16 | 23 | 19 | 15.915.652,69 |
| 28 | 83.763 | 5.284 | 16 | 23 | 19 | 15.852.097,65 |
| 29 | 82.193 | 5.234 | 16 | 23 | 19 | 15.702.834,16 |
| 30 | 85.868 | 5.350 | 16 | 23 | 19 | 16.050.046,73 |
| 31 | 83.810 | 5.286 | 16 | 23 | 19 | 15.856.544,46 |
| 32 | 81.167 | 5.202 | 16 | 23 | 19 | 15.602.518,65 |
| 33 | 83.220 | 5.267 | 16 | 23 | 19 | 15.800.632,90 |
| 34 | 83.314 | 5.270 | 16 | 23 | 19 | 15.809.554,08 |
| 35 | 82.888 | 5.256 | 16 | 23 | 19 | 15.769.083,71 |
| 36 | 83.037 | 5.261 | 16 | 23 | 19 | 15.783.250,62 |
| 37 | 82.642 | 5.249 | 16 | 23 | 19 | 15.745.666,13 |
| 38 | 82.472 | 5.243 | 16 | 23 | 19 | 15.729.462,81 |
| 39 | 82.530 | 5.245 | 16 | 23 | 19 | 15.734.992,85 |
| 40 | 82.306 | 5.238 | 16 | 23 | 19 | 15.713.624,67 |
| 41 | 82.644 | 5.249 | 16 | 23 | 19 | 15.745.856,64 |
| 42 | 84.635 | 5.311 | 16 | 23 | 19 | 15.934.396,82 |
| 43 | 80.701 | 5.187 | 16 | 23 | 19 | 15.559.659,44 |
| 44 | 82.875 | 5.256 | 16 | 23 | 19 | 15.767.847,03 |
| 45 | 83.831 | 5.286 | 16 | 23 | 19 | 15.858.530,84 |
| 46 | 83.150 | 5.265 | 16 | 23 | 19 | 15.793.986,23 |
| 47 | 82.659 | 5.249 | 16 | 23 | 19 | 15.747.285,48 |
| 48 | 81.174 | 5.202 | 16 | 23 | 19 | 15.605.191,46 |
| 49 | 82.995 | 5.260 | 16 | 23 | 19 | 15.779.258,56 |
| 50 | 82.224 | 5.235 | 16 | 23 | 19 | 15.705.795,13 |

Tabel A-2 Hasil Uji Coba dengan Periode Mingguan

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 51 | 81.984 | 5.228 | 16 | 23 | 19 | 15.682.856,92 |
| 52 | 83.029 | 5.261 | 16 | 23 | 19 | 15.782.490,31 |
| 53 | 84.460 | 5.306 | 16 | 23 | 19 | 15.917.914,44 |
| 54 | 82.871 | 5.256 | 16 | 23 | 19 | 15.767.466,51 |
| 55 | 82.559 | 5.246 | 16 | 23 | 19 | 15.737.757,15 |
| 56 | 81.263 | 5.205 | 16 | 24 | 20 | 15.613.744,00 |
| 57 | 82.517 | 5.245 | 16 | 23 | 19 | 15.733.753,57 |
| 58 | 80.820 | 5.190 | 16 | 23 | 19 | 15.571.127,17 |
| 59 | 84.069 | 5.294 | 16 | 23 | 19 | 15.881.026,45 |
| 60 | 82.990 | 5.260 | 16 | 23 | 19 | 15.778.783,27 |
| 61 | 83.071 | 5.262 | 16 | 23 | 19 | 15.786.481,57 |
| 62 | 82.727 | 5.251 | 16 | 23 | 19 | 15.753.761,47 |
| 63 | 79.633 | 5.152 | 16 | 24 | 20 | 15.456.357,92 |
| 64 | 83.298 | 5.269 | 16 | 23 | 19 | 15.808.035,97 |
| 65 | 83.235 | 5.267 | 16 | 23 | 19 | 15.802.056,86 |
| 66 | 85.406 | 5.336 | 16 | 23 | 19 | 16.006.811,09 |
| 67 | 83.065 | 5.262 | 16 | 23 | 19 | 15.785.911,44 |
| 68 | 82.966 | 5.259 | 16 | 23 | 19 | 15.776.501,52 |
| 69 | 83.787 | 5.285 | 16 | 23 | 19 | 15.854.368,50 |
| 70 | 81.817 | 5.222 | 16 | 23 | 19 | 15.666.875,91 |
| 71 | 83.682 | 5.281 | 16 | 23 | 19 | 15.844.431,26 |
| 72 | 83.831 | 5.286 | 16 | 23 | 19 | 15.858.530,84 |
| 73 | 82.185 | 5.234 | 16 | 23 | 19 | 15.702.069,93 |
| 74 | 83.808 | 5.285 | 16 | 23 | 19 | 15.856.355,25 |
| 75 | 84.982 | 5.322 | 16 | 23 | 19 | 15.967.028,56 |
| 76 | 82.969 | 5.259 | 16 | 23 | 19 | 15.778.786,75 |
| 77 | 84.673 | 5.313 | 16 | 23 | 19 | 15.937.973,56 |

Tabel A-2 Hasil Uji Coba dengan Periode Mingguan

| | | | | | | |
|------------------|--------|-------|----|----|----|---------------|
| 78 | 82.280 | 5.237 | 16 | 23 | 19 | 15.711.142,54 |
| 79 | 82.211 | 5.235 | 16 | 23 | 19 | 15.704.533,49 |
| 80 | 84.831 | 5.318 | 16 | 23 | 19 | 15.952.836,78 |
| 81 | 85.159 | 5.328 | 16 | 23 | 19 | 15.983.647,90 |
| 82 | 81.877 | 5.224 | 16 | 23 | 19 | 15.672.619,45 |
| 83 | 83.870 | 5.287 | 16 | 23 | 19 | 15.862.219,31 |
| 84 | 80.808 | 5.190 | 16 | 23 | 19 | 15.569.971,10 |
| 85 | 83.625 | 5.280 | 16 | 23 | 19 | 15.839.034,09 |
| 86 | 86.151 | 5.359 | 16 | 23 | 19 | 16.076.473,50 |
| 87 | 84.094 | 5.294 | 16 | 23 | 19 | 15.883.387,61 |
| 88 | 83.556 | 5.277 | 16 | 23 | 19 | 15.832.498,29 |
| 89 | 83.932 | 5.289 | 16 | 23 | 19 | 15.868.081,21 |
| 90 | 82.019 | 5.229 | 16 | 23 | 19 | 15.686.204,15 |
| 91 | 82.141 | 5.233 | 16 | 23 | 19 | 15.697.866,14 |
| 92 | 82.723 | 5.251 | 16 | 23 | 19 | 15.753.380,59 |
| 93 | 82.872 | 5.256 | 16 | 23 | 19 | 15.767.561,64 |
| 94 | 82.765 | 5.252 | 16 | 23 | 19 | 15.757.379,28 |
| 95 | 82.827 | 5.254 | 16 | 23 | 19 | 15.763.280,17 |
| 96 | 83.330 | 5.270 | 16 | 23 | 19 | 15.811.072,11 |
| 97 | 84.457 | 5.306 | 16 | 23 | 19 | 15.917.631,74 |
| 98 | 83.248 | 5.268 | 16 | 23 | 19 | 15.803.290,81 |
| 99 | 85.311 | 5.333 | 16 | 23 | 19 | 15.997.906,15 |
| 100 | 85.426 | 5.336 | 16 | 23 | 19 | 16.008.685,16 |
| Rerata | 83.082 | 5.262 | 16 | 23 | 19 | 15.784.410,82 |
| Deviasi standard | 1.273 | 40 | 0 | 0 | 0 | 132.393,32 |

Tabel A-2 Hasil Uji Coba dengan Periode Mingguan

| No | Total Permintaan (unit per tahun) | Kuantitas Pesanan (unit) setiap kali pesan | Frekuensi Pemesanan (kali) dalam 1 tahun | Cycle time (hari) | ROP (hari) setelah pesanan diterima | Total Biaya Inventory (Rupiah) dalam 1 tahun |
|----|-----------------------------------|--|--|-------------------|-------------------------------------|--|
| 1 | 84.106 | 5.295 | 16 | 23 | 19 | 15.884.540,77 |
| 2 | 80.228 | 5.171 | 16 | 24 | 20 | 15.513.993,71 |
| 3 | 87.381 | 5.397 | 16 | 23 | 19 | 16.190.830,74 |
| 4 | 82.653 | 5.249 | 16 | 23 | 19 | 15.746.713,95 |
| 5 | 78.412 | 5.112 | 15 | 24 | 20 | 15.337.405,32 |
| 6 | 79.494 | 5.148 | 15 | 23 | 20 | 15.442.862,47 |
| 7 | 83.549 | 5.277 | 16 | 23 | 19 | 15.831.835,04 |
| 8 | 84.537 | 5.308 | 16 | 23 | 19 | 15.925.168,80 |
| 9 | 84.603 | 5.310 | 16 | 23 | 19 | 15.931.384,18 |
| 10 | 81.298 | 5.206 | 16 | 23 | 19 | 15.617.106,03 |
| 11 | 83.030 | 5.261 | 16 | 23 | 19 | 15.782.585,35 |
| 12 | 83.275 | 5.269 | 16 | 23 | 19 | 15.805.853,39 |
| 13 | 84.540 | 5.308 | 16 | 24 | 20 | 15.925.451,39 |
| 14 | 76.063 | 5.035 | 15 | 23 | 19 | 15.105.926,02 |
| 15 | 80.860 | 5.192 | 16 | 23 | 19 | 15.574.979,97 |
| 16 | 82.155 | 5.233 | 16 | 23 | 19 | 15.699.203,80 |
| 17 | 83.842 | 5.287 | 16 | 23 | 19 | 15.859.571,31 |
| 18 | 83.468 | 5.275 | 16 | 23 | 19 | 15.824.158,77 |
| 19 | 82.833 | 5.255 | 16 | 24 | 20 | 15.763.851,09 |
| 20 | 76.702 | 5.056 | 15 | 24 | 20 | 15.169.245,25 |
| 21 | 78.175 | 5.105 | 15 | 23 | 19 | 15.314.209,11 |
| 22 | 86.414 | 5.367 | 16 | 23 | 19 | 16.100.996,76 |
| 23 | 84.183 | 5.297 | 16 | 23 | 19 | 15.891.790,35 |

Tabel A-3 Hasil Uji Coba dengan Periode Bulanan

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 24 | 83.671 | 5.281 | 16 | 23 | 19 | 15.843.389,79 |
| 25 | 80.814 | 5.190 | 16 | 23 | 19 | 15.570.549,13 |
| 26 | 83.185 | 5.266 | 16 | 23 | 19 | 15.797.309,91 |
| 27 | 81.549 | 5.214 | 16 | 23 | 19 | 15.641.195,63 |
| 28 | 80.577 | 5.183 | 16 | 22 | 18 | 15.547.700,85 |
| 29 | 89.418 | 5.459 | 16 | 23 | 19 | 16.378.461,53 |
| 30 | 86.512 | 5.370 | 16 | 23 | 19 | 16.110.121,04 |
| 31 | 85.007 | 5.323 | 16 | 23 | 19 | 15.969.376,95 |
| 32 | 81.957 | 5.227 | 16 | 23 | 19 | 15.680.274,25 |
| 33 | 82.569 | 5.246 | 16 | 23 | 19 | 15.738.710,26 |
| 34 | 80.992 | 5.196 | 16 | 23 | 19 | 15.587.687,45 |
| 35 | 82.666 | 5.249 | 16 | 23 | 19 | 15.747.952,28 |
| 36 | 80.740 | 5.188 | 16 | 23 | 19 | 15.563.418,66 |
| 37 | 91.636 | 5.217 | 16 | 21 | 17 | 15.649.536,80 |
| 38 | 83.572 | 5.278 | 16 | 23 | 19 | 15.834.014,02 |
| 39 | 85.555 | 5.340 | 16 | 23 | 19 | 16.020.767,79 |
| 40 | 84.832 | 5.318 | 16 | 23 | 19 | 15.952.930,80 |
| 41 | 84.012 | 5.292 | 16 | 23 | 19 | 15.875.641,72 |
| 42 | 83.572 | 5.278 | 16 | 23 | 19 | 15.834.014,62 |
| 43 | 82.717 | 5.251 | 16 | 23 | 19 | 15.752.809,27 |
| 44 | 84.817 | 5.317 | 16 | 23 | 19 | 15.951.520,31 |
| 45 | 79.506 | 5.148 | 15 | 24 | 20 | 15.444.027,97 |
| 46 | 83.998 | 5.291 | 16 | 23 | 19 | 15.874.318,94 |
| 47 | 81891 | 5.225 | 16 | 23 | 19 | 15.673.959,33 |
| 48 | 83.236 | 5.267 | 16 | 23 | 19 | 15.802.151,79 |
| 49 | 84.028 | 5.292 | 16 | 23 | 19 | 15.877.153,44 |
| 50 | 81.561 | 5.214 | 16 | 23 | 19 | 15.642.346,38 |

Tabel A-3 Hasil Uji Coba dengan Periode Bulanan

| | | | | | | |
|----|--------|-------|----|----|----|---------------|
| 51 | 79.584 | 5.151 | 15 | 24 | 20 | 15.451.601,92 |
| 52 | 82.343 | 5.239 | 16 | 23 | 19 | 15.717.156,23 |
| 53 | 80.381 | 5.176 | 16 | 24 | 20 | 15.528.779,75 |
| 54 | 86.162 | 5.359 | 16 | 23 | 19 | 16.077.499,81 |
| 55 | 82.595 | 5.247 | 16 | 23 | 19 | 15.741.188,01 |
| 56 | 84.819 | 5.317 | 16 | 23 | 19 | 15.951.708,39 |
| 57 | 84.225 | 5.299 | 16 | 23 | 19 | 15.895.754,20 |
| 58 | 80.150 | 5.169 | 16 | 24 | 20 | 15.506.450,28 |
| 59 | 88.214 | 5.423 | 16 | 22 | 18 | 16.267.821,04 |
| 60 | 80.406 | 5.177 | 16 | 24 | 20 | 15.531.194,42 |
| 61 | 83.352 | 5.271 | 16 | 23 | 19 | 15.813.159,08 |
| 62 | 83.485 | 5.275 | 16 | 23 | 19 | 15.825.770,14 |
| 63 | 83.226 | 5.267 | 16 | 23 | 19 | 15.801.202,49 |
| 64 | 85.826 | 5.349 | 16 | 23 | 19 | 16.046.121,05 |
| 65 | 83.701 | 5.282 | 16 | 23 | 19 | 15.846.229,84 |
| 66 | 80.176 | 5.170 | 16 | 24 | 20 | 15.508.956,18 |
| 67 | 79.081 | 5.134 | 15 | 24 | 20 | 15.402.694,59 |
| 68 | 85.840 | 5.349 | 16 | 23 | 19 | 16.047.429,71 |
| 69 | 85.756 | 5.347 | 16 | 23 | 19 | 16.039.576,12 |
| 70 | 84.118 | 5.295 | 16 | 23 | 19 | 15.885.653,92 |
| 71 | 83.277 | 5.269 | 16 | 23 | 19 | 15.806.043,18 |
| 72 | 81.246 | 5.204 | 16 | 23 | 19 | 15.612.110,68 |
| 73 | 84.154 | 5.299 | 16 | 23 | 19 | 15.898.490,56 |
| 74 | 85.364 | 5.334 | 16 | 23 | 19 | 16.002.781,03 |
| 75 | 86.770 | 5.378 | 16 | 23 | 19 | 16.134.125,33 |
| 76 | 84.147 | 5.296 | 16 | 23 | 19 | 15.888.391,99 |
| 77 | 82.520 | 5.245 | 16 | 23 | 19 | 15.734.039,56 |

Tabel A-3 Hasil Uji Coba dengan Periode Bulanan

| | | | | | | |
|------------------|--------|-------|----|----|----|---------------|
| 78 | 87.800 | 5.410 | 16 | 22 | 18 | 16.229.602,59 |
| 79 | 83.424 | 5.273 | 16 | 23 | 19 | 15.819.987,39 |
| 80 | 81.297 | 5.206 | 16 | 23 | 19 | 15.617.09,99 |
| 81 | 80.722 | 5.187 | 16 | 23 | 19 | 15.561.683,73 |
| 82 | 81.686 | 5.218 | 16 | 23 | 19 | 15.654.328,48 |
| 83 | 84.185 | 5.297 | 16 | 23 | 19 | 15.891.976,14 |
| 84 | 90.486 | 5.492 | 16 | 22 | 18 | 16.475.982,52 |
| 85 | 82.856 | 5.255 | 16 | 23 | 19 | 15.766.039,49 |
| 86 | 85.297 | 5.332 | 16 | 23 | 19 | 15.996.593,40 |
| 87 | 79.405 | 5.145 | 15 | 24 | 20 | 15.434.215,26 |
| 88 | 80.972 | 5.195 | 16 | 23 | 19 | 15.585.762,75 |
| 89 | 78.159 | 5.104 | 16 | 24 | 20 | 15.312.641,85 |
| 90 | 83.970 | 5.291 | 16 | 23 | 19 | 15.871.672,94 |
| 91 | 85.311 | 5.333 | 16 | 23 | 19 | 15.997.906,15 |
| 92 | 81.611 | 5.216 | 16 | 23 | 19 | 15.647.140,34 |
| 93 | 83.712 | 5.282 | 16 | 23 | 19 | 15.847.271,11 |
| 94 | 88.527 | 5.432 | 16 | 22 | 18 | 16.296.656,11 |
| 95 | 85.335 | 5.333 | 15 | 23 | 19 | 16.000.156,29 |
| 96 | 85.072 | 5.325 | 16 | 23 | 19 | 15.975.481,22 |
| 97 | 78.049 | 5.133 | 16 | 24 | 20 | 15.399.577,93 |
| 98 | 82.202 | 5.235 | 16 | 23 | 19 | 15.703.693,89 |
| 99 | 83.125 | 5.264 | 16 | 23 | 19 | 15.791.611,70 |
| 100 | 79.764 | 5.156 | 15 | 24 | 20 | 15.469.065,94 |
| Rerata | 83.103 | 5.260 | 16 | 23 | 19 | 15.784.584,56 |
| Deviasi Standard | 2.774 | 83 | 0 | 1 | 0 | 256.969,58 |

Tabel A-3 Uji Coba dengan Periode Bulanan

LAMPIRAN B
(Listing Program)

```
Option Explicit
Option Base 1
Dim data(), total_permintaan, q As Long
Dim frekuensi, cycle, rop As Long
Dim hasil() As Single
Dim p As Integer
```

FUNGSI PENGHITUNGAN *STOCK* PERIODE HARIAN

```
Function Hitung_stock_harian()
Dim hitung_hari As Integer
Dim c0, stockawal As Long
Dim order1, order, n As Integer
ReDim stockakhir(p), datang(p)
Dim waktu As Integer
```

```
List2.Clear
List3.Clear
```

```
order = 0
stockawal = txtstock.Text
n = 1
If q = 0 Then
    c0 = 0
Else
    c0 = CLng(txtrata.Text / q)
End If
hitung_hari = txttime.Text
waktu = 1
```

```
While n <= p
    While hitung_hari <= waktu And order < frekuensi
        If cycle = 0 Or (c0 > 1 And cycle < 0) Then
            order1 = 0
            While order1 < c0 And order < frekuensi
                stockawal = stockawal + q
                datang(n) = stockawal
                List3.AddItem datang(n)
                order = order + 1
                order1 = order1 + 1
            Wend
            hitung_hari = waktu + 1
        Else
            stockawal = stockawal + q
            datang(n) = stockawal
            List3.AddItem datang(n)
            order = order + 1
        End If
    End While
    n = n + 1
End While
```

```

        hitung_hari = hitung_hari + cycle
    End If
Wend
    stockakhir(n) = stockawal - data(n)
    List2.AddItem stockakhir(n)
    stockawal = stockakhir(n)
    n = n + 1
    waktu = waktu + 1
Wend
    MSCstock.ChartData = stockakhir
    MSCorderdatang.ChartData = datang
End Function

```

FUNGSI PENGHITUNGAN STOCK PERIODE MINGGUAN

```
Function hitung_stock_mingguan()
```

```
Dim hitung_hari As Integer
```

```
Dim c0, stockawal As Long
```

```
Dim order1, n As Integer
```

```
ReDim stockakhir(p), datang(p)
```

```
Dim waktu, order As Integer
```

```
List2.Clear
```

```
List3.Clear
```

```
order = 0
```

```
stockawal = txtstock.Text
```

```
n = 1
```

```
If q = 0 Then
```

```
    c0 = 0
```

```
Else
```

```
    c0 = CLng(txttrata.Text / q)
```

```
End If
```

```
hitung_hari = txttime.Text
```

```
waktu = 7
```

```
While n <= p
```

```
    While hitung_hari <= waktu And order < frekuensi And waktu <= 365
```

```
        If cycle = 0 Or (c0 > 7 And cycle < 0) Then
```

```
            order1 = 0
```

```
            While order1 < c0 And order < frekuensi
```

```
                stockawal = stockawal + q
```

```
                datang(n) = stockawal
```

```
                List3.AddItem datang(n)
```

```
                order = order + 1
```

```
                order1 = order1 + 1
```

```
            Wend
```



```

        hitung_hari = waktu + 1
    Else
        stockawal = stockawal + q
        datang(n) = stockawal
        List3.AddItem datang(n)
        order = order + 1
        hitung_hari = hitung_hari + cycle
    End If
Wend
    stockakhir(n) = stockawal - data(n)
    List2.AddItem stockakhir(n)
    stockawal = stockakhir(n)
    n = n + 1
    If n = 52 Then
        waktu = waktu + 8
    Else
        waktu = waktu + 7
    End If
Wend
    MSCstock.ChartData = stockakhir
    MSCorderdatang.ChartData = datang
End Function

```

FUNGSI PENGHITUNGAN STOCK PERIODE BULANAN

```

Function hitung_stock_bulanan()
Dim hitung_hari As Integer
Dim c0, stockawal As Long
Dim order1, n As Integer
ReDim stockakhir(p), datang(p)
Dim waktu, order As Integer

List2.Clear
List3.Clear
MSCorderdatang.Refresh
MSCstock.Refresh

order = 0
stockawal = txtstock.Text
n = 1
If q = 0 Then
    c0 = 0
Else
    c0 = CLng(txtrata.Text / q)
End If
hitung_hari = txttime.Text
waktu = 31

```

```

While n <= p
  While hitung_hari <= waktu And order < frekuensi And waktu <= 365
    If cycle = 0 Or (c0 > 28 And cycle < 0) Then
      order1 = 0
      While order1 < c0 And order < frekuensi
        stockawal = stockawal + q
        datang(n) = stockawal
        List3.AddItem datang(n)
        order = order + 1
        order1 = order1 + 1
      Wend
      hitung_hari = waktu + 1
    Else
      stockawal = stockawal + q
      datang(n) = stockawal
      List3.AddItem datang(n)
      order = order + 1
      hitung_hari = hitung_hari + cycle
    End If
  Wend
  stockakhir(n) = stockawal - data(n)
  List2.AddItem stockakhir(n)
  stockawal = stockakhir(n)
  n = n + 1
  If n = 2 Then
    waktu = waktu + 28
  ElseIf n = 4 Or n = 6 Or n = 9 Or n = 11 Then
    waktu = waktu + 30
  Else
    waktu = waktu + 31
  End If
Wend

MSCstock.ChartData = stockakhir
MSCorderdatang.ChartData = datang
End Function

```

FUNGSI PILIHAN PENGHITUNGAN STOCK

```
Function fungsi_stock()
```

```

List2.Clear 'Membersihkan list2
List3.Clear
If p = 365 Then
  Hitung_stock_harian
ElseIf p = 52 Then
  hitung_stock_mingguan

```

```
Else
  hitung_stock_bulanan
End If
End Function
```

PEMBANGKITAN BILANGAN ACAK & PENGHITUNGAN RUMUS

```
Function fungsi_generate()
```

```
Const pi = 3.14
```

```
Dim biaya As Currency
```

```
Dim u1, u2, x As Single
```

```
Dim i As Integer
```

```
Dim teks
```

```
On Error GoTo kesalahan:
```

```
List1.Clear
```

```
List1.Refresh
```

```
For i = 1 To p
```

```
  u1 = Rnd()
```

```
  u2 = Rnd()
```

```
  x = (Sqr(pi / 8)) * (Math.Log((1 + u1) / (1 - u1)))
```

```
  If u2 <= 0.5 Then
```

```
    hasil(i) = (-x)
```

```
  Else
```

```
    hasil(i) = (x)
```

```
  End If
```

```
Next i
```

```
Dim n As Integer
```

```
total_permintaan = 0
```

```
For n = 1 To p
```

```
  data(n) = CLng((hasil(n) * (10 / 100 * txtrata.Text)) + txtrata.Text)
```

```
  List1.AddItem data(n)
```

```
  total_permintaan = total_permintaan + data(n)
```

```
Next n
```

```
lbljumlah.Caption = total_permintaan
```

```
MSCpermintaan.ChartData = data
```

```
Dim teksb
```

```
If txtbiayahold.Text = 0 Then
```

```
  teksb = MsgBox("Biaya penyimpanan tidak boleh NOL", vbCritical +  
vbOKOnly, "Pesan Kesalahan")
```

```
Else
```

```
  If total_permintaan = 0 Then
```

```

teksb = MsgBox("Tidak ada permintaan", vbOKOnly, "Pesan")
q = 0
lblq.Caption = q
Else
q = CLng(Sqr((2 * Txtbiayaorder.Text * total_permintaan) /
txtbiayahold.Text))
lblq.Caption = q
If q > total_permintaan Then
teksb = MsgBox("Kuantitas pesanan lebih besar dari Total Permintaan ",
vbOKOnly, "Pesan")
End If
End If

If q = 0 Then
teksb = MsgBox("Tidak ada pemesanan.", vbOKOnly, "Pesan")
frekuensi = 0
lblfrekuensi.Caption = frekuensi
cycle = 0
lblcycleharian.Caption = frekuensi
rop = 0
lblrop.Caption = frekuensi
biaya = 0
lblbiaya.Caption = 0
fungsi_stock
Else
frekuensi = CLng(total_permintaan / q)
lblfrekuensi.Caption = frekuensi

If p = 365 Then
cycle = CLng((q / total_permintaan) * 365)
lblcycleharian.Caption = cycle
ElseIf p = 52 Then
cycle = CLng((q / total_permintaan) * 365)
lblcycleharian.Caption = cycle
ElseIf p = 12 Then
cycle = CLng((q / total_permintaan) * 365)
lblcycleharian.Caption = cycle
End If

If Val(txttime.Text) > cycle Then
teks = MsgBox("Lead time lebih besar dari cycle time." & Chr(13) &
Chr(10) & " Input ulang lead time ?", vbYesNo + vbQuestion, "Pesan")
If teks = vbNo Then
rop = (cycle - txttime.Text)
If rop = 0 Then
lblrop.Caption = ""

```

```

        lblketeranganrop.Caption = "Saat pesanan diterima"
    Else
        lblrop.Caption = rop
        lblketeranganrop.Caption = "hari setelah pesanan diterima"
    End If

    biaya = ((txtbiayahold.Text * q) / 2) + ((Txtbiayaorder.Text *
total_permintaan) / q)
    lblbiaya.Caption = FormatCurrency(biaya, 2)
    fungsi_stock
End If
Else
    rop = (cycle - txttime.Text)
    If rop = 0 Then
        lblrop.Caption = ""
        lblketeranganrop.Caption = "Saat pesanan diterima"
    Else
        lblrop.Caption = rop
        lblketeranganrop.Caption = "hari setelah pesanan diterima"
    End If

    biaya = ((txtbiayahold.Text * q) / 2) + ((Txtbiayaorder.Text *
total_permintaan) / q)
    lblbiaya.Caption = FormatCurrency(biaya, 2)
    fungsi_stock
End If
End If
Exit Function
kesalahan:
If txtrata.Text = "" Then
    MsgBox "Inputkan rata-rata permintaan!", , "Pesanan"
End If
If Txtbiayaorder.Text = "" Then
    MsgBox "Inputkan biaya pemesanan!", , "Pesanan"
End If
If txtbiayahold.Text = "" Then
    MsgBox "Inputkan biaya penyimpanan!", , "Pesanan"
End If
If txtstock.Text = "" Then
    MsgBox "Inputkan stock awal!", , "Pesanan"
End If
If txttime.Text = "" Then
    MsgBox "Inputkan Lead time!", , "Pesanan"
End If
End Function

```

LIST BERIKUT DIJALANKAN SAAT TOMBOL SIMULASI DI-CLICK

```
Private Sub cmdgenerate_Click()
```

```
On Error GoTo kesalahan:
```

```
Dim teks
```

```
    If p = 365 Then
```

```
        lblketeranganrata.Caption = "per hari"
```

```
    ElseIf p = 52 Then
```

```
        lblketeranganrata.Caption = " per minggu"
```

```
    ElseIf p = 12 Then
```

```
        lblketeranganrata.Caption = "per bulan"
```

```
    End If
```

```
    fungsi_generate
```

```
Exit Sub
```

```
kesalahan:
```

```
    MsgBox "Lengkapi input data anda", , "Pesan"
```

```
End Sub
```

LIST BERIKUT DIJALANKAN SAAT TOMBOL HITUNG DI-CLICK

```
Private Sub cmdhitung_Click()
```

```
Dim biaya As Currency
```

```
Dim i As Integer
```

```
Dim teks
```

```
On Error GoTo kesalahan:
```

```
List1.Clear
```

```
If txtbiayahold.Text = 0 Then
```

```
    teks = MsgBox("Biaya penyimpanan tidak boleh NOL", vbCritical +  
vbOKOnly, "Pesan kesalahan")
```

```
Else
```

```
    total_permintaan = 0
```

```
    For i = 1 To p
```

```
        data(i) = CLng((hasil(i) * (10 / 100 * txtrata.Text)) + txtrata.Text)
```

```
        List1.AddItem data(i)
```

```
        total_permintaan = total_permintaan + data(i)
```

```
    Next i
```

```
    lbljumlah.Caption = total_permintaan
```

```
    MSCpermintaan.ChartData = data
```

```
    If total_permintaan = 0 Then
```

```
        teks = MsgBox("Tidak ada permintaan", vbOKOnly, "Pesan")
```

```
        q = 0
```

```
        lblq.Caption = 0
```

```
    Else
```

```

    q = CLng(Sqr((2 * Txtbiayaorder.Text * total_permintaan) /
txtbiayahold.Text))
    lblq.Caption = q
    If q > total_permintaan Then
        teks = MsgBox("Kuantitas pesanan lebih besar dari Total Permintaan ",
vbOKOnly, "Pesanan")
    End If
End If

If q = 0 Then
    teks = MsgBox("Tidak ada pemesanan. ", vbOKOnly, "Pesanan")
    frekuensi = 0
    lblfrekuensi.Caption = frekuensi
    cycle = 0
    lblcycleharian.Caption = cycle
    rop = 0
    lblrop.Caption = rop
    biaya = 0
    lblbiaya.Caption = biaya
    fungsi_stock
Else
    frekuensi = CLng(total_permintaan / q)
    lblfrekuensi.Caption = frekuensi

    If p = 365 Then
        cycle = CLng((q / total_permintaan) * 365)
        lblcycleharian.Caption = cycle
    ElseIf p = 52 Then
        cycle = CLng((q / total_permintaan) * 365)
        lblcycleharian.Caption = cycle
    ElseIf p = 12 Then
        cycle = CLng((q / total_permintaan) * 365)
        lblcycleharian.Caption = cycle
    End If

    If Val(txttime.Text) > cycle Then
        teks = MsgBox("Lead time lebih besar dari cycle time." & Chr(13) &
Chr(10) & " Input ulang lead time ?", vbYesNo + vbQuestion, "Pesanan")
        If teks = vbNo Then
            rop = (cycle - txttime.Text)
            If rop = 0 Then
                lblrop.Caption = ""
                lblketeranganrop.Caption = "Saat pesanan diterima"
            Else
                lblrop.Caption = rop
                lblketeranganrop.Caption = "hari setelah pesanan diterima"
            End If
        End If
    End If

```

```

        End If

        biaya = ((txtbiayahold.Text * q) / 2) + ((Txtbiayaorder.Text *
total_permintaan) / q)
        lblbiaya.Caption = FormatCurrency(biaya, 2)
        fungsi_stock
        End If
    Else
        rop = (cycle - txttime.Text)
        If rop = 0 Then
            lblrop.Caption = ""
            lblketeranganrop.Caption = "Saat pesanan diterima"
        Else
            lblrop.Caption = rop
            lblketeranganrop.Caption = "hari setelah pesanan diterima"
        End If

        biaya = ((txtbiayahold.Text * q) / 2) + ((Txtbiayaorder.Text *
total_permintaan) / q)
        lblbiaya.Caption = FormatCurrency(biaya, 2)
        fungsi_stock
        End If

    End If
End If
Exit Sub

```

kesalahan:

```

    If txtrata.Text = "" Then
        MsgBox "Inputkan rata-rata permintaan!", , "Pesan"
    End If
    If Txtbiayaorder.Text = "" Then
        MsgBox "Inputkan biaya pemesanan!", , "Pesan"
    End If
    If txtbiayahold.Text = "" Then
        MsgBox "Inputkan biaya penyimpanan!", , "Pesan"
    End If
    If txtstock.Text = "" Then
        MsgBox "Inputkan stockawal!", , "Pesan"
    End If
    If txttime.Text = "" Then
        MsgBox "Inputkan Lead time!", , "Pesan"
    End If
End Sub

```


LIST BERIKUT DIJALANKAN SAAT TOMBOL INFO DI-CLICK

```
Private Sub cmdinfo_Click()  
    frminfo.Show  
End Sub
```

LIST BERIKUT DIJALANKAN SAAT TOMBOL EXIT DI-CLICK

```
Private Sub cmdkeluar_Click()  
    Unload Me  
End Sub
```

LIST BERIKUT DIJALANKAN SAAT PILIHAN COMBO BOX DI UBAH

```
Private Sub Combo1_Click()  
    Dim j, periode As Integer
```

```
    For j = 0 To 2  
        If (StrComp(Combo1.Text, Combo1.List(j))) = 0 Then  
            periode = j  
        End If  
    Next j
```

```
    If periode = 0 Then  
        p = 365  
        ReDim data(p), hasil(p)  
    ElseIf periode = 1 Then  
        p = 52  
        ReDim data(p), hasil(p)  
    Else  
        p = 12  
        ReDim data(p), hasil(p)  
    End If
```

```
End Sub
```

PROSEDUR YANG DIMUAT TIAP KALI PROGRAM DIJALANKAN

```
Private Sub Form_Load()
```

```
    Randomize  
    Combo1.AddItem "Harian"  
    Combo1.AddItem "Mingguan"  
    Combo1.AddItem "Bulanan"  
    Combo1.Text = Combo1.List(1)
```

```
    lblketeranganata.Caption = "per minggu"
```

fungsi_generate

End Sub

LIST BERIKUT DIJALANKAN UNUTK PENGECEKAN INPUT

Private Sub txtbiayahold_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then

 Beep

 KeyAscii = 0

End If

End Sub

Private Sub txtbiayaorder_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then

 Beep

 KeyAscii = 0

End If

End Sub

Private Sub txttrata_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then

 Beep

 KeyAscii = 0

End If

End Sub

Private Sub txtstock_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then

 Beep

 KeyAscii = 0

End If

End Sub

Private Sub txttime_KeyPress(KeyAscii As Integer)

If Not (KeyAscii >= Asc("0") And KeyAscii <= Asc("9") Or KeyAscii = vbKeyBack) Then

 Beep

 KeyAscii = 0

End If

End Sub

