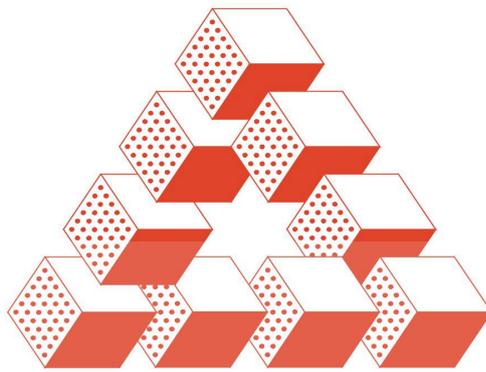
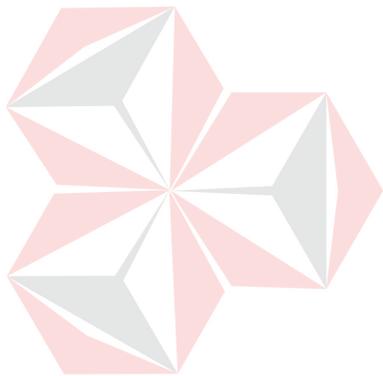


**PEMBUATAN APLIKASI PENENTUAN TITIK PESAN KEMBALI
DENGAN MENERAPKAN METODE FUZZY LOGIC
PADA PT. DITA JAYA PERKASA**



UNIVERSITAS
STIKOM
Dinamika

Oleh :

Nama : DIAZ PENMIL INDRAWANTO

NIM : 00.41010.0036

Program : S1 (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

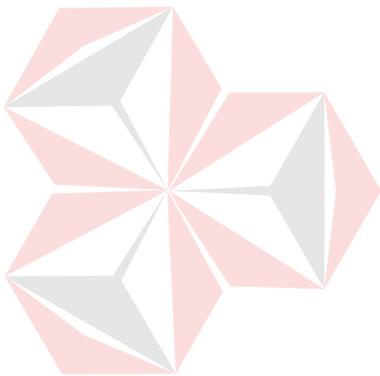
2006

**PEMBUATAN APLIKASI PENENTUAN TITIK PESAN KEMBALI
DENGAN MENERAPKAN METODE FUZZY LOGIC
PADA PT. DITA JAYA PERKASA**

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



UNIVERSITAS
Dinamika

Oleh :

Nama : DIAZ PENMIL INDRAWANTO

NIM : 00.41010.0036

Program : SI (Strata Satu)

Jurusan : Sistem Informasi

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**

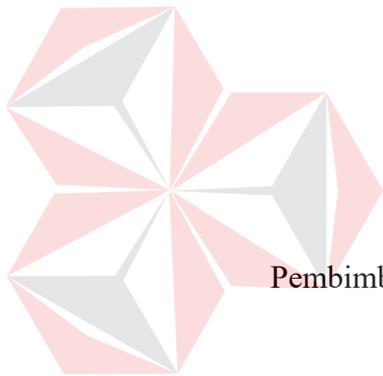
2006

**PEMBUATAN APLIKASI PENENTUAN TITIK PESAN KEMBALI
DENGAN MENERAPKAN METODE FUZZY LOGIC
PADA PT. DITA JAYA PERKASA**

Disusun Oleh :

Nama : Diaz Penmil Indrawanto

NIM : 00.41010.0036



Pembimbing I

Surabaya, Agustus 2006

Telah diperiksa, diuji dan disetujui :

Pembimbing II

Asti Dwi Irfianti, M.Kom
NIDN 0717027301

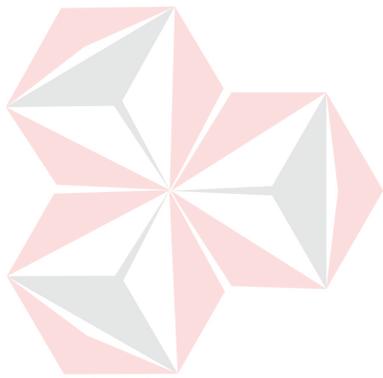
Romeo, ST
NIDN 0705087301

Mengetahui :

Wakil Ketua Bidang Akademik

Drs. Antok Supriyanto, M.MT
NIDN 0726106201

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA**



Ku persembahkan kepada

Ayah & Ibu yang tercinta

dan

Tri Astutik Julianingsih

yang tersayang

UNIVERSITAS
Dinamika

ABSTRAKSI

Pengawasan dan pemeliharaan persediaan adalah masalah yang sering dijumpai dalam semua organisasi disetiap sektor ekonomi. Masalah persediaan tidak hanya terbatas pada perusahaan pencari keuntungan saja tetapi juga dialami oleh perusahaan *non profit oriented*.

Salah satu alasan utama mengapa perusahaan mempunyai persediaan adalah agar perusahaan dapat membeli atau membuat item dalam jumlah yang paling ekonomis. Pertanyaan mendasar yang harus dijawab adalah “berapa banyak” dan “kapan” melakukan pemesanan. Untuk menjawab pertanyaan tersebut sangat tergantung pada parameter seperti : permintaan, tenggang waktu dan biaya persediaan. Untuk mengatasi hal tersebut, diperlukan suatu alat bantu aplikasi yang dapat mengetahui titik pesan kembali suatu item barang. Salah satu metode yang dapat digunakan untuk mengetahui titik pesan kembali adalah *fuzzy logic*.

Fuzzy logic adalah suatu metode yang dapat menghasilkan nilai output yang pasti berdasarkan inputan yang tidak pasti atau samar, karena parameter penentuan pemesanan bersifat tidak pasti.

Dengan adanya penelitian ini, diharapkan dapat meningkatkan pendapatan perusahaan dengan menekan biaya-biaya yang seharusnya tidak perlu dikeluarkan.

Kata kunci: Fuzzy Logic, persediaan, titik pesan kembali.

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul “Pembuatan Aplikasi Penentuan Titik Pesan Kembali dengan Menerapkan Metode Fuzzy Logic pada PT. Dita Jaya Perkasa” ini dengan baik dan lancar.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk meraih gelar Sarjana Sistem Informasi pada Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya. Terima kasih kepada Bapak Romeo, ST selaku Dosen Pembimbing II yang telah meluangkan waktu untuk memberikan bimbingan dan petunjuk selama pelaksanaan tugas akhir dengan sabar & Ibu Asti Dwi Irfianti, M.Kom., selaku Dosen Pembimbing I yang telah banyak mendukung dan memberikan kepercayaan penuh kepada penulis dalam menyelesaikan tugas akhir ini.

Penyelesaian laporan tugas akhir ini tidak lepas dari bantuan banyak pihak yang benar-benar memberikan masukan dan dukungan kepada penulis. Untuk ini penulis juga mengucapkan banyak terima kasih kepada:

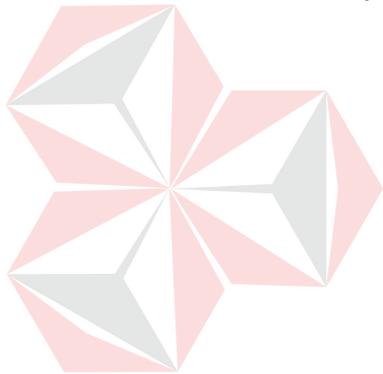
1. Bapak Haryanto Tanuwijaya, S.Kom., M.MT., selaku Ketua Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya.
2. PT Dita Jaya Perkasa, yang telah memberikan ijin untuk mengadakan penelitian dan telah memberikan data-data yang dibutuhkan untuk penyusunan Tugas Akhir ini.
3. Kru GAPTEK tanpa terkecuali, atas kritik, gangguan dan bantuannya.

4. Rekan mahasiswa STIKOM Surabaya tanpa terkecuali, atas perkenalan kita.
5. Orang tua penulis yang telah memberikan dorongan baik secara moril maupun materiil kepada kami.
6. Semua pihak yang secara langsung maupun tidak langsung, yang tidak bisa penulis sebutkan satu-persatu yang telah membantu memberikan dukungan moral spiritual dan material kepada penulis.

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan, baik dari materi maupun teknik penyajiannya. Untuk itu segala kritik dan saran membangun, sangat penulis harapkan semoga laporan ini dapat memberikan guna dan manfaatnya.

Surabaya, Agustus 2006

Penulis

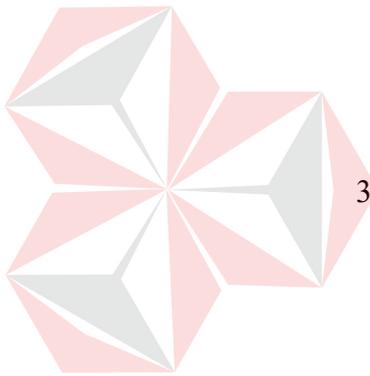


UNIVERSITAS
Dinamika

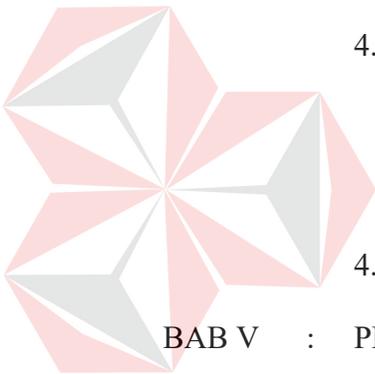
DAFTAR ISI

	Halaman
ABSTRAKSI	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xvi
BAB I : PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Sistematika Penulisan	3
BAB II : LANDASAN TEORI	4
2.1 Persediaan	4
2.2 Distribusi Persediaan.....	5
2.3 Teknik Peramalan	9
2.4 Economic Order Quantity	12
2.5 Fuzzy Logic	13
2.5.1 Himpunan Fuzzy	13
2.5.2 Fungsi Keanggotaan Fuzzy	14

2.5.3 Operasi Himpunan Fuzzy.....	15
2.5.4 Variabel Linguistik.....	15
2.5.5 Fuzzy Rule-Base	17
2.5.6 Perhitungan Fuzzy.....	18
2.6 Konsep Basis Data	21
2.7 Data Flow Diagram (DFD).....	22
2.8 Entity Relationship Diagram (ERD).....	27
BAB III : PERANCANGAN SISTEM	30
3.1 Analisis Sistem	30
3.1.1 Alur Proses Perhitungan Menggunakan Fuzzy Logic	31
3.1.2 Penentuan Rulebase	33
3.2 Data Flow Diagram.....	36
3.2.1 Context Diagram.....	36
3.2.2 Diagram Berjenjang	37
3.2.3 DFD Level 0 Aplikasi Penentuan Titik Pesan Kembali	38
3.2.4 DFD Level 1 Proses Migrasi.....	39
3.2.5 DFD Level 1 Proses Konfigurasi Fuzzy	39
3.2.6 DFD Level 1 Proses Perhitungan Fuzzy	40
3.3 Entity Relational Diagram	41
3.3.1 Conceptual Data Model	41
3.3.2 Physical Data Model	42
3.4 Struktur Database.....	42
3.5 Rancangan Input Output	46



3.5.1	Rancangan Input.....	46
3.5.2	Rancangan Output.....	50
3.6	Desain Uji Coba.....	50
3.6.1	Desain Uji Coba Fungsional	50
3.6.2	Desain Uji Coba Validasi.....	55
BAB IV	: IMPLEMENTASI DAN EVALUASI SISTEM.....	57
4.1	Kebutuhan Sistem	57
4.2	Instalasi dan Konfigurasi	57
4.3	Implementasi Aplikasi Penentuan Titik Pesan Kembali	58
4.4	Hasil Uji Coba.....	63
4.4.1	Hasil Uji Coba Fungsional.....	63
4.4.2	Hasil Uji Coba Validasi	71
4.5	Analisa Hasil Uji Coba.....	74
BAB V	: PENUTUP	77
5.1	Kesimpulan	77
5.2	Saran	77
DAFTAR PUSTAKA	78
LAMPIRAN	79

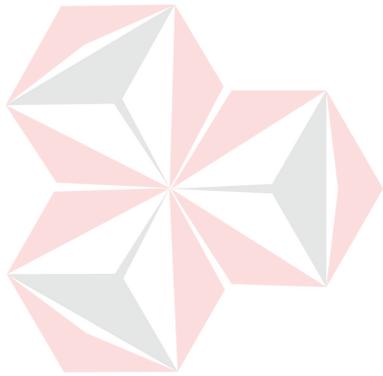


UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 3.1 Bahasa Fuzzy	33
Tabel 3.2 Kombinasi Rulebase Rata-rata Delay Pembelian.....	34
Tabel 3.3 Kombinasi Rulebase Penjualan	35
Tabel 3.4 Tabel Obat	43
Tabel 3.5 Tabel Penjualan	43
Tabel 3.6 Tabel Pembelian	44
Tabel 3.7 Tabel Periode.....	44
Tabel 3.8 Tabel Fuzzy_Set.....	44
Tabel 3.9 Tabel Fs_Detail	45
Tabel 3.10 Tabel Rule_Base.....	45
Tabel 3.11 Tabel dbUser	45
Tabel 3.12 Tabel Uji Main Menu.....	51
Tabel 3.13 Tabel Uji Automasi	51
Tabel 3.14 Tabel Uji Batasan Fuzzy	52
Tabel 3.15 Tabel Uji Aturan Fuzzy	53
Tabel 3.16 Tabel Uji Maintenance User.....	54
Tabel 3.17 Tabel Uji Proses	54
Tabel 3.18 Perhitungan Pembelian Amoxan 500mg	55
Tabel 3.19 Perhitungan Penjualan Amoxan 500mg	55
Tabel 3.20 Hasil Perhitungan Titik Pesan Kembali Amoxan 500mg	55
Tabel 3.21 Perhitungan Pembelian Hemaviton	55
Tabel 3.22 Perhitungan Penjualan Hemaviton	56

	Halaman
Tabel 3.23 Hasil Perhitungan Titik Pesan Kembali Hemaviton.....	56
Tabel 4.1 Tabel Hasil Uji Main Menu.....	63
Tabel 4.2 Tabel Hasil Uji Automasi.....	64
Tabel 4.3 Tabel Hasil Uji Batasan Fuzzy	65
Tabel 4.4 Tabel Hasil Uji Aturan Fuzzy	67
Tabel 4.5 Tabel Hasil Uji Maintenance User	69
Tabel 4.6 Tabel Hasil Uji Proses	70
Tabel 4.7 Tabel TPK dan Rata-rata Penjualan	75
Tabel 4.8 Tabel Simulasi Pembelian	75



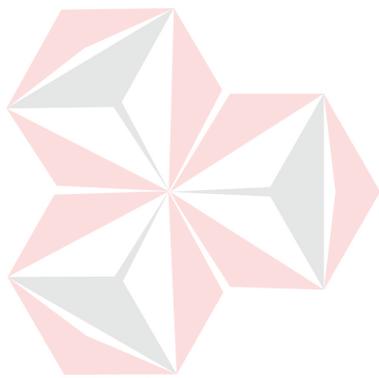
UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model EOQ	6
Gambar 2.2 Grafik Fungsi Keanggotaan Jumlah	17
Gambar 2.3 Diagram Alir Proses Pengaturan Himpunan <i>Fuzzy</i>	18
Gambar 2.3 Konsep Data Flow	22
Gambar 2.4 Contoh Ilustrasi Detil Child Diagram	24
Gambar 2.5 Kesalahan Penulisan Proses Dalam DFD	25
Gambar 3.1 Diagram Alir Aplikasi Penentuan Titik Pesan Kembali	31
Gambar 3.2 Alur Proses Perhitungan Menggunakan <i>Fuzzy Logic</i>	31
Gambar 3.3. Grafik Bahasa Fuzzy	33
Gambar 3.4 <i>Context Diagram</i>	36
Gambar 3.5 Diagram Berjenjang.....	37
Gambar 3.6 DFD Level 0 Aplikasi Penentuan Titik Pesan Kembali	38
Gambar 3.7 DFD Level 1 Proses Migrasi	39
Gambar 3.8 DFD Level 1 Proses Konfigurasi Fuzzy	39
Gambar 3.9 DFD Level 1 Proses Perhitungan Fuzzy	40
Gambar 3.10 <i>Conceptual Data Model</i>	41
Gambar 3.11 <i>Physical Data Model</i>	42
Gambar 3.12 Rancangan Form Automasi.....	46
Gambar 3.13 Rancangan Form Input Nilai Batasan Fuzzy	47
Gambar 3.14 Rancangan Form Input Aturan Fuzzy	48

	Halaman
Gambar 3.15 Rancangan Form Input User	48
Gambar 3.16 Rancangan Form Proses.....	49
Gambar 3.17 Rancangan Output.....	50
Gambar 4.1 Form Login	58
Gambar 4.2 Form Setup Fuzzy Nilai Batasan.....	59
Gambar 4.3 Form Setup Fuzzy Aturan.....	60
Gambar 4.4 Form Automasi	60
Gambar 4.5 Form ROP	61
Gambar 4.6 Hasil Perhitungan Fuzzy	62
Gambar 4.7 Grafik Perubahan Nilai	62
Gambar 4.8 Pesan Kesalahan User	63
Gambar 4.9 Popup Migrasi Data Pembelian.....	64
Gambar 4.10 Popup Migrasi Data Penjualan.....	65
Gambar 4.11 Popup Simpan Fuzzy Set	66
Gambar 4.12 Delete Detail Fuzzy Set.....	67
Gambar 4.13 Popup Simpan Aturan Fuzzy	68
Gambar 4.14 Popup Simpan User.....	69
Gambar 4.15 Maintenance User.....	70
Gambar 4.16 Hasil Perhitungan Periode Oktober 2005.....	71
Gambar 4.17 Hasil Perhitungan Periode November 2005.....	72
Gambar 4.18 Hasil Perhitungan Periode Desember 2005.....	72
Gambar 4.19 Hasil Perhitungan Periode Januari 2005	73

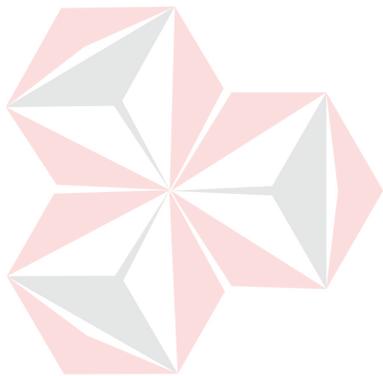
	Halaman
Gambar 4.20 Hasil Perhitungan Periode Februari 2005	73
Gambar 4.21 Hasil Perhitungan Periode Maret 2005	74



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Data Rata-Rata Delay Pembelian.....	79
Lampiran 2. Data Total Penjualan	86
Lampiran 3. Hasil Cetak Perhitungan.....	93
Lampiran 4. Coding	94
Lampiran 5. Biodata Penulis.....	133



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

PT. Dita Jaya Perkasa adalah perusahaan yang bergerak dibidang pemasaran obat-obatan. Dalam usaha untuk terus meningkatkan daya saing dan volume penjualan, diperlukan suatu manajemen pemesanan kembali yang baik. Hal ini dibutuhkan perusahaan agar tidak mengalami kehabisan persediaan, biaya penyimpanan yang tinggi, dan biaya-biaya yang dikarenakan obat rusak ataupun kadaluarsa, yang akhirnya dapat mengurangi keuntungan.

Dengan banyaknya produk yang dipasarkan, nilai flukstasi penjualan yang rata-rata tidak menentu, lama waktu order yang tidak pasti, banyaknya jumlah transaksi, dan permintaan pasar yang tidak pasti, penentuan titik pesan kembali menjadi permasalahan yang penting dalam usaha pemasaran. Berbagai faktor ketidakpastian seperti yang telah disebutkan diatas membuat penentuan titik pesan kembali menjadi sulit dilakukan.

Dari permasalahan yang ada, maka diperlukan pembuatan sebuah aplikasi yang mampu menentukan titik pesan kembali. Dalam pembuatan aplikasi tersebut, digunakan suatu metode pendukung. Metode yang digunakan adalah *fuzzy logic*. Metode *fuzzy logic* merupakan suatu metode pemecahan masalah yang mampu menghitung hal-hal yang bersifat tidak pasti menjadi suatu nilai yang pasti, berdasarkan parameter yang telah ditetapkan. Metode *fuzzy logic* juga mempunyai toleransi terhadap data-data yang kurang tepat.

Berdasarkan permasalahan tersebut, diharapkan dengan penentuan titik pesan kembali yang cepat, tepat dan akurat dapat meningkatkan pendapatan perusahaan dan menekan biaya-biaya yang ada.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, didapatkan suatu perumusan permasalahan sebagai berikut: “Bagaimana membuat suatu sistem aplikasi untuk menentukan titik pesan kembali dengan menggunakan fuzzy logic pada PT. Dita Jaya Perkasa?”

1.3 Batasan Masalah

Batasan permasalahan dari sistem yang dibahas adalah sebagai berikut:

1. Aplikasi ini hanya digunakan pada PT. Dita Jaya Perkasa.
2. Metode yang digunakan adalah *fuzzy logic*.
3. Proses analisa data menggunakan model *heuristik*.
4. Aplikasi yang dibuat tidak menyangkut proses transaksional.
5. Aplikasi dibuat dengan menggunakan Microsoft Visual Basic 6.0
6. Database menggunakan Microsoft SQL Server 7

1.4 Tujuan

Tujuan dari dibuatnya sistem ini adalah:

1. Membuat aplikasi yang dapat digunakan untuk menentukan titik pesan kembali pada PT. Dita Jaya Perkasa.
2. Memberikan informasi perkiraan jumlah order item.

1.5 Sistematika Penulisan

Didalam penulisan Tugas Akhir ini secara sistematika diatur dan disusun dalam lima bab, yaitu :

BAB I : PENDAHULUAN

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, serta sistematika penulisan.

BAB II : LANDASAN TEORI

Berisi teori-teori inventory, EOQ, serta metode Fuzzy Logic, khususnya dalam pemahaman penentuan titik pesan kembali yang digunakan sebagai landasan dalam desain dan implementasi aplikasi.

BAB III : PERANCANGAN SISTEM

Berisi tentang desain diagram alir (*system flow*), *block diagram*, DFD, ERD, dan struktur database yang digunakan.

BAB IV : IMPLEMENTASI DAN EVALUASI SISTEM

Berisi tentang cara untuk mengimplementasikan aplikasi dan pengujian terhadap aplikasi yang dibuat. Didalamnya juga berisi informasi tentang kebutuhan aplikasi baik berupa perangkat keras maupun perangkat lunak yang diperlukan.

BAB V : PENUTUP

Berisi kesimpulan dari Tugas Akhir, serta saran sehubungan dengan adanya kemungkinan pengembangan aplikasi pada masa yang akan datang.



BAB II

LANDASAN TEORI

Untuk melakukan pembuatan aplikasi penentuan titik pesan kembali diperlukan pemahaman terhadap teori dan konsep yang mendasarinya, antara lain persediaan, teknik peramalan, *economic order quantity* (EOQ), dan khususnya dalam pembahasan fuzzy logic dalam menentukan titik pesan kembali.

2.1 Persediaan

Istilah (*terminologi*) persediaan dapat digunakan dalam beberapa perbedaan seperti:

1. Persediaan bahan baku di tangan
2. Daftar persediaan secara fisik
3. Jumlah item di tangan
4. Nilai persediaan barang

Berbagai macam organisasi mempunyai kebutuhan persediaan yang berbeda.

Oleh karena itu tipe persediaan dapat dibedakan menjadi:

1. Persediaan alat-alat kantor

Persediaan yang diperlukan dalam menjalankan fungsi organisasi dan tidak menjadi bagian dari produk akhir.

2. Persediaan bahan baku

Item yang dibeli dari supplier untuk digunakan sebagai input dalam proses produksi. Bahan baku ini akan ditransformasi menjadi barang akhir.

3. Persediaan barang jadi

Persediaan produk akhir yang siap untuk dijual, didistribusikan atau disimpan.

4. Persediaan barang dalam proses

Bagian dari produk akhir tetapi masih dalam proses pengerjaan, karena masih menunggu item yang lain untuk diproses.

Tujuan manajemen persediaan adalah untuk menyediakan jumlah material yang tepat, lead time yang tepat, dan biaya rendah. Jenis biaya persediaan adalah sebagai berikut:

1. Biaya pembelian

Harga per unit apabila item dibeli dari pihak luar, atau biaya produksi per unit apabila diproduksi dalam perusahaan.

2. Biaya pemesanan

Biaya yang berasal dari pembelian pesanan dari supplier atau biaya persiapan (*setup cost*) apabila item diproduksi didalam perusahaan.

3. Biaya simpan

Biaya yang dikeluarkan atas investasi dalam persediaan dan pemeliharaan maupun investasi sarana fisik untuk menyimpan persediaan.

4. Biaya kekurangan persediaan

Konsekuensi ekonomis atas kekurangan dari luar maupun dari dalam perusahaan.

2.2 Distribusi Persediaan

Dalam keputusan ini perlu diketahui kapan di pesan dan berapa banyak yang dipesan. Semakin banyak barang yang dipesan, semakin jarang jarak pemesanan yang dilakukan. Sistem manajemen distribusi inventory dapat

diklasifikasikan sebagai sistem tarik atau sistem dorong. Dalam sistem tarik akan dibahas tentang sistem titik pesan kembali, dan *periodic review system*.

1. Sistem Tarik

Setiap pusat distribusi pada tingkat rendah menghitung kebutuhannya dan kemudian memesan dari pusat distribusi pada tingkat lebih tinggi. Metode yang sering digunakan adalah :

a. Reorder Point System (ROP)

ROP adalah titik waktu dimana pemesanan dilakukan kembali setelah persediaan mencapai jumlah tertentu, sehingga tidak terjadi kekurangan komponen.

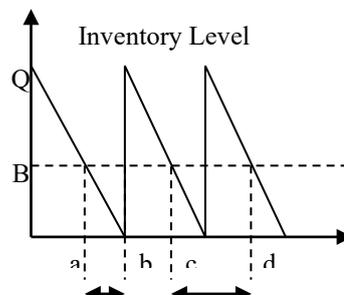
$$\text{ROP} = \text{DLT} + \text{SS}$$

Dimana : DLT = Permintaan selama waktu tunggu

SS = Safety Stock

ROP diterapkan di dalam EOQ (*Economic Order Quantity*), dimana setelah jumlah pemesanan tertentu yang optimal ditentukan, kemudian dihitung kapan ROP dilaksanakan, beserta frekuensi dan interval dari pemesanan, serta biaya yang dikeluarkan untuk pembelian dan penyimpanan.

Pada EOQ, transaksi permintaan diasumsikan diketahui terjadi setiap saat, serta jumlah sekali pesan selalu sama.



Gambar 2.1 Model EOQ

Q = Jumlah pemesanan

B = Titik Reorder Point (titik dimana dilakukan pemesanan ke supplier)

L = a-b = lead time = waktu antara pemesanan sampai barang tiba

T = c-d = Interval pemesanan

Jika Q terlalu banyak → biaya simpan tinggi, namun jika terlalu sedikit → biaya pesan tinggi. Untuk menghitung Q yang optimum dipergunakan perhitungan dengan metode EOQ, dimana EOQ ini akan didapatkan biaya inventory yang minimum.

$$EOQ = Q^* = \sqrt{\frac{2CR}{H}}$$

TC = biaya pembelian + biaya pemesanan + biaya simpan

$$TC = PR + \frac{CR}{Q} + \frac{HQ}{2}$$

Keterangan:

R = total permintaan per tahun

P = harga beli per unit

C = biaya pemesanan / pesan

H = P*F = biaya simpan / unit / tahun

Q = ukuran order

F = fraksi biaya penyimpanan / tahun

$$M = \frac{R}{Q^*} = \sqrt{\frac{HR}{2C}}$$

$$T = \frac{Q^*}{R} = \sqrt{\frac{2C}{HR}}$$

$$B = \frac{RL}{12}$$

Jika L dalam bulan

Minimum ongkos per tahun = $TC(Q^*) = PR + HQ^*$

Keterangan:

M = frekuensi pemesanan

T = interval pemesanan

B = reorder point

b. Periodic Review System

Diterapkan apabila ada kondisi – kondisi berikut:

1. Inventory dalam posisi independent demand
2. Item – item mudah busuk

$$Q = M - I$$

$$M = D(R + LT) + SS$$

Dimana:

Q = Kuantitas pesanan

I = Quantity on hand (kuantitas yang ditunjukkan dalam catatan inventory yang secara fisik ada dalam stock) + Quantity on order (total pesanan pengisian kembali yang belum diselesaikan)

M = Tingkat inventory maksimum

D = Demand

R = Lama Periodic Review (Waktu antara evaluasi status inventory yang berurutan untuk menentukan apakah perlu memesan kembali item inventory atau tidak perlu)

LT = Lead Time

SS = Safety Stock

2. Sistem Dorong

Dalam sistem ini kebutuhan pertama kali diperhitungkan melalui peramalan dari produk akhir, kemudian permintaan untuk level selanjutnya dapat diperkirakan.

2.3 Teknik Peramalan

Situasi peramalan sangat beragam dalam horison waktu peramalan, faktor yang menentukan hasil sebenarnya, tipe pola data dan berbagai aspek lainnya. Untuk menghadapi penggunaan yang luas seperti itu, berbagai teknik telah dikembangkan. Teknik tersebut dibagi dalam dua kategori utama, yaitu :

- a. Metode kuantitatif
- b. Metode kualitatif.

Metode kuantitatif dapat dibagi dalam deret berkala (*time series*) dan metode kausal. Sedangkan metode kualitatif dapat dibagi menjadi metode eksploratif dan normatif. Menurut Makridakis, Whellwright dan McGee (1995:8)

peramalan kuantitatif dapat diterapkan bila terdapat tiga kondisi berikut :

- a. Tersedia informasi tentang masa lalu
- b. Informasi tersebut dapat dikuantitatifkan dalam bentuk data numerik
- c. Dapat diasumsikan bahwa beberapa aspek pola masa lalu akan terus berlanjut di masa mendatang.

Suatu dimensi tambahan untuk mengklasifikasi metode peramalan kuantitatif adalah dengan memperhatikan model yang mendasarinya. Terdapat dua jenis model peramalan yang utama, yaitu model deret berkala dan model regresi (kausal). Pada jenis pertama, pendugaan masa depan dilakukan berdasarkan nilai

masa lalu dari suatu variabel dan/atau kesalahan masa lalu. Tujuan metode peramalan seperti itu adalah menemukan pola dalam deret data *historis* dan mengekspolasi data tersebut ke masa depan.

Model kausal dipihak lain mengasumsikan bahwa faktor yang diramalkan menunjukkan suatu hubungan sebab-akibat dengan suatu atau lebih variabel bebas.

Langkah penting dalam memilih suatu metode deret berkala yang tepat adalah dengan mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Menurut Makridakis, Whellwright dan McGee (1995:10) pola dapat dibedakan menjadi empat jenis siklis (*cyclical*) dan

trend yaitu :

a. Pola horisontal (H)

Pola horisontal terjadi bilamana nilai data berfluktuasi di sekitar nilai rata-rata konstan. Suatu produk yang penjualannya tidak meningkat atau menurun selama waktu tertentu termasuk jenis ini.

b. Pola musiman (S)

Pola musiman terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tertentu, bulan atau harian). Penjualan dari produk seperti minuman ringan, es krim dan bahan bakar pemanas ruangan, semuanya menunjukkan jenis pola ini.

c. Pola siklis (C)

Pola siklis terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis mobil, baja dan

peralatan utama lainnya. Penjualan produk seperti mobil, baja, dan peralatan utama lainnya menunjukkan jenis pola ini.

d. Pola trend (T)

Pola trend terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Penjualan banyak perusahaan, produk bruto nasional, dan berbagai indikator bisnis atau ekonomi lainnya mengikuti pola trend selama perubahannya sepanjang waktu.

Kesalahan peramalan didapat dengan menghitung selisih antara nilai data aktual dengan peramalan. Kesalahan peramalan tersebut dapat diukur dengan menggunakan beberapa alternatif, antara lain:

1. Mean Squared Errors (MSE)

Kesalahan rata-rata kuadrat atau MSE diperoleh dengan cara setiap kesalahan atau residual dikuadratkan, kemudian dijumlahkan dan dibagi dengan jumlah observasi.

$$MSE = \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{n}$$

2. Mean Absolute Percentage Error (MAPE)

Persentase kesalahan absolut rata-rata atau MAPE memberikan petunjuk seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya.

$$MAPE = \frac{\sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t}}{n}$$

2.4 Economic Order Quantity

Untuk kasus deterministik, total biaya persediaan per *planning horizon* dapat diformulasikan sebagai berikut:

$$TC(Q) = \text{Purchase cost} + \text{order cost} + \text{holding cost}$$

$$TC(Q) = P \cdot D + (C \cdot D) / Q + (h \cdot Q) / 2$$

Dimana :

Q = Lot size atau jumlah pesanan (unit)

D = Total demand per planning horizon

C = Biaya order per order (atau biaya setup kalau produksi sendiri)

h = Biaya simpan per unit per planning horizon

Dengan menggunakan *derivatif total cost* terhadap Q, maka didapatkan:

$$TC(Q) = P \cdot D + (C \cdot D) / Q + (h \cdot Q) / 2$$

$$\partial TC / \partial Q = -(C \cdot D) / Q^2 + h / 2$$

Syarat optimal titik kritis $\partial TC / \partial Q = 0$, maka didapatkan:

$$Q = \sqrt{\frac{2 \cdot C \cdot D}{h}}$$

Sebagai contoh, diketahui lead time 1 minggu, holding cost \$2 / unit / minggu, setup cost \$200.

$$D = (35 + 30 + 40 + 0 + 10 + 40 + 30 + 0 + 30 + 55) / 10$$

$$D = 27$$

$$Q = \sqrt{\frac{2 \cdot 200 \cdot 27}{2}} = 74$$

2.5 Fuzzy Logic

Fuzzy Logic (FL) adalah suatu teknologi pemecahan masalah, yang diinisialisasi oleh Lotfi Zadeh pada tahun 1965. FL menyajikan suatu cara yang sederhana dan bermanfaat untuk menggambarkan ketidakjelasan, kerancuan, atau informasi yang samar-samar. Pada dasarnya FL adalah suatu ekstensi dari logika boolean. Dalam FL, nilai kebenaran antara 1 dan 0. Dalam beberapa hal, FL menyerupai pengambilan keputusan yang digunakan manusia untuk pendekatan informasi dan menghasilkan suatu keputusan. Ada dua konsep utama pada Fuzzy Logic:

1. Linguistic variable, suatu variabel dimana nilainya adalah kata-kata atau bahasa alamiah.
2. Fuzzy IF-THEN rule.

2.5.1 Himpunan Fuzzy

Suatu himpunan *crisp* A dari semesta (U) didefinisikan oleh suatu fungsi karakteristik χ_A , dimana bernilai 1 untuk elemen didalam A dan 0 untuk elemen diluar A .

$$\chi_A(x): U \rightarrow \{0,1\}$$

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \notin A \\ 1 & \text{if } x \in A \end{cases}$$

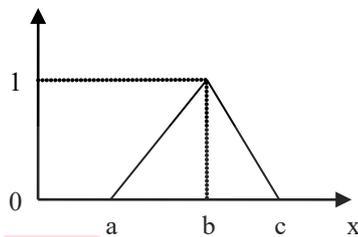
Suatu himpunan A dari U didefinisikan oleh suatu fungsi keanggotaan yang berhubungan dengan setiap elemen x dari semesta nilai keanggotaan $\mu_A(x)$. Nilai ini menjelaskan derajat keanggotaan dari x ke A .

$$\mu_A(x) : U \rightarrow [0,1]$$

2.5.2 Fungsi Keanggotaan Fuzzy

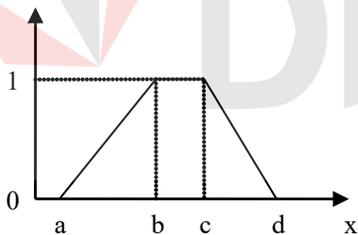
Fungsi keanggotaan menjelaskan derajat keanggotaan dari setiap elemen terhadap himpunan fuzzy. Beberapa fungsi keanggotaan yang populer adalah segitiga, trapesium, kurva, Gaussian, bell, fungsi S dan fungsi π .

a. Fungsi keanggotaan segitiga



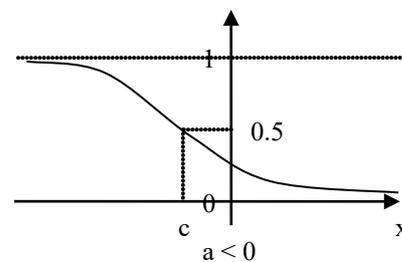
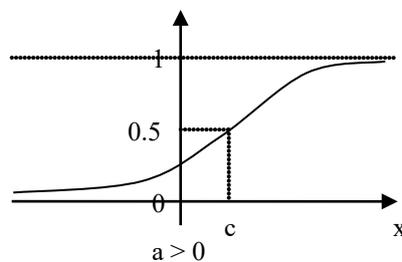
$$\text{triangle}(x;a,b,c) = \max \left\{ \min \left\{ \frac{x-a}{b-a}, \frac{c-x}{c-b} \right\}, 0 \right\}$$

b. Fungsi keanggotaan trapesium



$$\text{trapezoid}(x;a,b,c,d) = \max \left\{ \min \left\{ \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right\}, 0 \right\}$$

c. Fungsi keanggotaan kurva



$$\text{sigmoid}(x;a,c) = \frac{1}{1 + e^{-a(x-c)}}$$

2.5.3 Operasi Himpunan Fuzzy

Operasi dasar yang berhubungan dengan himpunan fuzzy sebagai berikut:

- a. $(A = B) \Rightarrow \mu_A(x) = \mu_B(x), x \in U$
- b. $(A = \bar{B}) \Rightarrow \mu_A(x) = \mu_{\bar{B}}(x), x \in U$
- c. $(A \subseteq B) \Rightarrow \mu_A(x) \leq \mu_B(x), x \in U$
- d. $(A \cup B) \Rightarrow \mu_{A \cup B}(x) = \vee(\mu_A(x), \mu_B(x)), x \in U$
- e. $(A \cap B) \Rightarrow \mu_{A \cap B}(x) = \wedge(\mu_A(x), \mu_B(x)), x \in U$
- f. $\mu_{Com(A)}(x) = (\mu_A(x))^2$

$$\mu_{Dil(A)}(x) = (\mu_A(x))^{1/2}$$

- g. $\mu_{not(A)}(x) = 1 - \mu_A$

$$\mu_{very(A)}(x) = (\mu_A)^2$$

$$\mu_{not\ very(A)}(x) = 1 - (\mu_A)^2$$

$$\mu_{more\ or\ less(A)}(x) = (\mu_A)^{0.5}$$

2.5.4 Variabel Linguistik

Variabel linguistik adalah variabel yang bernilai kata/kalimat, bukan angka. Sebagai alasan menggunakan kata/kalimat daripada angka karena peranan linguistik kurang spesifik dibandingkan angka, namun informasi yang disampaikan lebih informatif. Variabel linguistik ini merupakan konsep penting dalam logika samar dan memegang peranan penting dalam beberapa aplikasi.

Jika “jumlah” adalah variabel linguistik, maka nilai linguistik untuk variabel kecepatan adalah, misalnya “sedikit”, “sedang”, “banyak”. Hal ini sesuai dengan kebiasaan manusia sehari-hari dalam menilai sesuatu, misalnya : “Jumlah barang itu **banyak**”, tanpa memberikan nilai berapa jumlahnya.

Konsep tentang variabel linguistik ini diperkenalkan oleh Lofti Zadeh. Dalam variable linguistik ini menurut Zadeh dikarakteristikan dengan :

$$(X, T(x), U, G, M)$$

dimana:

X = nama variabel (variabel linguistik)

T(x) atau **T** = semesta pembicaraan untuk x atau disebut juga nilai linguistik dari x

U = jangkauan dari setiap nilai samar untuk x yang dihubungkan dengan variabel dasar U

G = Aturan sintaksis untuk memberikan nama (x) pada setiap nilai X

M = aturan semantik yang menghubungkan setiap X dengan artinya.

Sebagai contoh, jika:

X = “jumlah”, dengan $U[0, 100]$ dan $T(\text{jumlah}) = \{\text{sedikit, sedang, banyak}\}$

Maka M untuk setiap X, M(x) adalah M(sedikit), M(sedang), M(banyak).

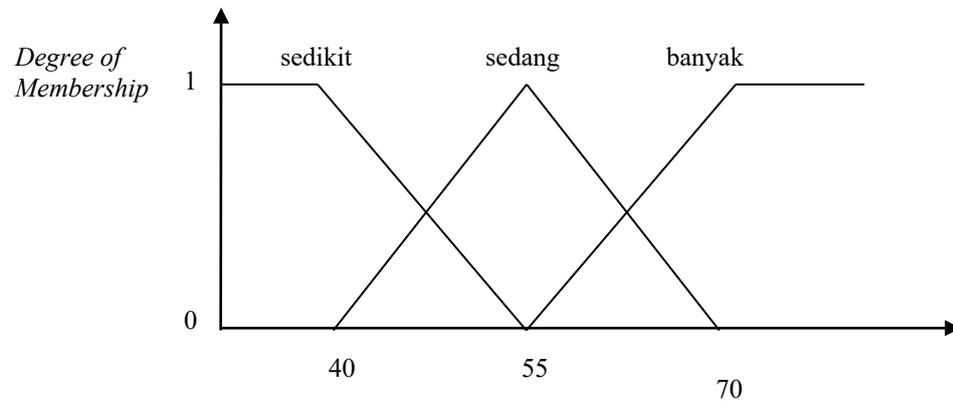
Dimana:

M(lambat) = himpunan samarnya “jumlah dibawah 40” dengan fungsi keanggotaan μ sedikit.

M(sedang) = himpunan samarnya “jumlah mendekati 55” dengan fungsi keanggotaan μ sedang.

$M(\text{cepat})$ = himpunan samarnya “jumlah diatas 70” dengan fungsi keanggotaan μ banyak.

Gambar grafik fungsi keanggotaannya sebagai berikut:



Gambar 2.2 Grafik Fungsi Keanggotaan Jumlah

2.5.5 Fuzzy Rule-Base

Aturan ini disebut sebagai aturan fuzzy IF-THEN. Definisi dari aturan fuzzy dijelaskan dibawah ini:

$$\text{IF } [X_1=A_1] \text{ AND } [X_2=A_2], \dots, \text{ AND } [X_m=A_m] \text{ THEN} \\ [Y_1=B_1] \text{ AND } [Y_2=B_2], \dots, \text{ AND } [Y_n=B_n]$$

dimana $X = (X_1, X_2, \dots, X_m)$ dan $Y = (Y_1, Y_2, \dots, Y_n)$ adalah variabel linguistik, dan (A_1, A_2, \dots, A_m) dan (B_1, B_2, \dots, B_n) adalah nilai pendukung dari nilai linguistik.

Contoh dari aturan fuzzy IF-THEN :

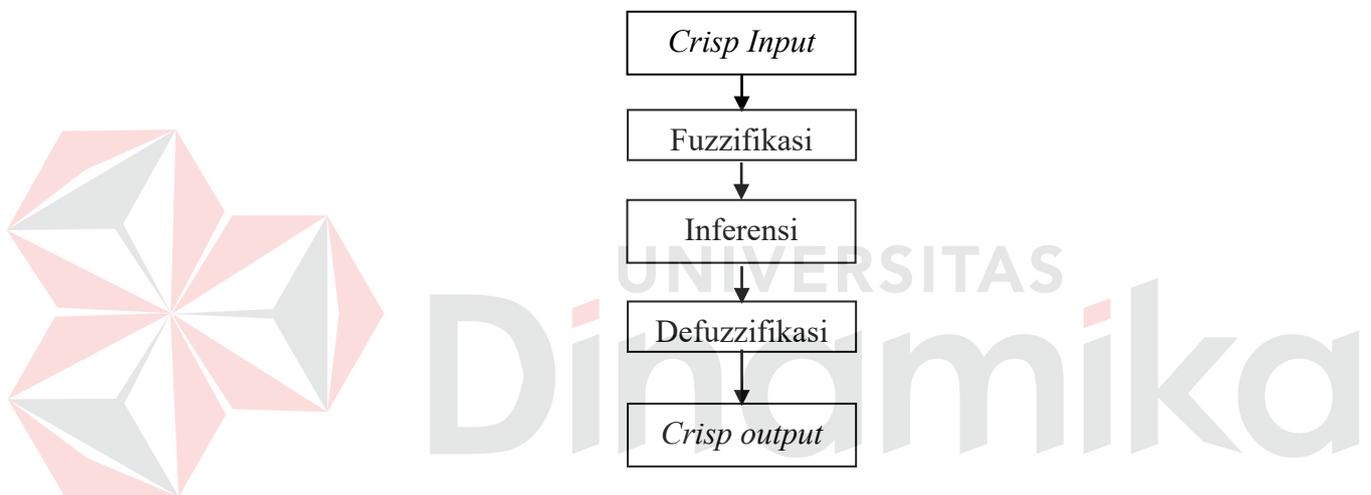
IF [tekanan="rendah"] AND [suhu="tinggi"] THEN [volume="besar"]

Kumpulan aturan fuzzy mendasari fuzzy rule based yang dapat digunakan sebagai pendukung keputusan untuk memasukkan query.

2.5.6 Perhitungan Fuzzy

Dalam teori *fuzzy* (*fuzzy*) untuk mendapatkan solusi yang eksak, maka ada tiga langkah umum yang dapat dilakukan:

1. Fuzzifikasi (*fuzzification*)
2. Penalaran/Evaluasi Kaidah (*rule evaluation*)
3. Defuzzifikasi (*Defuzzification*)



Gambar 2.3 Diagram Alir Proses Pengaturan Himpunan *Fuzzy*

a. Fuzzifikasi

Fuzzifikasi adalah fase pertama dari perhitungan *fuzzy* yaitu perubahan nilai tegas ke nilai *fuzzy*. Proses fuzzyfikasi ditulis sebagai berikut

$$x = \text{fuzzifier}(x_0)$$

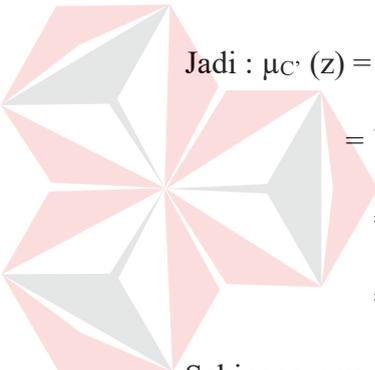
dengan x_0 adalah sebuah vektor nilai tegas dari satu variabel input, x adalah vektor himpunan *fuzzy* yang didefinisikan sebagai variabel, dan *fuzzifier* adalah sebuah operator *fuzzyfikasi* yang mengubah nilai tegas ke himpunan *fuzzy*.

Aturan samar pada premis 2 diatas dapat diubah menjadi bentuk yang lebih sederhana “ $A \times B \rightarrow C$ ”. Aturan samar tersebut dapat juga diubah ke dalam hubungan samar *ternary* R, dan fungsi keanggotaannya dapat ditulis sebagai berikut:

$$\begin{aligned}\mu_R(x,y,z) &= \mu_{(A \times B \rightarrow C)}(x,y,z) \\ &= \mu_{(A)}(x) \wedge \mu_{(B)}(y) \wedge \mu_{(C)}(z)\end{aligned}$$

dengan menghasilkan C' dengan ekspresi:

$$C' = (A' \times B') \circ (A \times B \rightarrow C)$$



$$\begin{aligned}\text{Jadi : } \mu_{C'}(z) &= \bigvee_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \bigvee_{x,y} \{[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)]\} \wedge \mu_C(z) \\ &= \{ \bigvee_x [\mu_{A'}(x) \wedge \mu_A(x)] \} \wedge \{ \bigvee_y [\mu_{B'}(y) \wedge \mu_B(y)] \} \wedge \mu_C(z) \\ &= w_1 \wedge w_2 \wedge \mu_C(z)\end{aligned}$$

$$\text{Sehingga : } w_1 = \bigvee_x [\mu_{A'}(x) \wedge \mu_A(x)]$$

= derajat kesesuaian antara A dan A'

$$w_2 = \bigvee_y [\mu_{B'}(y) \wedge \mu_B(y)]$$

= derajat kesesuaian antara B dan B'

$w_1 \wedge w_2$ merupakan kekuatan penyulutan (*firing strength*) atau derajat penyelesaian (*degree of fulfillment*) dari aturan samar.

c. Defuzzifikasi

Defuzzifikasi adalah fase terakhir dari proses perhitungan samar yaitu perubahan nilai samar hasil fuzzyfikasi ke nilai tegas. Proses defuzzifikasi ditulis sebagai berikut:

$$Y_0 = \text{defuzzifier}(y)$$

dengan y sebagai aksi pengendalian samar, sedangkan Y_0 sebagai aksi pengendalian tegas, dan defuzzifier sebagai operator defuzzifikasi. Strategi defuzzifikasi yang paling sering digunakan adalah pusat daerah (*Center of Area*), rata-rata maksimum *Mean Max membership* (MOM), dan *Center of Largest Area* (COA), *Center of Maximum* (COM).

2.6 Konsep Basis Data

Database adalah kumpulan dari komputer data yang diintegrasikan, diorganisir dan disimpan dengan cara yang mudah untuk diakses. Untuk ini digunakan direct access storage. Integrasi secara logical dari record di beberapa file disebut konsep database. Dua tujuan utama konsep database:

- a. Meminimalkan redundancy.
- b. Diperoleh data independent.

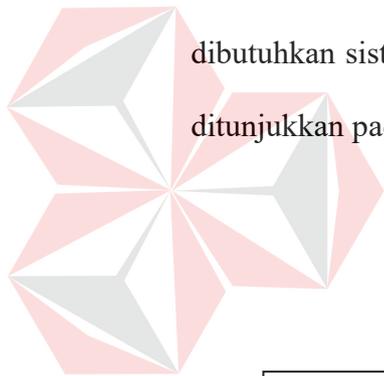
Kemampuan untuk mengubah struktur data tanpa mengubah program untuk memproses data. Hal ini dapat dicapai dengan menempatkan data specification pada suatu tabel yang secara fisik terpisah dari programnya.

2.7 Data Flow Diagram

Data *Flow* Diagram atau yang dapat disingkat dengan DFD adalah representasi dari sebuah sistem secara grafis yang digambarkan dengan sejumlah simbol tertentu untuk menunjukkan perpindahan data dalam proses-proses suatu sistem.

Dalam hal ini DFD menunjukkan perpindahan dan perubahan data dalam suatu sistem. Meskipun diberi nama Data *Flow* Diagram, namun penekanan pada DFD lebih pada prosesnya, bahkan DFD merupakan salah satu alat pemodelan proses dari sistem yang paling sering digunakan.

DFD dapat dan harus digambarkan secara sistematis . pertama, dibutuhkan sistem analisis untuk mengkonsep data *flow*, dari atas ke bawah seperti ditunjukkan pada gambar berikut:



Pembuatan DFD dengan pendekatan
dari atas ke bawah

1. Membuat sebuah daftar dari kegiatan bisnis dengan menggunakan beberapa variasi, yaitu :
 - *Entity* luar (*external entity*)
 - *Data flow*
 - Proses
 - *Data store*
2. Membuat sebuah context diagram dimana ditunjukkan *external entity* dan *data flow* ke dan dari sistem.
3. Menggambar diagram level 0, level selanjutnya.
4. Membuat sebuah *child* diagram untuk tiap-tiap proses pada level 0
5. Pengecekan *error*
6. Membangun sebuah DFD fisik dari DFD logika
7. Melakukan pemisahan data.

Gambar 2.3 Konsep Data Flow

Untuk memulai sebuah DFD dari suatu sistem biasanya dituangkan dalam sebuah daftar dengan empat kategori yaitu *entity* luar, arus data, proses, dan penyimpanan data. Daftar ini akan membantu menentukan batasan-batasan dari suatu sistem yang akan digambarkan. Pada dasarnya daftar itu berisi elemen-elemen data yang dikarang. Elemen-elemen tersebut terdiri dari:

a. Pembuatan *context* diagram

Context diagram adalah level yang tertinggi dalam sebuah DFD dan hanya berisi satu proses serta merupakan representasi dari sebuah sistem. Proses dimulai dengan penomoran ke-0 dan untuk seluruh *entity* luar akan ditunjukkan dalam *context* diagram yang sama seperti data awal yang dikirim dari *entity* luar. *Context* diagram tidak berisi penyimpanan data.

b. Pembuatan diagram level 0 serta level berikutnya

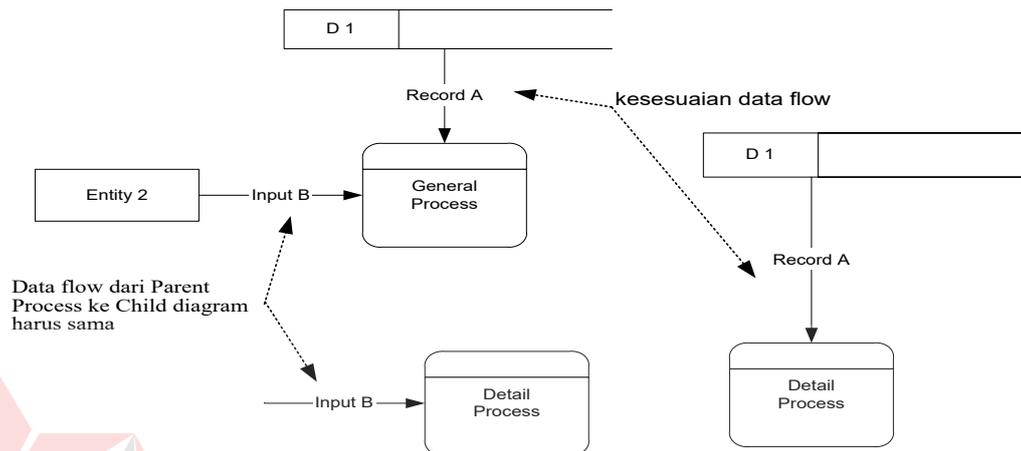
Diagram level 0 dihasilkan oleh *context* diagram dan berisi proses-proses. Pengisian proses-proses yang berlebihan pada level ini akan menghasilkan sebuah diagram yang salah, sehingga sulit untuk dimengerti.

Masing-masing proses diberikan penomoran dengan sebuah bentuk *integer*. Umumnya dimulai dari kiri atas dan penyelesaiannya di kanan bawah dalam sebuah bentuk diagram.

c. Pembuatan *child* diagram

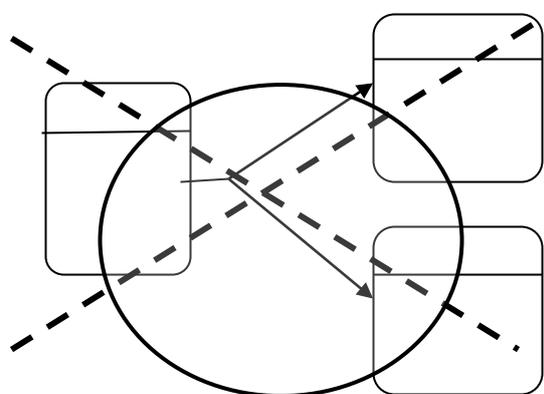
Child diagram diberikan nomor yang sama seperti proses di atasnya (parent proses) dalam diagram level 0. Contohnya, proses 3 harus diturunkan ke diagram 3, proses pada *child* diagram menggunakan penomoran unik untuk masing-masing proses dengan mengikuti penomoran proses di atasnya. Contohnya, dalam diagram 3 proses-proses diberikan nomor 3.1, 3.2, 3.3 dan

seterusnya. Konversi ini diikuti oleh analisis sistem untuk menelusuri seri-seri dari proses-proses yang dikeluarkan oleh beberapa level, jika pada proses diagram level 0 digambarkan sebagai 1, 2, , dan 3 maka *child* diagram-diagramnya adalah 1, 2, dan 3 pada level yang sama. ilustrasi level detil dengan sebuah *child* DFD dapat ditunjukkan pada gambar:

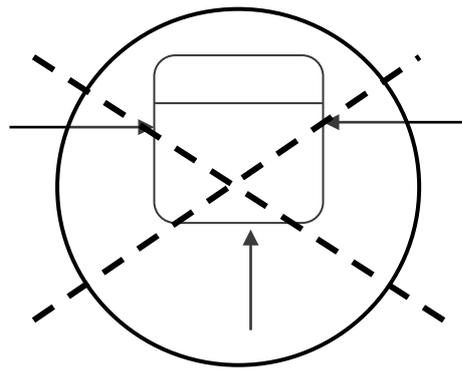


Gambar 2.4 Contoh Ilustrasi Detil Child Diagram

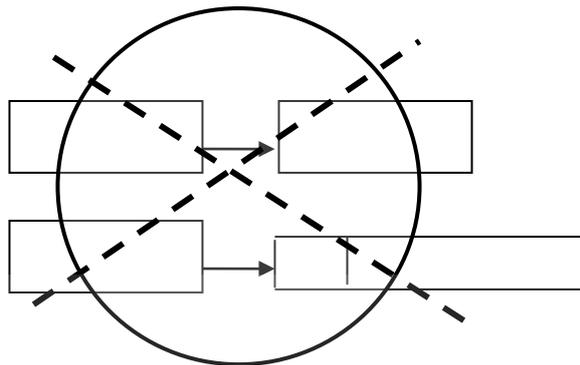
d. Pengecekan kesalahan-kesalahan pada diagram digunakan untuk melihat kesalahan-kesalahan yang terdapat pada sebuah DFD. Beberapa kesalahan-kesalahan yang umum terjadi ketika penggambaran / pembuatan DFD, ditunjukkan pada gambar berikut, adalah:



Sebuah data *flow* tidak diperbolehkan mempunyai percabangan / memisahkan diri (*flow*) ke dalam dua atau lebih data *flow* yang berbeda



Sebuah proses harus mempunyai minimal satu *input* data *flow* dan satu *output* data *flow*



Semua data *flow* salah satunya harus berasal atau berakhir pada sebuah proses

Gambar 2.5 Kesalahan Penulisan Proses Dalam DFD

1. Tidak menginputkan sebuah arus data atau arah panah langsung. Sebagai contoh adalah penggambaran proses yang menunjukkan sebuah data *flow* seperti *input* atau seperti *output*. Tiap-tiap proses pengubahan data harus menerima *input* dan *output*. Tipe kesalahan ini terjadi ketika sistem analis tidak memasukkan sebuah data *flow* atau meletakkan sebuah arah panah ditempat yang salah.
2. Hubungan penyimpanan data dan *entity* luar secara langsung satu sama lain. Data *store* dan *entity* tidak mungkin dikoneksikan satu sama lain ; data *store* dan *entity* luar harus dikoneksikan melalui sebuah proses.
3. Kesalahan penamaan (label) pada proses-proses atau data *flow*. Pengecekan DFD untuk memastikan bahwa tiap-tiap obyek atau data *flow* telah diberikan

label. Sebuah proses haruslah di indikasikan seperti nama dari sistem atau menggunakan format kata kerja. Tiap data *flow* haruslah dideskripsikan dengan sebuah kata benda.

4. Memasukkan lebih dari sembilan proses dalam sebuah DFD. Memiliki banyak proses akan mengakibatkan kekacauan pada diagram sehingga dapat menyebabkan kebingungan dalam pembacaan sebuah proses dan akan menghalangi tingkat komunikasi. Jika lebih dari sembilan proses dalam sebuah sistem, maka beberapa grup dalam proses dilakukan bersama-sama ke dalam sebuah sub sistem dan meletakkannya dalam sebuah *child* diagram.
5. Menghilangkan suatu arus data. Pengujian dari suatu diagram yang menunjukkan garis / arah (*flow*), dimana untuk setiap proses data *flow* hanya mempunyai *input* data, *output* kecuali dalam kasus dari detil (*child*). Setiap *child* data dari DFD, arah arus data seringkali digambarkan untuk mengidentifikasi bahwa diagram tersebut kehilangan data *flow*.
6. Membuat ketidaksesuaian komposisi dalam *child* diagram , dimana tiap *child* diagram harus mempunyai *input* dan *output* arus data yang sama seperti proses dilevel atasnya (*parent process*). Pengecualian untuk *rule* ini adalah kurangnya *output*, seperti kesalahan garis yang ada didalam *child* diagram.

Hal-hal yang perlu diperhatikan dalam dalam penyusunan DFD:

- a. Dalam DFD tidak dibedakan antara data dan informasi, semua dianggap data.
- b. Nama proses dalam *context diagram* harus sama dengan nama sistem.
- c. Setiap DFD harus muat dalam satu halaman.
- d. Setiap simbol harus diberi nama yang unik, namun konstan.
- e. Hindari garis yang berpotongan jika mungkin.

- f. DFD pasti selalu mempunyai *output*.
- g. Setiap proses dalam DFD harus mempunyai *input* dan *output*.
- h. Suatu kesatuan luar hanya boleh mempunyai *input* dan *output*, tapi tidak boleh kedua-duanya.
- i. Setiap simpanan data hanya boleh menerima *input* dari proses dan juga memberikan *output* ke proses saja.

2.8 Entity Relationship Diagram

Entity Relationship Diagram (ERD) digunakan untuk

menginterpretasikan, menentukan dan mendokumentasikan kebutuhan untuk sistem pemrosesan database. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari aplikasi. Adapun elemen dari ERD ini adalah:

1. *Entitas*, adalah sesuatu yang dapat diidentifikasi di dalam lingkup pemakai, sesuatu yang penting bagi pemakai dari sistem yang akan dikembangkan.
2. *Atribut*, entitas memiliki atribut yang berfungsi untuk menjelaskan karakteristik dari entitas.
3. *Identifikasi*, data-data entitas memiliki nama yang berfungsi untuk mengidentifikasikan mereka. Sebuah identifikasi dapat bersifat unik atau tidak unik. (Kendall & Kendall, 2002:46)

Hubungan atau relasi berfungsi untuk menunjukkan hubungan satu entitas dengan entitas yang lain. Hubungan ini boleh memiliki atribut. Banyaknya entitas dalam suatu relasi menunjukkan tingkat dari relasi yang bersangkutan,

namun yang banyak digunakan dalam aplikasi-aplikasi adalah model yang menggunakan relasi tingkat dua atau yang disebut dengan hubungan biner. Hubungan biner ini memiliki tipe yaitu hubungan biner satu ke satu, hubungan biner satu ke banyak, hubungan biner banyak ke banyak dan hubungan biner banyak ke satu. (Kendall & Kendall, 2002:52).

Ada beberapa macam hubungan yang terjadi antara dua *entity*:

a. *One to one relationship*

Hubungan antara satu *entity* mempunyai hubungan antara satu *entity* yang lain pada *entity* yang berbeda. Hubungan ini akan dibedakan menjadi dua macam yaitu *obligatory* dan *non obligatory*. *Obligatory* bila semua anggota dari suatu *entity* harus berpartisipasi atau mempunyai hubungan dengan *entity* yang lain, dan *non obligatory* bila semua anggota *entity* tidak harus mempunyai hubungan dengan anggota *entity* yang lain.

b. *One to many relationship*

Hubungan dimana suatu *entity* mempunyai hubungan dengan beberapa anggota *entity* yang berbeda.

c. *Many to many relationship*

Hubungan dimana beberapa *entity* mempunyai hubungan dengan beberapa *entity* yang lain pada *entity* yang berbeda. Jadi kedua belah pihak bisa memiliki hubungan dengan lebih dari satu anggota *entity* yang lain. Hubungan ini dapat dibedakan menjadi hubungan *obligatory* dan *non obligatory*.

Dengan menggunakan metode hubungan *entity* dalam perencanaan database, dapat dilakukan beberapa langkah untuk membedakan hubungan antara *file* yang ada:

- a. Menentukan *entity* yang ada dalam database dengan masing-masing *attribute*.
- b. Menentukan semua hubungan yang dapat terjadi antar *entity* yang ada.
- c. Menentukan hubungan yang ada, termasuk *one to one*, *one to many* atau *many to many* juga menentukan adalah apakah hubungan tersebut *obligatory* atau *non obligatory*.
- d. Dari jenis hubungan yang telah ditentukan tersebut, maka ditentukan jumlah relasi yang diperlukan.

Setelah tahap ini selesai dilaksanakan, maka telah tersedia suatu database yang telah didesain dengan baik dan siap digunakan dalam kondisi yang seefisien mungkin, dan dapat memberikan informasi secara tepat dan benar.

Jadi secara garis besar, langkah-langkah dalam pembuatan ERD adalah:

- a. Mengidentifikasi *entity*.
- b. Mengidentifikasi *relationship* antar tiap *entity*.
- c. Menyiapkan rancangan ERD.
- d. Memindahkan elemen-elemen data ke dalam *entity*.
- e. Melakukan analisis data, dengan proses yang disebut dengan normalisasi.
- f. Menyiapkan ERD yang telah dimodifikasi.
- g. *Me-review* ERD bersama *user* dan melakukan perbaikan jika diperlukan.

BAB III

PERANCANGAN SISTEM

Analisis terhadap suatu sistem yang sedang berjalan merupakan suatu langkah penting dalam pemahaman permasalahan yang ada sebelum dilakukannya pengambilan keputusan atau tindakan dalam menyelesaikan permasalahan tersebut.

Setelah dilakukan analisis terhadap sistem yang sedang berjalan, langkah berikutnya adalah perancangan sistem baru. Dimana dalam perancangan sistem ini dapat memberikan gambaran tentang sistem yang dibuat.

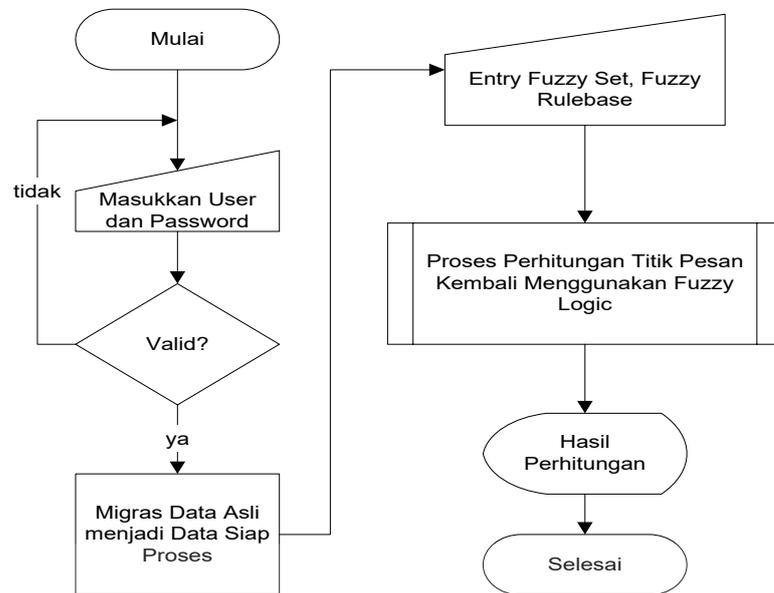
Dalam pembuatan sistem ini, dibuat suatu perancangan dengan menggunakan tahap dalam merancang sistem seperti pada umumnya. Tahapan tersebut adalah:

- a. *System flow*.
- b. Diagram berjenjang.
- c. *Data Flow Diagram* (DFD).
- d. *Entity Relationship Diagram* (ERD).
- e. Struktur database.

3.1. Analisa Sistem

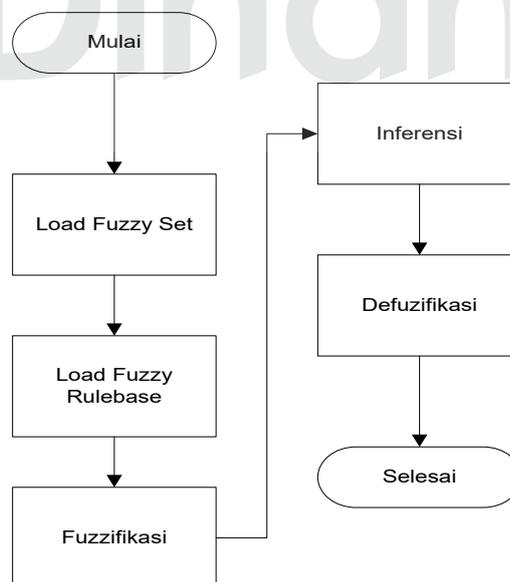
Setelah menganalisa permasalahan yang timbul maka tahap selanjutnya adalah mendesain sistem, dimana pada tahap desain sistem yang baru meliputi penerapan perhitungan menggunakan *fuzzy logic* untuk mengoptimalkan pembelian barang pada PT Dita Jaya Perkasa, DFD, ERD dan struktur database yang akan digunakan. Diagram alir sistem pendukung keputusan untuk

mengoptimalkan pembelian barang terkomputerisasi yang akan dibangun seperti tampak pada gambar 3.1.



Gambar 3.1 Diagram Alir Aplikasi Penentuan Titik Pesan Kembali

3.1.1. Alur Proses Perhitungan Menggunakan Fuzzy Logic



Gambar 3.2 Alur Proses Perhitungan Titik Pesan Kembali Menggunakan *Fuzzy*

Logic

Flowchart di atas menggambarkan bagaimana cara perhitungan menggunakan *fuzzy logic* untuk mengoptimalkan pembelian barang. Jika pembilang pada proses defuzzifikasi lebih besar dari 0, maka perhitungan telah menghasilkan pemecahan optimal. Unsur terpenting dari algoritma di atas adalah *fuzzy set* dan *fuzzy rulebase*. Berikut merupakan penjelasan *flowchart* dari proses perhitungan yang dilakukan untuk metode *fuzzy logic*:

1. Mulai

2. Load *Fuzzy Set*

Pada proses ini menentukan bahasa *fuzzy*, nilai minimum, nilai tengah, dan nilai maksimum setiap data

3. Tetapkan bahasa *fuzzy* untuk *fuzzy rule*

4. Menentukan derajat keanggotaan dan menetapkan nilai MF dan nilai Centre berdasarkan dari *fuzzy set*

5. Lakukan langkah 4 untuk setiap *fuzzy set* dan inputan

6. Lakukan pencocokan nilai MF tiap inputan dengan *fuzzy rulebase*, sehingga dapat ditetapkan nilai MF dan nilai Centre dari *fuzzy rulebase*

7. Hitung:

a. Pembilang baru = nilai MF * nilai Centre dari *fuzzy rulebase*

b. Penyebut baru = nilai MF dari *fuzzy rulebase*

8. Ulangi langkah 7 untuk setiap nilai dari *fuzzy rulebase*

9. Hitung:

Defuzzifikasi = Pembilang / Penyebut

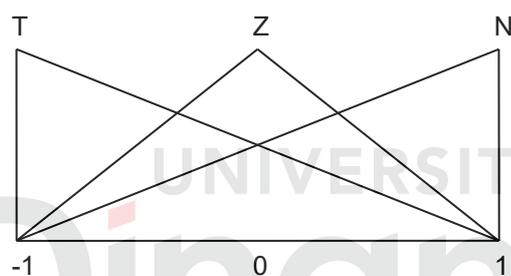
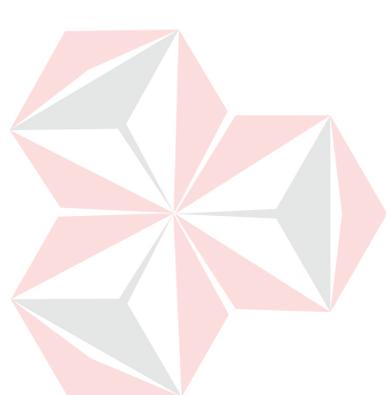
10. Selesai

3.1.2. Penentuan Rulebase

Untuk dapat menghasilkan *rulebase* yang optimal, perlu dilakukan analisa data. Dalam proses menganalisa data ini, data yang dianalisa adalah data 4 (empat) tahun pertama terhitung mulai September 2001 sampai Juli 2005. Bahasa fuzzy dibedakan menjadi 3 yaitu turun, zero, naik.

Tabel 3.1 Bahasa Fuzzy

Bahasa	Min	Center	Max
Turun (T)	-1	-1	1
Zero (Z)	-1	0	1
Naik (N)	1-	1	1



Gambar 3.3 Grafik Bahasa Fuzzy

Berikut adalah langkah-langkah dalam menganalisa data:

1. Query data penjualan dan delay pembelian

Data penjualan dan pembelian dikelompokkan berdasarkan item barang per periode. Untuk penjualan dicari total per periode per item yang ditulis sebagai RD. Sedangkan pembelian dicari selisih rata-rata waktu order dan waktu item datang per periode per item yang ditulis sebagai TJ.

2. Hitung nilai maksimum

Menentukan nilai maksimum dari total penjualan ataupun rata-rata delay pembelian.

3. Hitung batas atas dengan menambah nilai maksimum dengan 20% toleransi
4. Merubah data menjadi prosentase dengan cara membagi nilai data asli yang ke t dengan batas atas
5. Hitung selisih (t+1) terhadap (t), (t-1), (t-2) yang ditulis sebagai E1, E2, E3, E
6. Klasifikasikan terhadap bahasa yang ditulis sebagai S1, S2, S3, S.
Menyesuaikan nilai E1, E2, E3, E dengan bahasa fuzzy menjadi S1, S2, S3, S
7. Tiap produk dikumpulkan menjadi satu.
Semua nilai E1, E2, E3, E, S1, S2, S3, S dijadikan satu untuk mempermudah pemetaan data.
8. Memetakan menjadi rulebase dengan cara menghitung jumlah setiap kombinasi yang ada. Jumlah tiap kombinasi yang paling besar yang digunakan sebagai rulebase.

Berdasarkan langkah-langkah diatas, penjelasan perhitungan tampak seperti pada lampiran.

A. Analisa Rule Pembelian

Dari data rata-rata delay pembelian seperti pada lampiran didapatkan suatu kombinasi rulebase sebagai berikut:

Tabel 3.2 Kombinasi Rulebase Rata-rata Delay Pembelian

E1	E2	E3	T	Z	N	E
N	N	N				
N	N	Z	2			T
N	N	T	6	3		T
N	Z	N	2	1		T
N	Z	Z		1	1	Z
N	Z	T	11	6	1	T
N	T	N	13	6	7	T
N	T	Z	5	6	1	Z
N	T	T	3	2	1	T
Z	N	N	1	2		Z
Z	N	Z	6	1	1	T
Z	N	T	8	5	2	T

Tabel 3.2 Kombinasi Rulebase Rata-rata Delay Pembelian (Lanjutan)

E1	E2	E3	T	Z	N	E
Z	Z	N	6	1	1	T
Z	Z	Z	2	1		T
Z	Z	T		1	1	Z
Z	T	N	1	2	7	N
Z	T	Z	1		7	N
Z	T	T			3	N
T	N	N	1	2	5	N
T	N	Z		1	12	N
T	N	T	4	7	11	N
T	Z	N	4	6	6	Z
T	Z	Z		3	6	N
T	Z	T		1		Z
T	T	N		2	3	N
T	T	Z		2	3	N
T	T	T				

B. Analisa Rule Penjualan

Dari data total penjualan seperti pada lampiran didapatkan suatu kombinasi rulebase sebagai berikut:

Tabel 3.3 Kombinasi Rulebase Penjualan

E1	E2	E3	T	Z	N	E
N	N	N	10	2	7	T
N	N	Z	2			T
N	N	T	18		12	T
N	Z	N	2		1	T
N	Z	Z				
N	Z	T	2	2	1	Z
N	T	N	15	1	21	N
N	T	Z	2		2	Z
N	T	T	6		9	N
Z	N	N	1		1	Z
Z	N	Z			2	N
Z	N	T	1			T
Z	Z	N				
Z	Z	Z				
Z	Z	T				
Z	T	N	1		3	N
Z	T	Z				
Z	T	T	2		2	Z
T	N	N	12	2	16	N
T	N	Z			4	N
T	N	T	4	2	17	N

Tabel 3.3 Kombinasi Rulebase Penjualan (Lanjutan)

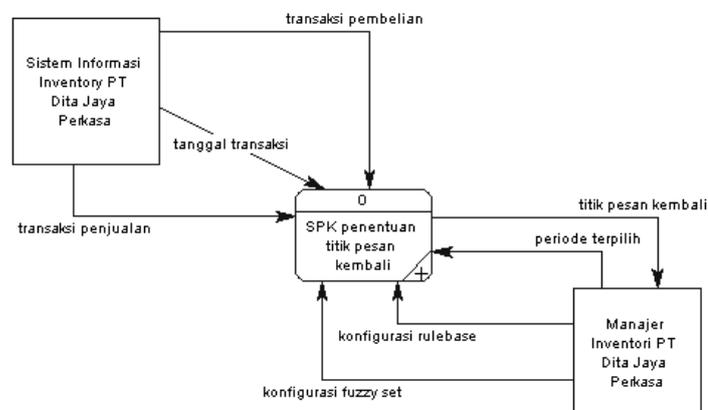
E1	E2	E3	T	Z	N	E
T	Z	N	1		1	Z
T	Z	Z				
T	Z	T			3	N
T	T	N	7	3	6	T
T	T	Z	1			T
T	T	T	7	1		T

3.2. Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu bagan yg memiliki arus data dalam suatu sistem yang ditunjukkan untuk membantu menganalisa suatu sistem ditinjau dari arus data yang ada dalam suatu sistem.

3.2.1 Context Diagram

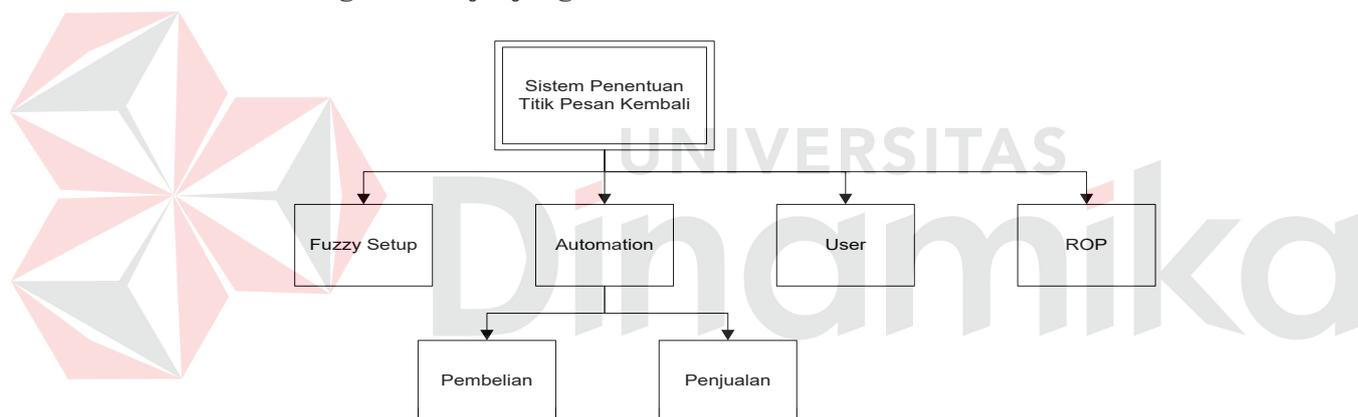
Sesuai dengan judul yang membahas tentang penentuan titik pesan kembali, maka pada sistem ini pokok pembahasan terdapat pada titik pemesanan kembali, pada *context diagram* dibawah ini penulis akan menjelaskan secara garis besar tentang aplikasi penentuan titik pesan kembali. Gambar 3.4 menunjukkan gambar *Context Diagram* untuk aplikasi penentuan titik pesan kembali di PT Dita Jaya Perkasa menggunakan *fuzzy logic*.

Gambar 3.4 *Context Diagram*

Sistem Informasi Inventory pada PT Dita Jaya Perkasa yang telah ada sebelumnya memberikan inputan berupa data total penjualan, rata-rata delay pembelian (dalam hitungan hari), yang nantinya data hasil dari migrasi tersebut digunakan untuk perhitungan titik pesan kembali.

Manajer inventori memasukkan konfigurasi aturan *fuzzy* serta batasan-batasan dari *fuzzy set*. Dan berdasarkan data hasil dari migrasi data asli, manajer inventori dapat mendapat informasi mengenai titik pesan kembali dari item barang yang dimaksud.

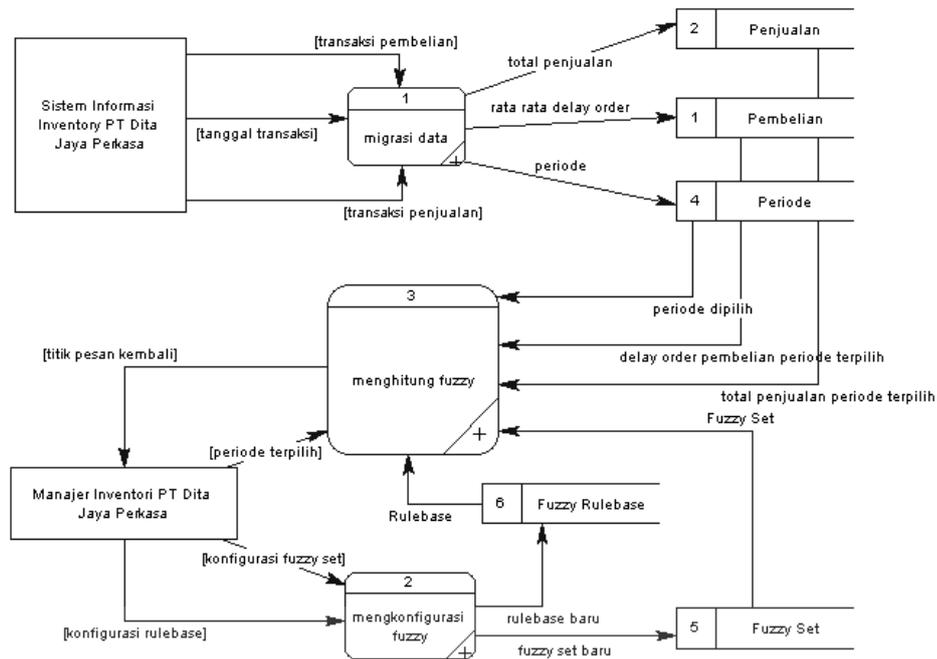
3.2.2 Diagram Berjenjang



Gambar 3.5 Diagram Berjenjang

Diagram berjenjang merupakan alat perancangan sistem yang dapat menampilkan seluruh proses yang terdapat pada suatu aplikasi tertentu dengan jelas dan terstruktur. Adapun secara garis besar, diagram berjenjang yang membangun aplikasi dapat digambarkan pada gambar 3.5.

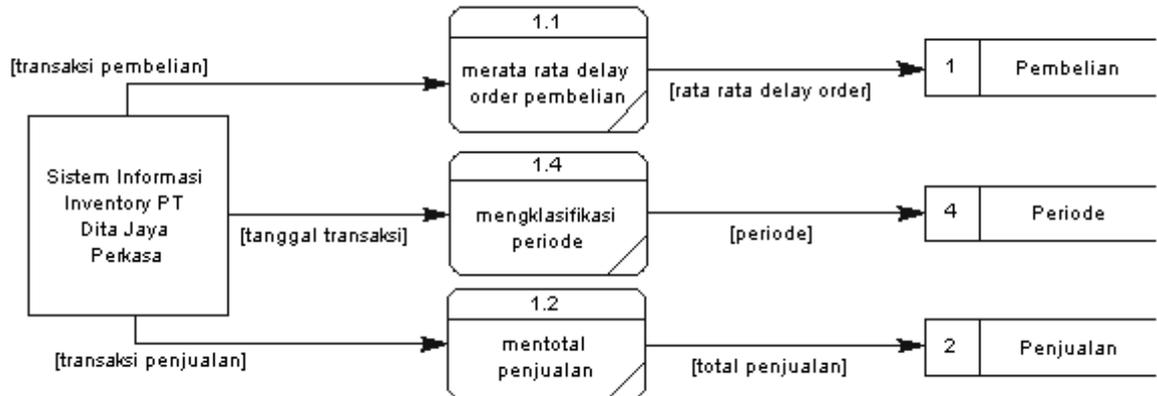
3.2.3 DFD Level 0 Aplikasi Penentuan Titik Pesan Kembali



Gambar 3.6 DFD Level 0 Aplikasi Penentuan Titik Pesan Kembali

Pada DFD Level 0 aplikasi penentuan titik pesan kembali dapat dijelaskan proses secara keseluruhan mulai dari proses automasi, proses konfigurasi *fuzzy*, sampai proses perhitungan *fuzzy*. Untuk lebih jelasnya dapat dilihat pada gambar-gambar DFD Level 1 yang merupakan *breakdown* dari DFD Level 0 terdiri dari proses automasi, proses konfigurasi *fuzzy*, dan proses perhitungan *fuzzy*.

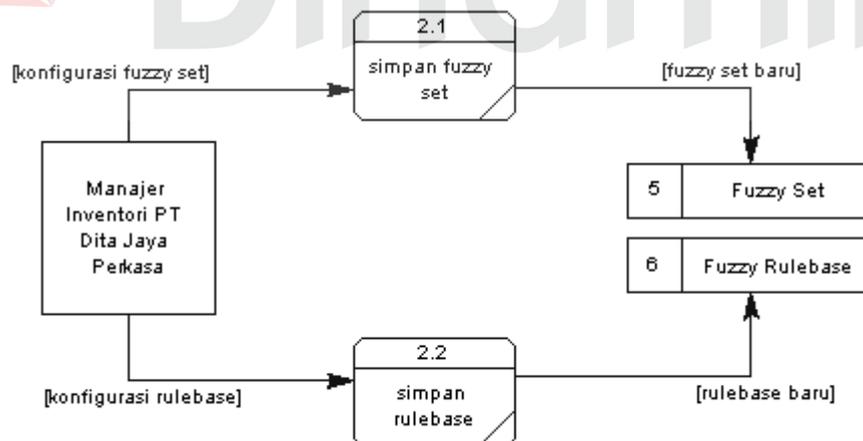
3.2.4 DFD Level 1 Proses Migrasi



Gambar 3.7 DFD Level 1 Proses Migrasi

Proses automasi bertujuan untuk migrasi dari data asli menjadi data olahan yang siap digunakan dalam proses penghitungan selanjutnya. Proses migrasi data asli dikelompokkan dalam periode bulanan.

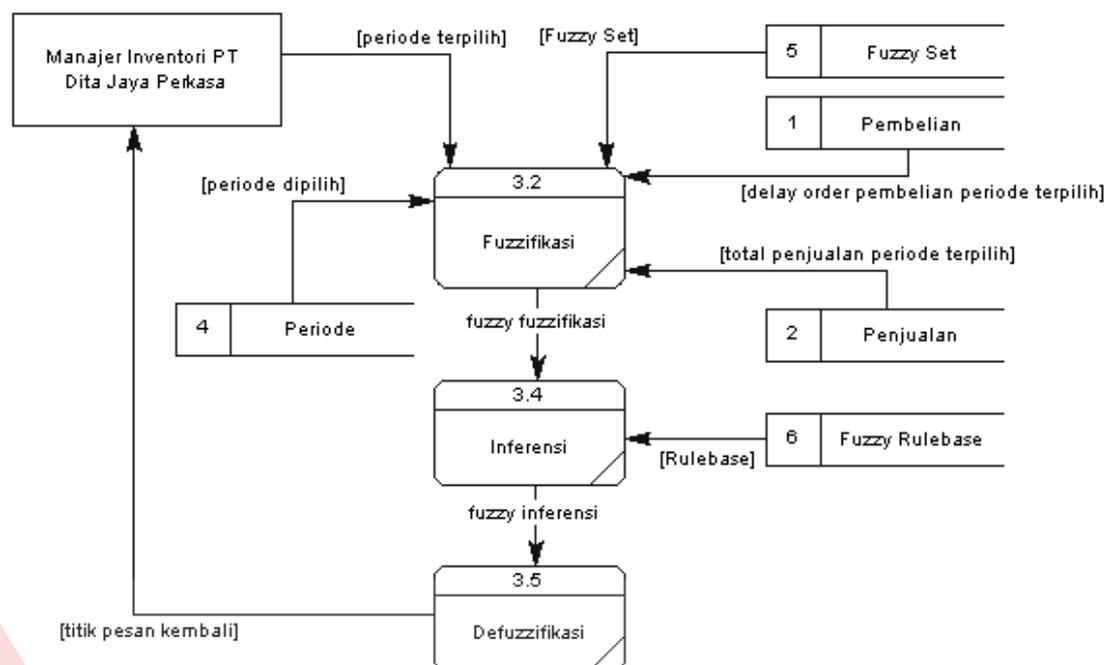
3.2.5 DFD Level 1 Proses Konfigurasi Fuzzy



Gambar 3.8 DFD Level 1 Proses Konfigurasi Fuzzy

Pada proses konfigurasi *fuzzy*, manajer inventori menetapkan batasan-batasan *fuzzy* dan aturan *fuzzy* yang nantinya digunakan dalam menentukan titik pesan kembali suatu barang.

3.2.6 DFD Level 1 Proses Perhitungan Fuzzy



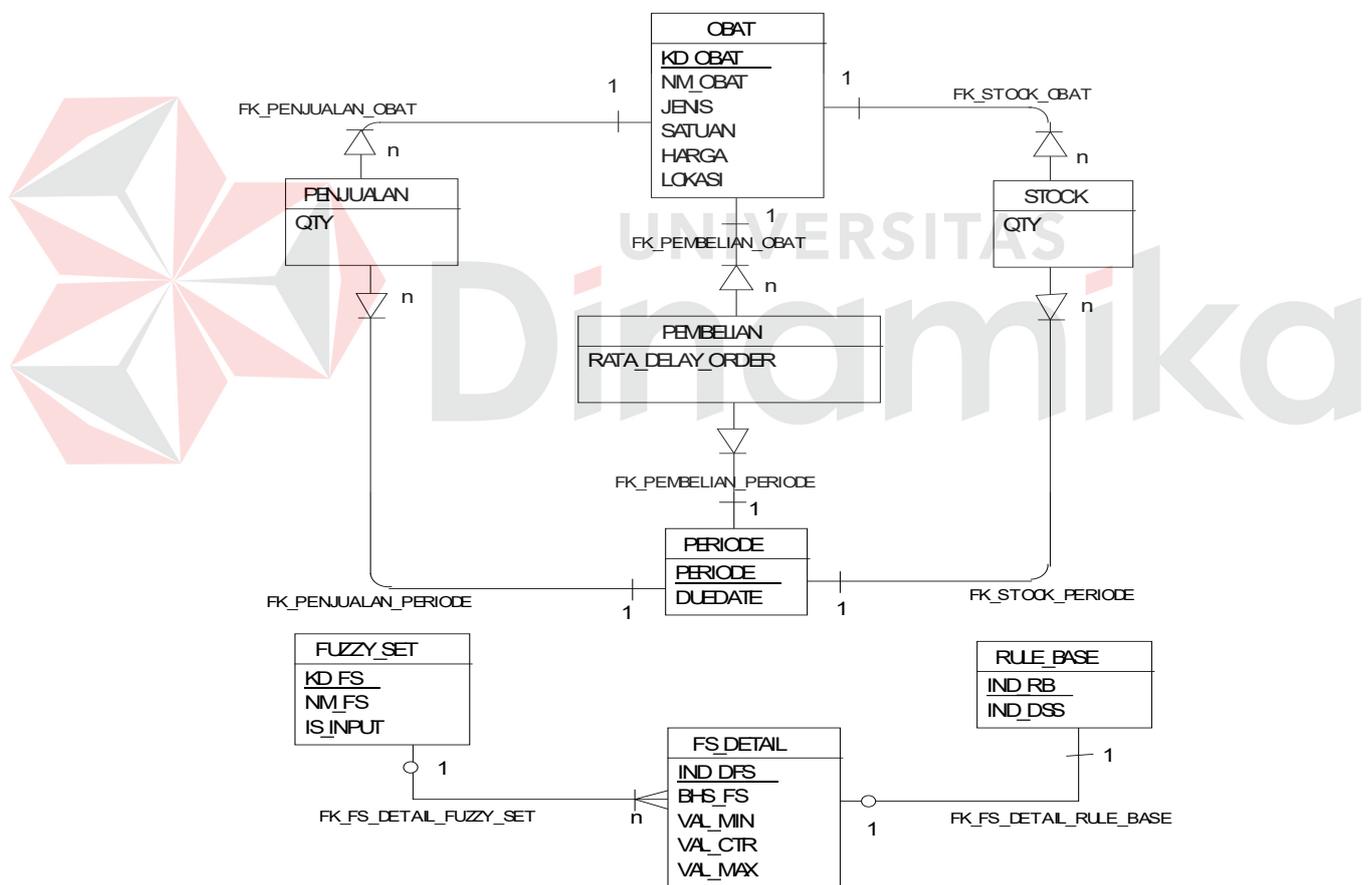
Gambar 3.9 DFD Level 1 Proses Perhitungan Fuzzy

Pada proses perhitungan *fuzzy*, pertama yang dilakukan adalah mengklasifikasikan *fuzzy set* dan *fuzzy rulebase* yang telah ditentukan oleh manajer inventori. Setelah itu dilakukan proses fuzzifikasi yang berguna untuk mengetahui derajat keanggotaan dari tiap inputan yaitu total penjualan, rata-rata delay order, dari *fuzzy set*. Setelah hasil diperoleh maka dilakukan proses inferensi yang bertujuan untuk mengetahui aturan-aturan yang sesuai dari hasil fuzzifikasi. Hasil dari proses fuzzifikasi tersebut masih berupa nilai *fuzzy*. Untuk merubah menjadi nilai *crisp*, dilakukan proses defuzzifikasi. Hasil proses defuzzifikasi ini adalah nilai dari titik pesan kembali yang digunakan sebagai pendukung keputusan oleh manajer inventori.

3.3. Entity Relational Diagram

Entity Relational Diagram (ERD) menggambarkan model data yang ada pada sistem, dimana terdapat entity dan relationship. Entitas merupakan objek yang ada dan terdefinisikan di dalam suatu organisasi. Bentuknya abstrak atau nyata, misalnya dapat berupa orang, objek atau waktu kejadian. Setiap entitas mempunyai mempunyai atribut atau karakteristik. Sedangkan relationship menjelaskan hubungan yang mewujudkan pemetaan antar entitas.

3.3.1. Conceptual Data Model

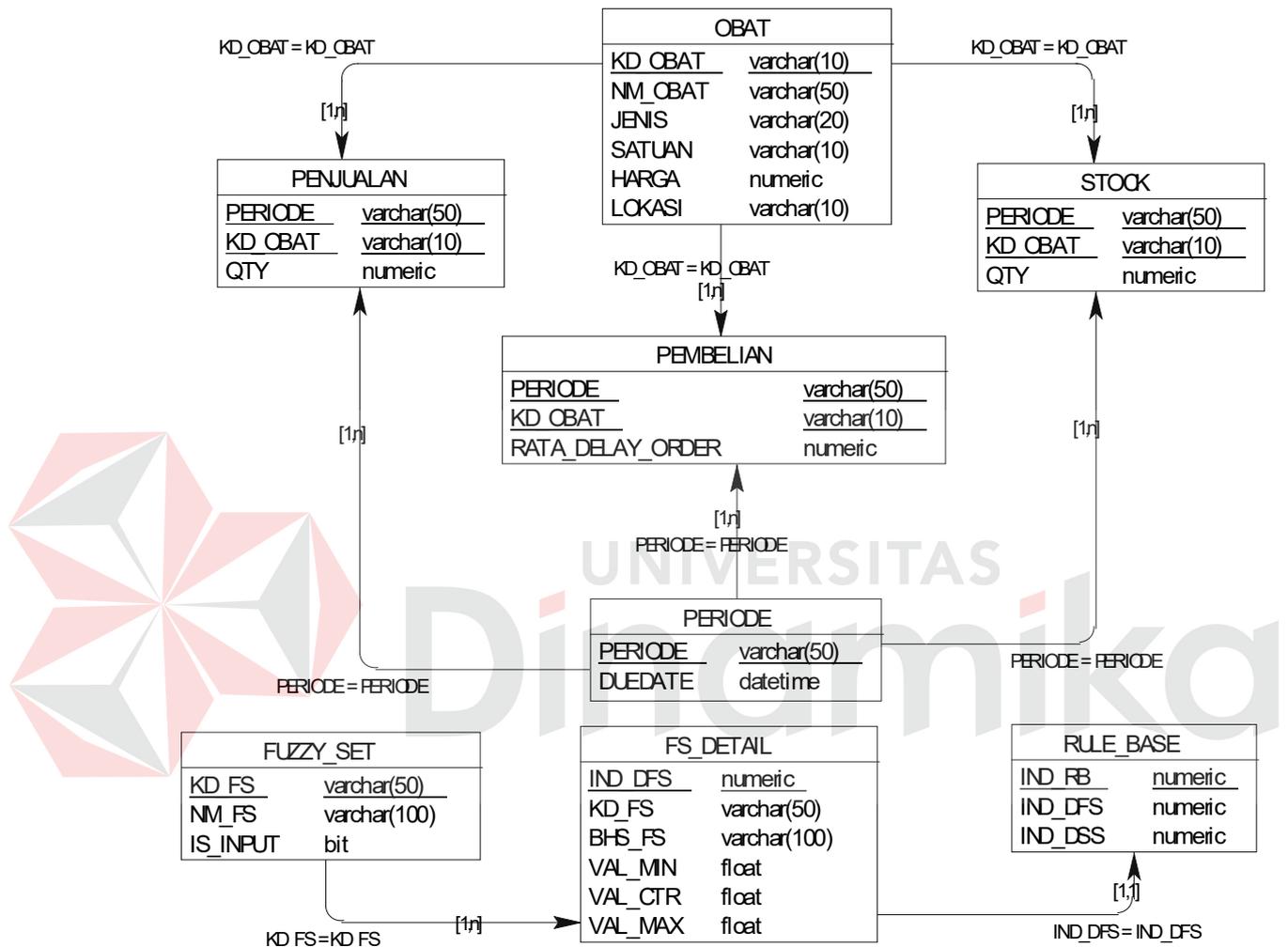


Gambar 3.10 *Conceptual Data Model*

Pada ERD *Conceptual Data Model* (CDM) dapat dijelaskan hubungan kardinalisasi yang terjadi antar tabel. Misal tabel fuzzy set mempunyai relasi *one*

to many ke tabel fuzzy detail sehingga satu fuzzy set bisa diisi beberapa detail fuzzy.

3.3.2. Physical Data Model



Gambar 3.11 Physical Data Model

Pada ERD *Physical Data Model* (PDM) dapat dijelaskan struktur database secara lengkap beserta nama *field* serta *primary key* dan *foreign key*.

3.4. Struktur Database

Struktur *database* merupakan uraian struktur fisik dari tabel-tabel yang terdapat pada database sistem yang berfungsi untuk menyimpan data-data yang

saling berhubungan. Adapun tabel-tabel pada struktur basis data yang dibentuk untuk membangun sistem ini dapat dijelaskan sebagai berikut:

1. Tabel Obat

Primary key : KD_OBAT

Foreign key : -

Fungsi : Menyimpan data obat

Tabel 3.4 Tabel Obat

No.	Nama	Key	Tipe	Keterangan
1.	<u>KD_OBAT</u>	PK	Varchar(10)	Kode obat
2.	<u>NM_OBAT</u>		Varchar(50)	Nama obat
3.	JENIS		Varchar(20)	Jenis obat
4.	SATUAN		Varchar(10)	Satuan obat
5.	HARGA		Money	Harga obat
6.	LOKASI		Varchar(10)	Lokasi obat

2. Tabel Penjualan

Primary key : PERIODE, KD_OBAT

Foreign key : PERIODE, KD_OBAT

Fungsi : Menyimpan migrasi data penjualan

Tabel 3.5 Tabel Penjualan

No.	Nama	Key	Tipe	Keterangan
1.	<u>PERIODE</u>	PK,FK	Varchar(50)	Periode
2.	<u>KD_OBAT</u>	PK,FK	Varchar(10)	Kode obat
3.	QTY		Numeric(18)	Jumlah penjualan

3. Tabel Pembelian

Primary key : PERIODE, KD_OBAT

Foreign key : PERIODE, KD_OBAT

Fungsi : Menyimpan migrasi data pembelian

Tabel 3.6 Tabel Pembelian

No.	Nama	Key	Tipe	Keterangan
1.	<u>PERIODE</u>	PK,FK	Varchar(50)	Periode
2.	<u>KD_OBAT</u>	PK,FK	Varchar(10)	Kode obat
3.	<u>RATA_DELAY_OR DER</u>		Numeric(18)	Rata-rata delay pembelian

4. Tabel Periode

Primary key : PERIOD

Foreign key : -

Fungsi : Menyimpan data periode

Tabel 3.7 Tabel Periode

No.	Nama	Key	Tipe	Keterangan
1.	<u>PERIOD</u>	PK	Varchar(50)	Periode
2.	<u>DUEDATE</u>		Datetime	Tanggal periode

5. Tabel Fuzzy_Set

Primary key : KD_FS

Foreign key : -

Fungsi : Menyimpan fuzzy set

Tabel 3.8 Tabel Fuzzy_Set

No.	Nama	Key	Tipe	Keterangan
1.	<u>KD_FS</u>	PK	Varchar(50)	Kode fuzzy
2.	<u>NM_FS</u>		Varchar(100)	Nama fuzzy
3.	<u>IS_INPUT</u>		Bit	Input / Output

6. Tabel Fs_Detail

Primary key : IND_DFS

Foreign key : KD_FS

Fungsi : Menyimpan detail dari fuzzy set

Tabel 3.9 Tabel Fs_Detail

No.	Nama	Key	Tipe	Keterangan
1.	<u>IND_DFS</u>	PK	Numeric(18)	Index detail
2.	<u>KD_FS</u>		Varchar(50)	Kode fuzzy
3.	<u>BHS_FS</u>		Varchar(100)	Bahasa fuzzy
4.	<u>VAL_MIN</u>		Float(53)	Nilai minimum
5.	<u>VAL_CTR</u>		Float(53)	Nilai tengah
6.	<u>VAL_MAX</u>		Float(53)	Nilai maksimum

7. Tabel Rule_Base

Primary key : IND_RB, IND_DFS, IND_DSS

Foreign key : IND_RB, IND_DFS, IND_DSS

Fungsi : Menyimpan aturan-aturan keputusan

Tabel 3.10 Tabel Rule_Base

No.	Nama	Key	Tipe	Keterangan
1.	<u>IND_RB</u>	PK,FK	Int	Index rulebase
2.	<u>IND_DFS</u>	PK,FK	Numeric(18)	Index detail fuzzy
3.	<u>IND_DSS</u>	PK,FK	Numeric(18)	Index keputusan

8. Tabel dbUser

Primary key : USER_NAME

Foreign key : -

Fungsi : Menyimpan data user

Tabel 3.11 Tabel dbUser

No.	Nama	Key	Tipe	Keterangan
1.	<u>USR_NAME</u>	PK	Varchar(50)	User name
2.	<u>USR_PWD</u>		Varchar(50)	Password
3.	<u>USR_LEVEL</u>		Varchar(50)	Level user
4.	<u>USR_EXP</u>		Bit	Validitas user

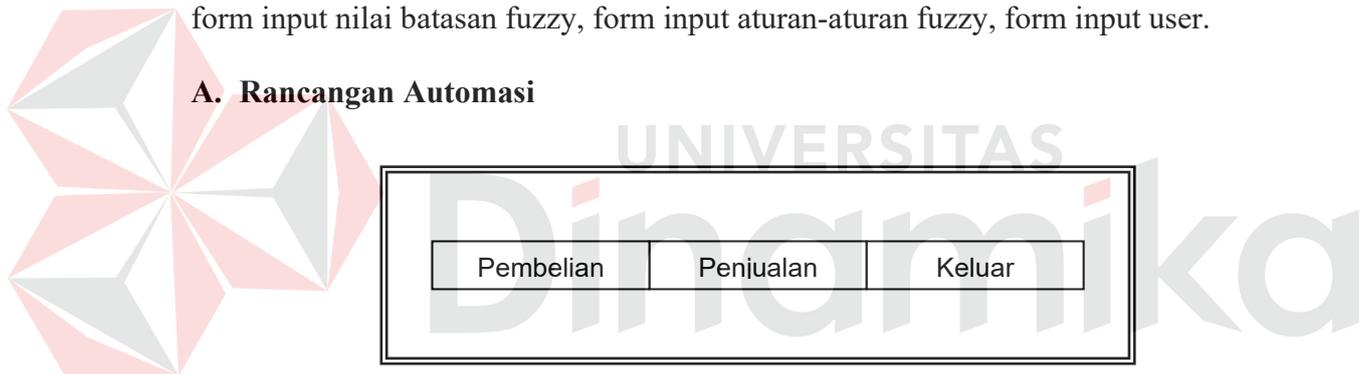
3.5. Rancangan Input Output

Untuk memberikan gambaran awal tentang sistem pendukung keputusan untuk menentukan titik pesan kembali pada PT Dita Jaya Perkasa menggunakan *fuzzy logic* serta menjaga konsistensi dari rancangan aplikasi, maka perlu dibuat suatu rancangan input dan output aplikasi sebagai berikut:

3.5.1. Rancangan Input

Rancangan masukan sistem dapat berupa form-form yang merupakan antar muka aplikasi dengan pengguna yaitu manajer inventori. Form masukan yang memerlukan interaksi langsung dengan pengguna adalah form automasi, form input nilai batasan fuzzy, form input aturan-aturan fuzzy, form input user.

A. Rancangan Automasi



Gambar 3.12 Rancangan Form Automasi

Form automasi digunakan untuk migrasi data asli menjadi data siap pakai. Pada rancangan form automasi terdapat beberapa tombol, yaitu :

1. Tombol Pembelian : untuk migrasi data pembelian
2. Tombol Penjualan : untuk migrasi data penjualan
3. Tombol Keluar : untuk keluar form automasi

B. Rancangan Input Nilai Batasan Fuzzy

Kelompok

Index	Bahasa Fuzzy	Min	Center	Max

Grafik

Simpan Keluar

Gambar 3.13 Rancangan Form Input Nilai Batasan Fuzzy

Form input nilai batasan *fuzzy* digunakan untuk menginputkan batasan-batasan *fuzzy* berdasarkan kelompok input yaitu pembelian, penjualan. Pada rancangan

form input nilai batasan terdapat beberapa tombol, yaitu:

1. Tombol Simpan : untuk menyimpan data batasan
2. Tombol Keluar : untuk keluar dari form input nilai batasan

Terdapat pula bagian untuk menampilkan grafik hasil inputan batasan-batasan *fuzzy*.

C. Rancangan Input Aturan Fuzzy

Index Rule	Index Pembelian	Index Penjualan	Index SPK

Gambar 3.14 Rancangan Form Input Aturan Fuzzy

Form input aturan *fuzzy* digunakan untuk menginputkan aturan-aturan fuzzy berdasarkan *fuzzy* set yang telah diinputkan pada form input batasan. Pada rancangan form input aturan *fuzzy* terdapat beberapa tombol, yaitu:

1. Tombol Simpan : untuk menyimpan data aturan fuzzy
2. Tombol Keluar : untuk keluar dari form input aturan

D. Rancangan Maintenance User

User Name	<input type="text"/>
Password	<input type="text"/>
<input type="button" value="Simpan"/> <input type="button" value="Keluar"/>	

Gambar 3.15 Rancangan Form Input User

Form input user digunakan untuk menginputkan user yang berhak menggunakan aplikasi. Pada rancangan form input user terdapat beberapa tombol, yaitu:

1. Tombo Simpan : untuk menyimpan data user
2. Tombo Keluar : untuk keluar dari form input user

E. Rancangan Form Proses

Kode Obat	Nama Obat	DB	TJ	ST

Gambar 3.16 Rancangan Form Proses

Form proses ini digunakan untuk memproses data-data yang akan dihitung titik pesan kembalinya. Pada rancangan form proses terdapat beberapa tombol, yaitu:

1. Tombol Proses : untuk melakukan proses penghitungan
2. Tombol Keluar : untuk keluar dari form proses

3.5.2. Rancangan Output

Kode Obat	Nama Obat	DB	TJ	ROP

Gambar 3.17 Rancangan Output

Form output ini digunakan untuk menampilkan nilai dari titik pesan kembali tiap barang yang telah diproses.

3.6. Desain Uji Coba

Uji coba validasi sistem bertujuan untuk memastikan bahwa sistem telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan sistem pada tahap ini akan dievaluasi sebelum diimplementasikan secara riil.

Proses pengujian menggunakan *Black Box Testing* dimana aplikasi akan diuji dengan berbagai percobaan untuk membuktikan bahwa aplikasi yang telah dibuat sudah sesuai dengan tujuan.

3.6.1 Desain Uji Coba Fungsional

A. Form Main Menu

Pada tabel 3.12 berikut merupakan desain test case dari main menu yang terdapat frame login yang berfungsi untuk memeriksa validasi kebenaran user dan password.

Tabel 3.12 Tabel Uji Main Menu

Test			
Case ID	Tujuan	Input	Output yang diharapkan
1	Deskripsi user dan password yang valid	User = "admin" Password = "admin" Klik tombol login	Frame login akan disable, menu akan muncul
2	Deskripsi user dan password yang tidak valid	User = "admin1" Password = "admin" Klik tombol login	Tampil pesan "Invalid User Account"

B. Form Automasi

Pada tabel 3.13 berikut merupakan desain test case dari automasi yang berfungsi untuk migrasi data.

Tabel 3.13 Tabel Uji Automasi

Test			
Case ID	Tujuan	Input	Output yang diharapkan
3	Migrasi data pembelian	Klik pada tombol "Pembelian"	Progress akan bergerak dan muncul popup "Migrasi Pembelian Sukses"
4	Migrasi data penjualan	Klik pada tombol "Penjualan"	Progress akan bergerak dan muncul popup "Migrasi Penjualan Sukses"

C. Form Batasan Fuzzy

Pada tabel 3.14 berikut merupakan desain test case dari batasan fuzzy yang berfungsi untuk memasukkan nilai-nilai batasan fuzzy.

Tabel 3.14 Tabel Uji Batasan Fuzzy

Test			
Case ID	Tujuan	Input	Output yang diharapkan
5	Memilih kelompok fuzzy detail	Klik pada combobox kelompok, kelompok = "Pembelian"	Nilai batasan fuzzy kelompok pembelian akan muncul pada list dan grafik batasan akan muncul
6	Mengisi index detail fuzzy	Pada list, index akan digenerate oleh sistem =1,2,3,dst	Akan tampil index detail fuzzy = 1,2,3,dst
7	Memasukkan nilai batasan fuzzy	Klik pada list, Bahasa = "Turun", Min = "-1", Center = "-1", Max = "1"	Bahasa = "Turun", Min = "-1", Center = "-1", Max = "1"
8	Membuat grafik nilai batasan fuzzy	Klik pada tombol "Simpan"	Grafik akan ditampilkan
9	Menyimpan nilai batasan fuzzy	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"
10	Merubah nilai batasan fuzzy	Klik pada list, Bahasa = "Zero", Min = "-1", Center = "0", Max = "1"	Bahasa = "Zero", Min = "-1", Center = "0", Max = "1"
11	Menghapus nilai batasan fuzzy	Klik kanan pada list dan pilih "Delete"	Nilai batasan fuzzy akan terhapus

D. Form Aturan Fuzzy

Pada tabel 3.15 berikut merupakan desain test case dari aturan fuzzy yang berfungsi untuk memasukkan aturan-aturan fuzzy.

Tabel 3.15 Tabel Uji Aturan Fuzzy

Test			
Case ID	Tujuan	Input	Output yang diharapkan
12	Memasukkan index rule	Pada list, index rule akan digenerate oleh sistem =1,2,3,dst	Akan tampil index rule = 1,2,3,dst
13	Memasukkan index pembelian	Pada list, index pembelian akan digenerate oleh sistem sesuai dengan index pembelian	Akan tampil index pembelian sesuai dengan index pembelian
14	Memasukkan index penjualan	Pada list, index penjualan akan digenerate oleh sistem sesuai dengan index penjualan	Akan tampil index penjualan sesuai dengan index penjualan
15	Memasukkan index spk	Klik pada list, pilih index spk, index spk = 10	Index spk = 10
16	Menyimpan aturan fuzzy	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"

E. Form Maintenance User

Tabel 3.16 Tabel Uji Maintenance User

Test			
Case ID	Tujuan	Input	Output yang diharapkan
17	Merubah user name	Klik textbox user name = "Saya"	User name = "Saya"
18	Merubah password	Klik textbox password = "Saya"	Password = "Saya"
19	Menyimpan perubahan account	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"

F. Form Proses

Pada tabel 3.17 berikut merupakan desain test case dari proses yang berfungsi untuk menghitung titik pesan kembali

Tabel 3.17 Tabel Uji Proses

Test			
Case ID	Tujuan	Input	Output yang diharapkan
20	Memilih periode	Klik pada combobox periode, periode = "0306"	Periode = "0306"
21	Menampilkan data transaksi	Klik pada combobox periode, periode = "0306"	Data transaksi periode "0306" akan muncul pada list
22	Menghitung titik pesan kembali	Klik pada tombol "Proses"	Akan menampilkan hasil perhitungan titik pesan kembali sesuai dengan data

3.6.2 Desain Uji Coba Validasi

Berikut desain uji coba validasi yang akan dilakukan pada proses menghitung titik pesan kembali. Pada sub bab ini akan dilakukan uji coba penghitungan 2 (dua) item barang yaitu “Amoxan 500 mg” dan “Hemaviton” pada periode September 2005 sampai dengan Februari 2006.

Tabel 3.18 Perhitungan Pembelian Amoxan 500mg

Periode	RD(t)	%	E1	E2	E3	E
0905	1	0.28	0.00	-0.28	0.00	0.56
1005	3	0.83	0.56	0.00	-0.28	-0.56
1105	1	0.28	-0.56	0.56	0.00	0.00
1205	1	0.28	0.00	-0.56	0.56	0.28
0106	2	0.56	0.28	0.00	-0.56	-0.28
0206	1	0.28	-0.28	0.28	0.00	0.56
Max	4					

Tabel 3.19 Perhitungan Penjualan Amoxan 500mg

Periode	TJ(t)	%	E1	E2	E3	E
0905	8	0.08	-0.16	-0.44	0.26	0.19
1005	26	0.27	0.19	-0.16	-0.44	-0.07
1105	19	0.20	-0.07	0.19	-0.16	-0.02
1205	17	0.18	-0.02	-0.07	0.19	-0.05
0106	12	0.13	-0.05	-0.02	-0.07	0.00
0206	12	0.13	0.00	-0.05	-0.02	-0.04
Max	95					

Tabel 3.20 Hasil Perhitungan Titik Pesan Kembali Amoxan 500mg

	1005	1105	1205	0106	0206	0306
RD(t+1)	1	3	3	3	2	2
TJ(t+1)	14	42	21	20	14	12
TPK	1	6	3	3	2	2

Tabel 3.21 Perhitungan Pembelian Hemaviton

Periode	RD(t)	%	E1	E2	E3	E
0905	0	0.00	0.00	-0.28	0.28	0.28
1005	1	0.28	0.28	0.00	-0.28	0.28
1105	2	0.56	0.28	0.28	0.00	-0.56
1205	0	0.00	-0.56	0.28	0.28	0.28
0106	1	0.28	0.28	-0.56	0.28	0.00

Tabel 3.21 Perhitungan Pembelian Hemaviton (Lanjutan)

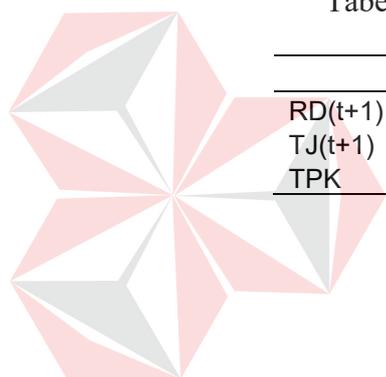
Periode	RD(t)	%	E1	E2	E3	E
0206	1	0.28	0.00	0.28	-0.56	0.00
Max	4					

Tabel 3.22 Perhitungan Penjualan Hemaviton

Periode	TJ(t)	%	E1	E2	E3	E
0905	17	0.18	-0.07	-0.16	0.09	0.13
1005	79	0.83	0.13	-0.07	-0.16	-0.13
1105	21	0.22	-0.13	0.13	-0.07	0.05
1205	45	0.47	0.05	-0.13	0.13	0.00
0106	44	0.46	0.00	0.05	-0.13	-0.04
0206	27	0.28	-0.04	0.00	0.05	0.03
Max	464					

Tabel 3.23 Hasil Perhitungan Titik Pesan Kembali Hemaviton

	1005	1105	1205	0106	0206	0306
RD(t+1)	0	1	1	2	2	1
TJ(t+1)	24	110	38	63	44	27
TPK	0	5	2	6	4	2



UNIVERSITAS
Dinamika

BAB IV

IMPLEMENTASI DAN EVALUASI SISTEM

4.1 Kebutuhan Sistem

Untuk menjalankan sistem yang telah dirancang ini, diperlukan beberapa perangkat baik itu perangkat keras (*hardware*) maupun perangkat lunak (*software*) agar sistem dapat berjalan seperti yang diharapkan.

A. Perangkat Keras

- a. CPU dengan kecepatan 633 Mhz atau lebih tinggi.
- b. Memory sebesar 256 MB atau lebih tinggi.
- c. VGA card 16 MB atau lebih tinggi.
- d. Harddisk 10 GB atau lebih tinggi.

B. Perangkat Lunak

- a. Sistem operasi menggunakan Microsoft Windows XP.
- b. Microsoft SQL Server 2000 sebagai database.
- c. Microsoft Visual Basic 6.0 sebagai *tool* pemrograman.
- d. Power Designer 6 32-bit untuk membuat rancangan DFD dan ERD.
- e. Microsoft Visio 2002 untuk membuat rancangan *system flow*.

4.2 Instalasi dan Konfigurasi

Untuk dapat mengoperasikan aplikasi ini, ada beberapa tahapan yang harus dilakukan agar tidak terjadi gangguan sewaktu program dijalankan. Tahapan yang harus dilakukan adalah sebagai berikut :

- a. Install sistem operasi Microsoft Windows XP.

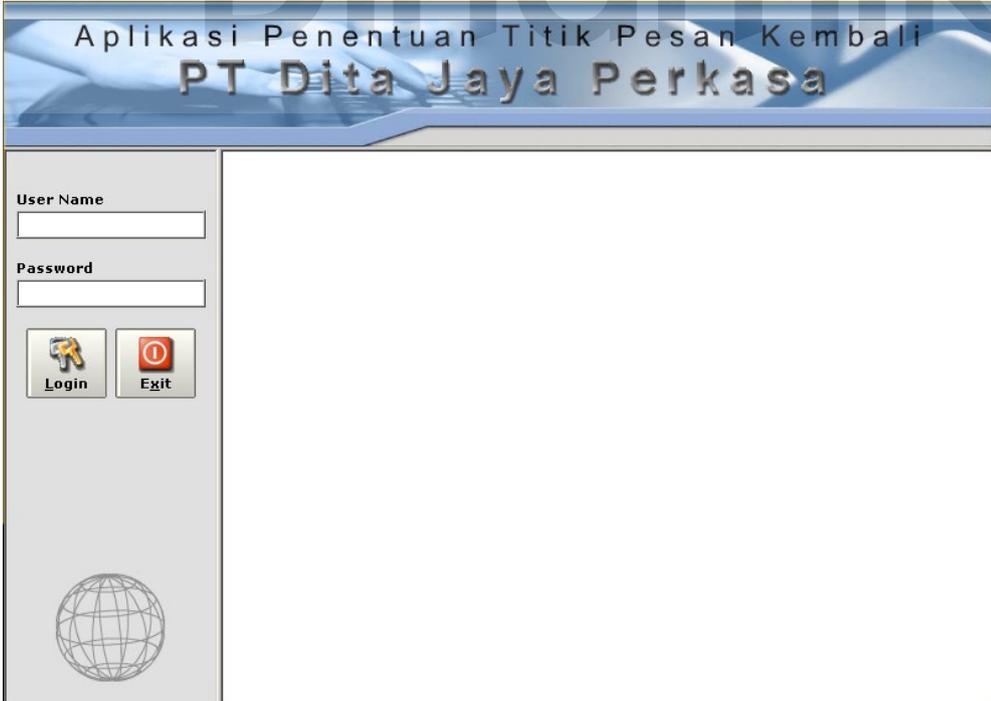
- b. Install Microsoft SQL Server 2000.
- c. Install aplikasi inventori yang sudah ada sebelumnya.
- d. Install aplikasi penentuan titik pesan kembali.

Sedangkan untuk dapat menggunakan program ini, setup database menggunakan *script* yang terdapat pada CD program.

4.3 Implementasi Aplikasi Penentuan Titik Pesan Kembali

A. Proses Login

Proses login dilakukan untuk mengidentifikasi user yang menggunakan aplikasi penentuan titik pesan kembali ini. User diminta untuk memasukkan nama *user* dan *password* sesuai dengan nama *user* dan *password* yang telah ada. Jadi, hanya *user* yang mempunyai kepentingan saja yang dapat menggunakan aplikasi sistem pendukung keputusan ini. Form login menjadi satu dengan form utama yang tampak seperti pada gambar 4.1.

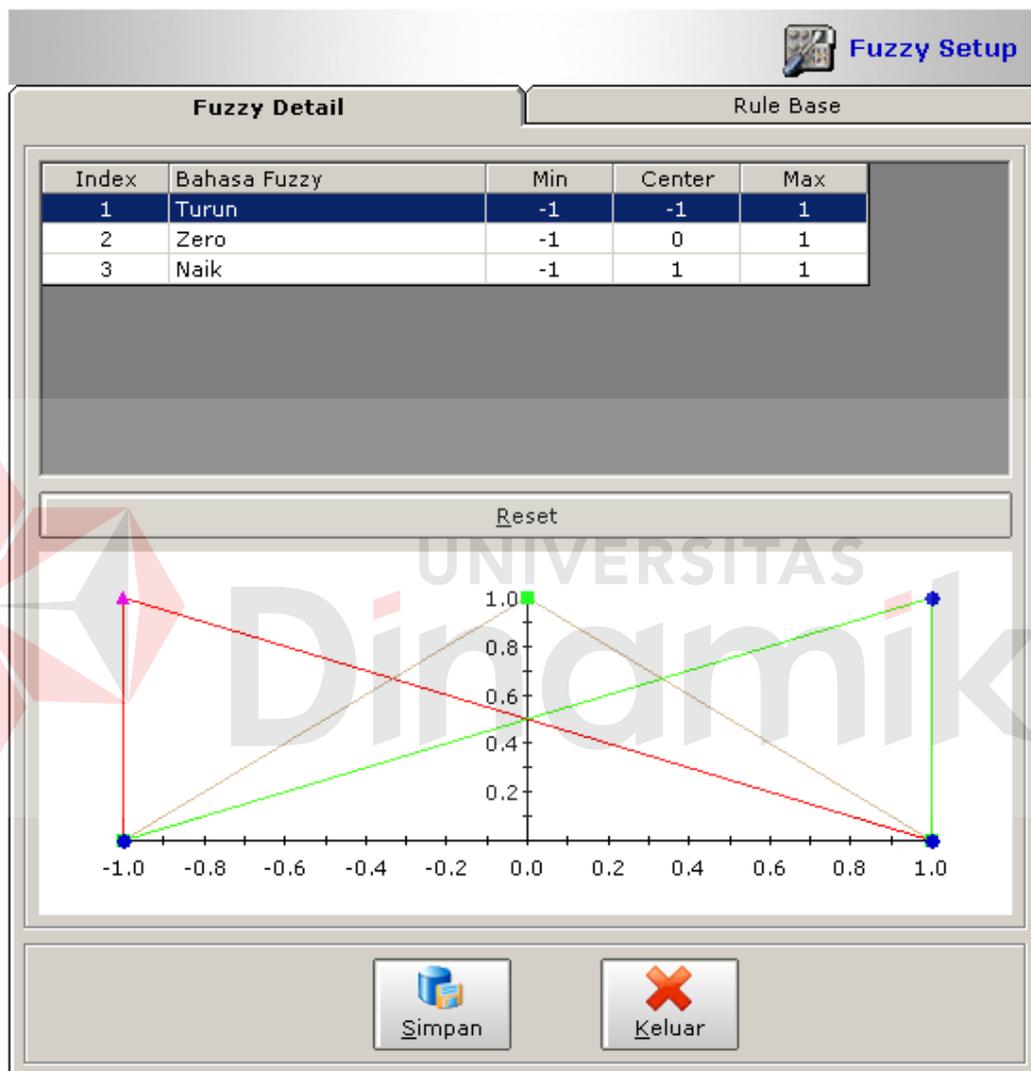


The image shows a web-based login form. At the top, there is a blue header with the text 'Aplikasi Penentuan Titik Pesan Kembali' and 'PT Dita Jaya Perkasa'. Below the header, the form is divided into two main sections. The left section contains a 'User Name' label above a text input field, a 'Password' label above another text input field, and two buttons: 'Login' with a key icon and 'Exit' with a red stop sign icon. At the bottom left of this section is a globe icon. The right section of the form is a large, empty white area.

Gambar 4.1 Form Login

B. Proses Konfigurasi Fuzzy

Konfigurasi fuzzy dilakukan pada form setup fuzzy. Form ini digunakan untuk menentukan nilai-nilai batasan fuzzy dan menentukan aturan fuzzy. Proses konfigurasi fuzzy tampak seperti pada gambar 4.2 dan 4.3.



Gambar 4.2 Form Setup Fuzzy Nilai Batasan

Fuzzy Setup

Fuzzy Detail Rule Base

Kelompok: Pembelian

Index Rule	E1	E2	E3	TPK
1	Naik	Naik	Turun	Turun
2	Naik	Naik	Zero	Turun
3	Naik	Turun	Naik	Turun
4	Naik	Turun	Turun	Turun
5	Naik	Turun	Zero	Zero
6	Naik	Zero	Naik	Turun
7	Naik	Zero	Turun	Turun
8	Naik	Zero	Zero	Turun
9	Turun	Naik	Naik	Zero
10	Turun	Naik	Turun	Naik
11	Turun	Naik	Zero	Naik
12	Turun	Turun	Naik	Naik
13	Turun	Turun	Zero	Naik
14	Turun	Zero	Naik	Zero
15	Turun	Zero	Turun	Naik
16	Turun	Zero	Zero	Naik
17	Zero	Naik	Naik	Turun
18	Zero	Naik	Turun	Turun
19	Zero	Naik	Zero	Zero
20	Zero	Turun	Naik	Zero

Analisa

Simpan Keluar

Gambar 4.3 Form Setup Fuzzy Aturan

C. Proses Migrasi Data

Proses migrasi data dilakukan pada form automasi. Pada form ini, kita bisa melakukan migrasi data pembelian dan penjualan. Gambar 4.4 menunjukkan form automasi.

Automation

Pembelian Penjualan Keluar

Gambar 4.4 Form Automasi

D. Proses Penghitungan

Proses penghitungan penentuan titik pesan kembali dilakukan pada form ROP. Pada form ini, user memilih periode dan menentukan item yang dicari dan akan ditampilkan hasil perhitungan titik pesan kembali. Gambar 4.5 menunjukkan form ROP.

Kode Obat	Nama Obat	DB	TJ	Semua
AMO-000037	AMOXAN 500 MG *NO RE...	2	8	<input type="checkbox"/>
BAQ-000007	BAQUINOR FORTE	2	16	<input type="checkbox"/>
HEM-000028	HEMAVITON ACTION CAP	3	40	<input type="checkbox"/>
INS-000015	INSAAR	2	11	<input type="checkbox"/>
MEF-000013	MEFINAL 500	2	1	<input type="checkbox"/>

Gambar 4.5 Form ROP

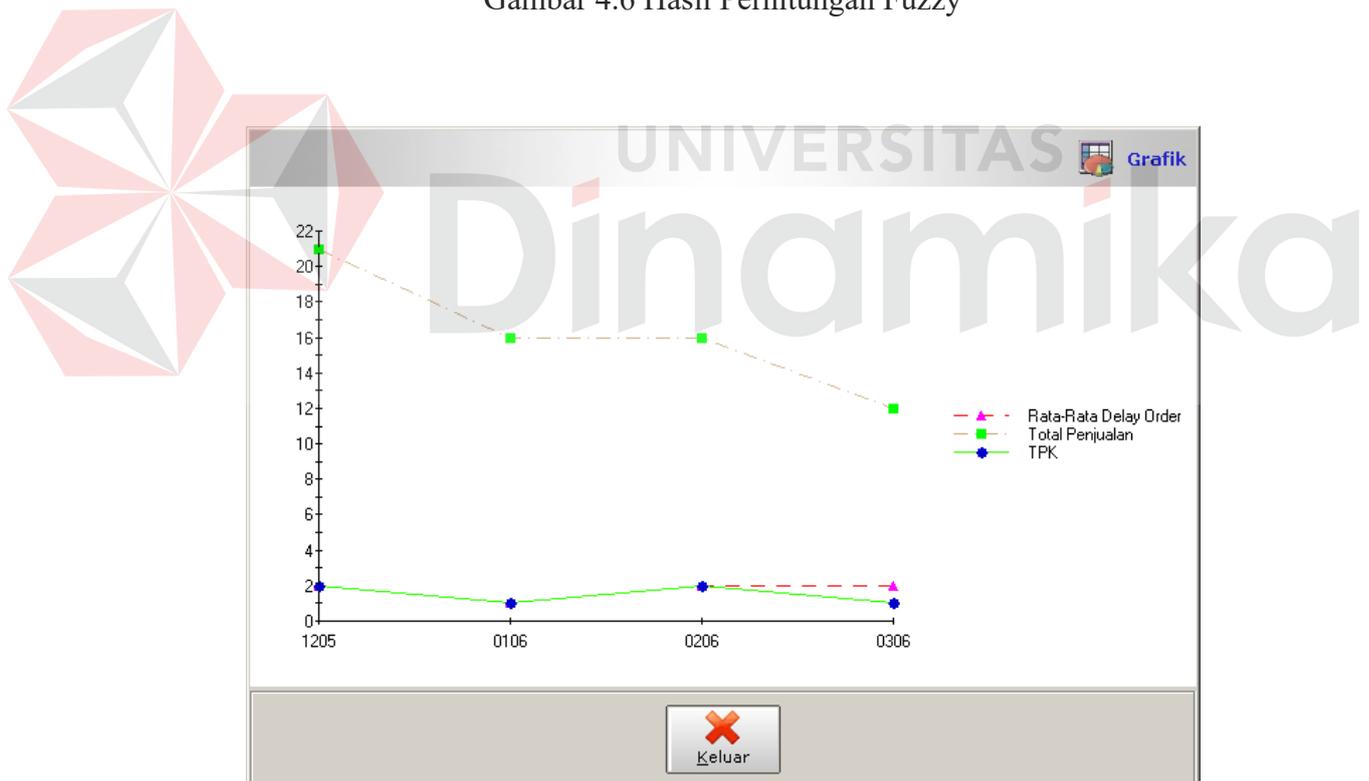
Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	2	8	2	12	1
2	BAQ-000007	BAQUINOR FORTE	2	16	2	29	3
3	HEM-000028	HEMAVITON ACTION CAP	3	40	3	83	10
4	INS-000015	INSAAR	2	11	2	14	2
5	MEF-000013	MEFINAL 500	2	1	2	13	2

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode April Tahun 2006 : 8
 Perkiraan Total Penjualan Periode April Tahun 2006 : 2
 Titik Pesan Kembali : 12




Gambar 4.6 Hasil Perhitungan Fuzzy



Gambar 4.7 Grafik Perubahan Nilai

4.4 Hasil Uji Coba

Berdasarkan pada desain uji coba yang terdapat pada pembahasan sebelumnya (bab 3), sistem aplikasi akan di uji cobakan terhadap data inputan sebenarnya.

4.4.1 Hasil Uji Coba Fungsional

A. Form Main Menu

Tabel 4.1 Tabel Hasil Uji Main Menu

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
1	Deskripsi user dan password yang valid	User = "admin" Password = "admin" Klik tombol login	Frame login akan disable, menu akan muncul	Frame login akan disable, menu akan muncul Gambar 4.1
2	Deskripsi user dan password yang tidak valid	User = "admin1" Password = "admin" Klik tombol login	Tampil pesan "Invalid User Account"	Tampil pesan "Invalid User Account" Gambar 4.8



Gambar 4.8 Pesan Kesalahan User

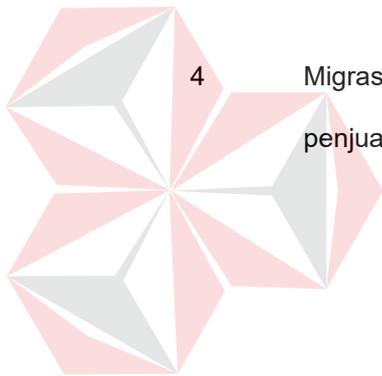
B. Form Automasi

Tabel 4.2 Tabel Hasil Uji Automasi

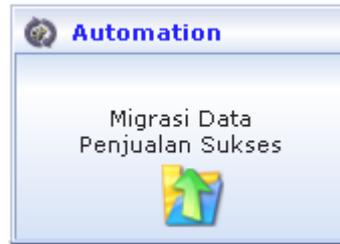
Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
3	Migrasi data pembelian	Klik pada tombol "Pembelian"	Progress akan bergerak dan muncul popup "Migrasi Pembelian Sukses"	Progress akan bergerak dan muncul popup "Migrasi Pembelian Sukses"
4	Migrasi data penjualan	Klik pada tombol "Penjualan"	Progress akan bergerak dan muncul popup "Migrasi Penjualan Sukses"	Progress akan bergerak dan muncul popup "Migrasi Penjualan Sukses"

Gambar 4.9

Gambar 4.10



Gambar 4.9 Popup Migrasi Data Pembelian



Gambar 4.10 Popup Migrasi Data Penjualan

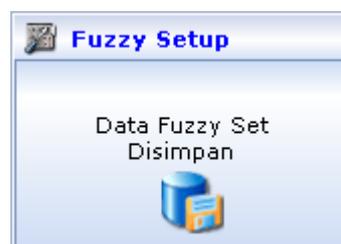
C. Form Batasan Fuzzy

Tabel 4.3 Tabel Hasil Uji Batasan Fuzzy

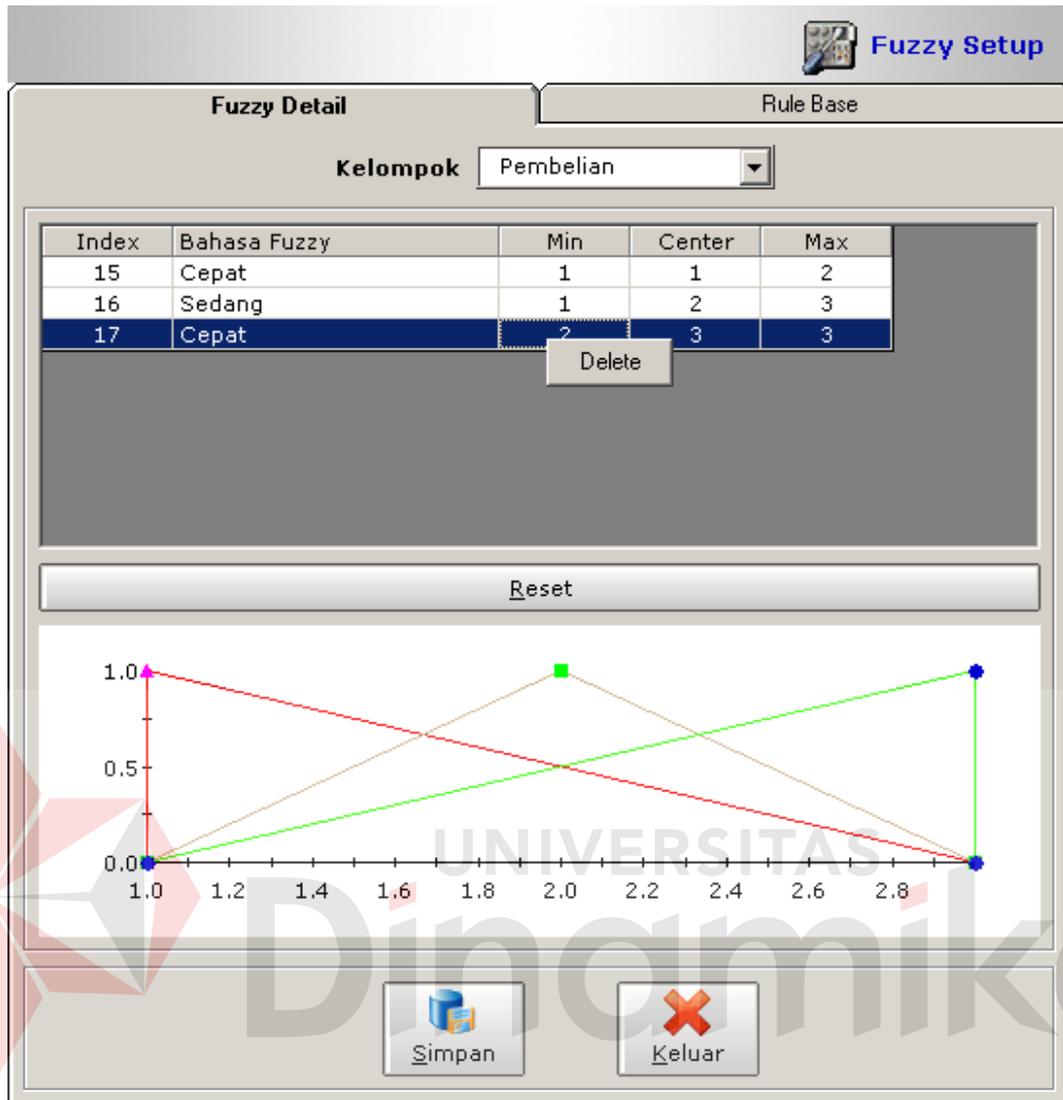
Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
5	Memilih kelompok fuzzy detail	Klik pada combobox kelompok, kelompok = "Pembelian"	Nilai batasan fuzzy kelompok pembelian akan muncul pada list dan grafik batasan akan muncul	Nilai batasan fuzzy kelompok pembelian akan muncul pada list dan grafik batasan akan muncul Gambar 4.2
6	Mengisi index detail fuzzy	Pada list, index akan digenerate oleh sistem =1,2,3,dst	Akan tampil index detail fuzzy = 1,2,3,dst	Akan tampil index detail fuzzy = 1,2,3,dst Gambar 4.2
7	Memasukkan nilai batasan fuzzy	Klik pada list, Bahasa = "Cepat", Min = "1", Center = "1", Max = "1"	Bahasa = "Cepat", Min = "1", Center = "1", Max = "1"	Bahasa = "Cepat", Min = "1", Center = "1", Max = "2"

Tabel 4.3 Tabel Hasil Uji Batasan Fuzzy (Lanjutan)

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
8	Menyimpan nilai batasan fuzzy	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"	Akan muncul popup "Data Fuzzy Set Disimpan" Gambar 4.2 Gambar 4.11
9	Merubah nilai batasan fuzzy	Klik pada list, Bahasa = "Sedang", Min = "1", Center = "2", Max = "3"	Bahasa = "Sedang", Min = "1", Center = "2", Max = "3"	Bahasa = "Sedang", Min = "1", Center = "2", Max = "3"
10	Menghapus nilai batasan fuzzy	Klik kanan pada list dan pilih "Delete"	Nilai batasan fuzzy akan terhapus	Nilai batasan fuzzy akan terhapus Gambar 4.12



Gambar 4.11 Popup Simpan Fuzzy Set



Gambar 4.12 Delete Detail Fuzzy Set

D. Form Aturan Fuzzy

Tabel 4.4 Tabel Hasil Uji Aturan Fuzzy

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
11	Memasukkan index rule	Pada list, index rule akan digenerate oleh sistem =1,2,3,dst	Akan tampil index rule = 1,2,3,dst	Akan tampil index rule = 1,2,3,dst

Tabel 4.4 Tabel Hasil Uji Aturan Fuzzy (Lanjutan)

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
				Gambar 4.3
12	Memasukkan index pembelian	Pada list, index pembelian akan digenerate oleh sistem sesuai dengan index pembelian	Akan tampil index pembelian sesuai dengan pembelian	Akan tampil index pembelian sesuai dengan index pembelian
				Gambar 4.3
13	Memasukkan index penjualan	Pada list, index penjualan akan digenerate oleh sistem sesuai dengan index penjualan	Akan tampil index penjualan sesuai dengan penjualan	Akan tampil index penjualan sesuai dengan index penjualan
				Gambar 4.3
14	Memasukkan index spk	Klik pada list, pilih index spk, index spk = 10	Index spk = 10	Index spk = 10
15	Menyimpan aturan fuzzy	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"	Akan muncul popup "Data Rulebase Disimpan"
				Gambar 4.12



Gambar 4.13 Popup Simpan Aturan Fuzzy

E. Form Maintenance User

Tabel 4.5 Tabel Hasil Uji Maintenance User

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
16	Merubah user name	Klik textbox user name = "Saya"	User name = "Saya"	User name = "Saya" Gambar 4.14
17	Merubah password	Klik textbox password = "Saya"	Password = "Saya"	Password = "Saya" Gambar 4.14
18	Menyimpan perubahan account	Klik pada tombol "Simpan"	Akan muncul popup "Data Disimpan"	Akan muncul popup "Data Disimpan" Gambar 4.13



Gambar 4.14 Popup Simpan User

Gambar 4.15 Maintenance User

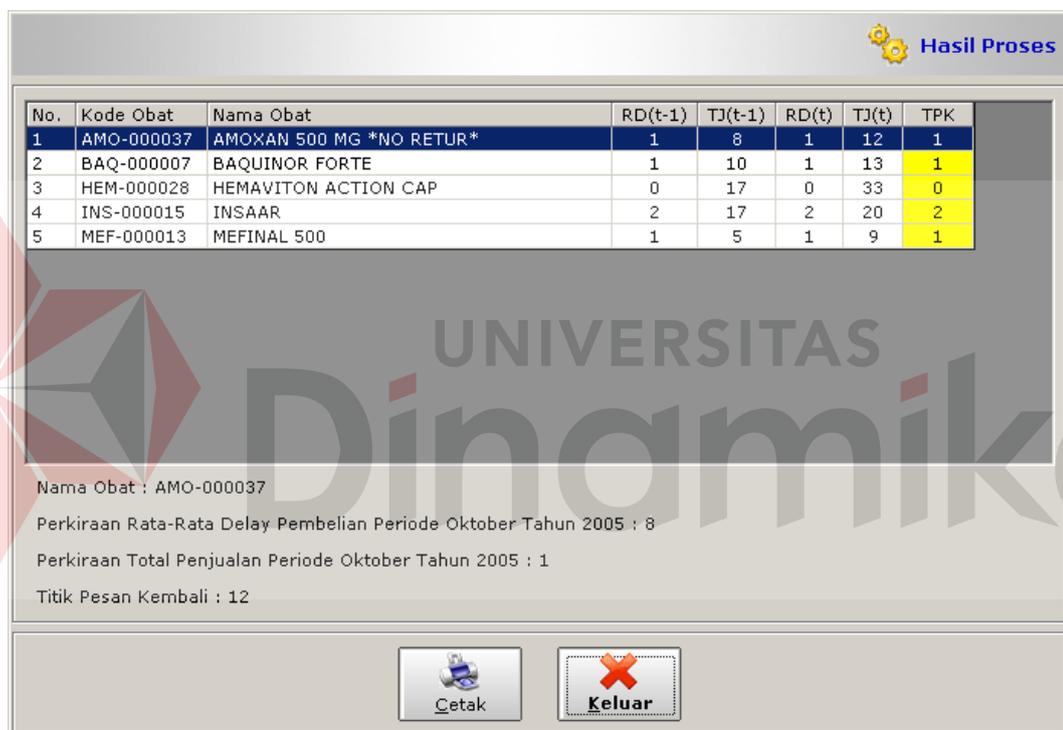
F. Form Proses

Tabel 4.6 Tabel Hasil Uji Proses

Test Case ID	Tujuan	Input	Output yang diharapkan	Hasil keluaran system
19	Memilih periode	Klik pada combobox periode, periode = "0306"	Periode = "0306"	Periode = "0306" Gambar 4.5
20	Menampilkan data transaksi	Klik pada combobox periode, periode = "0306"	Data transaksi periode "0306" akan muncul pada list	Data transaksi periode "0306" akan muncul pada list Gambar 4.5
21	Menghitung titik pesan kembali	Klik pada tombol "Proses"	Akan menampilkan hasil perhitungan titik pesan kembali sesuai dengan data	Akan menampilkan hasil perhitungan titik pesan kembali sesuai dengan data Gambar 4.6

4.4.2 Hasil Uji Coba Validasi

Berdasarkan pada desain uji coba validasi yang terdapat pada pembahasan sebelumnya (bab 3), sistem aplikasi akan di uji cobakan terhadap data inputan barang periode Oktober 2005 sampai Maret 2006 guna menentukan titik pesan kembali (TPK). Uji coba yang akan dilakukan menggunakan data sampel yang telah ditentukan.



No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TPK
1	AMO-000037	AMOXAN 500 MG *NO RETUR*	1	8	1	12	1
2	BAQ-000007	BAQUINOR FORTE	1	10	1	13	1
3	HEM-000028	HEMAVITON ACTION CAP	0	17	0	33	0
4	INS-000015	INSAAR	2	17	2	20	2
5	MEF-000013	MEFINAL 500	1	5	1	9	1

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode Oktober Tahun 2005 : 8
 Perkiraan Total Penjualan Periode Oktober Tahun 2005 : 1
 Titik Pesan Kembali : 12

Cetak Keluar

Gambar 4.16 Hasil Perhitungan Periode Oktober 2005

 Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	1	26	1	30	2
2	BAQ-000007	BAQUINOR FORTE	2	25	2	28	3
3	HEM-000028	HEMAVITON ACTION CAP	1	79	1	95	4
4	INS-000015	INSAAR	2	30	2	33	3
5	MEF-000013	MEFINAL 500	0	0	0	4	0

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode November Tahun 2005 : 26
 Perkiraan Total Penjualan Periode November Tahun 2005 : 1
 Titik Pesan Kembali : 30




Gambar 4.17 Hasil Perhitungan Periode November 2005

 Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	2	19	2	23	2
2	BAQ-000007	BAQUINOR FORTE	2	38	2	41	4
3	HEM-000028	HEMAVITON ACTION CAP	3	21	3	52	7
4	INS-000015	INSAAR	3	28	3	32	4
5	MEF-000013	MEFINAL 500	3	16	2	20	2

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode Desember Tahun 2005 : 19
 Perkiraan Total Penjualan Periode Desember Tahun 2005 : 2
 Titik Pesan Kembali : 23




Gambar 4.18 Hasil Perhitungan Periode Desember 2005

 Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	2	17	2	21	2
2	BAQ-000007	BAQUINOR FORTE	2	42	2	47	4
3	HEM-000028	HEMAVITON ACTION CAP	0	45	2	30	3
4	INS-000015	INSAAR	2	24	2	27	3
5	MEF-000013	MEFINAL 500	1	9	2	3	1

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode Tahun 2005 : 17
 Perkiraan Total Penjualan Periode Tahun 2005 : 2
 Titik Pesan Kembali : 21

 Cetak
  Keluar

Gambar 4.19 Hasil Perhitungan Periode Januari 2006

 Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	1	12	1	16	1
2	BAQ-000007	BAQUINOR FORTE	2	9	2	20	2
3	HEM-000028	HEMAVITON ACTION CAP	2	44	2	91	8
4	INS-000015	INSAAR	1	16	1	19	1
5	MEF-000013	MEFINAL 500	3	10	2	21	2

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode Februari Tahun 2006 : 12
 Perkiraan Total Penjualan Periode Februari Tahun 2006 : 1
 Titik Pesan Kembali : 16

 Cetak
  Keluar

Gambar 4.20 Hasil Perhitungan Periode Februari 2006

 Hasil Proses

No.	Kode Obat	Nama Obat	RD(t-1)	TJ(t-1)	RD(t)	TJ(t)	TP
1	AMO-000037	AMOXAN 500 MG *NO RETUI	2	12	2	16	2
2	BAQ-000007	BAQUINOR FORTE	2	12	2	25	2
3	HEM-000028	HEMAVITON ACTION CAP	3	27	3	70	9
4	INS-000015	INSAAR	2	10	2	13	2
5	MEF-000013	MEFINAL 500	0	0	2	-10	0

Nama Obat : AMO-000037
 Perkiraan Rata-Rata Delay Pembelian Periode Maret Tahun 2006 : 12
 Perkiraan Total Penjualan Periode Maret Tahun 2006 : 2
 Titik Pesan Kembali : 16

 Cetak
  Keluar

Gambar 4.21 Hasil Perhitungan Periode Maret 2006

Berdasarkan output yang dikeluarkan sistem, aplikasi ini dikatakan berhasil 100 %, karena hasil yang dikeluarkan sesuai dengan desain sistem pada bab 3.

4.5 Analisa Hasil Uji Coba

Berdasarkan uji coba yang telah dilakukan maka dapat dihasilkan suatu perbedaan mendasar dari perbandingan pembelian barang menggunakan sistem yang lama yaitu dengan teori “kira-kira” dibandingkan dengan melakukan perhitungan menggunakan metode *fuzzy logic*. Tabel 4.14 menjelaskan simulasi perbandingan pembelian Amoxan 500, dimana asumsi pembelian pada simulasi didapat dari 2 kali rata-rata penjualan periode sebelumnya.

Tabel 4.7 Tabel TPK dan Rata-rata Penjualan

	2005				2006		
	September	Oktober	November	Desember	Januari	Februari	Maret
TPK	-	1	6	2	2	2	1
Rata2 Penjualan	2	2	2	2	2	1	-

Tabel 4.8 Tabel Simulasi Pembelian

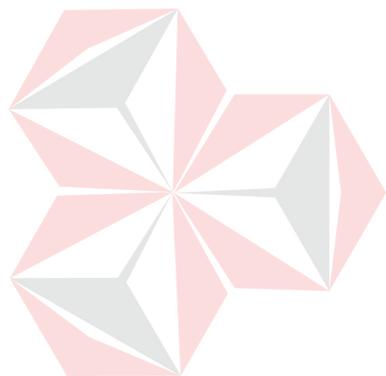
Tahun	Bulan	Tanggal	Penjualan	Simulasi		Kenyataan	
				Pembelian	Stok	Pembelian	Stok
2005	10	1			25		25
2005	10	11	15		10	21	31
2005	10	12	5		5		26
2005	10	13	1		4		25
2005	10	19	1		3		24
2005	10	20	2	4	5		22
2005	10	21			5	5	27
2005	11	16		4	9	8	35
2005	11	18	8	4	5		27
2005	11	21	1	4	8	1	27
2005	11	22			8	1	28
2005	11	23	1		7		27
2005	11	24	2	4	9	2	27
2005	11	26			9	10	37
2005	11	28	4	4	9		33
2005	11	30	2		7		31
2005	12	1	2		5	1	30
2005	12	3	2	4	7		28
2005	12	9	1	4	10	2	29
2005	12	10	1		9		28
2005	12	27	6	4	7	11	33
2005	12	28	5	4	6		28
2006	1	2			6	2	30
2006	1	14	5	4	5	9	34
2006	1	16	1		4		33
2006	1	20	1	4	7	1	33
2006	1	24	1		6	1	33
2006	1	30	1		5	1	33
2006	2	8	1		4	1	33
2006	2	15		4	8	10	43
2006	2	16	7	4	5		36
2006	2	17			5		36
2006	2	23			5	1	37
2006	2	24	1		4		36
2006	3	8			4		36
2006	3	10	2	2	4		34
2006	3	13	1		3		33
2006	3	16	2	2	3		31

Tabel 4.8 Tabel Simulasi Pembelian (Lanjutan)

Tahun	Bulan	Tanggal	Penjualan	Simulasi		Kenyataan	
				Pembelian	Stok	Pembelian	Stok
2006	3	17	1	2	4		30
2006	3	18	2	2	4		28

Seperti tampak pada tabel 4.8, jika tidak melakukan penghitungan dengan *fuzzy*, sisa stok terlalu banyak. Hal ini mempengaruhi keuntungan perusahaan. Berbeda dengan hasil simulasi, stok akhir relatif rendah.

Setelah dilakukan uji validasi, didapatkan kesimpulan bahwa sistem aplikasi secara validasi dinyatakan berhasil. Hal ini dibuktikan dengan output nilai TPK yang mempengaruhi stok, sehingga stok tidak terlalu besar dan tidak kurang.



BAB V

PENUTUP

5.1 Kesimpulan

Setelah pembuatan sistem pendukung keputusan untuk menentukan titik pesan kembali pada PT Dita Jaya Perkasa, maka dapat ditarik kesimpulan bahwa dengan menerapkan titik pesan kembali hasil keluaran sistem, maka dapat memperkecil biaya penyimpanan barang, karena item pada gudang tidak berlebihan, tetapi juga menyukupi untuk transaksi selanjutnya, dan biaya pengeluaran untuk pengadaan barang dapat diminimumkan.

5.2 Saran

Adapun saran-saran yang dapat disampaikan oleh penulis berkaitan dengan penulisan Tugas Akhir ini adalah sebagai berikut :

1. Sistem ini akan lebih sempurna jika dilengkapi dengan sistem akuntansi guna menghitung biaya yang akan dikeluarkan untuk pengadaan barang.
2. Sebagai bahan pembandingan bisa dicoba menggunakan metode lainnya selain *Fuzzy Logic*, misal *Neural Network*, *Genetic Algorithm*.

DAFTAR PUSTAKA

Buckley, James J., 2005, *Fuzzy Expert System and Fuzzy Reasoning*, John Wiley & Sons, Inc., Birmingham.

Bonde, Allen, 2005, *Fuzzy Logic Basics*, GTE Government Systems Corp., Needham, (Online),
(<http://www.austinlinks.com/Fuzzy/index.html>, diakses 19 Juli 2006).

Kaehler, Steven D., 2004. *Fuzzy Logic Tutorial*, Seattle Robotics Society, Seattle, (Online),
(<http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html>, diakses 4 Juni 2006).

Kendall & Kendall, 2002, *Systems Analysis And Design Fifth Edition*, Prentice Hall, New Jersey.

Pal, Sankar K. and Shiu, Simon C.K., 2004, *Foundations of Soft Case-Based Reasoning*, John Wiley & Sons, Inc., New Jersey.

Winston, Wayne L., 2004, *Operation Research Applications and Algorithms Fourth Edition*, Thomson, Indiana University.

Wolkenhauer, Olaf, 2001, *Fuzzy Mathematics in System Theory ad Data Analysis*, John Wiley & Sons, Inc., New York.

Yamit, Zulian, 2003, *Manajemen Persediaan*, Ekonisa, Yogyakarta.