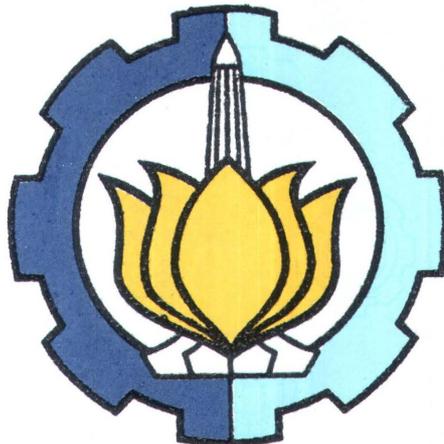


27006/H106



**RANCANG BANGUN SISTEM INFORMASI PELAPORAN  
PENJUALAN UNTUK EVALUASI KINERJA MEDICAL  
REPRESENTATIVE PT KALBE FARMA SURABAYA**

**TUGAS AKHIR**



RSSI  
005.3  
Ass  
r-1  
2006

Disusun Oleh :  
**SOFYAN ASSAURI**  
5201.100.008

PUSKAPUS PUSKAPUS ITS	
Tgl. Terima	16-2-06
Terima Dari	H
No. Agenda Prp.	224099

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2006**

**RANCANG BANGUN SISTEM INFORMASI PELAPORAN  
PENJUALAN UNTUK EVALUASI KINERJA MEDICAL  
REPRESENTATIVE PT KALBE FARMA SURABAYA**

**TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Sarjana Komputer**

**Pada**

**Program Studi Sistem Informasi**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya**

**Mengetahui/Menyetujui**

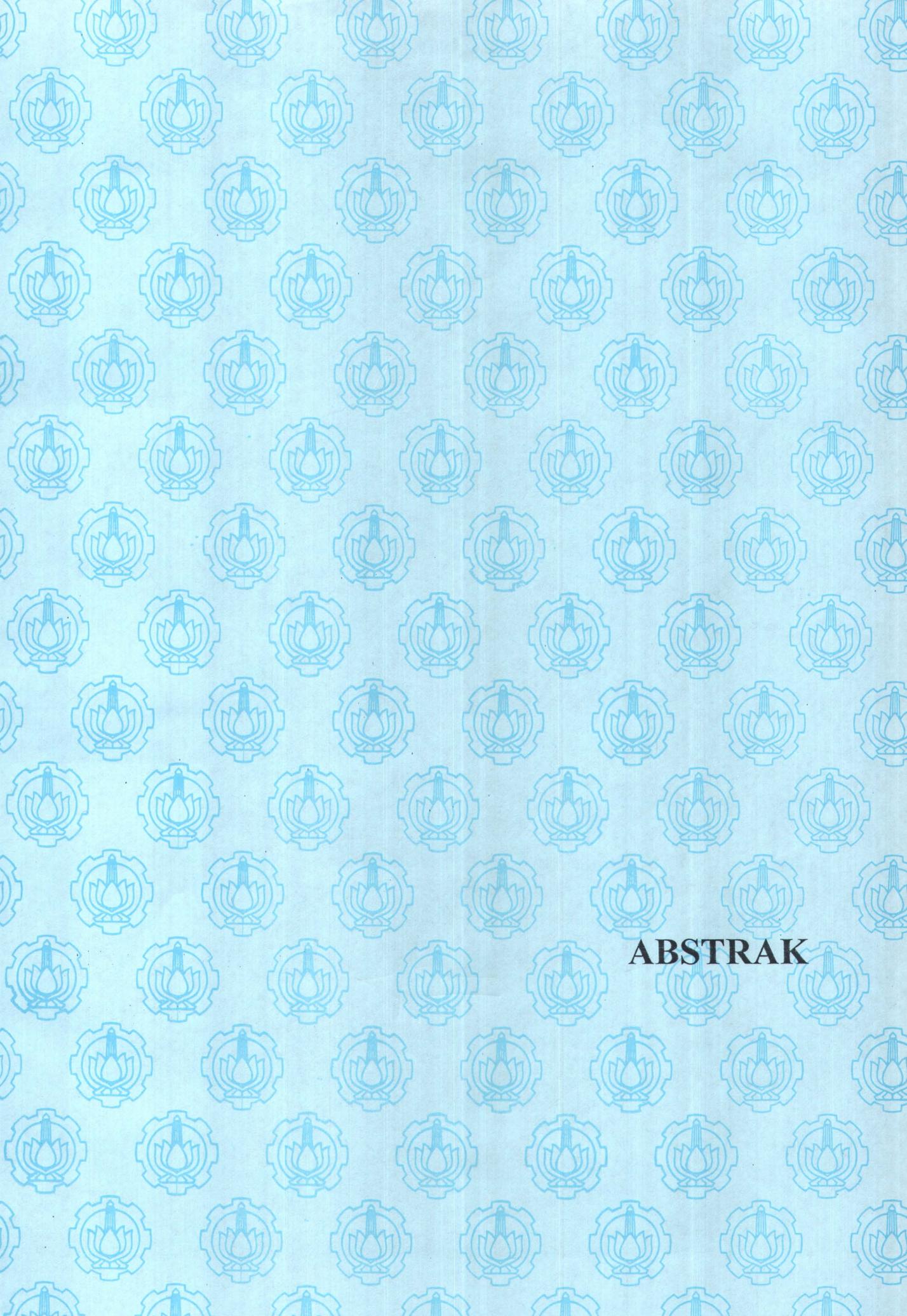
**Dosen Pembimbing I**



**Ir. Khakim Ghozali**  
**NIP. 131.846.108**

**SURABAYA**

**JANUARI 2006**



**ABSTRAK**

## ABSTRAK

*Armada penjualan merupakan inti dari sebuah kegiatan pemasaran perusahaan. Medical Representative adalah nama yang biasa digunakan untuk menyebut armada penjualan di perusahaan farmasi, begitu pula dengan PT. Kalbe Farma. Kalbe menggunakan pembagian daerah kerja berdasar region-region yang di dalamnya terdapat tiga sampai empat Medical Representative dengan seorang supervisor sebagai pimpinannya. Saat ini pengukuran kinerja Medical Representative masih dilakukan secara kolektif tiap region. Tugas Akhir ini bertujuan merancang dan membangun sebuah sistem pengukuran kinerja Medical Representative yang dapat dimanfaatkan untuk mengevaluasi kinerja Medical Representative secara personal. Manfaat yang didapat dari Tugas Akhir ini adalah menyajikan detail seluruh hasil kerja seorang Medical Representative dalam sebuah sistem yang terintegrasi sekaligus menjadi alat monitoring penjualan dan pemasaran yang nantinya dapat digunakan pihak manajemen untuk mengambil keputusan strategis pemasaran.*

*Penelitian diawali dengan analisa terhadap kebutuhan informasi sistem yang didapat dari survey dan wawancara. Identifikasi kebutuhan dan desain sistem didefinisikan dengan menggunakan UML dan ERD. Pengembangan perangkat lunak dikembangkan dengan berbasis web dengan menggunakan JSP, MySQL sebagai basis data, dan Tomcat sebagai web server. Uji coba dilakukan dengan melakukan pengambilan contoh dari setiap fungsi untuk mengevaluasi data masukan, dan kevalidan hasil evaluasi kinerja Medical Representative yang telah dilakukan.*

*Tugas Akhir ini menghasilkan aplikasi yang dapat menilai kinerja Medical Representative, melalui proses perbandingan antara hasil kerja Medical Representative dengan parameter yang telah ditetapkan oleh pihak manajemen, fleksibilitas dalam hal perubahan bobot tiap parameter memberikan informasi secara detail sekaligus dinamis terhadap kinerja Medical Representative.*

*Kata kunci : pengukuran kinerja, Medical Representative, monitoring*





**KATA PENGANTAR**

## KATA PENGANTAR

*Dengan menyebut nama Allah yang Maha Pengasih lagi Maha Penyayang  
Segala puji bagi Allah, Tuhan semesta alam  
Maha Pemurah lagi Maha Penyayang  
Yang menguasai hari pembalasan  
Hanya Engkalulah yang kami sembah dan hanya kepada Engkaulah kami  
mohon pertolongan  
Tunjukilah kami jalan yang lurus  
Jalan orang-orang yang telah Engkau anugerahkan nikmat kepada mereka;  
bukan jalan mereka yang dimurkai dan bukan mereka yang sesat.*

Segala puji dan syukur semata ditujukan ke hadirat Allah SWT yang atas rahmat dan hidayah-Nya penulis dapat menyelesaikan tugas akhir berjudul

### **"Rancang Bangun Sistem Infomasi Pelaporan Penjualan untuk Evaluasi Kinerja Medical Representative PT Kalbe Farma Surabaya"**

Mata kuliah Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan program strata satu (S-1) pada Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan Tugas Akhir ini, penulis berusaha untuk menerapkan ilmu yang didapat selama menjalani perkuliahan di Sistem Infomasi tidak terlepas dari petunjuk, bimbingan, bantuan, dan dukungan berbagai pihak.

Terima kasih penulis sampaikan kepada :

- Pemimpin besar Nabi Muhammad SAW.
- Ayah dan Ibuku tersayang atas segala doa, bimbingan, dukungan, dan kasih sayangnya yang tiada pernah terputus.
- Adik-adikku, Muhammad Alfarizi dan Muhammad Fachrurrozy, yang sudah merelakan sebagian waktunya untuk membantu penulis
- Anugerah terindah yang pernah kumiliki, Lula Permata Sari, *We'll rent a little world.*
- Seluruh keluarga besar penulis atas dukungannya.

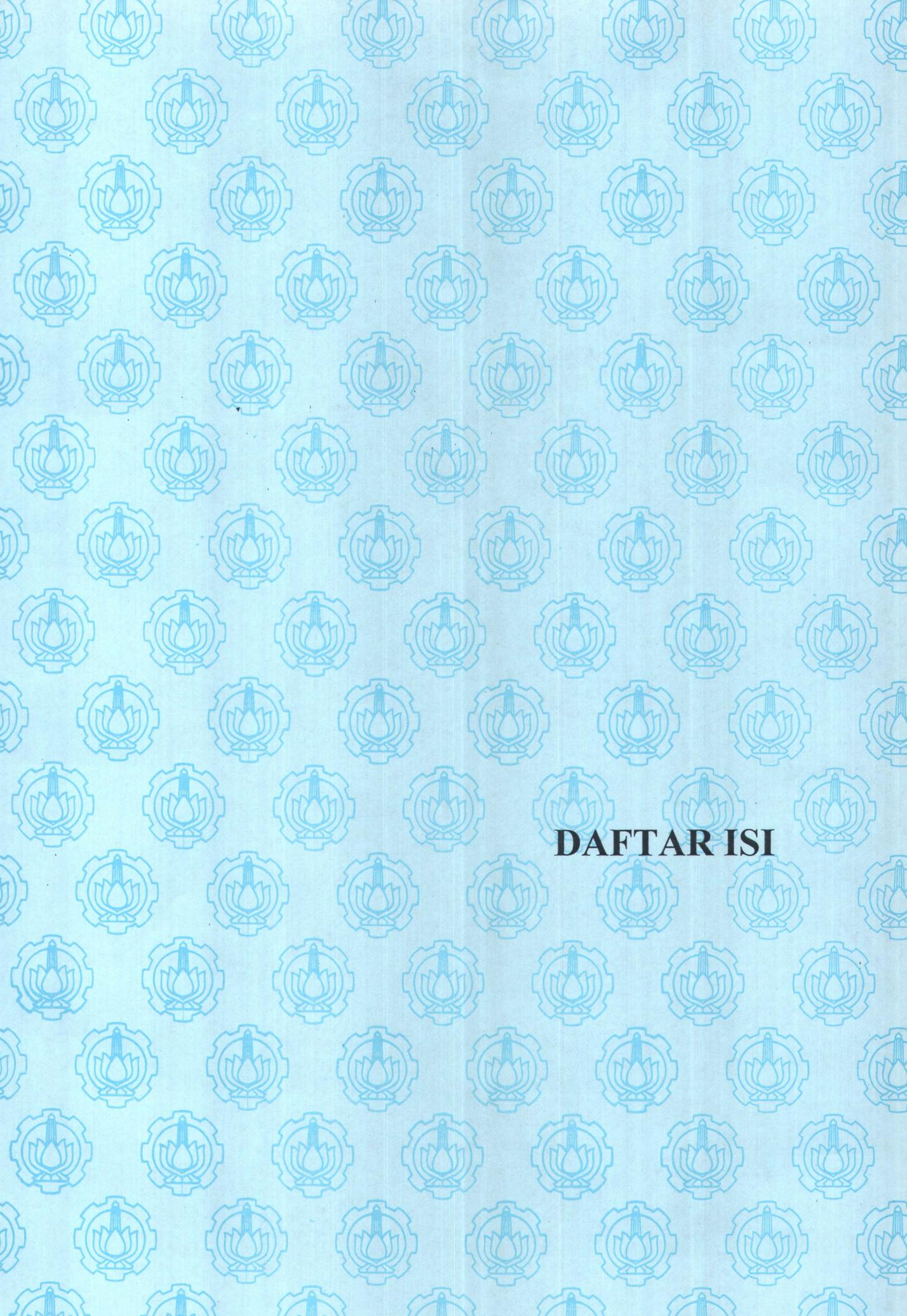
- Bapak Ir. Khakim Ghozali, selaku Dosen Pembimbing I, yang selalu memberikan bimbingan, petunjuk dan dorongan kepada penulis sampai Tugas Akhir ini selesai.
- Bapak Ir. Achmad Holil Noor Ali, M.Kom., Bapak Bambang Setiwan, S.Kom. M.Kom., dan Bapak Edwin Riksakomara, S.Kom. MT atas bantuan dan arahannya yang sangat membantu penulis.
- Bapak Prof. Dr. Ir. Arif Djunaidy, M.Sc, PhD selaku Dekan Fakultas Teknologi Informasi, ITS.
- Bapak Ir. Aris Tjahyanto M.Kom, selaku dosen wali, yang telah membimbing dan mengarahkan penulis semasa kuliah di Sistem Informasi.
- Bpk. Rully Soelaiman, S.Kom, M.Kom selaku Sekretaris Program Studi Sistem Informasi ITS
- Dosen-dosen Sistem Informasi: Bpk. Febriliyan Samopa, S.Kom, M.Kom, Bpk. Mudjahidin, ST, MT, Bpk. Faizal Johan A, S.Kom, Ibu Wiwik Anggraeni S.Si, M.Kom, Ibu Erma Suryani, S.Kom, MT, Bpk. Rully Agus, S.Kom, dan Ibu Nur Aini, S.Kom, atas segala bimbingan dan arahan selama ini
- Seluruh Staf dan Karyawan Program Studi Sistem Informasi – ITS yang telah banyak membantu.
- Para “pendahulu”: Deptha, Cemit, Dudut, Soni, Beny, Icha, Yudi, Tante, Gesti, dan Phipu yang telah memacu semangat penulis untuk lulus.
- Teman-teman seperjuangan : Arin, Yeni, Ari, Angga-Om, Anshar, Fatwa, Kindi, Mbok Yah, Risa, Eka, Mo, Ryan, Didit, Titus, Ratih, Mbeth. *“Where do you go? Friend.*
- Saudara 2001 yang laen Poleng, Irfan, Adhi, Gofs, Miko, Boz, Bagus, Reza, Bambang Mamang, Priza, Dimas, Aik, Mirza, Pangestu. *Wish You All the Best. We love 2001*
- Segenap Keluargaku Sistem Informasi seluruh angkatan 2002, 2003, 2004, dan 2005 atas bantuan dan kenangan indah bersama kalian.

Dan pihak-pihak yang mempunyai peranan penting dalam kehidupan penulis, yang tidak dapat disebutkan satu-persatu.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih terdapat banyak kekurangan sehingga penulis berharap adanya kritik dan saran yang membangun dari rekan pembaca sehingga menjadikan Tugas Akhir ini lebih baik. Oleh karena itu penulis sangat berterima kasih jikalau ada saran maupun kritik yang membangun.

Surabaya, Januari 2006

Penulis



**DAFTAR ISI**

## DAFTAR ISI

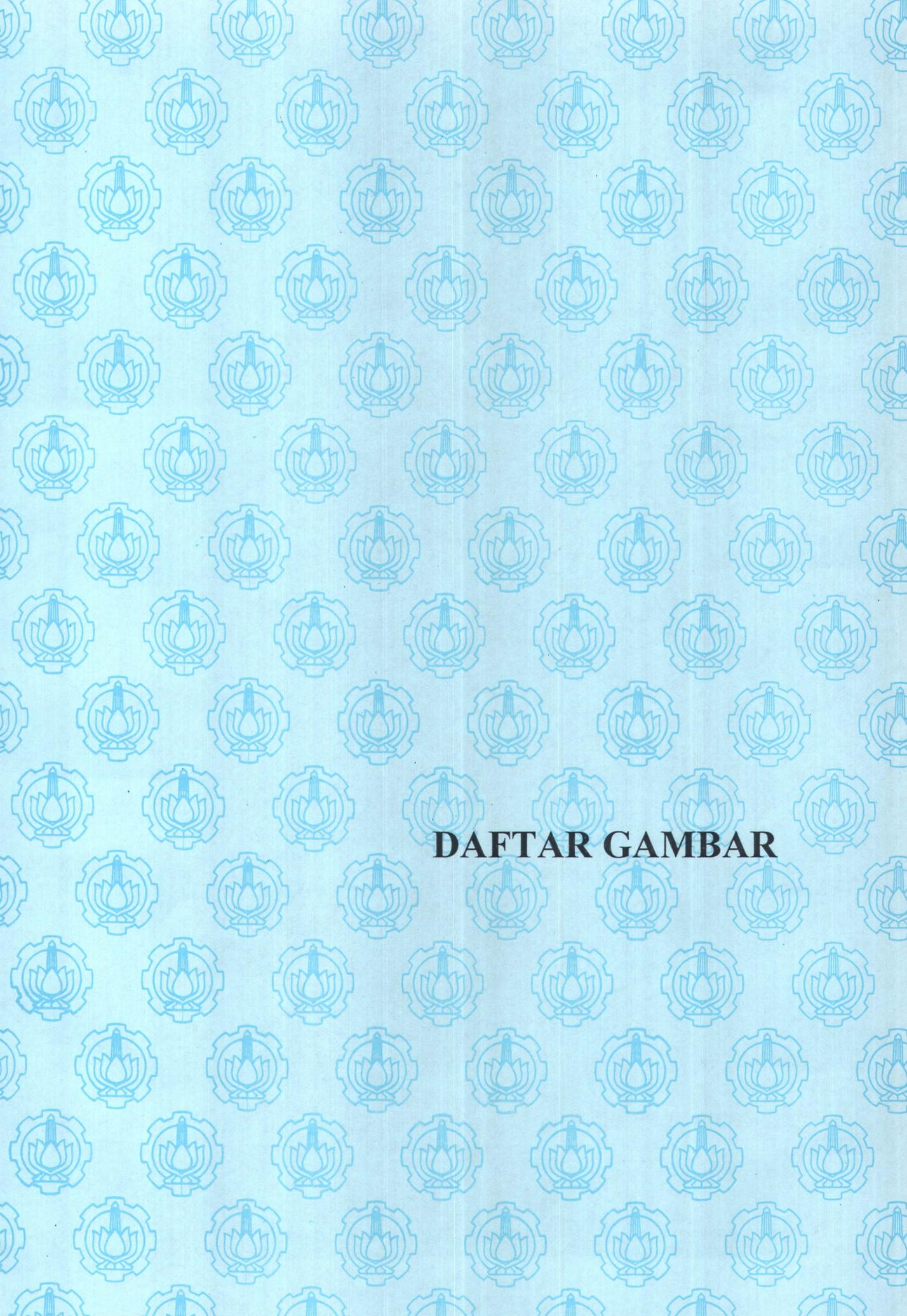
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>ABSTRAK .....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>v</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xi</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I. PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Tujuan .....	2
1.3. Manfaat .....	2
1.4. Permasalahan.....	3
1.5. Batasan Masalah.....	3
1.6. Metodologi .....	4
1.7. Sistematika Penulisan.....	5
<b>BAB II. DASAR TEORI.....</b>	<b>7</b>
2.1. Armada Penjualan .....	7
2.1.1. Jenis Armada Penjualan .....	7
2.1.2. Tugas Armada Penjualan .....	8
2.1.3. Struktur Armada Penjualan.....	9
2.2. Teori Pengukuran Kinerja.....	11
2.2.1 Pengukuran Kinerja Secara Umum.....	11
2.2.2 Pengukuran Kinerja Medrep.....	12
2.3. UML (UNIFIED MODELLING LANGUAGE) .....	14
2.3.1. Definisi UML .....	15
2.3.2. Konsep Dasar UML .....	16
2.3.3. <i>Use Case Diagram</i> .....	17
2.3.4. <i>Sequence Diagram</i> .....	18
2.3.4. <i>Activity Diagram</i> .....	20
2.4. JSP ( <i>Java Server Pages</i> ) .....	21
2.4.1. <i>Tag Deklarasi</i> .....	23
2.4.2. <i>Tag Ekspresi</i> .....	23
2.4.3. <i>Tag Scriptlet</i> .....	23
2.4.4. <i>Directives</i> .....	24
2.4.5. Obyek Implisit.....	25
2.4.6. Komentari.....	26

2.5. MySQL.....	26
2.5.1. Definisi MySql.....	26
2.5.2. Fitur MySQL.....	27
2.5.3. Tipe Data MySQL.....	28
2.5.4. <i>Structure Query Language</i> (SQL) untuk MySQL.....	30
2.5.4.1. <i>Data Definition Language</i> (DDL).....	30
2.5.4.2. <i>Data Manipulation Language</i> (DML).....	30
2.5.4.3. <i>Security</i> .....	31
2.5.4.4. <i>Integrity</i> .....	31
2.5.4.5. <i>Auxiliary</i> .....	31
<b>BAB III. ANALISA KEBUTUHAN SISTEM.....</b>	<b>32</b>
3.1. PT. Kalbe Farma.....	32
3.1.1. Visi dan Misi.....	33
3.1.2. Divisi <i>Ethycal</i> PT. Kalbe Farma.....	34
3.2. Pengukuran Kinerja di PT. Kalbe Farma.....	35
3.2.1. Pembagian Region.....	35
3.2.2. Volume Penjualan.....	36
3.2.3. Kuota Penjualan.....	38
3.3. Kinerja Pemasaran di PT. Kalbe Farma.....	40
3.3.1. Sistem Berjalan.....	42
3.3.2. Permasalahan Saat Ini.....	43
3.3.3. Analisa kinerja <i>Medical Representative</i> .....	45
3.4. Kebutuhan Informasi.....	49
3.4.1. Identifikasi Pengguna.....	50
3.4.2. Kebutuhan Sistem.....	51
<b>BAB IV. DESAIN DAN PERANCANGAN PERANGKAT LUNAK.....</b>	<b>55</b>
4.1. Desain Sistem.....	55
4.1.1. <i>Use Case Diagram</i> .....	55
4.1.1.1. <i>Actor</i> .....	55
4.1.1.2. <i>Use Case</i> .....	56
4.1.2. <i>Activity Diagram</i> .....	58
4.1.2.1. <i>Activity Diagram</i> Mengelola Data Apotek.....	58
4.1.2.2. <i>Activity Diagram</i> Mengelola Data Penjualan.....	59
4.1.2.3. <i>Activity Diagram</i> Mengelola Data <i>Medrep</i> .....	60
4.1.2.4. <i>Activity Diagram</i> Mengelola Data Dokter.....	61
4.1.2.5. <i>Activity Diagram</i> Mengelola Data Visitasi.....	62
4.1.2.6. <i>Activity Diagram</i> Mengelola Bobot.....	63
4.1.2.7. <i>Activity Diagram</i> Mengelola Data Region.....	64
4.1.2.8. <i>Activity Diagram</i> Analisa.....	64
4.1.2.9. <i>Activity Diagram</i> Mengelola Data Obat.....	66
4.1.2.9. <i>Activity Diagram</i> Mengelola Data <i>User</i> .....	67
4.1.2.10. <i>Activity Diagram</i> Validasi.....	67

4.1.3. <i>Sequence Diagram</i> .....	68
4.1.3.1. <i>Sequence Diagram Insert Data Apotek</i> .....	68
4.1.3.2. <i>Sequence Diagram Update Data Apotek</i> .....	68
4.1.3.3. <i>Sequence Diagram Delete Data Apotek</i> .....	70
4.1.3.4. <i>Sequence Diagram Lihat Data Apotek</i> .....	70
4.1.3.5. <i>Sequence Diagram Insert Data Medrep</i> .....	71
4.1.3.6. <i>Sequence Diagram Update Data Medrep</i> .....	71
4.1.3.7. <i>Sequence Diagram Delete Data Medrep</i> .....	73
4.1.3.8. <i>Sequence Diagram Lihat Data Medrep</i> .....	73
4.1.3.9. <i>Sequence Diagram Insert Data Dokter</i> .....	74
4.1.3.10 <i>Sequence Diagram Update Data Dokter</i> .....	75
4.1.3.11 <i>Sequence Diagram Delete Data Dokter</i> .....	75
4.1.3.12 <i>Sequence Diagram Lihat Data Dokter</i> .....	76
4.1.3.13 <i>Sequence Diagram Insert Data Transaksi</i> .....	76
4.1.3.14 <i>Sequence Diagram Update Data Transaksi</i> .....	77
4.1.3.15 <i>Sequence Diagram Delete Data Transaksi</i> .....	78
4.1.3.16 <i>Sequence Diagram Lihat Data Transaksi</i> .....	78
4.1.3.17 <i>Sequence Diagram Update Data Region</i> .....	80
4.1.3.18 <i>Sequence Diagram Lihat Data Region</i> .....	80
4.1.3.19 <i>Sequence Diagram Lihat Target Penjualan</i> .....	81
4.1.3.20 <i>Sequence Diagram Insert Data Visitasi</i> .....	81
4.1.3.21 <i>Sequence Diagram Update Data Visitasi</i> .....	83
4.1.3.22 <i>Sequence Diagram Delete Data Visitasi</i> .....	83
4.1.3.23 <i>Sequence Diagram Lihat Data Visitasi</i> .....	84
4.1.3.24 <i>Sequence Diagram Update Bobot Perhitungan</i> .....	84
4.1.3.25 <i>Sequence Diagram Lihat Data Penjualan</i> .....	86
4.1.3.26 <i>Sequence Diagram Analisa Kinerja Medrep</i> .....	86
4.1.3.27 <i>Sequence Diagram Lihat Data Obat</i> .....	86
4.1.4. ERD ( <i>Entity Relational Diagram</i> ) .....	89
4.2. Implementasi Sistem .....	96
4.2.1. Modul <i>Sales</i> .....	97
4.2.2. Modul <i>Apotek</i> .....	99
4.2.3. Modul <i>Supervisor</i> .....	101
4.2.4. Modul <i>Medrep</i> .....	106
4.2.5. Modul <i>Manager</i> .....	106
4.2.6. Modul <i>Administrator</i> .....	114
<b>BAB V. UJI COBA DAN EVALUASI</b> .....	<b>116</b>
5.1. Lingkungan Uji Coba.....	116
5.2. Skenario dan Proses Uji Coba.....	116
5.2.1. Uji Coba Validasi.....	117
5.2.2. Uji Coba Masukan Data <i>User</i> .....	118
5.2.3. Uji Coba Integritas Data.....	119



5.2.4. Uji Coba Validitas Data .....	121
5.2.5. Uji Coba Evaluasi Kinerja <i>Medrep</i> .....	122
<b>BAB VI. PENUTUP</b> .....	<b>128</b>
6.1. Kesimpulan .....	128
6.2. Saran .....	129
<b>DAFTAR PUSTAKA</b> .....	<b>130</b>
<b>LAMPIRAN A : Narasi <i>Use Case</i></b> .....	<b>A-1</b>
<b>LAMPIRAN B : Data Gathering</b> .....	<b>B-1</b>
<b>LAMPIRAN C : Daftar Penjualan Unit</b> .....	<b>C-1</b>
<b>LAMPIRAN D : Tabel Kebutuhan Sistem</b> .....	<b>D-1</b>



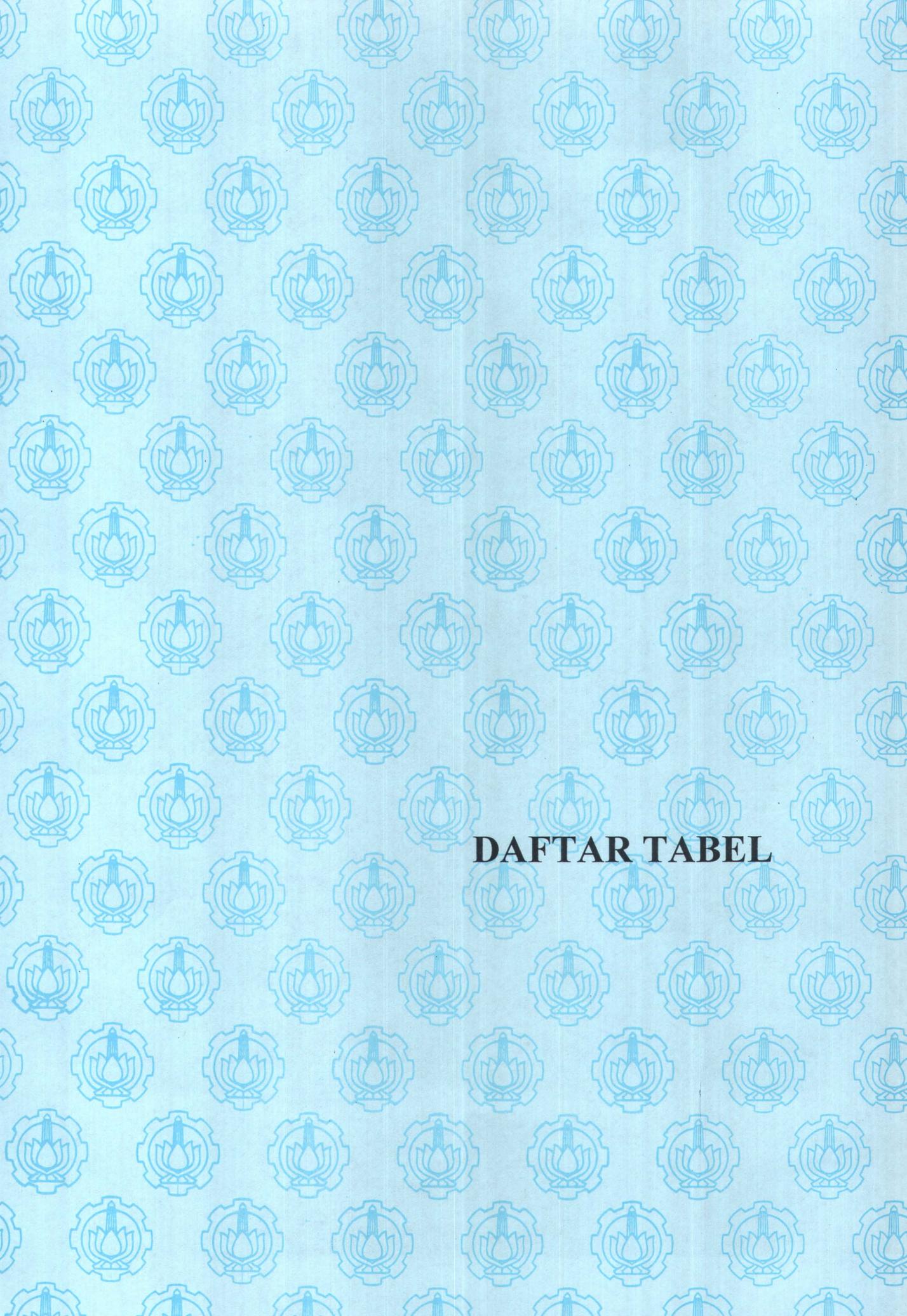
**DAFTAR GAMBAR**

## DAFTAR GAMBAR

Gambar 2.1. Unsur-unsur <i>Use case Diagram</i> .....	17
Gambar 2.2. Contoh <i>Sequence Diagram</i> .....	18
Gambar 2.3. Contoh <i>Activity Diagram</i> .....	20
Gambar 2.4. Contoh <i>Tag</i> Ekspresi JSP .....	23
Gambar 2.5. Contoh <i>Tag Scriptlet</i> JSP .....	24
Gambar 2.6. Contoh Komentar pada JSP.....	26
Gambar 3.1. Core Values PT. Kalbe Farma.....	34
Gambar 3.2. Formulasi perhitungan volume Penjualan.....	36
Gambar 3.3. Formulasi perhitungan penjualan obat .....	37
Gambar 3.4. Formulasi perhitungan Kuota Penjualan .....	38
Gambar 3.5 Struktur Organisasi PT. Kalbe Farma .....	40
Gambar 3.6. Divisi Penjualan dan Pemasaran <i>Ethycal</i> .....	41
Gambar 3.7. Diagram Alir proses penilaian kinerja pemasaran .....	43
Gambar 3.8. Formulasi Perhitungan target Medrep.....	47
Gambar 3.9. Formulasi Indeks Kinerja Medrep.....	49
Gambar 4.1. <i>Use Case Diagram</i> Sistem .....	57
Gambar 4.2. <i>Use Case Diagram</i> Sistem beserta Relasinya .....	58
Gambar 4.3. <i>Activity Diagram</i> Mengelola Data Apotek.....	59
Gambar 4.4. <i>Activity Diagram</i> Mengelola Data Penjualan.....	60
Gambar 4.5. <i>Activity Diagram</i> Mengelola Data <i>Medrep</i> .....	61
Gambar 4.6. <i>Activity Diagram</i> Mengelola Data Dokter .....	62
Gambar 4.7. <i>Activity Diagram</i> Mengelola Data Visitasi .....	63
Gambar 4.8. <i>Activity Diagram</i> Mengelola Bobot .....	64
Gambar 4.9. <i>Activity Diagram</i> Mengelola Data Region.....	65
Gambar 4.10. <i>Activity Diagram</i> Analisa .....	65
Gambar 4.11. <i>Activity Diagram</i> Mengelola Data Obat.....	66
Gambar 4.12. <i>Activity Diagram</i> Mengelola Data User .....	67
Gambar 4.13. <i>Activity Diagram</i> Validasi .....	68
Gambar 4.14. <i>Sequence Diagram</i> insert Data Apotek.....	69
Gambar 4.15. <i>Sequence Diagram</i> Update Data Apotek .....	69
Gambar 4.16. <i>Sequence Diagram</i> Delete Data Apotek.....	70
Gambar 4.17. <i>Sequence Diagram</i> Lihat data apotek.....	71
Gambar 4.18. <i>Sequence Diagram</i> Insert Data <i>Medrep</i> .....	72
Gambar 4.19. <i>Sequence Diagram</i> Update Data <i>Medrep</i> .....	72
Gambar 4.20. <i>Sequence diagram</i> Delete Data <i>Medrep</i> .....	73
Gambar 4.21. <i>Sequence Diagram</i> Lihat Data <i>Medrep</i> .....	74
Gambar 4.22. <i>Sequence Diagram</i> Insert Data Dokter .....	74
Gambar 4.23. <i>Sequence Diagram</i> Update Data Dokter.....	75
Gambar 4.24. <i>Sequence Diagram</i> Delete Data Dokter .....	76
Gambar 4.25. <i>Sequence Diagram</i> Lihat Data Dokter .....	77

Gambar 4.26. <i>Sequence Diagram</i> Insert Data Transaksi .....	77
Gambar 4.27. <i>Sequence Diagram</i> Update Data Transaksi .....	78
Gambar 4.28. <i>Sequence Diagram</i> Delete Data Transaksi .....	79
Gambar 4.29. <i>Sequence Diagram</i> Lihat Data Transaksi .....	79
Gambar 4.30. <i>Sequence Diagram</i> Update Data Region .....	80
Gambar 4.31. <i>Sequence Diagram</i> Lihat Data Region .....	81
Gambar 4.32. <i>Sequence Diagram</i> Lihat Target Penjualan .....	82
Gambar 4.33. <i>Sequence Diagram</i> Insert Data Visitasi .....	82
Gambar 4.34. <i>Sequence Diagram</i> Update Data Visitasi .....	83
Gambar 4.35. <i>Sequence Diagram</i> Delete Data Visitasi .....	84
Gambar 4.36. <i>Sequence diagram</i> Lihat Data Visitasi .....	85
Gambar 4.37. <i>Sequence Diagram</i> Update Bobot Perhitungan .....	85
Gambar 4.38. <i>Sequence Diagram</i> Lihat Data Penjualan .....	87
Gambar 4.39. <i>Sequence diagram</i> Lihat Data Obat .....	87
Gambar 4.40. <i>Sequence diagram</i> Analisa Kinerja <i>Medrep</i> .....	88
Gambar 4.41. <i>CDM (Conceptual Data Model)</i> Sistem .....	89
Gambar 4.42. <i>PDM (Physical Data Model)</i> Sistem .....	90
Gambar 4.43. Desain Antar Muka Sistem Informasi Kinerja <i>Medrep</i> .....	96
Gambar 4.44. Halaman Manajemen Transaksi .....	97
Gambar 4.45. Input Data Transaksi .....	98
Gambar 4.46. View Data Transaksi .....	98
Gambar 4.47. Daftar Seluruh Apotek Pada Sebuah Region .....	99
Gambar 4.48. Update Data Apotek .....	99
Gambar 4.49. Halaman Penjualan Sebuah Apotek .....	100
Gambar 4.50. <i>Pseudocode</i> Halaman Penjualan Apotek .....	100
Gambar 4.51. Halaman Penjualan Obat .....	101
Gambar 4.52. Manajemen Region .....	102
Gambar 4.53. <i>View</i> Target Penjualan .....	102
Gambar 4.54. <i>Pseudocode</i> Target Penjualan Tahun .....	103
Gambar 4.55. Halaman Manajemen <i>Medrep</i> .....	103
Gambar 4.56. Form Pemberian Rekomendasi .....	104
Gambar 4.57. Halaman Manajemen Dokter .....	104
Gambar 4.58. Insert Data Visitasi .....	105
Gambar 4.59. <i>View</i> Data Visitasi .....	105
Gambar 4.60. Halaman Manajemen Bobot .....	107
Gambar 4.61. Halaman Region <i>Overview</i> .....	107
Gambar 4.62. Evaluasi Kinerja <i>Medrep</i> .....	108
Gambar 4.63. Halaman Detail Evaluasi Kinerja <i>Medrep</i> .....	108
Gambar 4.64. Contoh Rekomendasi Supervisor .....	109
Gambar 4.65. <i>Pseudocode</i> Analisa Kinerja <i>Medrep</i> .....	110
Gambar 4.66. Halaman Detail Visitasi .....	113
Gambar 4.67. Laporan Penjualan Tiap Region .....	113
Gambar 4.68. Detail Penjualan Tiap Obat .....	114

Gambar 4.69. Detail Penjualan Tiap Dokter .....	114
Gambar 4.70. Detail Penjualan Tiap Apotek .....	114
Gambar 4.71. Halaman Manajemen <i>User</i> .....	115
Gambar 4.72. Manajemen Obat .....	115
Gambar 5.1. Contoh Login.....	117
Gambar 5.2. Contoh Login <i>Sales</i> yang Sukses .....	118
Gambar 5.3. Contoh Login yang Gagal .....	118
Gambar 5.4. <i>Form Insert Medrep</i> baru .....	118
Gambar 5.5. Contoh Pesan Konfirmasi.....	119
Gambar 5.6. Detail Bobot pada Modul Manajer.....	120
Gambar 5.7. <i>Form</i> Evaluasi Kinerja Medrep .....	120
Gambar 5.8. Target Penjualan pada <i>Form</i> Manajemen Region.....	121
Gambar 5.9. Target Penjualan pada <i>Form</i> Analisa Kinerja <i>Medrep</i> .....	121
Gambar 5.10. Nilai Penjualan sebuah Region sebuah Periode .....	122
Gambar 5.11. Data Region 8.....	123
Gambar 5.12. Bobot Periode tahun 2005 .....	123
Gambar 5.13. Contoh target penjualan medrep .....	124
Gambar 5.14. Contoh daftar dokter didikan medrep.....	124
Gambar 5.15. <i>Form</i> Penilaian Kinerja medrep .....	125
Gambar 5.16. Detail Visitasi medrep.....	126
Gambar 5.17. indeks kinerja medrep .....	126
Gambar 5.18. <i>Form</i> Pemilihan periode grafik .....	127
Gambar 5.19. Grafik nilai penjualan medrep.....	127



**DAFTAR TABEL**

## DAFTAR TABEL

Tabel 2.1. Konsep Dasar UML [DHA-03] .....	16
Tabel 2.2. Obyek Implisit pada JSP .....	25
Tabel 2.3. Tipe Data MySQL.....	28
Tabel 3.1. Pembagian Region .....	36
Tabel 3.2. Klasifikasi Jumlah Penduduk Region .....	38
Tabel 3.3. Klasifikasi Jenis Region.....	38
Tabel 3.4. Klasifikasi Jumlah Dokter di Region.....	38
Tabel 3.5. Nilai Bobot Konversi .....	39
Tabel 3.6. Nilai Konversi Kunjungan per Hari.....	47
Tabel 3.7. Nilai Konversi Pendapatan per Kunjungan.....	47
Tabel 3.8. Nilai Konversi Biaya Kunjungan.....	48
Tabel 3.9. Nilai Konversi Target Penjualan.....	49
Tabel 4.1. Deskripsi Tabel jnsReg .....	91
Tabel 4.2. Deskripsi Tabel jumPenReg .....	91
Tabel 4.3. Deskripsi Tabel Region.....	91
Tabel 4.4. Deskripsi Tabel Obat .....	92
Tabel 4.5. Deskripsi Tabel Dokter.....	92
Tabel 4.6. Deskripsi Tabel Medrep.....	92
Tabel 4.7. Deskripsi Tabel Apotek .....	93
Tabel 4.8. Deskripsi Tabel Lap.....	93
Tabel 4.9. Deskripsi Tabel Visitasi .....	94
Tabel 4.10. Deskripsi Tabel Transaksi.....	94
Tabel 4.11. Deskripsi Tabel detTrans .....	95
Tabel 4.12. Deskripsi Tabel Bobot .....	95
Tabel 4.13. Deskripsi Tabel <i>User</i> .....	96



**BAB I**  
**PENDAHULUAN**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Tak seorangpun menyanggah pentingnya armada penjualan dalam bauran pemasaran. Namun, perusahaan-perusahaan sangat sensitif terhadap biaya yang tinggi dan meningkat (gaji, komisi, bonus, biaya perjalanan, tunjangan) untuk memelihara armada penjualan. Perusahaan perlu mempertimbangkan secara hati-hati kapan dan bagaimana menggunakan wiraniaga. Tidak mengherankan, jika perusahaan berusaha beralih ke penjualan melalui pos, telepon, dan internet demi menurunkan biaya penjualan lapangan. Dan perusahaan mencoba meningkatkan produktivitas armada penjualan yang ada melalui pelatihan, motivasi, dan kompensasi yang lebih baik.

Tugas-tugas armada penjualan bervariasi sesuai dengan keadaan perekonomian. Ketika terjadi kekurangan produk, wakil penjualan tidak mengalami masalah dalam menjual. Beberapa perusahaan berkesimpulan bahwa dibutuhkan lebih sedikit armada penjualan pada periode tersebut. Tetapi pemikiran tersebut mengabaikan peran lain dari wiraniaga - mengalokasikan produk, membuat tenang pelanggan yang kecewa, mengkomunikasikan rencana perusahaan untuk mengatasi kekurangan itu, dan menjual produk perusahaan yang persediannya cukup banyak.

Sedangkan selama periode kelimpahan produk, perwakilan perusahaan bersaing secara besar-besaran untuk memenangkan pilihan pelanggan. Perusahaan semakin banyak menilai wakil perwakilan penjualan mereka bukan hanya berdasarkan masalah volume penjualan, melainkan juga berdasarkan kemampuan mereka untuk menciptakan kepuasan pelanggan dan laba.

PT. Kalbe Farma, sebuah perusahaan farmasi besar yang memiliki pangsa pasar yang luas baik pada obat-obatan *OTC (Over The Counter)*/obat bebas maupun pada obat-obatan *Ethycal*/Obat yang harus ditebus dengan resep dokter, tentu membutuhkan sebuah informasi pemasaran yang akurat dan tepat yang bersifat

*update* dan dapat digunakan sebagai bahan pertimbangan pembuatan keputusan bagi pihak manajer. Berbeda dengan obat-obatan OTC yang murni menggunakan media iklan di media massa sebagai sarana pemasarannya, penjualan dan pemasaran obat-obatan *Ethycal* sangat bergantung pada kemampuan barisan *medrep* (*Medical Representative*) yang harus turun langsung ke lapangan untuk memasarkan obat-obatan tersebut pada para dokter.

Saat ini, para manajer merasa kesulitan untuk dapat menganalisa keefektifan seorang *medrep* dan hubungannya dengan hasil penjualan obat-obatan *Ethycal* di sebuah lokasi. Sumber-sumber informasi yang ada saat ini cenderung terpisah-pisah dan berdiri sendiri, sehingga pihak manajer seringkali merasa kesulitan untuk dapat mengintegrasikan berbagai informasi seperti laporan penjualan per tiap-tiap apotek, data dokter beserta spesialisasinya, maupun informasi mengenai *medrep* itu sendiri.

Berdasarkan latar belakang tersebut, dalam Tugas Akhir ini dibangun sebuah sistem Evaluasi Kinerja *Medrep* baru yang nantinya bukan hanya dapat membantu perusahaan untuk meningkatkan laba penjualan namun juga dapat membantu para *medrep* menggunakan waktunya secara lebih efisien.

## **1.2 Tujuan**

Tujuan dari tugas akhir ini adalah merancang dan membangun sebuah aplikasi Sistem Informasi yang dapat menganalisa kinerja bagian pemasaran pada bagian obat-obatan *Ethycal* khususnya keefektifan seorang *medrep* berdasarkan data-data transaksional yang tersedia.

## **1.3 Manfaat**

Beberapa manfaat yang bisa diperoleh dengan diimplementasikannya tugas akhir ini, diantaranya adalah

- a. Pihak supervisor dapat memonitor dan menganalisa keefektifan kinerja *medrep* yang berada di bawahnya.

- b. Pihak manajer mendapatkan informasi penjualan obat-obatan secara lebih baik dan lebih detail sebagai bahan pertimbangan pembuatan keputusan pemasaran.
- c. Pihak *medrep* akan mendapatkan informasi yang lebih baik mengenai pelanggan (dokter) sehingga nantinya diharapkan dapat membantu proses pemasaran obat itu sendiri.

#### 1.4 Permasalahan

Permasalahan yang diangkat dalam tugas akhir ini adalah:

- Apa saja informasi yang dibutuhkan pihak manajemen terkait dengan pengukuran kinerja *medrep*?
- Bagaimana desain aplikasi dari Sistem Informasi Pemasaran yang terkomputerisasi dan terintegrasi pada setiap unit perusahaan?
- Bagaimana implementasi dari desain yang ada agar dapat memenuhi kebutuhan para pengguna sekaligus terhindar dari ketidaksesuaian dengan sistem sebelumnya?

#### 1.5 Batasan Masalah

Dari permasalahan yang telah disebutkan di atas, maka batasan-batasan dalam tugas akhir ini adalah:

- Tugas akhir ini akan mengambil data-data transaksional dari bagian *Ethycal* PT. Kalbe Farma.
- Pengembangan aplikasi akan dititikberatkan pada proses analisa dan evaluasi kinerja *medrep*.
- Tidak menangani proses transaksi dan distribusi penjualan.
- Desain dan perancangan sistem menggunakan *use case*, *activity* dan *sequence diagram* dari UML.
- Aplikasi yang dibuat adalah aplikasi berbasis web, dengan menggunakan JSP sebagai server side scriptnya dan MySQL sebagai databasenya.

## 1.6 Metodologi

Pembuatan tugas akhir ini terbagi menjadi beberapa tahap pengerjaan yang tertera sebagai berikut :

### 1. Studi Literatur dan Identifikasi Permasalahan

Tahap ini dilakukan untuk memahami metode konsep armada penjualan (*Medical Representative*), pengukuran kinerja pemasaran, konsep desain basis data dan penanganan data, serta pemrograman sisi server yang akan digunakan dalam pembangunan aplikasi. Studi literatur diperoleh dari referensi pada berbagai sumber seperti buku literatur, dokumentasi Tugas Akhir, artikel dan sebagainya.

### 2. Analisa Kebutuhan Sistem

Tahap ini dilakukan untuk memahami keadaan dan kebutuhan informasi yang diperlukan pihak pasien selaku customer, digunakan untuk memperoleh informasi sebanyak-banyaknya mengenai kebutuhan pengguna pada aplikasi. Analisa kebutuhan ini menggunakan metode observasi langsung, survey, dan wawancara.

### 3. Perancangan dan Desain Sistem

Hal yang dilakukan oleh tahap ini didapatkan dari tahap sebelumnya. Tahap ini menerjemahkan kebutuhan pengguna yang telah didefinisikan menjadi rancangan aplikasi yang sesuai dengan keinginan perusahaan. Desain sistem dibuat dengan benar-benar memperhatikan serta berpedoman pada rancangan aplikasi.

### 4. Implementasi Aplikasi

Tahap ini merupakan tahap pembuatan dan pengembangan aplikasi sesuai dengan dengan desain sistem yang ditetapkan pada tahap sebelumnya. Sistem informasi evaluasi kinerja *medrep* merupakan aplikasi berbasis web, yang dibangun dengan JSP sebagai *server side script*-nya dan MySQL sebagai DBMS.

## 5. Uji Coba dan Evaluasi

Tahap ini mengevaluasi sistem evaluasi kinerja *medrep* yang dilakukan oleh pihak-pihak yang berkepentingan, sehingga dapat menunjukkan taraf kehandalan sekaligus memperbaiki kekurangan yang ada untuk mencapai hasil optimal. Dalam ujicoba aplikasi, dibuat beberapa skenario testing untuk membuktikan bahwa sistem yang dikembangkan memenuhi kebutuhan perusahaan.

## 6. Penyusunan Buku Tugas Akhir

Tahap terakhir ini merupakan dokumentasi pelaksanaan tugas akhir. Diharapkan, buku tugas akhir ini bermanfaat bagi pembaca yang ingin mengembangkan sistem ini lebih lanjut maupun pada lain kasus.

### 1.7 Sistematika Pembahasan

Pembahasan Tugas Akhir ini dibagi menjadi beberapa bab sebagai berikut :

Bab I, Pendahuluan. Bab ini berisi Bab ini berisi latar belakang, tujuan, manfaat, permasalahan, batasan masalah, metodologi dan sistematika pembahasan.

Bab II, Dasar Teori. Membahas teori yang digunakan sebagai dasar dan ilmu penunjang dalam pengerjaan tugas akhir ini, seperti teori pengukuran pemaaran perusahaan, teori Armada penjualan, UML model, pemrograman web, dan *database management system*.

Bab III, Analisa kebutuhan Informasi. Bab ini berisi analisa kebutuhan informasi dari perspektif manajer pemasaran sebagai pengguna, menentukan ruang lingkup kebutuhan informasi terhadap *medrep*, identifikasi pengguna, jenis informasi yang tersedia di perusahaan, serta disertai dengan wawancara dengan *medrep*, *supervisor*, *sales*, dan apotek sebagai metode untuk mencari data riil di lapangan.

Bab IV, Desain dan Perancangan Perangkat Lunak, pada bab ini berisi perancangan dan desain perangkat lunak termasuk penjelasan proses manajemen data informasi serta rancangan antarmuka untuk menampilkan hasil dari proses manajemen informasi.

Bab V, Uji Coba dan Evaluasi, Menjelaskan tentang skenario uji coba aplikasi, hasil uji coba, serta evaluasi terhadap aplikasi berdasarkan uji coba yang dilakukan.

Bab VI, Penutup, pada bab ini berisi tentang kesimpulan yang dapat diambil dari tugas akhir ini beserta saran untuk pengembangan di masa depan.



**BAB II**  
**DASAR TEORI**

## BAB II DASAR TEORI

Pada bab ini dibahas teori-teori yang mendukung pembuatan tugas akhir ini, yaitu pengertian tentang armada penjualan, pengukuran kinerja armada penjualan, Unified Modelling Language (UML), Java Server Pages (JSP) dan MySQL.

### 2.1 Armada Penjualan

Personalia penjualan (*sales personel*) berfungsi sebagai orang yang menjadi penghubung antara perusahaan dan pelanggan. Wakil penjualan (*Sales Representative*) adalah perusahaan itu sendiri bagi banyak pelanggannya dan wakil penjualanlah yang akan membawa banyak informasi mengenai pelanggan yang dibutuhkan perusahaan.

#### 2.1.1 Jenis Armada Penjualan

Istilah wakil penjualan (*Sales Representattive*) mencakup posisi yang luas. McMurry membedakan enam posisi penjualan, yang berkisar dari tipe penjualan yang paling tidak kreatif sampai yang paling kreatif.

1. **Pengirim** (*deliver*): Wiraniaga yang tugas utamanya adalah mengirimkan produk (misalnya susu, roti, bahan bakar, minyak).
2. **Penerima Pesanan** (*order taker*): Wiraniaga dalam perusahaan yang bertindak sebagai penerima pesanan (misalnya penjaga toko yang berdiri di belakang counter) atau sebagai penerima pesanan luar (misalnya wiraniaga sabun yang melakukan kunjungan singkat ke manajer toko swalayan).
3. **Pembawa Misi** (*missionary*): Wiraniaga yang tidak diharapkan atau tidak diizinkan untuk menerima pesanan tetapi hanya ditugaskan untuk membangun nama baik atau untuk mendidik pemakai aktual dan potensial (misalnya *detailer* medis yang mewakili perusahaan farmasi).

4. **Teknisi** (*Techician*): Wiraniaga dengan tingkat pengetahuan teknis yang tinggi (misalnya wiraniaga teknis yang bertugas sebagai konsultan pada perusahaan klien).
5. **Pencipta Permintaan** (*demand creator*): Wiraniaga yang mengandalkan metode-metode kreatif untuk menjual produk-produk yang berwujud (misalnya penyedot debu, lemari es) atau produk tak berwujud (misalnya asuransi, jasa periklanan).
6. **Penjual Solusi** (*solution vendor*): Wiraniaga yang mengkhususkan diri dalam memecahkan masalah pelanggan, seringkali berkaitan dengan sistem produk dan jasa perusahaan (misalnya sistem komputer dan komunikasi).

### 2.1.2 Tugas Armada Penjualan

Gagasan lama yang ada adalah bahwa armada penjualan harus "menjual, menjual dan menjual". Wiraniaga mempunyai kuota penjualan, dan wiraniaga yang lebih baik dapat memenuhi atau melebihi kuotanya. Kemudian timbul gagasan bahwa wiraniaga harus terampil dalam memecahkan masalah pelanggan, bahwa mereka harus mengerti bagaimana mendiagnosa masalah pelanggan, dan mengajukan solusinya. Wiraniaga pada awalnya tidak berusaha menjual produk tertentu atau memecahkan masalah tertentu. Sebaliknya, mereka menunjukkan kepada calon pelanggan bagaimana perusahaannya dapat membantu meningkatkan kemampuan pelanggan untuk menghasilkan laba. Mereka berusaha untuk menggabungkan perusahaannya dengan perusahaan pelanggan sebagai "mitra untuk mencapai laba". Terlepas dari konteks penjualan, wiraniaga memiliki satu atau beberapa tugas yang harus diselesaikan berikut ini :

1. *Mencari calon pembeli*: Wakil perusahaan mencari calon pembeli.
2. *Menetapkan Sasaran*: Wakil penjualan memutuskan seara mengalokasikan waktu ke para calon pembeli dan pelanggannya.



3. *Berkomunikasi*: Wakil penjualan mengkomunikasikan informasi tentang produk dan jasa perusahaan.
4. *Melayani*: Wakil penjualan memberikan berbagai pelayanan kepada pelanggan, mengkonsultasikan masalah, memberikan bantuan teknis, membantu pembiayaan, melakukan pengiriman.
5. *Menjual*: Wakil penjualan mendekati, mempresentasikan, menjawab keberatan-keberatan, dan menutup penjualan.
6. *Mengumpulkan informasi*: Wakil penjualan melakukan riset pasar dan tugas intelijen serta menyusun laporan kunjungan.
7. *Mengalokasikan*: Wakil penjualan memutuskan pelanggan mana yang akan memperoleh produk langka selama terjadi kelangkaan produk.

Perusahaan perlu mendefinisikan tujuan-tujuan khusus yang harus dicapai oleh armada penjualannya. Misalnya suatu perusahaan menginginkan wakil penjualannya menggunakan 80% waktunya untuk pelanggan yang sekarang dan 20% untuk calon pelanggan, serta 85% waktunya untuk produk yang sudah ada dan 15% untuk produk baru. Jika aturan-aturan tersebut tidak ditetapkan, wakil penjualan akan menggunakan sebagian besar waktunya untuk menjual produk yang mapan kepada pelanggan yang sekarang dan mengabaikan produk baru dan calon pelanggan baru.

### 2.1.3 Struktur Armada Penjualan

Strategi armada penjualan memiliki implikasi pada struktur armada penjualan. Misalnya jika sebuah perusahaan menjual satu lini produk kepada satu industri pengguna akhir dengan pelanggan di berbagai lokasi, perusahaan dapat menggunakan struktur armada penjualan teritorial. Secara garis besar ikhtisar struktur wiraniaga yang paling umum, yaitu :

1. **Teritorial**, tiap perwakilan penjualan diberikan satu daerah eksklusif. Struktur ini memiliki beberapa keunggulan. Pertama, memberikan definisi yang jelas atas tanggung jawab wiraniaga. Kedua, tanggung jawab teritorial mendorong wakil penjualan itu untuk mengembangkan

hubungan bisnis lokal dan ikatan pribadi. Ketiga, biaya perjalanan relatif kecil, karena tiap wakil penjualan hanya berkeliling dalam daerah geografis yang kecil.

*Ukuran Wilayah:* wilayah dapat dirancang supaya memberikan potensi dan beban kerja yang sebanding. Wilayah dengan potensi dan beban kerja yang sama memberikan peluang bagi masing-masing wakil penjualan untuk mendapatkan penghasilan yang sama, sehingga dapat digunakan perusahaan untuk mengevaluasi kinerja wakil penjualan.

*Bentuk Wilayah:* Rancangan bentuk wilayah harus mempertimbangkan lokasi dari hambatan-hambatan alam, kesesuaian dengan daerah tetangga, kelayakan transportasi dan lain-lain.

2. **Produk**, karena penting bagi wakil penjualan untuk mengetahui produk serta karena pentingnya pengembangan divisi produk dan manajemen produk, menyebabkan banyak perusahaan membentuk armada penjualannya berdasarkan lini produk. Spesialisasi produk terutama dibutuhkan untuk produk yang secara teknis rumit, sangat tidak berhubungan atau sangat banyak jenisnya.
3. **Pasar**, perusahaan sering menspesialisasi armada penjualan menurut industri atau lini pelanggan. Armada penjualan sendiri terbentuk untuk industri yang berbeda bahkan untuk pelanggan yang berbeda. Keuntungan paling jelas dari spesialisasi pasar adalah armada penjualan dapat sangat memahami kenutuhan tiap pelanggan. Kekurangan utama timbul jika pelanggan tersebar di seluruh negeri, sehingga membutuhkan banyak biaya perjalanan.
4. **Gabungan**, perusahaan menggunakan lini teritorial, produk dan pasar sekaligus untuk melakukan pemasaran dengan jumlah produk yang banyak dan dengan pelanggan yang tersebar.

## 2.2 Teori Pengukuran Kinerja

Ilmu sosial dan riset aplikasi bisnis sering memunculkan fenomena baru dalam dunia bisnis. Permasalahan yang seringkali muncul adalah bagaimana mengelola berbagai macam komponen bisnis menjadi sebuah nilai tunggal. Dalam beberapa kasus, beberapa komponen tersebut dinilai bersama dengan *range* dan dimensi yang sama, hal yang menyulitkan adalah mengambil nilai rata-rata dari tiap komponen. Sehingga seringkali proses peng-*index*-an menjadi sebuah proses studi perbandingan. Studi perbandingan membutuhkan banyak data statistik dan informasi berkualitas yang berhubungan dengan proses bisnis baik dalam maupun antar perusahaan. Sehingga saat ini, perbandingan menjadi jantung dari penilaian kinerja pada semua perusahaan di seluruh dunia.

### 2.2.1 Pengukuran Kinerja Secara Umum

Manajemen memperoleh informasi mengenai wakil penjualan dengan berbagai cara. Sumber yang paling penting adalah laporan penjualan. Informasi tambahan diperoleh dari pengamatan pribadi, surat dan keluhan pelanggan, survei pelanggan serta percakapan dengan wakil penjualan lainnya.

Saat ini banyak perusahaan yang mewajibkan wiraniaga mereka untuk membuat *rencana pemasaran wilayah* tahunan yang menjelaskan garis besar program mereka untuk mengembangkan pelanggan baru dan meningkatkan bisnis dari pelanggan yang sudah ada. Jenis laporan itu menempatkan wiraniaga sebagai manajer pasar atau pusat laba. Manajer mereka mempelajari rencana-rencana itu, memberikan usulan, dan menggunakannya untuk menentukan kuota penjualan.

Semakin banyak jumlah perusahaan yang mengukur kepuasan pelanggan tidak hanya dari produk dan pelayanan dukungan pelanggan, tetapi juga dengan wiraniaga mereka. Pendapat pelanggan tentang wiraniaga, produk dan pelayanan dapat diukur dengan kuisioner melalui surat atau telepon.

Evaluasi juga dapat menilai pengetahuan wiraniaga mengenai perusahaan, produk, pelanggan, pesaing, wilayah, dan tanggung jawab. Manajer penjualan juga dapat mempelajari masalah yang berkaitan dengan motivasi atau ketaatan wiraniaga terhadap hukum. Manajer terutama memperbandingkan kinerja armada penjualan melalui kontribusi penjualan wiraniaga baik terhadap wiraniaga lain, maupun dengan kontribusi penjualan yang mereka dapat pada periode sebelumnya.

### 2.2.2 Pengukuran Kinerja *Medrep*

Dalam dunia farmasi, khususnya obat-obatan *ethycal* terjadi kesulitan dalam hal mengevaluasi kinerja. Kesulitan tersebut muncul disebabkan banyaknya kriteria yang bisa digunakan, selain itu disebabkan adanya perbedaan antara *medrep*, konsumen, maupun kondisi bisnis yang ada. Pada umumnya penilaian kinerja *medrep* diukur berdasarkan nilai dari laba yang dihasilkan. Biasanya pihak manajer membuat satu diantara lima kesalahan berikut 1) manajer mengevaluasi *medrep* hanya berdasar volume penjualan. 2) manajer sangat bergantung pada jumlah kunjungan yang dilakukan tiap *medrep* kepada dokter. 3) manajer memperbandingkan nilai penjualan dari tiap *medrep* nilai penjualan sebelumnya pada periode yang sama, misal penjualan bulan Mei tahun ini dibandingkan dengan penjualan bulan Mei tahun lalu. 4) manajer meminta seluruh *medrep* untuk melakukan metode kunjungan yang sama seperti ketika terjadi lonjakan penjualan. 5) manajer memberi terlalu banyak kebebasan kepada *medrep*.

Kesalahan dari bagian pertama adalah bahwa volume penjualan sendiri tidak dapat menjelaskan berapa banyak keuntungan atau kerugian yang didapat dari seorang *medrep*. Hingga pihak manajer tahu fakta tersebut, *medrep* dapat meningkatkan biaya kunjungan tanpa disadari pihak manajer. Misalkan sebuah daerah yang memiliki beberapa *medrep* merugi sampai manajer menganalisa penjualan. Manajer menyadari bahwa salah satu dari *medrep* melakukan banyak visitasi tanpa didapat hasil yang signifikan dari dokter didikannya. *Medrep* ini berkonsentrasi pada

salah seorang dokter yang sudah identik dengan produk dari perusahaan farmasi lainnya, sehingga pada akhirnya mengurangi volume penjualan perusahaan.

Kesalahan dari bagian kedua adalah konsep dari visitasi itu sendiri. Melakukan kunjungan ke dokter adalah hal penting yang harus dilakukan *medrep*, tetapi seharusnya *medrep* dapat melakukan kunjungan potensial yang tidak hanya dapat meningkatkan volume penjualan dan laba perusahaan, tetapi juga menjadikan dokter sebagai konsumen loyal perusahaan. Melakukan penjadwalan kunjungan rutin per hari juga merupakan hal yang penting, perusahaan juga dapat memastikan berapa biaya yang dibutuhkan untuk membiayai kunjungan tiap hari. Namun membiayai seorang *medrep* untuk melakukan kunjungan rutin ke dokter tiap hari, sebagai contoh, dapat sangat mahal.

Kesalahan dari bagian ketiga adalah dapat menyebabkan terjadinya *misleading*, kesalahpahaman. Beberapa bulan memiliki sejumlah hari kerja yang berbeda antara satu dengan lainnya. Perubahan dari jenis produk, harga, kompetisi, dan *assignment* membuat perbandingan dengan periode sebelumnya tidak *fair*, baik kepada *medrep* maupun kepada manajer. Adalah hal yang lebih baik jika manajer mengukur kemajuan secara kumulatif – dua mingguan, bulanan, tiga bulanan, atau tahunan.

Kesalahan dari bagian keempat adalah jika suatu saat *medrep* melanggar kebijakan perusahaan atau tidak memenuhi target. Masalah lain juga muncul ketika manajer menunjukkan metode penjualan yang dimilikinya kepada para *medrep*. Adalah hal yang alamiah bahwa *medrep* ingin tahu metode penjualan yang baik, tapi sayangnya kondisi pemasaran berubah atau *medrep* menghadapi masalah yang berbeda. Misalnya ada seorang *medrep* yang introversi – yang menghindari bersosialisasi sebisa mungkin, yang mampu melakukan penjualan yang selalu melebihi target perusahaan. Sedangkan pimpinan *medrep* tersebut adalah orang yang extroversi dan dia tidak memahami apa yang terjadi dengan *medrep*-nya.

Kesalahan dari bagian terakhir disebabkan adanya keterbatasan pengetahuan mengenai penjualan dan pemasaran yang dimiliki manajer. Manajer kemudian memberikan kebebasan tiap *medrep* untuk mengembangkan metode penjualan yang

paling efektif untuk dirinya sendiri padahal pengetahuan memasarkan *medrep* sangat terbatas, bahkan beberapa manajer tidak mengetahui apa yang sebenarnya mereka inginkan dari *medrep*.

Pengukuran kinerja *medrep* adalah hal yang kompleks. Pengukuran kinerja *medrep* tidak hanya didasarkan pada besar volume penjualan mereka saja, dengan memperhatikan *market share*, teritorial, kualitas dan kuantitas visitasi yang *medrep* lakukan akan memberikan pandangan yang lebih baik terhadap kinerja *medrep*.

Saat ini dibutuhkan hubungan yang lebih kuat antara *medrep* dengan dokter. Seiring meningkatnya jumlah *medrep* dan ratusan obat baru yang tercipta setiap tahun, beban pengukuran kinerja *medrep* seakan mencapai puncaknya. Sebuah riset menunjukkan bahwa hampir 75% waktu dari *medrep* digunakan untuk kegiatan non penjualan, menunggu, melakukan perjalanan, dan melakukan kegiatan administratif lainnya. Dengan adanya sistem baru ini 75% kegiatan non penjualan tersebut akan berubah menjadi produktif, dengan lebih banyak aktifitas, sehingga *medrep* akan memiliki bekal yang lebih baik sebelum melakukan kunjungan ke dokter.

### 2.3 Unified Modelling Language (UML)

Saat ini dunia industri membutuhkan sebuah *framework* untuk mengukur dan menguji teknik perancangan. UML adalah standar untuk membuat sebuah model yang merepresentasikan *Object-Oriented Software* dan sistem bisnis. UML mengkombinasikan teknik-teknik penggambaran diagram yang dilakukan oleh para pengembang software selama 40 tahun terakhir.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

### 2.3.1 Definisi UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Kebutuhan adanya standardisasi metode telah menjadi kebutuhan berdasarkan berbagai alasan. Terdapat banyaknya metode dan notasi yang berkompetisi satu sama lain telah membingungkan pengguna dalam memutuskan metode mana yang harus dipakai. Sejak standarisasi metodologi menjadi isu, Object Management Group (OMG) telah mengeluarkan *Request for Proposal* (RFP) pada Juni 1996 untuk mendorong perusahaan-perusahaan sistem informasi, developer software, dan pengguna sistem komputer membuat sebuah RFP bersama. Satu perusahaan software, *Rational Software*, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling System* (UML) sebagai bahasa standar dalam OOAD. Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Kontribusi untuk UML telah dihasilkan dari perusahaan-perusahaan besar, diantaranya adalah Hewlett-Packard, IBM, Microsoft, Oracle, Rational Software, IntelliCorp, dan Electronic Data Services Corporation. Kolaborasi ini telah menjadikan UML sebagai bahasa pemodelan yang *well-defined*, ekspresif, tangguh, dan dapat digunakan secara luas.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

### 2.3.2 Konsep Dasar UML

Secara garis besar konsep dasar dari UML dapat dijelaskan pada tabel berikut:

Tabel 2.1 Konsep Dasar UML

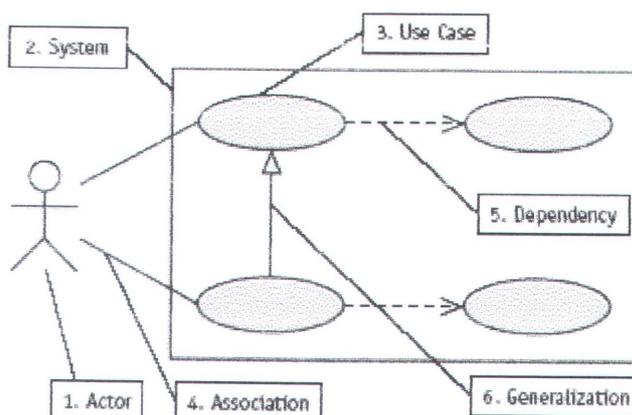
MAJOR AREA	VIEW	DIAGRAMS	MAIN CONCEPTS
Structural	Static view	Class diagram	Class, association, generalization, dependency, realization, interface
	Use case view	Use case diagram	Use case, actor, association, extend, include, use case generalization
	Implementation view	Component diagram	Component, interface, dependency, location
	Deployment view	Deployment diagram	Node, component, dependency, location
Dynamic	State machine view	State chart diagram	State, event, transition, action
	Activity view	Activity diagram	State, activity, completion transition, fork, join
	Interaction view	Sequence diagram	Interaction, object, message, activation
		Collaboration diagram	Collaboration, interaction, collaboration role, message
Model management	Model management view	Class diagram	Package, subsystem, model
Extensibility	all	all	Constraint, stereotype, tagged values

Dari tabel sebelumnya diketahui bahwa UML merupakan turunan dari beberapa metode yang mempunyai kumpulan diagram grafis sebagai kombinasi dari konsep pemodelan data (*entity relationship diagram*), pemodelan bisnis (*work flow*), pemodelan objek, dan pemodelan komponen. Diagram grafis tersebut merupakan tampilan dari beberapa level abstraksi yang dapat digunakan secara bersama oleh semua proses pada seluruh *lifecycle* pengembangan *software* serta pada implementasi ke beberapa teknologi yang berbeda.

### 2.3.3 Use Case Diagram

*Use case diagram* berisi mengenai interaksi antara sekelompok proses dengan sekelompok aktor, menggambarkan fungsionalitas dari sebuah sistem yang dibangun dan bagaimana sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menangkap kebutuhan sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

*Use Case diagram* terdiri dari 6 elemen pemodelan utama, yaitu *System*, *Actor*, *Use Case*, *Association*, *Dependency*, dan *Generalization*.



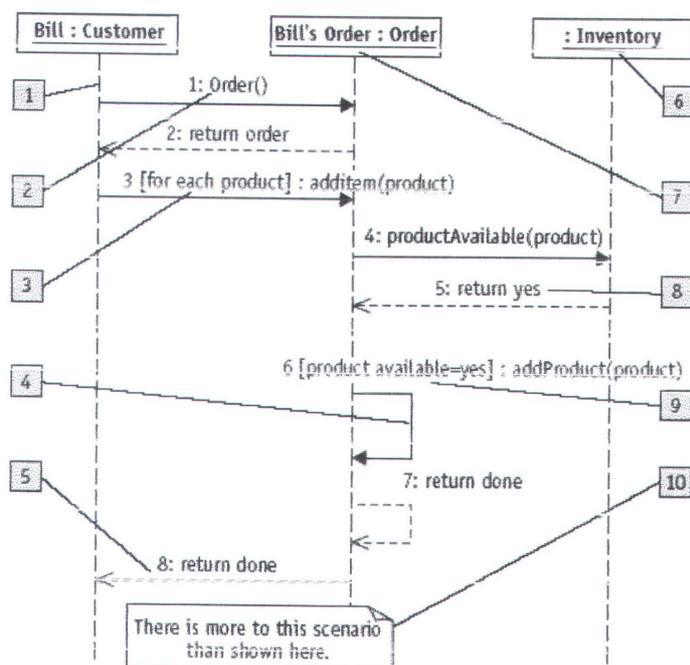
Gambar 2.1 Unsur-unsur *Use case Diagram*

1. **System**, sebuah diagram yang membatasi hubungan antara *Actor* yang menggunakan sistem itu (di luar sistem) dengan fitur yang disediakan sistem (di dalam sistem).
2. **Actor**, menggambarkan pengguna software aplikasi (*user*) yang bisa berupa orang, sistem lain, maupun perangkat lainnya yang memiliki peran dalam berjalannya sistem.
3. **Use Case**, menggambarkan fitur utama dari sebuah sistem, tanpa fitur ini, sistem akan gagal/tidak mampu memenuhi kebutuhan aktor (*user*). Setiap *Use Case* menunjukkan tujuan yang harus dicapai sistem.

4. **Association**, menunjukkan hubungan antara aktor dengan *Use Case*. Setiap *Association* akan menjadi dialog yang harus dijelaskan dalam naratif *Use Case*. Setiap naratif akan menjadi sebuah set skenario yang berfungsi menjadi semacam *test case* ketika mengevaluasi analisa, desain, dan implementasi dari *Use Case*.
5. **Dependency**, menggambarkan hubungan komunikasi antara dua *Use Case*.
6. **Generalization**, menunjukkan hubungan antara dua aktor atau dua *Use Case*, dimana properti dari sebuah *Use Case* diturunkan dan menambah atau meng-*override* properti *Use case* yang lain.

### 2.3.4 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang disusun dalam suatu urutan waktu. Secara khusus, diagram ini berasosiasi dengan *use case*.



Gambar 2.2 Contoh *Sequence Diagram*

1. **Object Lifeline**, adalah garis putus-putus vertikal yang terletak di bawah object yang menggambarkan dimensi waktu.
2. **Message/Stimulus**, biasanya berupa panggilan, sinyal, atau respon. Digambarkan sebagai panah. Jenis panah menggambarkan jenis *message*. Panah lurus menunjukkan *message* yang membutuhkan respon, sedangkan panah putus-putus menunjukkan respon. *Message* diletakkan berurutan dari atas ke bawah berdasar waktu terjadinya.
3. **Iteration**, menunjukkan bahwa operasi harus dilakukan berulang. Gunakan kurung siku untuk menunjukkan banyaknya perulangan atau kondisi yang mengontrol perulangan.
4. **Self Reference**, menggambarkan operasi yang *me-reference* pada dirinya sendiri.
5. **Return**, menunjukkan respon terhadap *message*. Beberapa orang tidak menggambarkannya. Padahal tujuan dari pemodelan adalah menampilkan informasi, tidak membuat asumsi. *Return* dibutuhkan untuk mengetahui bahwa hasil operasi sesuai dengan yang diminta pada *message*.
6. **Anonymous Object**, menunjukkan Object yang tidak bernama.
7. **Object Name**, menunjukkan nama dari tiap object.
8. **Sequence Number**, adalah pilihan, namun sangat berperan ketika berdiskusi atau akan mengubah diagram.
9. **Condition**, kurung siku juga dapat menunjukkan jika *message* telah dikirim. Seperti pada *step 6* yang menguji (*product available = yes*) untuk memastikan bahwa *step* sebelumnya telah dijalankan sebelum melakukan operasi di *step 6*.
10. **Basic Comment**, digunakan untuk memberikan informasi tambahan yang bukan bagian dari notasi.

### 2.3.5 Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.3 Contoh Activity Diagram

## 2.4 Java Server Pages (JSP)

JSP adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, serta merupakan bagian teknologi J2EE (*Java 2 Enterprise Edition*). JSP sangat sesuai dan tangguh untuk menangani presentasi di web. Sedangkan J2EE merupakan platform Java untuk pengembangan sistem aplikasi enterprise dengan dukungan API (*Application Programming Interface*) yang lengkap dan portabilitas serta memberikan sarana untuk membuat suatu aplikasi yang memisahkan antara *business logic* (sistem), presentasi dan data. JSP merupakan bagian dari J2EE dan secara khususnya merupakan komponen *web* dari aplikasi J2EE secara keseluruhan. Sedangkan teknologi J2EE secara keseluruhan adalah:

- *Java Server Pages*
- Servlets
- Dukungan Java terhadap XML yaitu JAXP
- *Enterprise Java Beans* (EJB)
- *Java Messaging*
- *Java Transaction*
- *Java Mail*
- *Java Naming and Directory Interface*
- JDBC (*Java Database Connectivity*)
- Dukungan Java terhadap CORBA dan RMI

Beberapa alasan digunakannya JSP sebagai bahasa scripting untuk membangun web site antara lain adalah:

- *Multi platform*, sesuai dengan semboyan Java yang juga multi platform dengan portabilitas pada semua mesin.
- Komponen dapat digunakan kembali (*reusable*) oleh JavaBean dan EJB.
- Kemampuan yang dimiliki oleh Java.



JSP memerlukan JVM (*Java Virtual Machine*) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal JVM di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut *Web Container*.

JVM adalah *software* yang berfungsi untuk menerjemahkan program Java agar dapat dimengerti komputer. Sebagaimana bahasa pemrograman lain, Java merupakan program yang dikompilasi dan diinterpretasikan. Dalam Java, proses kompilasi hanya terjadi sekali sedangkan interpretasi terjadi setiap program dijalankan. Untuk memiliki JVM di komputer kita, maka kita perlu *men-download* JDK (*Java Development Kit*), karena kita memerlukan class-class API untuk *development*.

Sedangkan *web container* adalah *services* yang dijalankan oleh suatu *Java Application Server* khususnya untuk servis yang kompatibel dengan servlet dan JSP. *Apache Tomcat* merupakan *web container* yang digunakan pada referensi pengimplementasian resmi dari teknologi *Java Servlet* dan *Java Server Pages*. Spesifikasi *Java Servlet* dan *JavaServer Pages* dikembangkan oleh Sun dibawah *Java Community Process*. *Apache Tomcat* dikembangkan pada lingkungan yang terbuka dan partisipatif serta dikeluarkan dibawah pengawasan *Apache Software License*. *Apache Tomcat* dimaksudkan untuk menjadi kolaborasi pengembangan terbaik dari pengembang perangkat lunak dari berbagai belahan bumi.

#### 2.4.1 Tag Deklarasi

Digunakan untuk mendefinisikan variabel dan *method*. Jangkauan deklarasi dalam file JSP adalah meliputi file lain yang diinclude dalam file JSP tersebut. Sintaks untuk deklarasi variabel adalah sebagai berikut:

```
<%! TipeData nama_variabel; %>
```

Contohnya adalah sebagai berikut:

```
<%! String nama_user; %>
```

Sintaks untuk deklarasi variabel adalah sebagai berikut:

```
<%! [modifier] TipeData NamaMethod(variabel){} %>
```

Contohnya adalah sebagai berikut:

```
<%! public int hitung(int x){ x = x + 1; return x;}; %>
```

### 2.4.2 Tag Ekspresi

Digunakan untuk mengevaluasi hasil ekspresi kemudian dikonversikan menjadi String. Tag ini dapat disisipkan ke dalam kode HTML. Sintaksnya adalah sebagai berikut:

```
<%= ekspresi %>
```

```
<html>
  <head>
    <title>Contoh halaman JSP</title>
  </head>
  <body>
    Selamat datang, Anda adalah :
    <%= request.getParamater(user) %>
  </body>
</html>
```

Gambar 2.4 Contoh Tag Ekspresi JSP

### 2.4.3 Tag Scriptlet

Digunakan untuk menulis kode JSP dalam JSP yang bercampur dengan HTML. Sintaksnya adalah sebagai berikut:

```
<% statement;%>
```

```

<html>
  <head>
    <title>Contoh halaman JSP</title>
  </head>
  <body>
    <% java.util.Date tgl = new java.util.Date(); %>
    Sekarang tanggal : <%= tgl.getDate()+-
      +tgl.getMonth()+-+tgl.getYear() %>
  </body>
</html>

```

Gambar 2.5 Contoh Tag Scriptlet

#### 2.4.4 Directives

*Directives* adalah pengiriman pesan pada JSP *container*, akan tetapi *directives* ini tidak melakukan cetak pada halaman. Sintaks *directives* adalah sebagai berikut:

```
<%@ directive {attribut=nilai} %>
```

*Directives* memiliki banyak jenis, akan tetapi *directives* yang paling penting ada beberapa yang harus diingat dalam pembangunan situs menggunakan JSP.

*Directive* tersebut antara lain:

- **page:** menghubungkan dependensi atribut halaman dan menghubungkannya pada *container* JSP. Contohnya adalah sebagai berikut: `<%@ page import=java.util.*%>`
- **include:** digunakan untuk memasukkan teks atau kode dalam halaman JSP. Contohnya adalah sebagai berikut:  
`<%@ include file=header.html %>`
- **taglib:** menjelaskan mengenai tag *library* yang harus diterjemahkan oleh JSP. Contohnya adalah sebagai berikut  
`<%@ taglib uri=mytags prefix=codecamp %>`

Tujuan dari penggunaan directive adalah memberikan informasi tingkat tinggi pada servlet tentang hasil dari halaman JSP. Adapun beberapa pengaturan terhadap halaman JSP yang perlu diingat adalah:

- `import class` dari Java untuk menunjang fungsi JSP. Contohnya adalah sebagai berikut:

```
<%@ page import=java.util.* %>
```

- Pengenal MIME (*Multipurpose Internet Mail Extension*). MIME adalah suatu standar mengenai dokumen yang dapat diakses dalam internet. Contohnya adalah sebagai berikut:

```
<%@ page contentType=MIME-Type %>
```

- Pengaturan `thread`. Secara otomatis akan diatur sebagai berikut:

```
<%@ page isThreadSafe=true %>
```

- Pengaturan halaman jika terjadi kesalahan/*error* pada JSP. Contohnya adalah sebagai berikut:

```
<%@ page errorPage=errorpage.jsp %>
```

#### 2.4.5 Obyek Implisit

Dalam pengembangan aplikasi menggunakan Java, para *programmer* dapat menggunakan Javabeans untuk berinteraksi dengan objek Java. Ada beberapa objek yang tersedia dalam JSP yang dikenal sebagai obyek implisit. Obyek implisit yang dimiliki JSP dapat dilihat pada berikut:

Tabel 2.2 Obyek Implisit pada JSP

Variabel	Tipe dari
Request	<code>Javax.servlet.http.HttpServletRequest</code>
Response	<code>Javax.servlet.http.HttpServletResponse</code>
Out	<code>Javax.servlet.jsp.JspWriter</code>
Session	<code>Javax.servlet.http.Session</code>

Variabel	Tipe dari
Pagecontent	Javax.servlet.jsp.httpcontext
Application	Javax.servlet.http.servletcontext
Config	Javax.servlet.http.servletconfig
Page	Java.lang.object
Exception	Java.lang.throwable

### 2.4.6 Komentar

Komentar digunakan untuk memberikan catatan khusus pada kode JSP untuk mengetahui apa yang sedang dilakukan, pemberian komentar adalah opsional. Maksudnya adalah boleh diberikan boleh tidak. Berikut adalah contoh sintaks dari komentar.

```
<%-- komentar --%>
<% /** komentar **/ %>
```

Gambar 2.6 Contoh Komentar pada JSP

## 2.5 MySQL

### 2.5.1 Definisi MySQL

MySQL adalah sebuah sistem manajemen *database*. *Database* adalah sekumpulan data yang terstruktur. Untuk menambah, mengakses, dan memproses data yang tersimpan dalam *database* komputer, dibutuhkan sebuah sistem *database* manajemen seperti MySQL. Sejak komputer menjadi alat yang sangat bagus untuk menangani sejumlah besar data, manajemen *database* memerankan peranan penting di dalam pengolahan data, sebagai utiliti yang stand-alone atau sebagai bagian dari suatu aplikasi.

MySQL adalah sebuah sistem manajemen *database*. Sebuah *database* menyimpan data didalam tabel yang terpisah daripada menyimpan semua data didalam satu ruang penyimpanan yang besar. Cara seperti ini akan menambah kecepatan dan *exibility*. Tabel-tabel ini terhubung oleh relasi yang telah didefinisikan sehingga memungkinkan untuk menghubungkan data dari beberapa tabel sesuai

permintaan. SQL yang merupakan bagian dari MySQL, merupakan kependekan dari “*Structured Query Language*” bahasa standar yang digunakan untuk mengakses *database*.

MySQL adalah sebuah *software* yang *Open Source*. Arti dari *open source* disini berarti MySQL bebas dipakai dan dimodifikasi oleh semua orang. Setiap orang dapat men-*download* MySQL dari internet dan menggunakannya tanpa perlu membayar. Setiap orang juga dapat mempelajari *source code*-nya dan merubahnya sesuai keinginannya. Ini menjadi nilai tambah tersendiri bagi *database* MySQL.

### 2.5.2 Fitur MySQL

Berikut ini adalah beberapa karakteristik penting dari MySQL :

- Mendukung penuh *multi-threaded* dengan menggunakan *thread* dari kernel. Ini juga memungkinkan untuk dengan mudahnya menggunakan banyak CPU.
- Didukung sepenuhnya oleh bahasa pemrograman C, C++, Eiffel, Java, Perl, PHP, Python dan Tcl untuk mengakses *database* MySQL.
- Dapat bekerja pada banyak platform berbeda, termasuk juga windows.
- Banyak tipe kolom : signed/unsigned integer 1, 2, 3, 4, dan 8 bytes, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, dan tipe ENUM.
- Dapat melakukan join dengan cepat menggunakan optimisasi *one-sweep multi-join*.
- Didukung berbagai macam operator dan fungsi untuk melakukan *query* dalam perintah SELECT dan WHERE. Contoh :
 

```
SELECT CONCAT (first_name, " ", last_name) FROM tbl_name
WHERE income/dependents > 10000 AND age > 30;
```
- Fungsi SQL diimplementasikan dengan optimisasi tinggi pada class library dan mencapai kecepatan yang maksimal. Biasanya tidak perlu ada pengalokasian memori sama sekali setelah inisialisasi *query*.

- Mendukung sepenuhnya parameter SQL GROUP BY dan ORDER BY. Fungsi yang dapat dipakai dalam group *query* : (COUNT (), COUNT (DISTINCT), AVG (), STD (), SUM (), MAX () and MIN ()).
- Mendukung LEFT OUTER JOIN dan RIGHT OUTER JOIN dengan perintah ANSI SQL dan ODBC.
- Dapat dilakukan penggabungan tabel dari beberapa *database* yang berbeda dalam satu *query*.
- Sistem *privilege* dan *password* dapat terjaga kerahasiaannya dapat di verifikasi berdasarkan nama *host*-nya. *Password* terjaga kerahasiaannya karena semua *password* disimpan dalam keadaan terenkripsi.
- Mendukung ODBC (*Open-Database-Connectivity*) untuk Win32 (*with source*). Sehingga, dimungkinkan untuk connect ke MySQL *server* menggunakan MS Access.
- Tabel dengan struktur B-tree dan dengan indeks yang terkompresi memungkinkan akses dengan kecepatan tinggi.
- Indeks dapat dilakukan hingga 32 indeks per tabel. Dengan tiap indeks terdiri atas 1 sampai 16 kolom. Batas panjang maksimum dari indeks adalah 500 byte yang dapat diubah dengan meng-compile ulang MySQL.
- MySQL dapat menyimpan tabel dalam *database* secara terpisah pada direktori *database*. Ukuran maksimum dari tabel adalah antara 4 GB sampai batas maksimum yang diperbolehkan oleh sistem operasi untuk sebuah *file*.

### 2.5.3 Tipe data MySQL

Tabel 2.3 Tipe Data MySQL

Tipe Data	Panjang / Jangkauan
TINYINT [ (M) ]	0 sampai 255. -128 sampai 127
SMALLINT[ (M) ]	0 sampai 65535. -32768 sampai 32767
MEDIUMINT[ (M) ]	0 sampai 16777215 -8388608 sampai 8388607

<b>Tipe Data</b>	<b>Panjang / Jangkauan</b>
<i>INT</i> [ (M) ]	0 sampai 4294967295 -2147483648 sampai 2147483647
<i>BIGINT</i> [ (M) ]	0 sampai 18446744073709551615 -9223372036854775808 sampai 9223372036854775807
<i>FLOAT</i> [ (M,D) ]	-3.402823466E+38 sampai 1.175494351E-38
<i>DOUBLE</i> [ (M,D) ]	-1.7976931348623157E+308 sampai 2.2250738585072014E-308
<i>DECIMAL</i> [ (M[,D]) ]	-1.7976931348623157E+308 sampai 2.2250738585072014E-308
<i>DATE</i>	'1000-01-01' sampai '9999-12-31'
<i>DATETIME</i>	'1000-01-01 00:00:00' sampai '9999-12-31 23:59:59'
<i>TIMESTAMP</i> [ (M) ]	'1970-01-01 00:00:00' sampai tahun 2037
<i>TIME</i>	'-838:59:59' sampai '838:59:59'
<i>YEAR</i> [ (2 4) ]	1901 sampai 2155
<i>TINYBLOB</i> / <i>TINYTEXT</i>	255 huruf
<i>BLOB</i> / <i>TEXT</i>	65535 huruf
<i>LOB</i> / <i>LONGTEXT</i>	4294967295 huruf
<i>ENUM</i> ('nilai', 'nilai',...)	Maksimal : 65535 nilai yang berbeda
<i>SET</i> ('nilai', 'nilai',...)	Maksimal : 64 anggota

Kode huruf yang digunakan di dalam sintaks :

- *M* menunjukkan ukuran maksimum yang dapat ditampilkan. Nilai maksimum yang dapat ditampilkan adalah 255.
- *D* dipakai untuk tipe data floating-point dan menunjukkan jumlah digit desimalnya. Nilai maksimum yang dapat dipakai adalah 30, tetapi tidak melebihi dari M-2.

#### 2.5.4 Structure Query Language (SQL) untuk MySQL

Bahasa SQL adalah bahasa standar yang digunakan untuk memodifikasi struktur dari *database* dan untuk mengakses atau mengupdate informasi dari *database* relasi.. Untuk mengetahui deskripsi statemen SQL selengkapnya, dapat dilihat pada dokumentasi SQL dari *database* yang kita gunakan. Hal yang dapat dilakukan oleh seorang *user* maupun administrator terhadap suatu *database* dengan menggunakan SQL, adalah :

- Mengubah struktur sebuah tabel
- Mengatur *security* dari sistem
- Memberikan hak akses kepada *user* untuk mengakses *database* atau tabel
- Membaca data dari tabel yang ada pada *database*
- Meng-*update* data dari tabel yang ada pada *database*

Perintah-perintah SQL dikelompokkan menjadi 5 (lima) macam :

##### 2.5.4.1 Data Definition Language (DDL)

Adalah perintah SQL yang digunakan untuk menjelaskan atau mendefinisikan struktur dari *database*. Perintahnya adalah :

- *Create* : Untuk membuat objek *database*
- *Alter* : Untuk memodifikasi objek *database*
- *Drop* : Untuk menghapus objek *database*

objek *database* yang dimaksud adalah *database*, tabel, indeks dan view

##### 2.5.4.2 Data Manipulation Language (DML)

Adalah perintah yang digunakan untuk memanipulasi isi *database*. SQL menyediakan empat perintah DML, yaitu :

- *Select* : Digunakan untuk mengambil data dari *database*
- *Delete* : Digunakan untuk menghapus data dari *database*

- *Insert* : Menambahkan data ke *database*
- *Update* : Memodifikasi data pada *database*

#### 2.5.4.3 Security

Adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Perintah tersebut adalah :

- *Grant* : Memberi hak akses kepada *user* tertentu untuk mengakses ke *database*
- *Revoke* : Mencabut hak akses dari *user*

#### 2.5.4.4 Integrity

Adalah perintah-perintah yang digunakan untuk menjaga keutuhan data. Contoh : *Recover Table* : untuk memperbaiki tabel pada *database*.

#### 2.5.4.5 Auxiliary

Adalah perintah-perintah pelengkap atau tambahan seperti : *unload* dan *rename*. Pada bab ini, bahasa SQL yang dibahas adalah bahasa SQL yang dikhususkan untuk MySQL.



**BAB III**  
**ANALISA KEBUTUHAN SISTEM**

### **BAB III**

#### **ANALISA KEBUTUHAN SISTEM**

Pada bab ini akan dibahas perusahaan yang menjadi tempat penelitian dalam Tugas Akhir ini, yaitu PT. Kalbe Farma Surabaya. Profil. dan strategi bisnis perusahaan, model proses bisnis dan struktur manajemen dalam fungsi pemasaran, serta peranan teknologi informasi dalam mendukung proses bisnis perusahaan.

#### **3.1 PT. Kalbe Farma**

Kalbe Farma (“Kalbe”) didirikan 38 tahun yang lalu atau tepatnya pada tanggal 10 September 1966 dengan tujuan ikut berpartisipasi dalam meningkatkan kesehatan masyarakat Indonesia secara luas. Selama masa 38 tahun beroperasi, Kalbe telah menjalani berbagai macam rute dan berbagai fluktuasi bisnis seiring dengan perubahan situasi makro Indonesia saat itu, seperti misalnya perubahan portofolio bisnis, perubahan peraturan perundang-undangan, devaluasi nilai rupiah dan terakhir krisis keuangan pada tahun 1997. Masa baik dan sulit telah dijalani oleh Kalbe dengan sukses sehingga mampu terus menerus meningkatkan nilai shareholders value perusahaan.

Bisnis Kalbe sekarang ini tetap fokus pada bisnis farmasi, makanan kesehatan dan kemasan untuk produk farmasi dan produk konsumen. Dari segi nilai penjualan tahun 2004, bisnis farmasi dan makanan kesehatan memberikan kontribusi sekitar 94%. Aktivitas ketiga bisnis tersebut dikerjakan oleh perusahaan termasuk Kalbe sebagai perusahaan induk. Diantara perusahaan tersebut terdapat 2 anak perusahaan yang telah mencatatkan sahamnya di bursa yaitu PT Dankos Laboratories Tbk. dan PT Kageo Igar Jaya Tbk.

Dalam hal sumber daya manusia, secara konsolidasi Kalbe memiliki sekitar 8100 karyawan yang terdiri dari sekitar 2000 di pemasaran, 5000 di pabrikan dan sekitar 1100 di kantor.

Dalam hal kualitas, Kalbe dan seluruh anak perusahaan yang mempunyai pabrik telah mendapatkan sertifikasi ISO 9001 : 2000. Khusus perusahaan farmasi ditambah sertifikasi Cara Pembuatan Obat yang Baik (CPOB) , dan dalam tahun 2004 ada 3 perusahaan telah mendapatkan sertifikasi ISO 14001 dan OHSAS 18001/SMK3 (Standard Keselamatan Kerja) yaitu Kalbe Farma sebagai Induk perusahaan, Dankos Laboratories dan Bintang Toedjoe.

Pada tanggal 30 Juli 1991, Kalbe mencatatkan sahamnya di Bursa Efek Jakarta dan Surabaya agar publik dapat ikut berpartisipasi memiliki saham Kalbe tersebut. Kepemilikan public sampai saat ini mencapai sekitar 47,3% dengan diwakili oleh sekitar 2250 pihak baik individual maupun institusi dari total saham yang beredar sebanyak 8.121.600.000 saham.

Sebagai salah satu perusahaan tercatat dan dalam kerangka Tata Kelola Perusahaan yang Baik (Good Corporate Governance), Kalbe wajib menjalankan seluruh ketentuan peraturan yang berlaku di pasar modal Indonesia secara konsisten, termasuk di dalamnya ketentuan tentang keterbukaan informasi material kepada publik dan pelaporan kinerja keuangan secara periodik.

### **3.1.1 Visi dan Misi**

Seiring dengan perkembangan kondisi pasar domestik maupun global, maka dalam tahun 2004, Manajemen melakukan penyesuaian atas misi, visi dan core values Kalbe yang disusun di tahun 2000. Misi, Visi dan Core Values yang baru ini akan menjadi pedoman menuju tahun 2010.

- MISI : Meningkatkan Kesehatan untuk Kehidupan yang lebih baik
- VISI : Menjadi Perusahaan yang Dominan dalam bidang kesehatan di Indonesia dan memiliki eksistensi di pasar global dengan merek dagang yang kuat, didasarkan oleh Manajemen, Ilmu dan Tehnologi yang unggul.

## CORE VALUES :



Gambar 3.1 Core Values PT. Kalbe Farma

### 3.1.2 Divisi Ethical PT. Kalbe Farma

Obat resep sangat berbeda dengan obat bebas dimana dalam obat resep meliputi lebih banyak kategori penyakit yang membutuhkan konsultasi dengan dokter. Para pasien harus mendapatkan resep dari dokter sebelum membeli obat tersebut di apotik atau rumah sakit.

Untuk memenuhi kebutuhan penyembuhan berbagai penyakit, sampai hari ini Kalbe memiliki sekitar 350 obat resep yang di pasarkan oleh sekitar 1800 medical representative di Indonesia dan luar negeri. Kekuatan *team medical representative* ini adalah jaringan terbesar di Indonesia.

Kontribusi penjualan masing-masing produk dalam segmen ini lebih merata sehingga kalau diambil 25 obat resep utama hanya memberikan kontribusi sekitar 50% dari total penjualan obat resep. Beberapa obat resep yang utama adalah BROADCED, CRAVIT, BACTECYN, CEFSPAN, CEFOTAXIME dari golongan antibiotika, ANGIOTEN dari golongan cardiovascular, NEUROTAM, NEUROCET, NEURALGIN dari golongan pusat susunan syaraf, KALTROFEN dari golongan rheumatism, FARMORUBICIN dari golongan oncology dan HEPASIL dari golongan hepatologi.

Di tahun 2004, obat resep membukukan penjualan sebesar Rp. 977 miliar atau tumbuh 20,1% dari tahun 2003 yang berjumlah Rp.813,2 miliar. Pertumbuhan ini berasal dari peningkatan volume penjualan sekitar 15% dan dari kenaikan harga.

Kalbe mulai memberikan service dalam bentuk informasi kesehatan kepada profesi kesehatan dan public melalui website dan berbagai seminar, sebagai pendukung kepada para Medical Representative yang secara proaktif melakukan kunjungan ke institusi kesehatan termasuk para dokter.

### **3.2 Pengukuran Kinerja di PT. Kalbe Farma**

Sistem pengukuran kinerja *medrep* yang akan dibangun dalam tugas akhir ini berdasarkan hasil pengamatan di PT. Kalbe Farma Surabaya bagian *Ethycal*. Di Kalbe Surabaya sendiri adalah kantor cabang terbesar kedua setelah kantor pusat di Jakarta.

Untuk pengukuran penjualan selama ini masih menggunakan patokan catatan penjualan dari Enseval selaku distributor penjualan obat-obatan. Pihak Enseval menghasilkan Daftar Penjualan Unit (DPU) yang berisi catatan volume penjualan untuk masing-masing divisi dari Kalbe yaitu divisi *Ethycal*, OTC, makanan kesehatan, dan makanan ternak. Nantinya masing-masing divisi akan melakukan rekapitulasi dan dibandingkan dengan target penjualan yang dibuat masing-masing divisi.

#### **3.2.1 Pembagian Region**

PT. Kalbe Farma Surabaya membagi wilayah kerja di Surabaya menjadi tiap 13 region, dimana tiap region terdiri dari 2 sampai 3 kecamatan. Pembagian region didasarkan pada luas geografis wilayah dan kedekatan lokasi. Tiap divisi memiliki 13 supervisor atau pemimpin yang bertanggung jawab atas sebuah region. Rincian pembagian region dapat dilihat pada tabel di halaman berikut.

Tabel 3.1 Pembagian Region

Region	Kecamatan	Region	Kecamatan	Region	Kecamatan
I	Pabean Cantikan	VI	Wiyung	X	Dukuh Pakis
	Semampir		Karang Pilang		Sukomanunggal
	Krembangan	VII	Gayungan	XI	Gubeng
Kenjeran	Jambangan		Tegalsari		
II	Simokerto	VIII	Wonocolo	XII	Genteng
	Asemrowo		Tenggilis Mejoyo		Bubutan
III	Tandes	IX	Rungkut	XIII	Sawahlan
	Pakal		Gunung Anyar		Wonokromo
IV	Benowo	X	Sukolilo	XI	Bulak
	Sambikerep		Mulyorejo		Tambaksari
V	Lakarsantri				

### 3.2.2 Volume Penjualan

Volume penjualan dari tiap-tiap divisi yang dihasilkan dalam DPU didapat dari akumulasi penjualan masing-masing obat di tiap apotek. DPU berisi (1) nilai penjualan tiap apotek yang didapatkan dari akumulasi nilai penjualan per obat di tiap apotek dan (2) nilai penjualan untuk tiap jenis obat yang berasal dari akumulasi penjualan sebuah obat di semua apotek. Berikut adalah formulasi yang digunakan untuk menghitung volume penjualan di tiap apotek.

$$V_1 = (V_{1a} * H_a) + (V_{1b} * H_b) + \dots + (V_{1n} * H_n)$$

Dengan :

- $V_1$  = Volume penjualan di apotek "1"
- $V_{1a}$  = Jumlah penjualan obat "a" di apotek "1"
- $H_a$  = Harga penjualan obat "a"

Gambar 3.2 Formulasi perhitungan volume Penjualan

Untuk kepentingan pihak manajer, DPU juga menampilkan total nilai penjualan sebuah obat di seluruh apotek. Fungsi ini digunakan pihak manajer untuk mengevaluasi perkembangan penjualan sebuah obat. Sehingga nantinya dapat digunakan pihak manajer untuk membantu membuat keputusan pemasaran. Untuk menghitung nilai penjualan dari sebuah obat digunakan fungsi `sum()` kemudian

dikalikan dengan harga penjualan satuan obat. Berikut adalah formulasi yang digunakan

$$V_a = (V_{1a} + V_{2a} + \dots + V_{na}) * H_a$$

Dengan :

- $V_a$  = Nilai penjualan obat "a"
- $V_{na}$  = Jumlah penjualan obat "a" di apotek "n"
- $H_a$  = Harga Satuan Obat

Gambar 3.3 Formulasi perhitungan penjualan obat

DPU dari Enseval berisi daftar penjualan obat di seluruh wilayah kerja Kalbe Surabaya. Sedangkan di wilayah kerja Surabaya dibagi menjadi 13 region. Tugas bagian *Ethycal (Supervisor)* adalah melakukan analisa penjualan di tiap region. berikut adalah contoh penghitungan volume penjualan di tiap region beserta tabel region.

Dari DPU

$V_1$	= Rp 1.000.000	$V_{1a}$	= Rp 250.000	$V_{1b}$	= Rp 100.000
$V_2$	= Rp 2.600.000	$V_{2a}$	= Rp 1.100.000	$V_{2b}$	= Rp 125.000
$V_3$	= Rp 900.000	$V_{3a}$	= Rp 125.000	$V_{3b}$	= Rp 50.000
$V_4$	= Rp 1.450.000	$V_{4a}$	= Rp 400.000	$V_{4b}$	= Rp 0

Dokumentasi supervisor

Supervisor : "X"

Region : I

Tanggal : 9 Nopember 2005

Apotek	Nilai penjualan	Obat "a"	Obat "b"
1	Rp 1.000.000	Rp 250.000	Rp 100.000
2	Rp 2.600.000	Rp 1.100.000	Rp 125.000
4	Rp 1.450.000	Rp 400.000	Rp 0

### 3.2.3 Kuota Penjualan

Kuota penjualan adalah target penjualan yang diberikan pihak *supervisor* kepada tiap *medrep* yang berada di bawah regionnya. Kuota penjualan diinisialisasi oleh pihak *supervisor*. Parameter-parameter yang digunakan untuk menghitung kuota penjualan adalah (1) jumlah penduduk di region (2) jenis dari region dan (3) jumlah dokter yang ada di region. Parameter-parameter tersebut dikelompokkan menurut klasifikasinya sebelum dikalikan dengan bobot perhitungan yang telah dibuat bersama oleh pihak manajer dan *supervisor*. Berikut adalah klasifikasi dari masing-masing parameter.

Tabel 3.2 Klasifikasi jumlah penduduk region

Jumlah Penduduk di region	Nilai
0-100.000	5
100.000-200.000	10
200.000-300.000	15
> 300.000	20

Tabel 3.3 Klasifikasi Jenis region

Jenis Region	Nilai
Daerah Perifer	5
Daerah Pendukung	10
Tengah kota	15

Tabel 3.4 Klasifikasi Jumlah Dokter di Region

Jumlah Dokter di region	Nilai
0-45	5
46-90	10
91-135	15
> 135	20

$$KPa = (\Sigma \text{Penduduk} * b_1) + (\text{Jenis Region} * b_2) + (\Sigma \text{Dokter} * b_3)$$

Dengan :  $Kpa$  = Kuota penjualan daerah "a"

$b_1, b_2, b_3$  = Bobot untuk masing-masing parameter

Gambar 3.4 Formulasi perhitungan Kuota Penjualan

Sedangkan untuk proses penghitungan, nilai yang didapat dari masing-masing parameter akan dikalikan dengan bobot tertentu. Dan terakhir, hasil yang didapat dari perkalian antara bobot dengan parameter tersebut akan “dikonversi” menjadi target penjualan region. Berikut adalah konversi target penjualan yang digunakan oleh PT. Kalbe Farma Surabaya.

Tabel 3.5 Nilai Bobot Konversi

Nilai yang Dikonversi	Target Penjualan Region per Hari
0-4	Rp 4.000.000
4-8	Rp 6.000.000
8-12	Rp 8.000.000
> 12	Rp 10.000.000

Contoh perhitungan target penjualan pada sebuah region dengan masing-masing bobot untuk jumlah penduduk adalah 0.5, untuk jenis region 0.3, dan untuk jumlah dokter adalah 0.2.

Region : I  
 Jumlah Penduduk : 100.000-200.000  
 Jenis Region : Tengah Kota  
 Jumlah Dokter : 45-90

I	Jumlah Penduduk	Jenis Region	Jumlah Dokter
Klasifikasi	100.000-200.000	Tengah Kota	46-90
Nilai	10	15	10

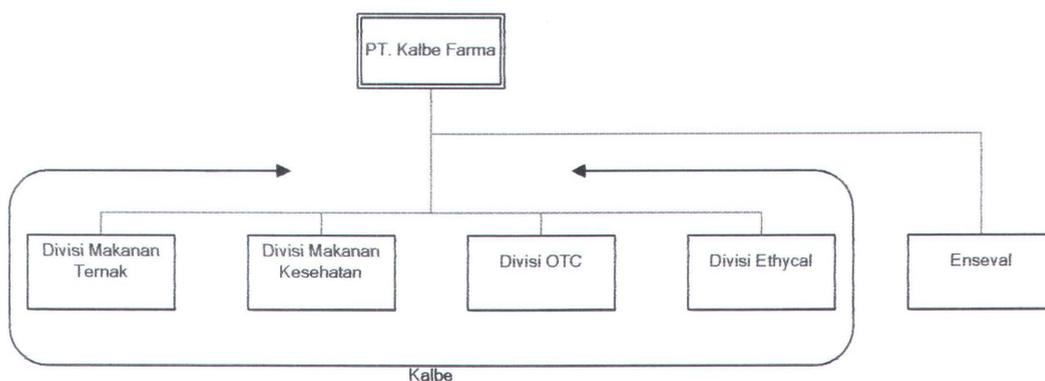
$$\begin{aligned}
 KP_I &= (10 * 0.5) + (15 * 0.3) + (10 * 0.2) \\
 &= 5 + 4.5 + 2 \\
 &= 11.5
 \end{aligned}$$

Berdasar perhitungan  $KP_I$  didapat nilai sebesar 11.5. Dari hasil tersebut dapat diambil kesimpulan bahwa target penjualan region I per hari adalah Rp 10.000.000



### 3.3 Kinerja Pemasaran di PT. Kalbe Farma

PT. Kalbe Farma adalah sebuah perusahaan yang bergerak di bidang farmasi. Produk perusahaan dapat dibedakan menjadi empat. Yaitu, *Ethycal* (obat resep), OTC (obat bebas), makanan kesehatan, dan makanan ternak. Selain itu Kalbe juga mempunyai anak perusahaan yang khusus menangani masalah distribusi dan penjualan produk-produk perusahaan yaitu Enseval Mega Trading (Enseval). Meskipun berstatus anak perusahaan, namun Enseval berhak menangani proses distribusi dan penjualan perusahaan farmasi lain. Secara umum struktur organisasi PT. Kalbe Farma adalah sebagai berikut



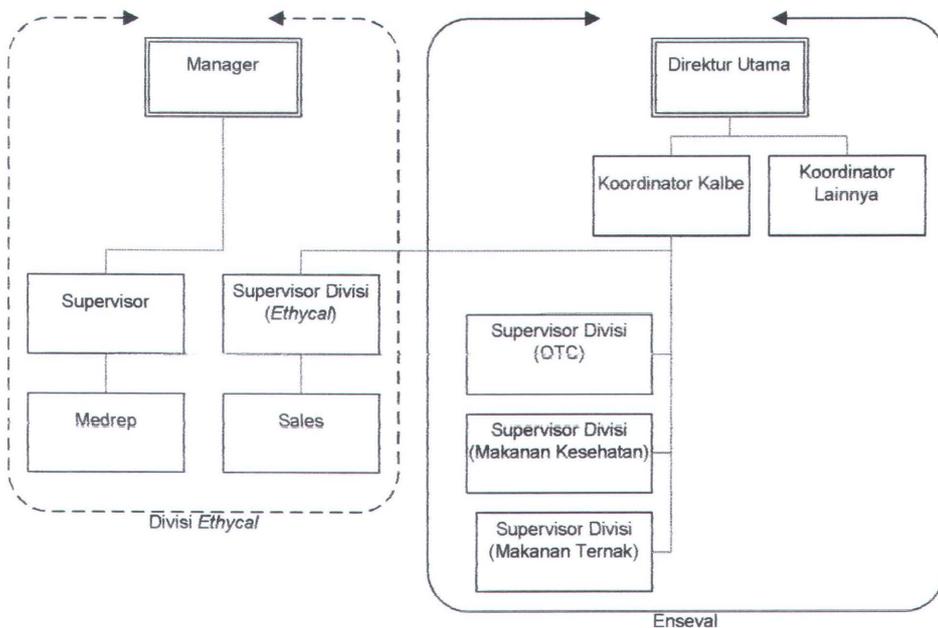
Gambar 3.5 Struktur Organisasi PT. Kalbe Farma

Dengan adanya anak perusahaan yang khusus menangani masalah distribusi penjualan menyebabkan semakin “lincah”-nya Enseval dalam melakukan proses penjualan, di samping itu divisi-divisi yang lain pun tidak lagi dipusingkan dengan masalah distribusi. Namun dampak signifikan yang didapat dengan adanya Enseval adalah adanya proses bisnis baru yang akan membawa keuntungan (laba) kepada pihak Kalbe. Di lain pihak, adanya pemisahan antara divisi-divisi dengan Enseval menyebabkan kompleksnya usaha pemasaran yang harus dikelola oleh Kalbe.

Proses pemasaran dan penjualan dilakukan oleh unit yang berbeda, untuk pemasaran dilakukan sendiri oleh tiap-tiap divisi sedangkan penjualan dari tiap divisi ditangani oleh Enseval. Hal tersebut menyebabkan harus adanya koordinasi yang baik

antara Enseval dengan divisi lainnya. Pihak Enseval dan divisi lain harus sepakat mengenai pembagian wilayah kerja/region. Setelah pembagian region disepakati, Enseval akan menerjunkan para *Sales Representative (Sales)* untuk tiap divisi. *Sales* di tiap divisi dipimpin oleh seorang *Supervisor Divisi*. Sehingga nantinya untuk pihak Kalbe sendiri akan ada empat *Supervisor Divisi* yang dipimpin oleh seorang koordinator.

Pada divisi *Ethycal* kedudukan *supervisor* sejajar dengan *supervisor divisi* dari pihak Enseval. Selain bertanggung jawab kepada koordinatornya, *supervisor divisi* juga harus berkomunikasi dengan *supervisor* untuk mengkoordinasikan kinerja antara *sales* dan *medrep* masing-masing. Secara garis besar struktur divisi penjualan dan pemasaran pada Enseval adalah sebagai berikut



Gambar 3.6 Divisi Penjualan dan Pemasaran *Ethycal*

### 3.3.1 Sistem Berjalan

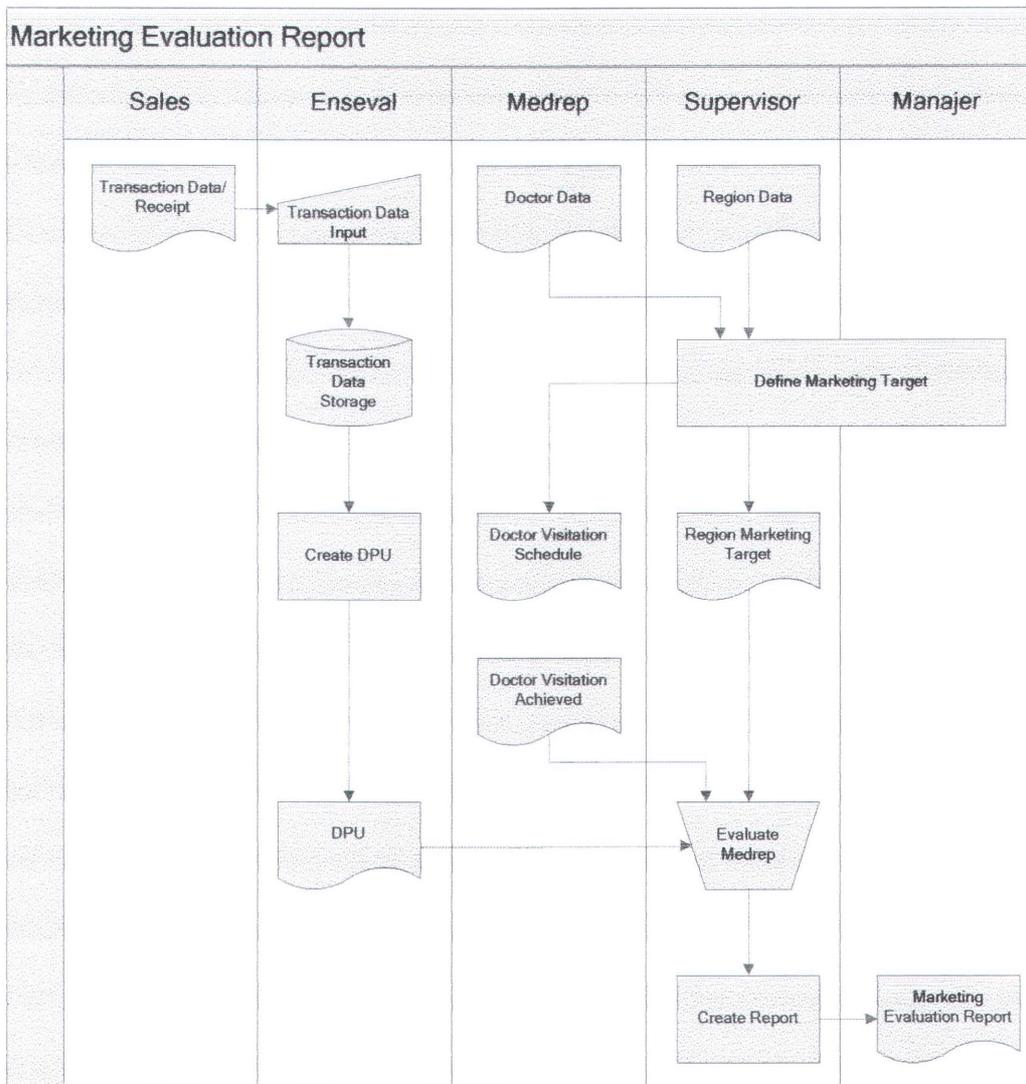
Sistem pemasaran bagian *Ethycal* sangat unik. Karena berbeda dengan proses pemasaran pada umumnya, yaitu dengan menggunakan *medrep* yang datang ke dokter sebagai alat pemasaran pengganti iklan dan promosi di media massa. Ketika *medrep* melakukan visitasi ke dokter, *medrep* tidak diperkenankan membawa obat untuk dijual begitu pula *medrep* tidak diperkenankan untuk menerima pesanan pembelian obat dari dokter. *Medrep* hanya bertugas “mendidik” dokter agar menuliskan resep obat produk dari Kalbe. Ketika *medrep* melakukan visitasi tidak ada *deal* sama sekali, tidak ada kesepakatan bahwa setelah “dididik” dokter akan menuliskan resep obat dari Kalbe, karena itu dibutuhkan keterampilan dan kreatifitas yang tinggi dari para *medrep*. Sedangkan proses transaksi obat sendiri berlangsung di apotek. Pasien mendatangi apotek untuk membeli obat yang dituliskan dokter pada resep. Disini bisa dilihat sangat krusialnya fungsi *medrep* sebagai ujung tombak perusahaan dalam pemasaran. Secara umum, tugas *medrep* sendiri ada tiga yaitu

1. Memperkenalkan produk baru.
2. Mengingatkan produk lama.
3. Kondisional, ini adalah tugas sampingan *medrep* seperti memberikan undangan simposium, seminar atau yang lainnya kepada dokter.

Sebelum melakukan visitasi (kunjungan) kepada para dokter, para *medrep* sudah diberikan jadwal rencana visitasi oleh pihak *supervisor*. Pihak *supervisor* juga yang menentukan dokter-dokter yang akan “dididik” oleh tiap *medrep*. Saat ini tiap *medrep* biasanya “mendidik” 40 sampai 50 orang dokter yang berada pada satu region.

Untuk saat ini proses pengukuran kinerja bagian *Ethycal* baru terbatas pada penilaian pencapaian target tiap region yang didasarkan pada DPU yang dikeluarkan oleh pihak Enseval. Dari DPU tersebut pihak *supervisor* kemudian memilah-milah dan mengelompokkan hasil penjualan tiap regionnya. Kemudian pihak *supervisor* melakukan hasil evaluasi penjualan di apotek dengan hasil visitasi *medrep* dan target penjualan di regionnya. Hasil evaluasi tersebut kemudian diserahkan kepada pihak

manajer untuk digunakan bahan evaluasi kinerja penjualan. Berikut adalah gambaran secara umum proses penilai kinerja penjualan.



Gambar 3.7 Diagram Alir proses penilaian kinerja pemasaran

### 3.3.2 Permasalahan Saat Ini

Permasalahan yang terjadi saat ini adalah banyaknya informasi yang harus dihimpun dan dikelola dalam sebuah sistem yang belum terintegrasi. Informasi yang dibutuhkan tersebut untuk saat ini berada dalam sebuah sistem yang berdiri sendiri-sendiri. Misalnya untuk mengetahui data transaksi tiap obat di apotek saat ini harus

menunggu DPU dari Enseval, sedangkan data mengenai wilayah, target penjualan dan kuota penjualan dibuat oleh seorang supervisor berdasar instruksi yang diberikan pihak manajer, sedangkan data mengenai dokter sebagai pelanggan utama produk dimiliki oleh seorang medrep. Sehingga selama ini pihak manajer membutuhkan waktu yang cukup panjang untuk dapat menilai kinerja para *medrep*-nya. Hal tersebut disebabkan antara lain

1. Pihak Enseval membutuhkan waktu satu hari untuk merekapitulasi penjualan hari ini. Sehingga DPU untuk hari ini baru bisa disampaikan kepada pihak manajer keesokan harinya.
2. Daftar Penjualan per Unit (DPU) yang diberikan pihak enseval kepada supervisor hanya berisi nama obat dan data transaksi penjualan tiap apotek.
3. Supervisor harus memilah-milah data berdasar lokasi dan region penjualan tiap apotek yang ada dalam DPU.
4. Pihak supervisor juga harus melakukan cross check terhadap hasil DPU dengan target dan kuota yang telah diberikan kepada medrep sebelum memberikan penilaian terhadap kinerja medrep di bawahnya.

Kesulitan lain yang dihadapi adalah terpecahnya lokasi bagian *Ethycal*. Di Surabaya, bagian pemasaran *Ethycal* berada pada tiga lokasi yang berbeda, yaitu di Brebek Industri, Klampis, dan Perak. Sedangkan lokasi dari kantor Enseval sendiri berada di Brebek Industri. Sehingga di kantor-kantor cabang lain masih menggunakan tenaga manual (kurir) untuk mengantar DPU setiap harinya.

Dari sistem yang sudah ada, ditemukan beberapa kekurangan dari sistem penilaian kinerja pemasaran. Karena itu dari sistem yang lama akan dikembangkan sebuah sistem baru yang diharapkan dapat meningkatkan kecepatan dan keakuratan dari kinerja sistem sebelumnya. Tujuan dari pengembangan sistem tersebut adalah (1) meningkatkan kecepatan dan mengurangi kesalahan dari sistem manual, dan juga memberikan kebebasan untuk menggunakan sistem baru (2) memberikan detail yang lebih baik atas kinerja *medrep* (3) memberikan variasi laporan, baik dari segi isi

(laporan penjualan tiap *medrep*, tiap dokter, tiap wilayah, tiap apotek) maupun dari segi bentuk (tabel dan grafik) sehingga dapat digunakan sebagai bahan pertimbangan pembuatan keputusan pemasaran.

### 3.3.4 Analisa Kinerja Medrep

Berbeda dengan penilaian kinerja sebelumnya yang didasarkan pada kinerja tiap region. Peningkatan penjualan di sebuah region merupakan keberhasilan kolektif begitu pula sebaliknya. Hal ini tidaklah salah, hanya saja penilaian akan terasa kurang *fair* bila dalam region tersebut hanya seorang *medrep* saja yang benar-benar aktif, dimungkinkan peningkatan nilai penjualan hanya berasal dari dokter-dokter yang dididik oleh seorang *medrep* saja. Pada sistem yang baru nanti, kinerja *medrep* akan dievaluasi dengan lebih detail dan personal. Setiap *medrep* dapat memantau hasil penjualan dari dokter yang “dididik” setiap hari. Selain meningkatkan transparansi bagi *medrep*, hal tersebut juga dapat membantu pihak *supervisor* dan manajer untuk menentukan langkah kebijakan pemasaran selanjutnya.

Dalam sistem baru nanti, target penjualan *medrep* didapat dari perhitungan persentase dokter yang “dididik” seorang *medrep* dengan jumlah keseluruhan dokter yang berada dalam region yang sama dengan *medrep* dikalikan dengan target penjualan region. Formulasi yang digunakan adalah

$$T_{m1} = \frac{\sum D_d}{\sum D} \times TP_1$$

Dengan :      $T_{m1}$      : Target penjualan *medrep* “1”  
                $\sum D_d$    : Jumlah dokter yang “dididik” *medrep* “1”  
                $\sum D$      : Jumlah dokter di region  
                $TP_1$     : target penjualan di region

Gambar 3.9 Formulasi Perhitungan target Medrep

Contoh penghitungan dengan melanjutkan dari contoh target penjualan region sebelumnya dapat dilihat pada halaman berikut

Target Penjualan region I : Rp 10.000.000

$\Sigma$  dokter yang dididik *medrep* “1” : 56

$$T_{m1} = \frac{56}{90} \times 10.000.000$$

$$T_{m1} = 6.222.222$$

Sehingga target penjualan untuk *medrep* “1” adalah sebesar Rp 6.222.222

Sedangkan indikator-indikator kunci yang digunakan untuk mengukur kinerja *medrep* adalah jumlah rata-rata kunjungan *medrep* per hari, pendapatan rata-rata per kunjungan penjualan, biaya kunjungan penjualan, dan jumlah nilai produk yang berhasil dijual. Dari tiap kategori tersebut akan muncul nilai yang nantinya akan dikonversi sistem menjadi nilai huruf – A, B, C, D, E. Dengan nilai A adalah nilai terbaik, dengan bobot masing-masing 4 untuk nilai A, 3 untuk B dan seterusnya hingga 0 untuk nilai E. Hal ini digunakan untuk menjaga obyektifitas dari supervisor dalam memberikan rekomendasi terhadap kinerja *medrep*. Nilai yang ditampilkan sistem adalah nilai dengan kurva normal. Dalam artian jika pada suatu saat kondisi perekonomian secara umum memburuk dan nilai penjualan menurun, supervisor dapat memberikan rekomendasi dengan baik tanpa harus menghilangkan kesan *fair* dalam mengevaluasi *medrep*.

Jumlah rata-rata kunjungan *medrep* per hari, berupa banyaknya kunjungan yang dilakukan *medrep* kepada dokter, baik yang berupa kunjungan penjualan (Mengenalkan atau mengingatkan produk perusahaan) maupun kunjungan-kunjungan yang bersifat kondisional (undangan, seminar, simposium, dll). Pada umumnya target visitasi yang diberikan kepada *medrep* adalah sepuluh kunjungan per hari. Dari rata-rata target tersebut akan dibuat kalkulasi untuk menghitung nilai visitasi *medrep*. Dengan asumsi, seorang *medrep* akan mendapat nilai A jika ia melakukan minimal 10 kunjungan perhari. Namun yang perlu dicermati adalah batasan yang “masuk akal” dari kunjungan per hari. Dengan penilaian seperti tidak berarti *medrep* yang

terbanyak melakukan visitasi adalah *medrep* terbaik, karena nantinya biaya kunjungan juga menjadi salah satu faktor penilaian. Semakin banyak visitasi akan semakin meningkatkan biaya visitasi. Poin penilaian yang ingin dicapai disini adalah bagaimana cara *medrep* mengefektifan waktu kunjungan tanpa harus mengorbankan jumlah visitasi yang harus dilakukan. Berikut adalah tabel yang digunakan untuk mengkonversi nilai dari jumlah kunjungan per hari.

Tabel 3.6 Nilai Bobot Konversi rata-rata kunjungan per hari

Jumlah kunjungan per hari	Nilai
$\geq 10$	A
9 – 10	B
8 – 9	C
7 – 8	D
$< 7$	E

Pendapatan rata-rata per kunjungan penjualan, yang nilainya dihitung dari total keuntungan yang dihasilkan *medrep* dikurangi biaya kunjungan dibagi dengan jumlah kunjungan. Hasil yang didapat oleh *medrep* akan dikonversikan sistem menjadi nilai dapat diperbandingkan. Tabel berikut adalah hasil konversi dari sistem.

Tabel 3.7 Nilai Bobot Konversi pendapatan per kunjungan penjualan dua mingguan

Pendapatan per kunjungan penjualan	Nilai
$> \text{Rp } 3.000.000$	A
$\text{Rp } 2.500.000 - \text{Rp } 3.000.000$	B
$\text{Rp } 2.000.000 - \text{Rp } 2.500.000$	C
$\text{Rp } 1.500.000 - \text{Rp } 2.000.000$	D
$< \text{Rp } 1.500.000$	E

Biaya kunjungan penjualan, berupa nilai dari biaya rata-rata per kunjungan penjualan dikalikan dengan banyaknya kunjungan penjualan. Dari sini pihak supervisor dapat mengetahui berapa alokasi biaya kunjungan yang dibutuhkan perusahaan untuk membiayai visitasi. Seorang *medrep* akan dikatakan baik jika ia mampu melakukan kunjungan yang efektif. Sehingga semakin sedikit biaya yang

harus dikeluarkan perusahaan akan semakin baik nilai yang didapat *medrep*. Berikut adalah tabel yang mengkonversi biaya kunjungan penjualan.

Tabel 3.8 Nilai Bobot Konversi biaya kunjungan penjualan dua mingguan

Biaya kunjungan penjualan	Nilai
Rp 700.000 – Rp 800.000	A
Rp 800.000 – Rp 900.000	B
Rp 900.000 – Rp 1.000.000	C
Rp 1.000.000 – Rp 1.100.000	D
> Rp 1.100.000	E

Jumlah nilai produk yang berhasil dijual. Dihitung dari penjumlahan obat yang terjual di apotek dari semua dokter yang “dididik” *medrep*. Nilai ini adalah hasil dari penjualan kotor yang didapat *medrep*, nilai ini pula yang akan dibandingkan dengan target penjualan yang didapat *medrep*. Tabel 3.9 akan menggambarkan nilai konversi dari nilai penjualan *medrep*.

Tabel 3.9 Nilai Bobot Konversi target penjualan dua mingguan

Persentase dari target penjualan	Nilai
> 97%	A
94% - 97%	B
91% - 94%	C
88% - 91%	D
< 88%	E

Dari tiap-tiap parameter tersebut akan diberikan bobot yang ditentukan oleh pihak manajer setiap bulan. Manajer dapat menentukan untuk bulan ini, parameter mana yang lebih diutamakan, dapat diberikan bobot yang lebih besar. Namun juga tidak menutup kemungkinan untuk memberikan bobot yang sama besar untuk tiap parameter. Dari nilai yang didapat *medrep* dengan bobot yang diberikan manajer dibuat formulasi berikut untuk menentukan nilai kinerja *medrep*.

$$Kinerja = \frac{(b_1 \times P_1) + (b_2 \times P_2) + (b_3 \times P_3) + (b_4 \times P_4)}{b_1 + b_2 + b_3 + b_4}$$

Dengan :  $P_1$  = Nilai dari banyaknya kunjungan

$P_2$  = Nilai dari jumlah pendapatan bersih per kunjungan penjualan

$P_3$  = Nilai dari jumlah biaya kunjungan

$P_4$  = Nilai dari jumlah pendapatan total penjualan

$b_1$  = bobot untuk banyaknya kunjungan

$b_2$  = bobot untuk pendapatan bersih per kunjungan penjualan

$b_3$  = bobot untuk jumlah biaya kunjungan

$b_4$  = bobot untuk jumlah pendapatan total penjualan

Gambar 3.  formula untuk menghitung indeks kinerja medrep

Dengan nilai kinerja yang didapat dari setiap *medrep*, supervisor dapat membandingkan dan memberikan rekomendasi atas kinerja *medrep*-nya. Hal ini untuk menjaga keobyektifitasan seorang supervisor dan menunjukkan adanya transparansi yang dalam prosesnya dapat diamati oleh setiap *medrep*.

Laporan analisa kinerja medrep mencakup semua informasi yang dibutuhkan pihak supervisor dan manajer baik yang berupa kuantitatif seperti yang disebutkan sebelumnya maupun kualitatif seperti laporan kunjungan yang berisi aktifitas yang dilakukan *medrep* pada tiap dokter tiap kunjungan. Laporan-laporan tersebut akan di-*generate* setiap dua minggu oleh sistem. Dibutuhkan peran pihak supervisor untuk memberikan rekomendasi kinerja *medrep* agar manajer dapat melihat kinerja *medrep* dengan lebih obyektif.

### 3.4 Kebutuhan Informasi

Dalam menentukan sistem dan aplikasi yang dibangun, analisa kebutuhan pengguna merupakan kunci atas keberhasilan aplikasi tersebut. Dalam tugas akhir ini digunakan metode observasi dan wawancara dengan beberapa individu yang terkait. Detail observasi dan wawancara dapat dilihat pada lampiran B.

### 3.4.1 Identifikasi Pengguna

Adapun penggolongan pengguna sistem baru ini dikelompokkan berdasar pada status individu yang berada di atau berhubungan dengan proses penialain kinerja bagian *ethycal* PT. Kalbe Farma Surabaya.

1. Manajer, selaku pimpinan bagian *ethycal*, manajer perlu dan memiliki kewenangan penuh untuk memonitor dan mengevaluasi seluruh bagian sistem informasi kinerja *medrep* ini sebagai sumber pendukung pembuatan keputusan.
2. Supervisor, sebagai "kaki-tangan" manajer dalam hal pelaksanaan dan pengawasan pemasaran perusahaan khususnya mengenai *medrep*. *Supervisor* mengelola target penjualan region, mengevaluasi hasil visitasi *medrep* dan memberikan penilaian terhadap kinerja *medrep*.
3. *Medrep*, sebagai pihak yang dievaluasi berhak mengetahui hasil penjualan dan penilaian *supervisor* terhadap kinerjanya. *Medrep* akan mendapatkan laporan atas hasil kinerjanya dari sistem.
4. *Sales*, merupakan bagian input terpenting dalam sistem. Meskipun tidak mendapatkan sesuatu yang signifikan dari sistem, namun peran *sales* sebagai penginput data transaksi dan resep dari dokter sangat krusial bagi sistem.
5. Apotek, adalah pihak luar Kalbe yang berhubungan dengan sistem. *Sales* akan memberikan *login* dan *password* yang dapat digunakan untuk memantau hasil penjualan apoteknya.
6. Administrator, adalah pihak yang bertanggung jawab atas manajemen data *user*, dan manajemen data obat.

### 3.4.2 Kebutuhan Sistem

Dari hasil pengamatan yang dilakukan dengan menggunakan format survey dan wawancara, maka dapat didefinisikan menjadi kebutuhan sistem, yaitu

#### A. Manajer

Pada klasifikasi ini, analisa kebutuhan yang dilakukan terdiri dari 2 bagian utama yaitu fungsionalitas sistem dan batasan sistem. Adapun penjelasan mengenai masing-masing adalah sebagai berikut:

##### *Fungsionalitas Sistem*

Tersedianya fungsi untuk :

- Mengubah bobot, baik bobot untuk perhitungan target penjualan maupun bobot yang digunakan untuk mengevaluasi kinerja *medrep*. Selain itu manajer juga dapat mengubah alokasi biaya yang digunakan *medrep* untuk setiap kunjungan penjualan. Proses pemasukan bobot dilakukan setiap bulan.
- Melihat detail seluruh region, baik supervisornya, jenis region, jumlah penduduk, maupun detail setiap dokter, *medrep*, dan apotek yang ada di setiap region.
- Melihat detail penilaian kinerja seluruh *medrep* beserta laporan penjualannya setiap dua mingguan.
- Melihat detail nilai penjualan untuk setiap dokter, obat, dan apotek, di seluruh region.

##### *Batasan Sistem*

- Sistem tidak dapat melakukan perhitungan jika bobot tidak diisi.

#### B. Supervisor

Pada klasifikasi ini, analisa kebutuhan yang dilakukan terdiri dari 2 bagian utama yaitu fungsionalitas sistem dan batasan sistem. Adapun penjelasan mengenai masing-masing adalah sebagai berikut:

### *Fungsionalitas Sistem*

Tersedianya fungsi untuk :

- Mengubah detail region, yaitu jenis region dan jumlah penduduk di region, sedangkan untuk jumlah dokter di region dihitung dari *database*.
- Melihat, menambah, mengubah dan menghapus data *medrep* yang menjadi bawahannya. Supervisor juga dapat memberikan rekomendasinya atas kinerja *medrep* selama dua mingguan. Rekomendasi *supervisor* nantinya akan digunakan untuk membantu manajer dalam menilai kinerja *medrep* secara lebih *fair*.
- Melihat, menambah, mengubah dan menghapus data dokter yang ada di region. selain itu supervisor juga dapat meng-*assign* seorang *medrep* yang ditugaskan untuk mendidik dokter tersebut.
- Melihat, menambah, mengubah dan menghapus data visitasi yang telah dilakukan *medrep*. *Medrep* harus menyerahkan surat jalan yang telah berisi tanda tangan dokter.
- Melihat detail nilai penjualan untuk setiap dokter dan obat di regionnya.

### *Batasan Sistem*

- Diasumsikan lokasi dokter tetap, tidak berpindah region.
- Sistem tidak menangani kevalidan jenis dan jumlah penduduk region, isi detail region menjadi tanggung jawab supervisor.

### C. *Medrep*

Pada klasifikasi ini, analisa kebutuhan yang dilakukan terdiri dari 2 bagian utama yaitu fungsionalitas sistem dan batasan sistem. Adapun penjelasan mengenai masing-masing adalah sebagai berikut:

#### *Fungsionalitas Sistem*

Tersedianya fungsi untuk :

- Melihat evaluasi kinerja *medrep* dan rekomendasi yang diberikan supervisor untuk region dimana *medrep* berada.
- Melihat detail nilai penjualan untuk setiap dokter dan obat di regionnya.

### *Batasan Sistem*

- Diasumsikan lokasi *medrep* tetap, tidak berpindah region.

### D. Sales

Pada klasifikasi ini, analisa kebutuhan yang dilakukan terdiri dari 2 bagian utama yaitu fungsionalitas sistem dan batasan sistem. Adapun penjelasan mengenai masing-masing adalah sebagai berikut:

#### *Fungsionalitas Sistem*

Tersedianya fungsi untuk :

- Melihat, menambah, mengubah dan menghapus data transaksi yang diperoleh melalui resep yang didapat dari apotek.
- Melihat, menambah, mengubah dan menghapus data apotek yang berada di region tempat *sales* berada.
- Melihat detail nilai penjualan untuk setiap dokter dan obat di regionnya.

#### *Batasan Sistem*

- Diasumsikan semua resep memiliki data yang dibutuhkan sistem, yaitu tanggal, nama apotek, nama dokter, dan detail transaksi.
- Sistem tidak dapat menangani pembelian obat tanpa resep.

### E. Apotek

Pada klasifikasi ini, analisa kebutuhan yang dilakukan terdiri dari 2 bagian utama yaitu fungsionalitas sistem dan batasan sistem. Adapun penjelasan mengenai masing-masing adalah sebagai berikut:

#### *Fungsionalitas Sistem*

Tersedianya fungsi untuk :

- Meihat detail transaksi pada apoteknya untuk periode kapan pun
- Monitoring detail penjualan obat di region apotek, sebagai bahan pertimbangan penjualan.

#### *Batasan Sistem*

- Monitoring hanya dapat dilakukan untuk obat-obatan dari Kalbe.



**BAB IV**  
**DESAIN DAN PERANCANGAN**  
**PERANGKAT LUNAK**

## BAB IV

### DESAIN DAN PERANCANGAN PERANGKAT LUNAK

Bab ini membahas analisa kebutuhan perusahaan atas sistem informasi pengukuran kinerja yang akan dibangun serta rancangan sistem, desain sistem, sekaligus implementasi sistem.

#### 4.1 Desain Sistem

Tahap desain sistem menjelaskan secara detail penerjemahan kebutuhan sistem ke dalam suatu aplikasi, baik dari segi pemodelan proses bisnis yang berlangsung, pemodelan dari sisi *programming* dan aplikasi, maupun pemodelan alur data. Model yang digunakan dalam desain aplikasi ini adalah UML (*Unified Modelling Language*).

##### 4.1.1 Use Case Diagram

*Use case diagram* merupakan gambaran fungsionalitas sistem hasil penjabaran kebutuhan terhadap sistem yang ingin dibangun. Pembuatan *use case diagram* dimulai dari mendefinisikan *actor*, kemudian membuat *use case*-nya.

###### 4.1.1.1 Actor

Individu yang terlibat dalam sistem, dapat dibedakan menjadi beberapa *actor*, yaitu

1. *Sales*, adalah *actor* yang bertanggung jawab terhadap data transaksi obat (resep) dari apotek. Tugas utama dari *actor* ini adalah memasukkan dan memonitor data transaksi penjualan obat tiap apotek. Tiap apotek akan dimonitor oleh seorang *sales*.
2. *Supervisor*, merupakan *actor* yang bertanggung jawab terhadap data penjualan dan *medrep* yang berada di regionnya. Tugas *supervisor* adalah menginisialisasi data region dan mengevaluasi kinerja *medrep*

yang berada di bawahnya. *Supervisor* hanya bisa mengedit data dalam regionnya sendiri.

3. Manajer, *actor* yang memanfaatkan aplikasi untuk kepentingan analisa dan evaluasi kinerja pemasaran. Manajer bersama-sama dengan *supervisor* akan menentukan bobot perhitungan target penjualan.
4. Administrator, adalah *actor* yang bertanggung jawab terhadap jalannya aplikasi dengan baik. Tugas utama administrator adalah melakukan manajemen *user*. Secara umum, administrator menangani seluruh manajemen data dalam aplikasi.

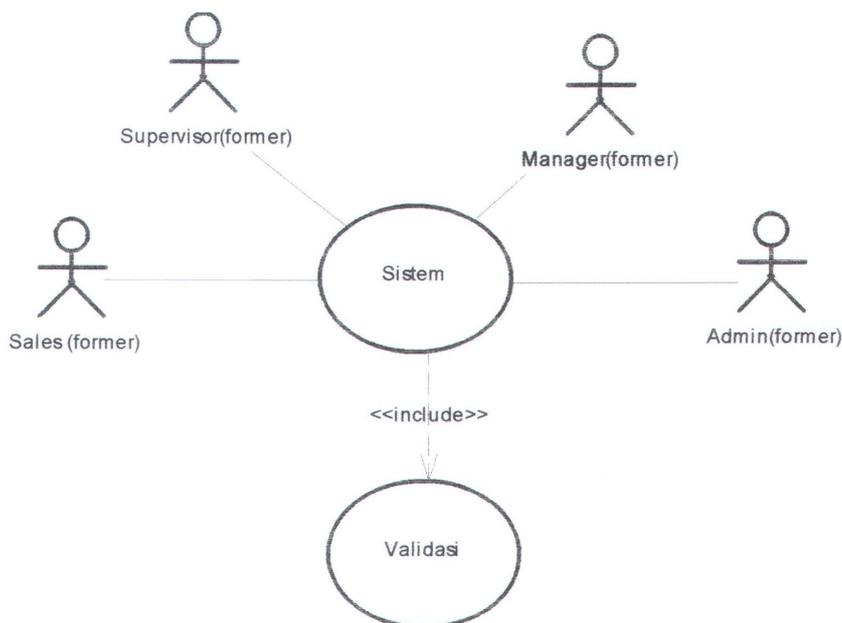
#### 4.1.1.2 Use Case

Dalam *Use Case* akan didefinisikan beberapa fungsionalitas yang ingin dibangun dalam sistem, yaitu

1. *Use Case* mengelola data *user*, yang mendefinisikan jenis *user* dan memberikan *user privilege* dalam aplikasi.
2. *Use Case* mengelola penjualan, adalah *use case* yang digunakan oleh *sales* untuk manajemen (*insert – update – delete*) data transaksi penjualan obat dari apotek. Data transaksi dari sebuah apotek akan dikelola oleh seorang *sales*.
3. *Use Case* mengelola data apotek sebuah fungsionalitas untuk manajemen (*insert – update – delete*) data apotek.
4. *Use Case* mengelola data dokter sebuah fungsionalitas untuk manajemen (*insert – update – delete*) data dokter.
5. *Use Case* mengelola data *medrep*, merupakan *use case* yang melakukan manajemen (*insert – update – delete*) terhadap data *medrep*.
6. *Use Case* mengelola bobot, adalah *use case* untuk melakukan pembobotan terhadap parameter-parameter target penjualan.
7. *Use Case* mengelola data obat, adalah *use case* yang digunakan untuk manajemen (*insert – update – delete*) data obat.

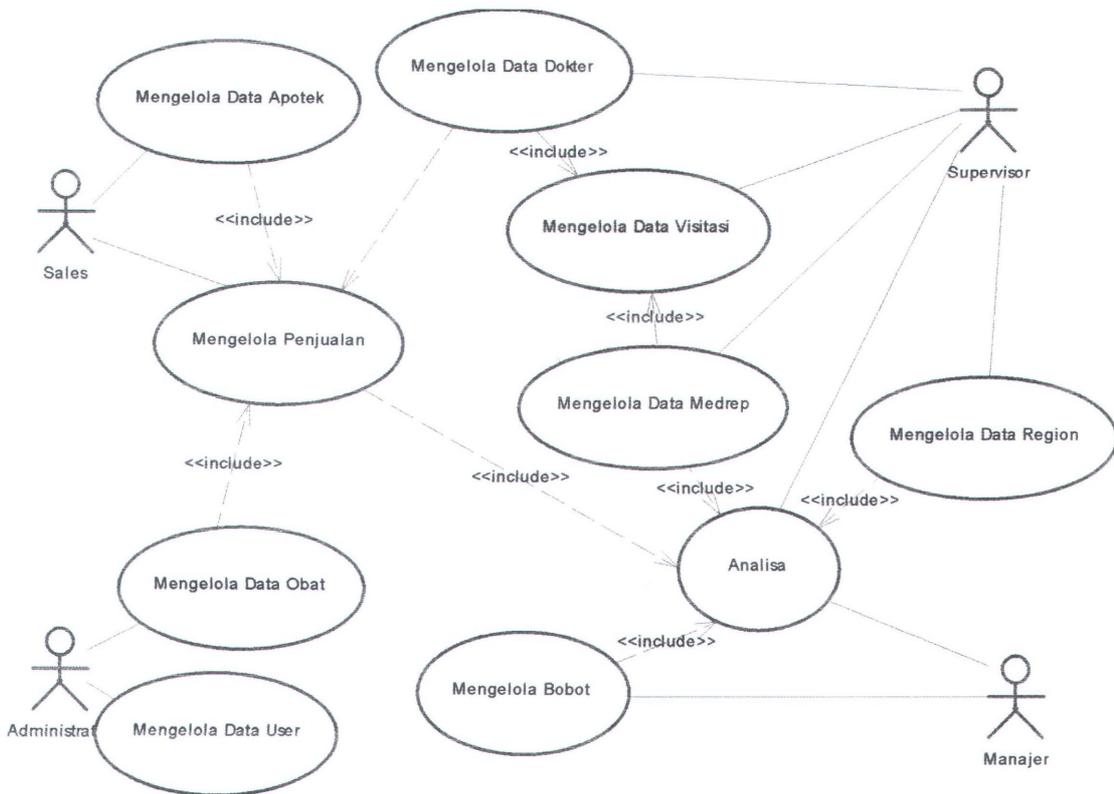
8. *Use Case* mengelola data visitasi, merupakan *use case* yang melakukan manajemen (*insert – update – delete*) terhadap data visitasi.
9. *Use Case* mengelola data region, merupakan fungsionalitas yang mengelola data region.
10. *Use Case* analisa, adalah *use case* yang berfungsi untuk melakukan analisa terhadap data-data yang telah tersimpan dengan menggunakan filter dan menghasilkan *output* berupa laporan dengan beberapa variasi.
11. *Use Case* Validasi, merupakan *use case* abstrak yang berfungsi sebagai filter bagi *user* yang menggunakan aplikasi ini, serta memberikan hak akses yang sesuai kepada *user*.

Setiap Aktor memiliki hak akses yang berbeda, sehingga digunakan *use case* validasi untuk memfilter hak akses *user*, untuk lebih jelasnya dapat dilihat pada gambar berikut



Gambar 4.1 Use Case Diagram Sistem

Spesifikasi *use case* yang digunakan pada sistem ini secara lengkap dapat dilihat dalam lampiran. Kumpulan use case dan actor di atas beserta relationship-nya dapat dilihat pada gambar 4.2. di halaman berikut



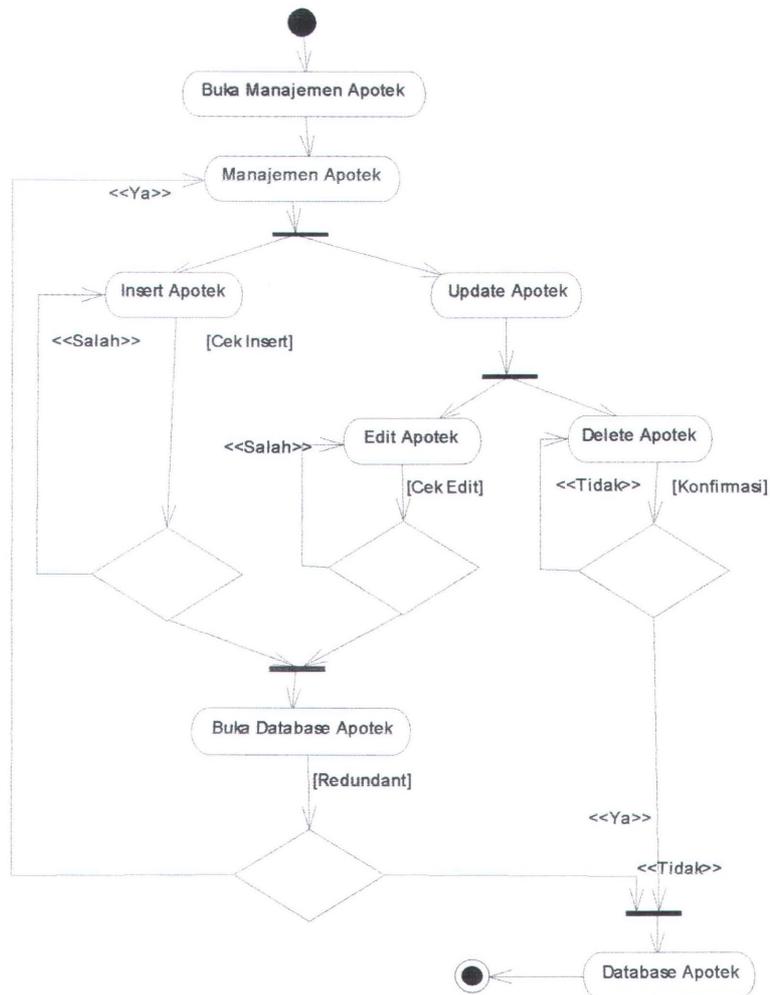
Gambar 4.2 Use Case Diagram Sistem beserta relasinya

### 4.1.2 Activity Diagram

Setelah menentukan Use Case dari sistem evaluasi kinerja medrep, selanjutnya untuk menentukan alur aplikasi didefinisikan *activity diagram*.

#### 4.1.2.1 Activity Diagram Mengelola Data Apotek

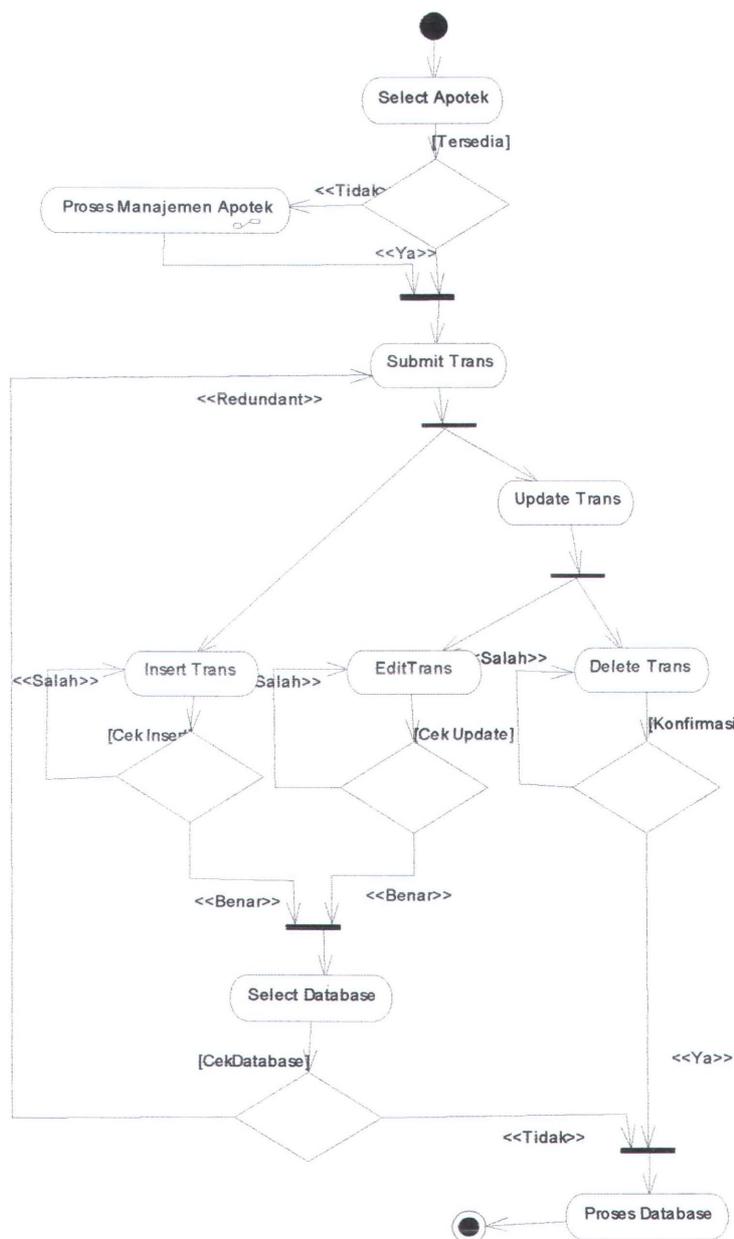
*Activity diagram* mengelola data apotek mendefinisikan proses penambahan, perubahan, dan penghapusan data apotek. Data apotek didapat dari pihak *sales* selaku pihak yang berhubungan langsung dengan pihak apotek. Jika *sales* akan menambah atau mengubah data apotek, data akan diverifikasi terlebih dahulu agar tidak ada data apotek yang sama dalam database.



Gambar 4.3 Activity Diagram Mengelola Data Apotek

#### 4.1.2.2 Activity Diagram Mengelola Data Penjualan

Activity diagram ini menggambarkan alur *insert*, *edit*, dan *delete* untuk data transaksi penjualan dari tiap apotek. Aktivitas utama proses ini adalah memasukkan data transaksi per hari dari tiap apotek yang dilakukan oleh *sales*. Data transaksi penjualan akan dikelompokkan tiap apotek. Lebih jelasnya dapat dilihat pada gambar 4.4 di halaman berikut.

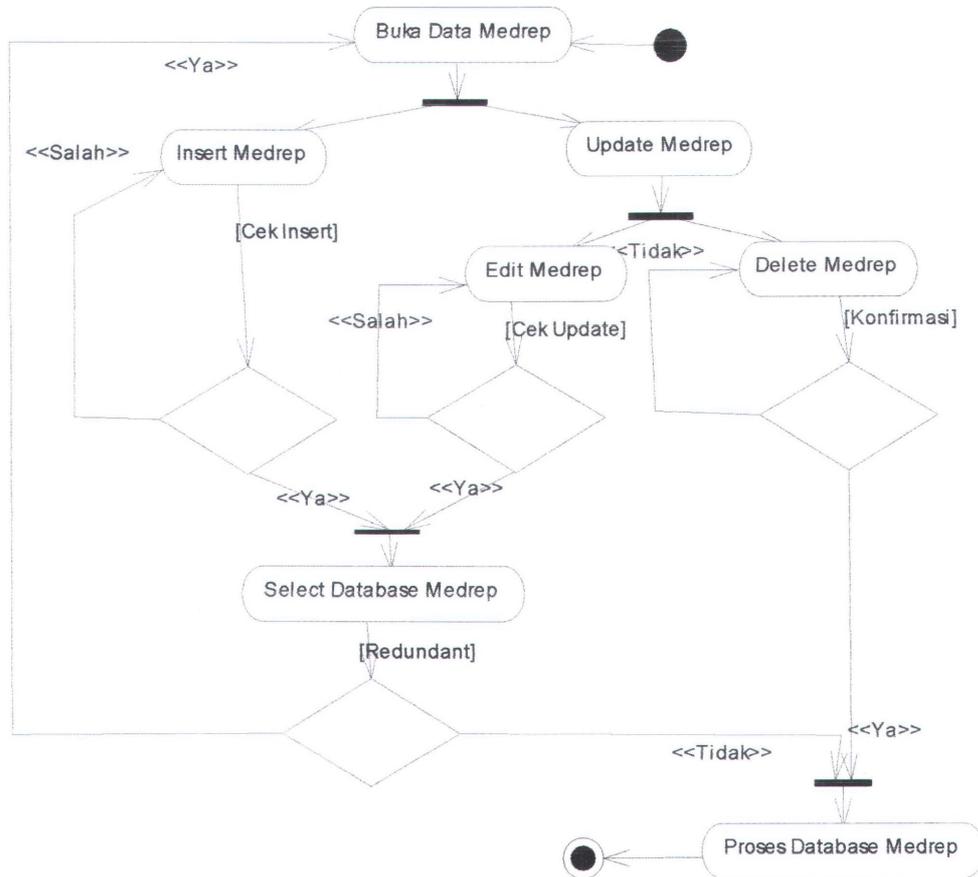


Gambar 4.4 Activity Diagram Mengelola Data Penjualan

#### 4.1.2.3 Activity Diagram Mengelola Data Medrep

Activity Diagram mengelola data Medrep mendefinisikan fungsi penambahan, perubahan atau penghapusan data medrep. Aktivitas utama proses ini adalah memasukkan data medrep yang dilakukan oleh pihak supervisor selaku pimpinan dari

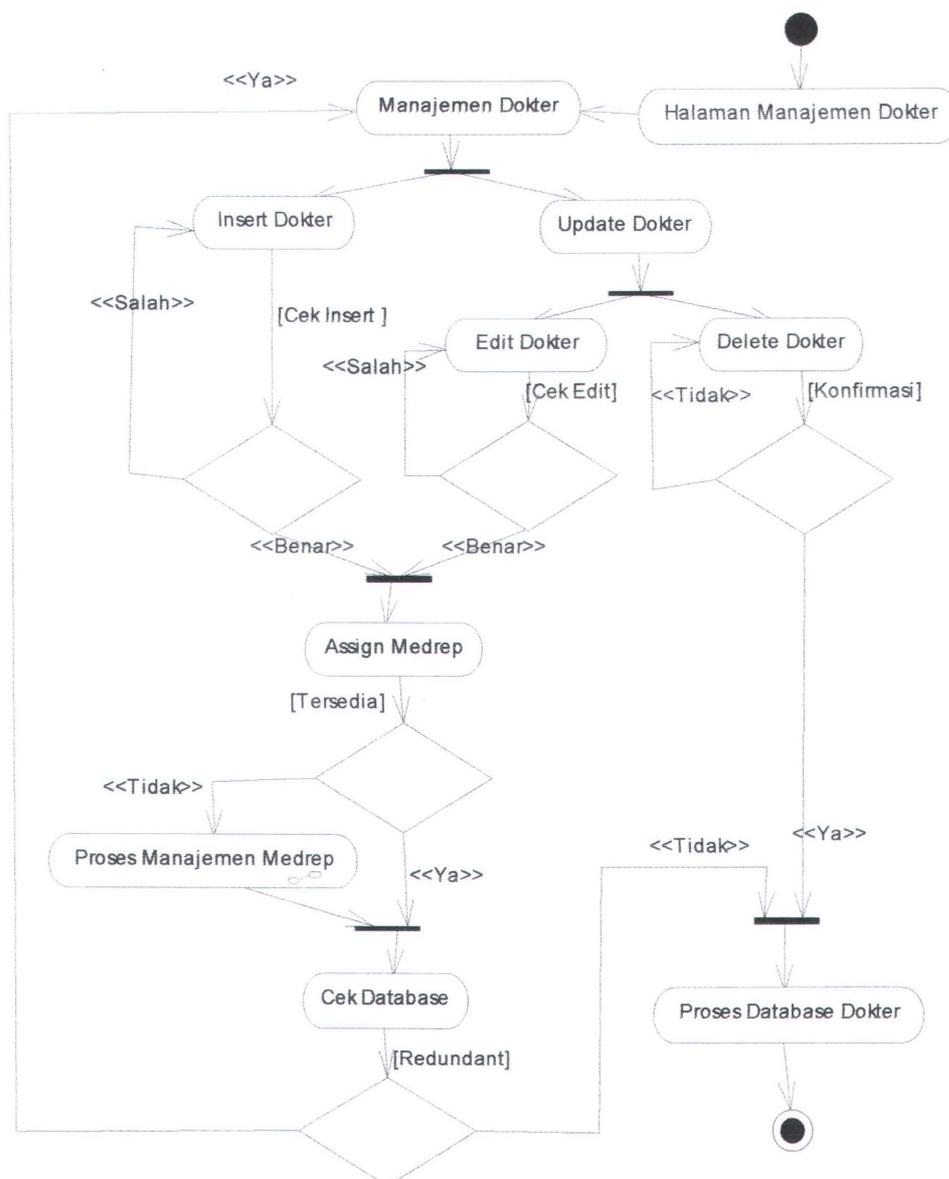
para *medrep*. Pihak supervisor hanya dapat mengelola data *medrep* yang berada di bawahnya.



Gambar 4.5 Activity Diagram Mengelola Data Medrep

#### 4.1.2.4 Activity Diagram Mengelola Data Dokter

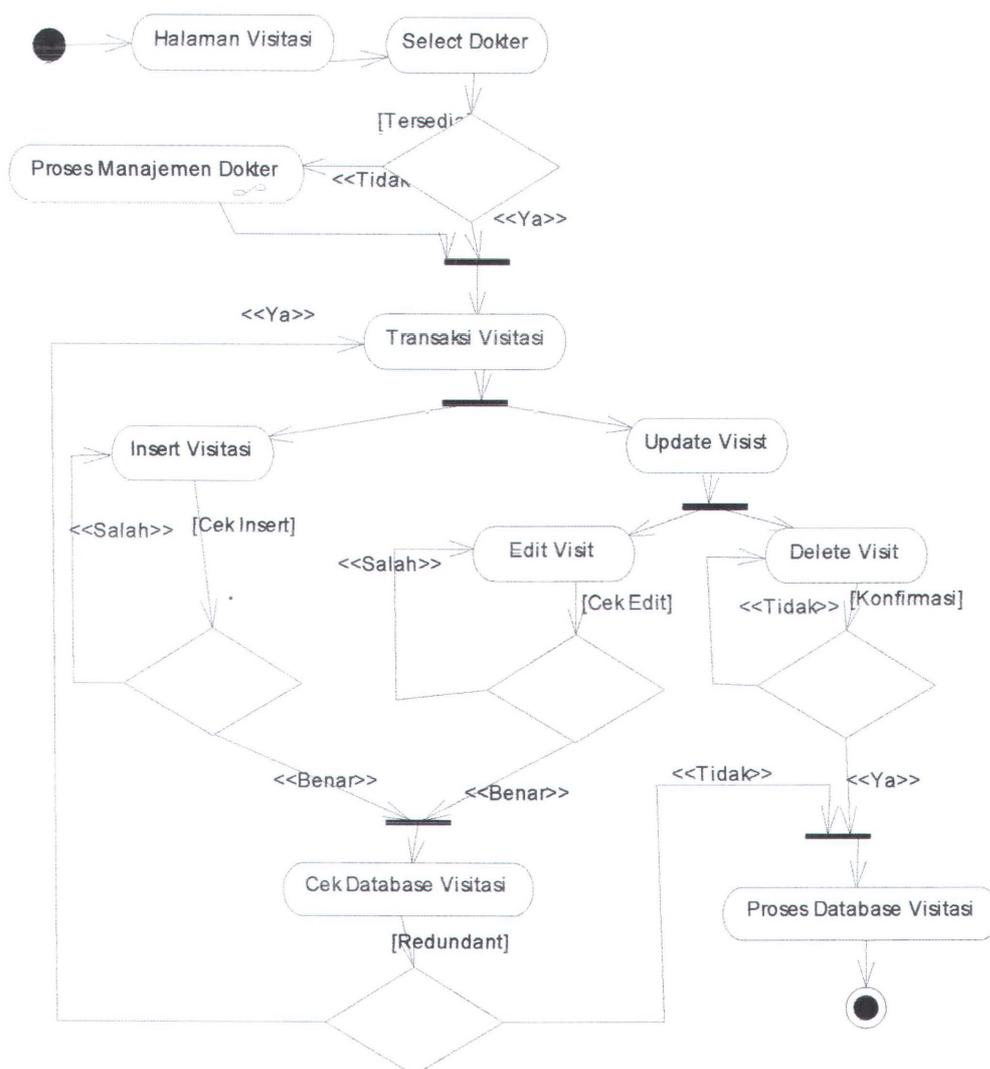
Activity Diagram Mengelola data dokter mendefinisikan fungsi *insert*, *update* dan *delete* data dokter. Aktivitas utama proses ini adalah memasukkan data dokter yang berada dalam region sekaligus meng-assign seorang *medrep* untuk mendidik dokter tersebut, proses ini dilakukan oleh pihak supervisor.



Gambar 4.6 Activity Diagram Mengelola Data Dokter

#### 4.1.2.5 Activity Diagram Mengelola Data Visitasi

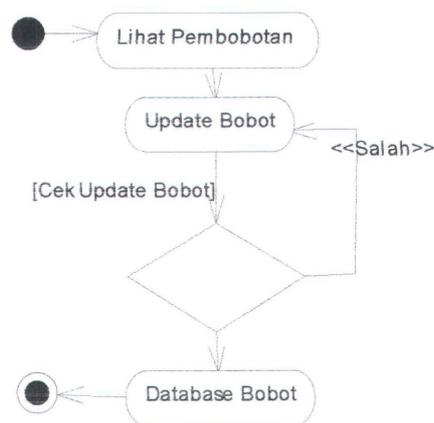
Activity Diagram mengelola data visitasi mendefinisikan fungsi *insert*, *update* dan *delete* data visitasi. Aktivitas utama proses ini adalah memasukkan data visitasi yang berisi detail kegiatan yang dilakukan oleh medrep ketika melakukan kunjungan ke dokter. Data visitasi dari seorang dokter akan dikelola oleh seorang supervisor yang menjadi pimpinan *medrep* yang mendidik dokter tersebut.



Gambar 4.7 Activity Diagram Mengelola Data Visitasi

#### 4.1.2.6 Activity Diagram Mengelola Bobot

Activity Diagram mengelola bobot digunakan untuk melakukan proses *update* atas nilai bobot yang nantinya akan digunakan untuk menghitung target penjualan region. tugas utama dari aktifitas ini adalah mengubah nilai bobot untuk tiga macam kriteria yaitu, jenis region, jumlah penduduk region, dan jumlah dokter. Meskipun proses penentuan bobot dilakukan bersama-sama antara pihak manajer dan supervisor, namun aktifitas ini dilakukan oleh pihak manajer.



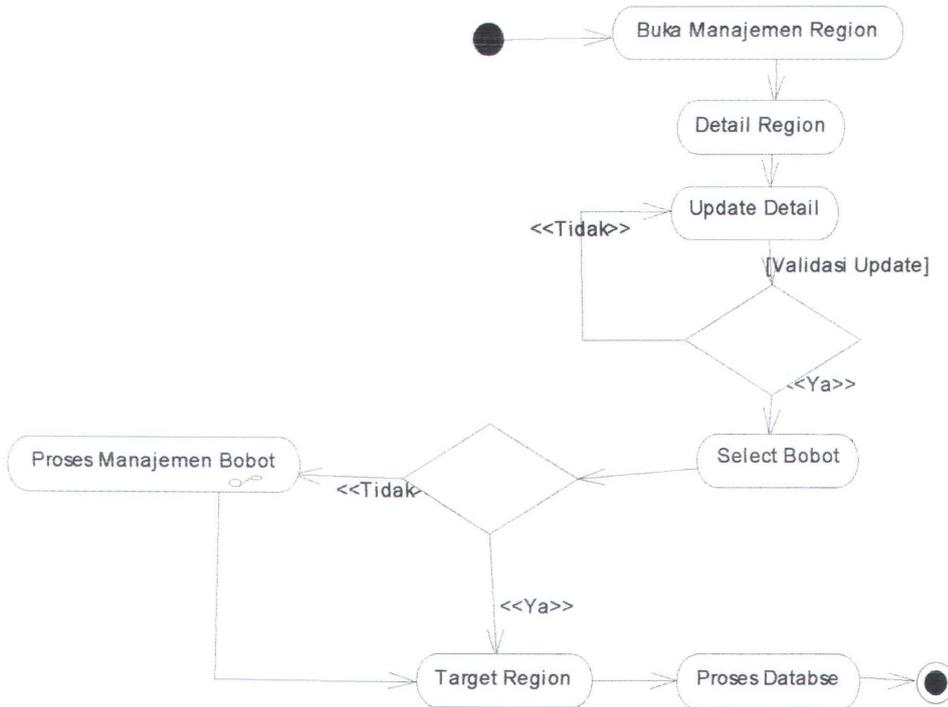
Gambar 4.8 *Activity Diagram* Mengelola Bobot

#### 4.1.2.7 *Activity Diagram* Mengelola Data Region

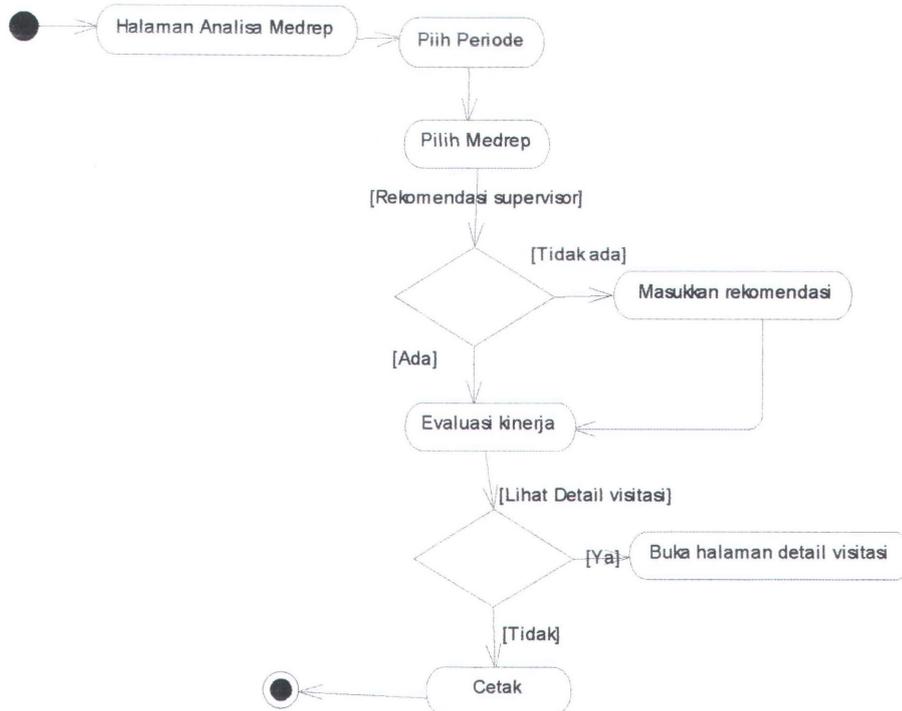
*Activity Diagram* mengelola data region mendefinisikan proses peng-update-an data detail region. Tugas utama dari aktifitas yang dilakukan oleh pihak supervisor ini adalah mengubah detail dari sebuah region. Untuk penghitungan target penjualan, target yang dapat diubah adalah jumlah penduduk dan jenis region. Dimana data keduanya didapat dari BPS ataupun survey langsung ke lapangan. Data sebuah region hanya bisa diakses oleh supervisor yang memimpinnya. *Activity Diagram* mengelola data region ditunjukkan pada gambar 4.9.

#### 4.1.2.8 *Activity Diagram* Analisa

Proses utama dalam sistem informasi evaluasi kinerja *medrep* ini adalah proses analisa data dengan menggunakan seperangkat indikator beserta target yang ditetapkan, sehingga menghasilkan kesimpulan. Proses analisa *medrep* sendiri terbagi menjadi dua bagian yaitu analisa kuantitatif yang berisi hasil penjualan yang berhasil dilakukan *medrep* dan analisa kualitatif yang berisi detail dari hasil vivitasi yang dilakukan oleh *medrep*. Proses ini akan di-generate dua minggu sekali oleh sistem dan hasil analisa yang ditampilkan berupa tabel dan grafik. Gambar 4.10 akan menunjukkan *Activity Diagram* Analisa



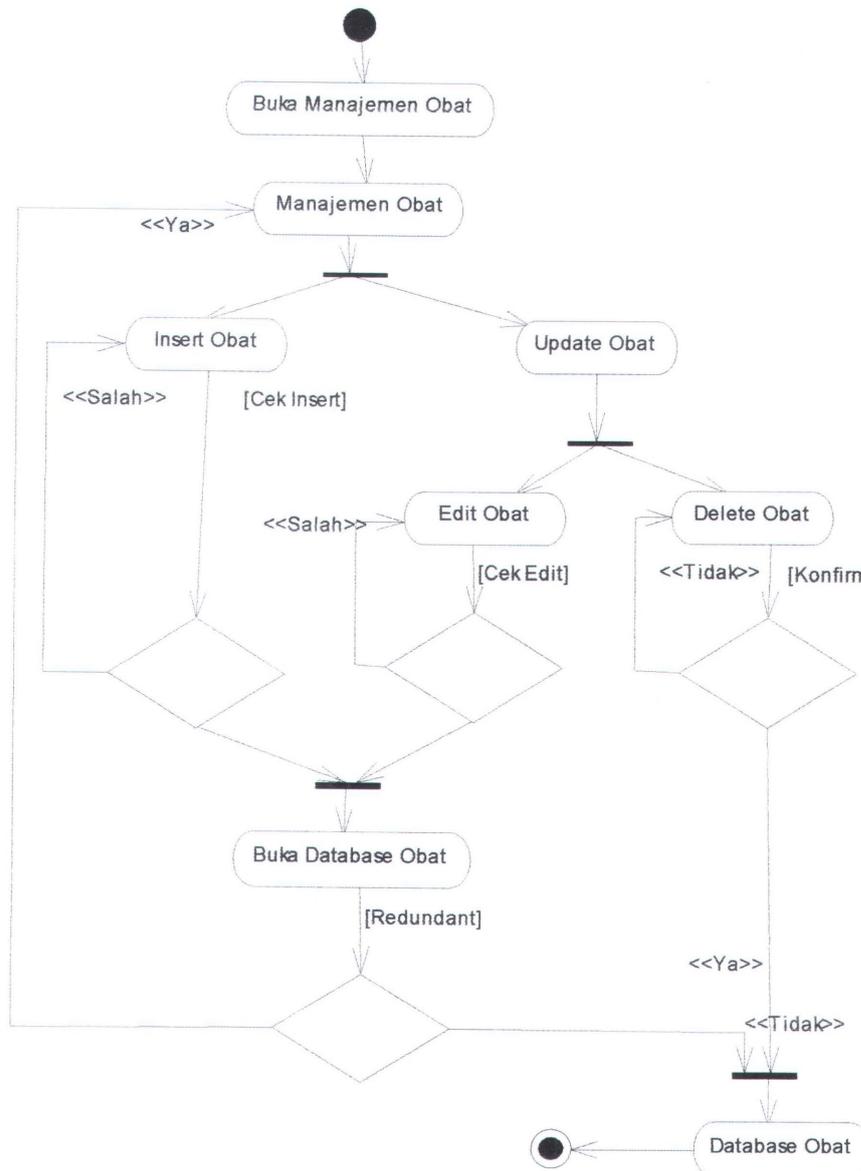
Gambar 4.9 Activity Diagram Maengelola Data Region



Gambar 4.10 Activity Diagram Analisa

#### 4.1.2.9 Activity Diagram Mengelola Data Obat

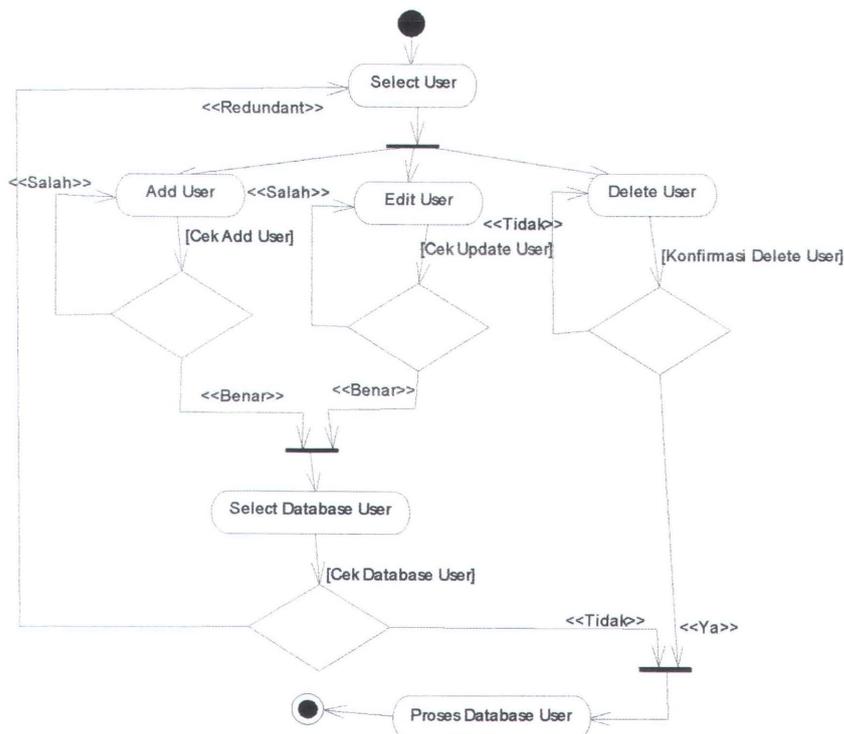
Activity Diagram mengelola data obat mendefinisikan fungsi *insert*, *update* dan *delete* data obat. Aktivitas utama proses ini adalah memasukkan data obat yang berisi detail informasi obat. Lebih jelasnya akan ditampilkan pada gambar berikut.



Gambar 4.11 Activity Diagram Mengelola Data Obat

#### 4.1.2.10 Activity Diagram Mengelola Data User

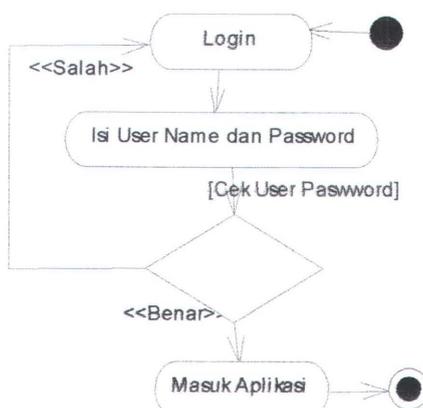
Activity Diagram manajemen user ini digunakan untuk mengelola data user yang akan menggunakan sistem. Tugas utama dari aktifitas yang dijalankan oleh pihak administrator ini adalah *insert*, *update* dan *delete* data user. Agar tidak terjadi duplikasi data, sebelum masuk database data akan diverifikasi terlebih dahulu.



Gambar 4.12 Activity Diagram Mengelola Data User

#### 4.1.2.11 Activity Diagram Validasi

Activity Diagram ini digunakan untuk memvalidasi setiap user yang akan menggunakan sistem. Tugas utama aktifitas ini adalah memberikan konfirmasi bila login user belum benar dan memberikan hak akses yang sesuai dengan login user. Dapat dilihat pada gambar dihalaman berikut



Gambar 4.13 Activity Diagram Validasi

### 4.1.3 Sequence Diagram

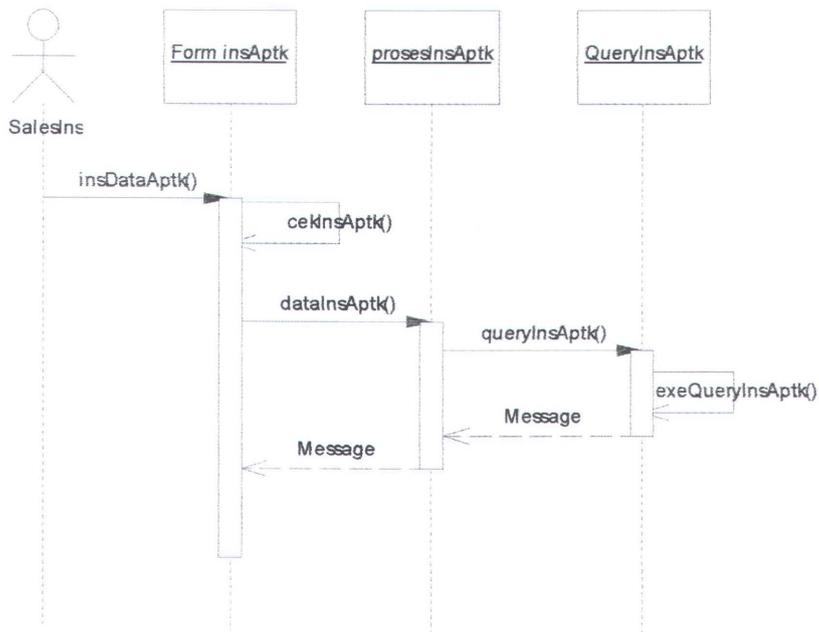
Berdasarkan *use case diagram* untuk model proses bisnisnya dan *activity diagram* untuk alur sistem yang akan dibangun, desain aplikasi berlanjut pada pembuatan *sequence diagram*. *Sequence diagram* yang didefinisikan di bawah ini merupakan skenario jalannya sistem.

#### 4.1.3.1 Sequence Diagram insert data apotek

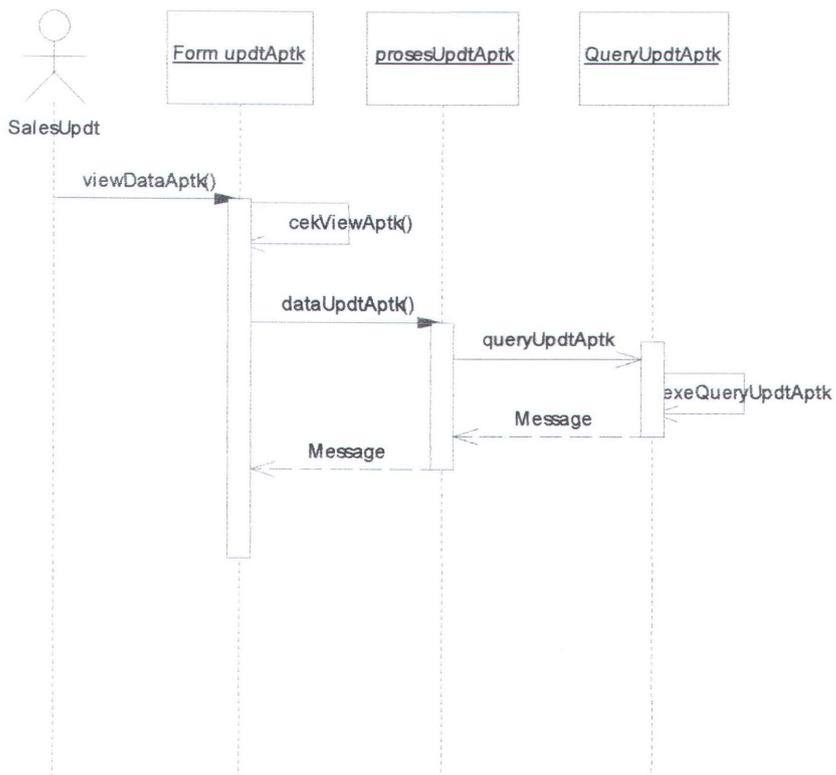
Skenario insert data apotek dimulai saat *sales* memilih menu manajemen apotek. Setelah *sales* memasukkan data apotek, proses *cekInsAptk()* dijalankan, bila data yang dimasukkan sudah sesuai dilanjutkan dengan proses *InsAptk()* yang menghasilkan query dan dieksekusi oleh *QueryInsAptk()* yang nantinya akan mengembalikan variabel berhasil atau tidak. *Sequence Diagram* insert data apotek dapat dilihat pada gambar 4.14.

#### 4.1.3.2 Sequence Diagram update data apotek

*Sequence* update data apotek diawali setelah *sales* memilih data apotek yang akan di-*update*. Proses *cekViewAptk()* dijalankan untuk memvalidasi data masukan yang akan dijadikan data pengganti. Kemudian dijalankan proses *UpdtAptk()* yang menghasilkan query yang akan dieksekusi oleh *QueryUpdtAptk()* yang akan menampilkan pesan bahwa data telah ter-*update*. Dapat dilihat pada gambar 4.15



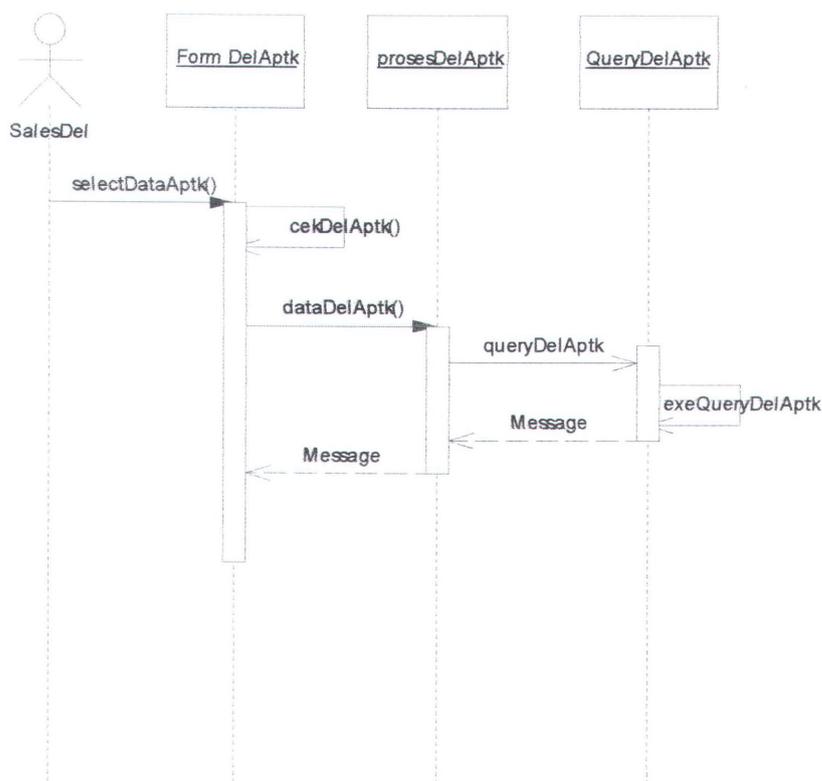
Gambar 4.14 *Sequence Diagram insert data apotek*



Gambar 4.15 *Sequence Diagram Update data apotek*

#### 4.1.3.3 Sequence Diagram delete data apotek

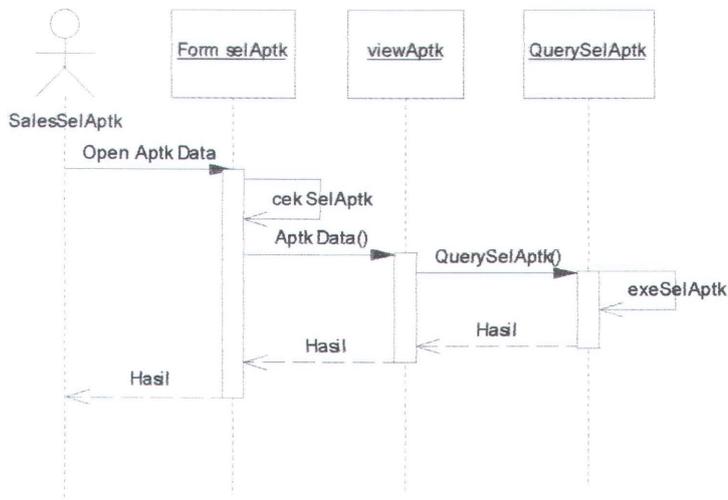
*Sequence Diagram Delete* data apotek diawali setelah *sales* melihat data yang tersedia dalam *database*. Setelah *sales* memilih data yang akan dihapus, akan diberikan konfirmasi yang menanyakan apakah *sales* benar-benar ingin menghapus data tersebut. Kemudian dijalankan *prosesDelAptk()* yang menghasilkan *query delete* dan akan dieksekusi oleh *QueryDelAptk()*. Dan menampilkan pesan bahwa data telah terhapus. Berikut adalah gambar *sequence diagram delete data apotek*



Gambar 4.16 *Sequence Diagram delete data apotek*

#### 4.1.3.4 Sequence Diagram lihat data apotek

*Sequence Diagram* lihat data apotek diawali ketika *sales* memilih menu manajemen apotek yang menampilkan *form* untuk melihat data apotek, kemudian dijalankan *Aptk Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelAptk()*. Berikut adalah *sequence diagram* lihat data apotek.



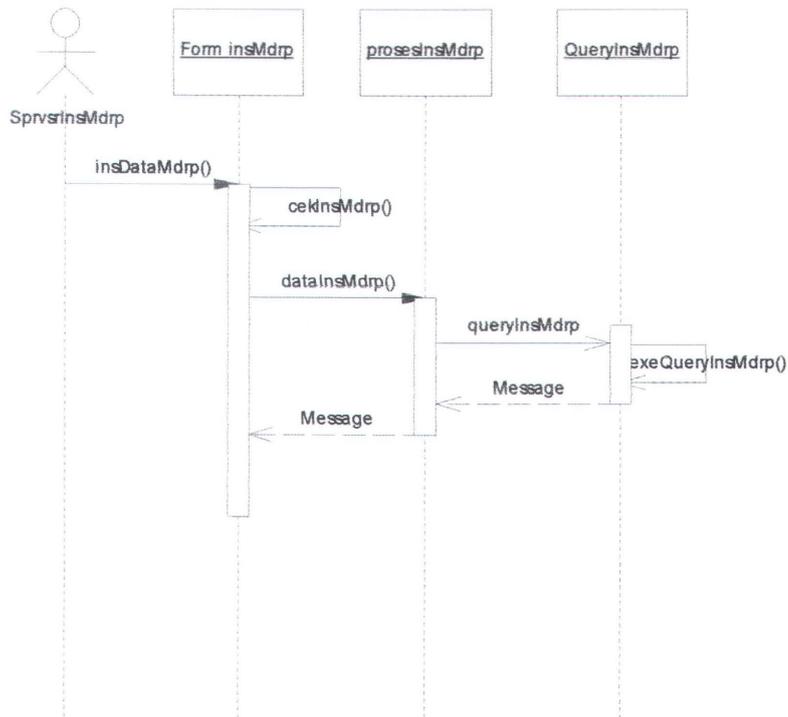
Gamabar 4.17 *Seuence Diagram* lihat data apotek

#### 4.1.3.5 *Sequence Diagram insert data medrep*

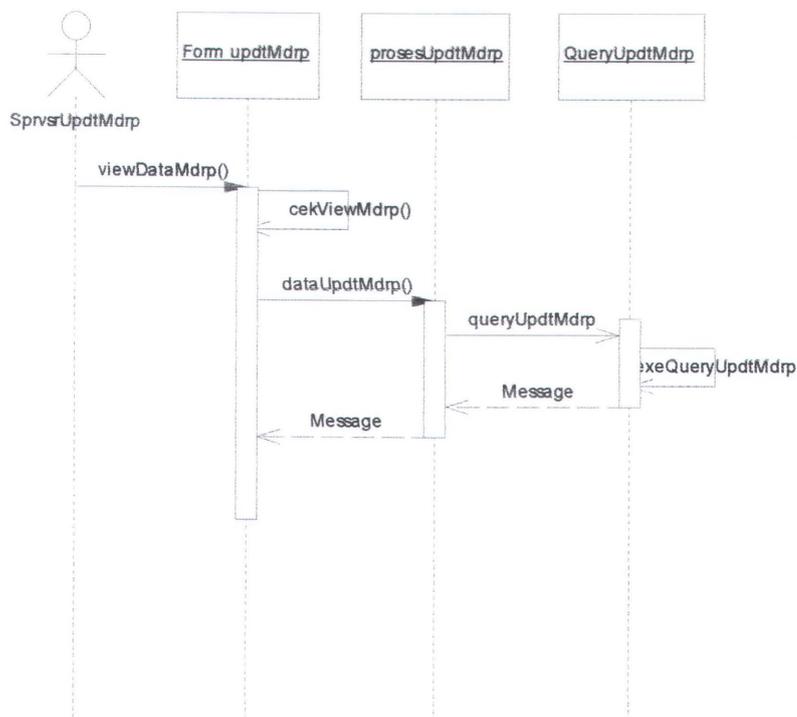
Skenario insert data *medrep* dimulai saat supervisor memilih menu manajemen *medrep*. Setelah supervisor memasukkan data apotek, proses *cekInsMdrp()* dijalankan, bila data yang dimasukkan sudah sesuai dilanjutkan dengan proses *InsMdrp()* yang menghasilkan query dan dieksekusi oleh *QueryInsMdrp()* yang nantinya akan mengembalikan variabel berhasil atau tidak. *Sequence Diagram* insert data *medrep* dapat dilihat pada gambar 4.1.8.

#### 4.1.3.6 *Sequence Diagram update data medrep*

*Sequence* update data *medrep* diawali setelah supervisor memilih data *medrep* yang akan di-*update*. Proses *cekViewMdrp()* dijalankan untuk memvalidasi data masukan yang akan dijadikan data pengganti. Kemudian dijalankan proses *UpdtMdrp()* yang menghasilkan query yang akan dieksekusi oleh *QueryUpdtMdrp()* yang akan menampilkan pesan bahwa data telah ter-*update*. *Sequence Diagram* update data *medrep* dapat dilihat pada 4.1.9.



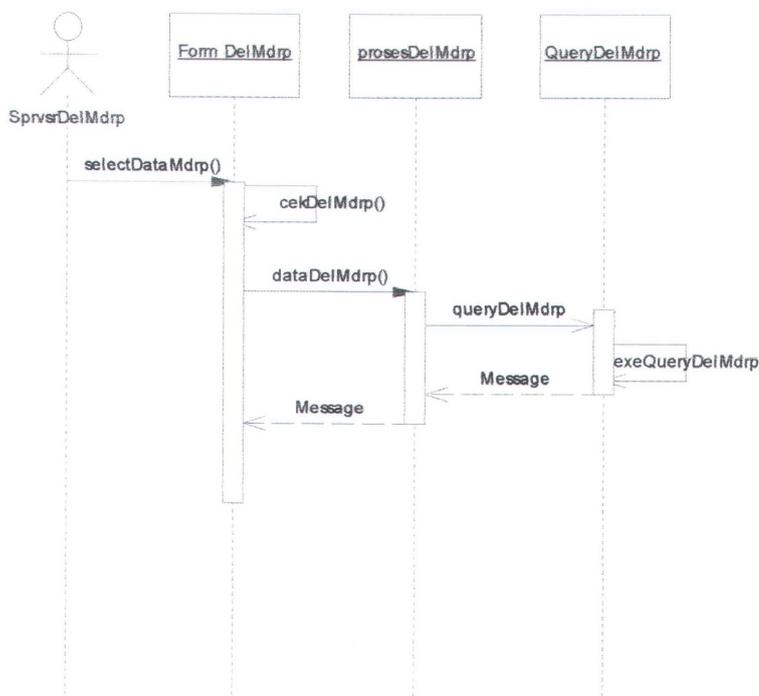
Gambar 4.18 Sequence Diagram insert data medrep



Gambar 4.19 Sequence Diagram Update data medrep

#### 4.1.3.7 Sequence Diagram delete data medrep

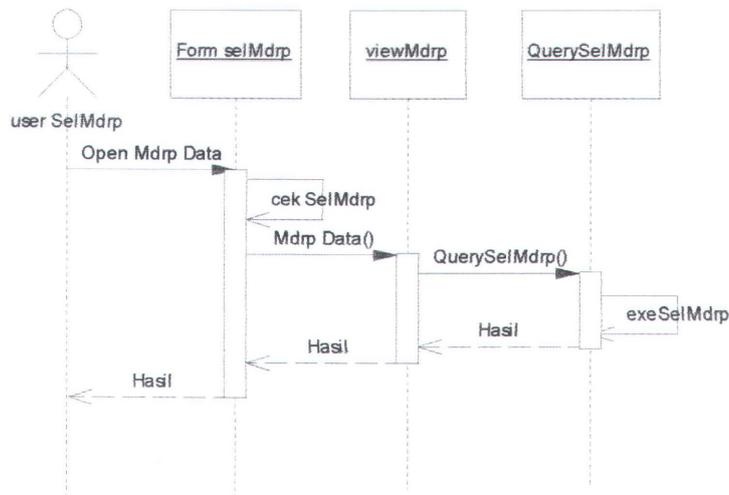
*Sequence Diagram Delete data medrep* diawali setelah supervisor melihat data yang tersedia dalam *database*. Setelah supervisor memilih data yang akan dihapus, akan diberikan konfirmasi yang menanyakan apakah supervisor benar-benar ingin menghapus data tersebut. Kemudian dijalankan *prosesDelMdrp()* yang menghasilkan query delete dan akan dieksekusi oleh *QueryDelMdrp()*. Dan menampilkan pesan bahwa data telah terhapus. *Sequence diagram delete data medrep* dapat dilihat pada gambar 4.20



Gambar 4.20 *Sequence diagram delete data medrep*

#### 4.1.3.8 Sequence Diagram lihat data medrep

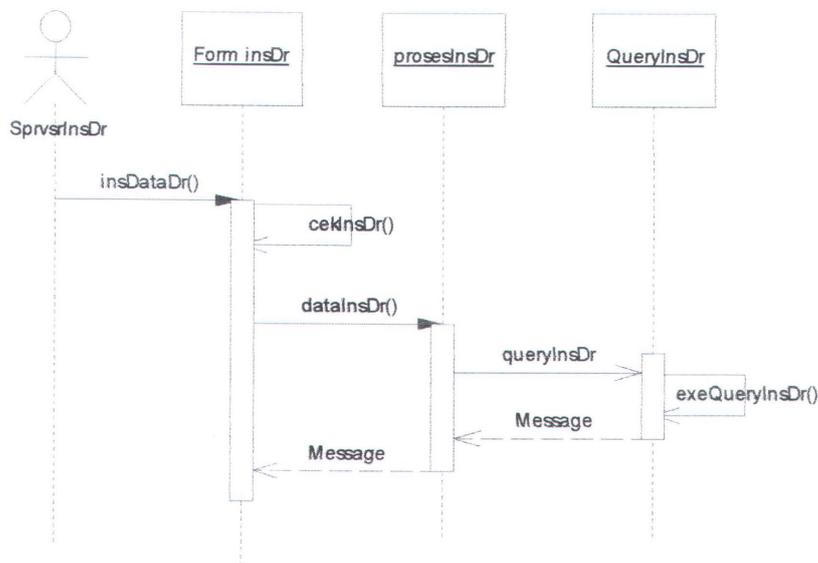
*Sequence Diagram* lihat data apotek diawali ketika *user* yang memiliki hak akses seperti supervisor, manajer atau *medrep* yang bersangkutan memilih menu manajemen *medrep* yang menampilkan *form* untuk melihat data *medrep*, kemudian dijalankan *Mdrp Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelMdrp()*. Berikut adalah *sequence diagram* lihat data *medrep*.



Gambar 4.21 *Sequence diagram* lihat data medrep

#### 4.1.3.9 *Sequence Diagram* insert data dokter

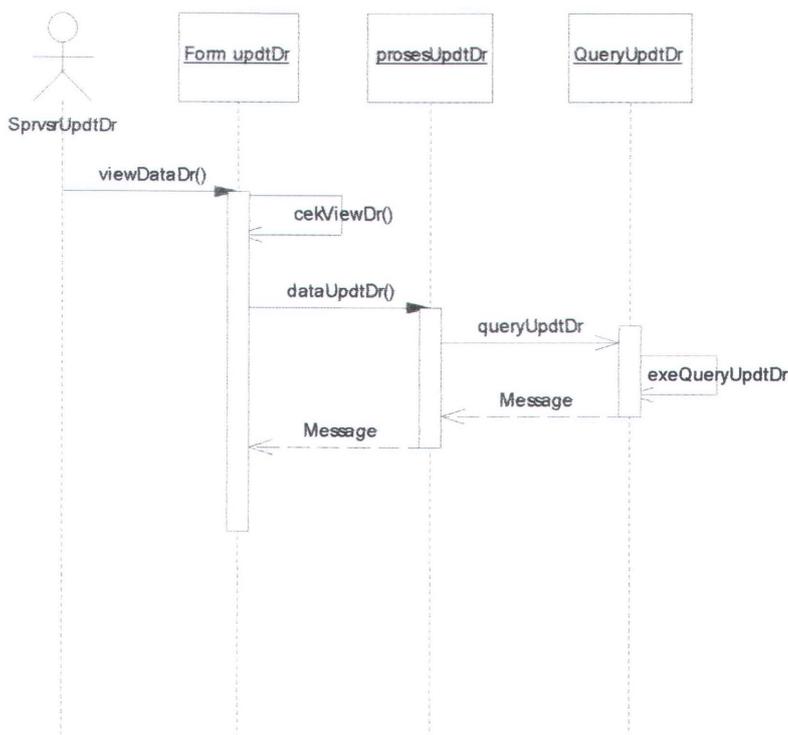
Skenario insert data *medrep* dimulai saat supervisor memilih menu manajemen dokter. Setelah supervisor memasukkan datadokter, proses *cekInsDr()* dijalankan, bila data yang dimasukkan sudah sesuai dilanjutkan dengan proses *InsDr()* yang menghasilkan query dan dieksekusi oleh *QueryInsDr()* yang nantinya akan mengembalikan variabel berhasil atau tidak. *Sequence Diagram* insert data dokter dapat dilihat pada gambar 4.22



Gambar 4.22 *Sequence diagram* insert data dokter

#### 4.1.3.10 Sequence Diagram update data dokter

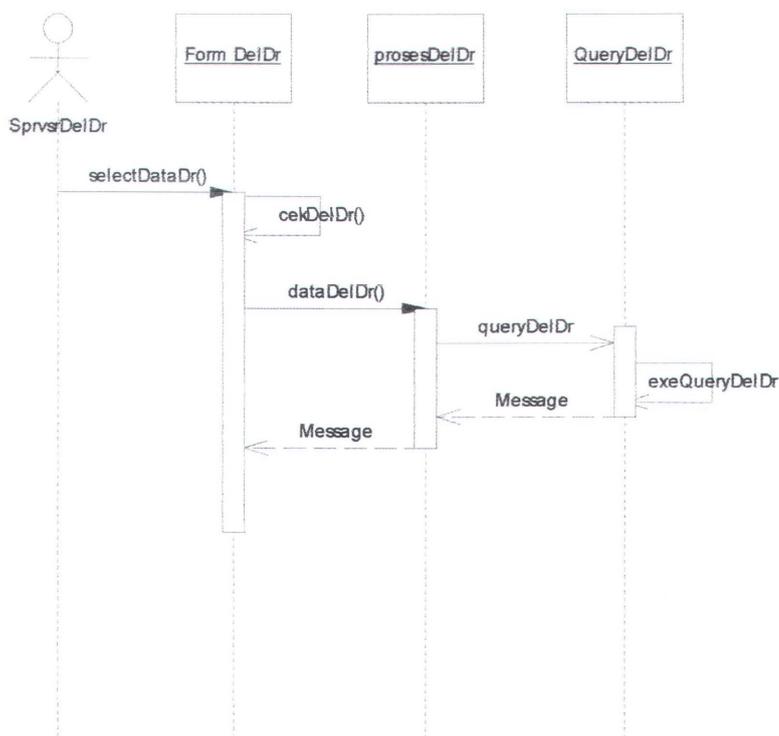
Sequence update data dokter diawali setelah supervisor memilih data dokter yang akan di-update. Proses *cekViewDr()* dijalankan untuk validasi data yang akan dijadikan data pengganti. Dilanjutkan proses *UpdtDr()* yang menghasilkan query yang dieksekusi oleh *QueryUpdtDr()* yang akan menampilkan pesan bahwa data telah ter-update. Sequence Diagram update data dokter dapat dilihat pada gambar berikut



Gambar 4.23 Sequence Diagram Update data dokter

#### 4.1.3.11 Sequence Diagram delete data dokter

Sequence Diagram Delete data dokter diawali setelah supervisor melihat data yang tersedia dalam database. Setelah supervisor memilih data yang akan dihapus, akan diberikan konfirmasi yang menanyakan apakah supervisor benar-benar ingin menghapus data tersebut. Kemudian dijalankan *prosesDelDr()* yang menghasilkan query delete dan dieksekusi oleh *QueryDelDr()*. Dan menampilkan pesan bahwa data telah terhapus. Sequence diagram delete data dokter dapat dilihat pada gambar berikut



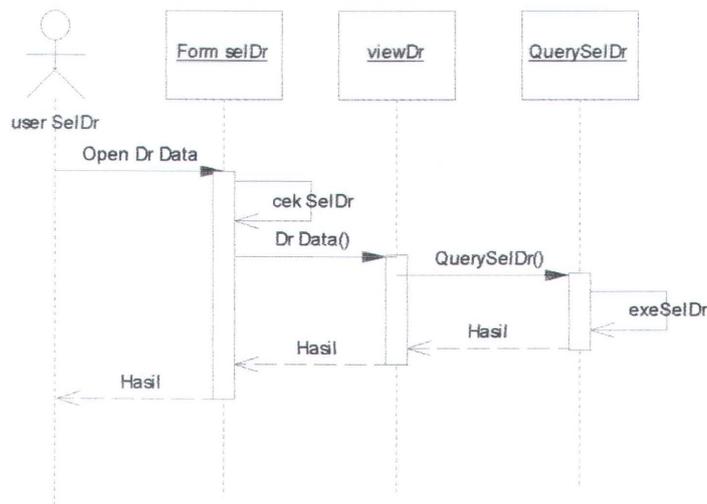
Gambar 4.24 *Sequence Diagram delete data dokter*

#### 4.1.3.12 *Sequence Diagram* lihat data dokter

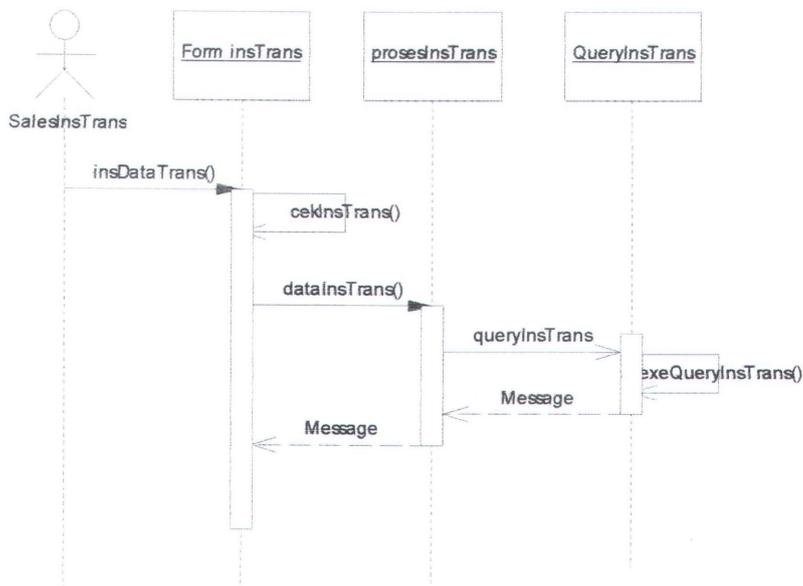
*Sequence Diagram* lihat data dokter diawali ketika *user* yang memiliki hak akses seperti supervisor, manajer atau *medrep* memilih menu manajemen dokter yang menampilkan *form* untuk melihat data dokter, kemudian dijalankan *Dr Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelDr()*. *Sequence diagram* lihat data dokter dapat dilihat pada gambar 4.25.

#### 4.1.3.13 *Sequence Diagram insert data transaksi*

Skenario insert data transaksi dimulai saat *sales* memilih menu manajemen transaksi. Setelah *sales* memasukkan data transaksi, proses *cekInsTrans()* dijalankan, bila data yang dimasukkan sudah sesuai dilanjutkan dengan proses *InsTrans()* yang menghasilkan *query* dan dieksekusi oleh *QueryInsTrans()* yang nantinya akan mengembalikan variabel berhasil atau tidak. *Sequence Diagram* insert data apotek dapat dilihat pada gambar 4.26



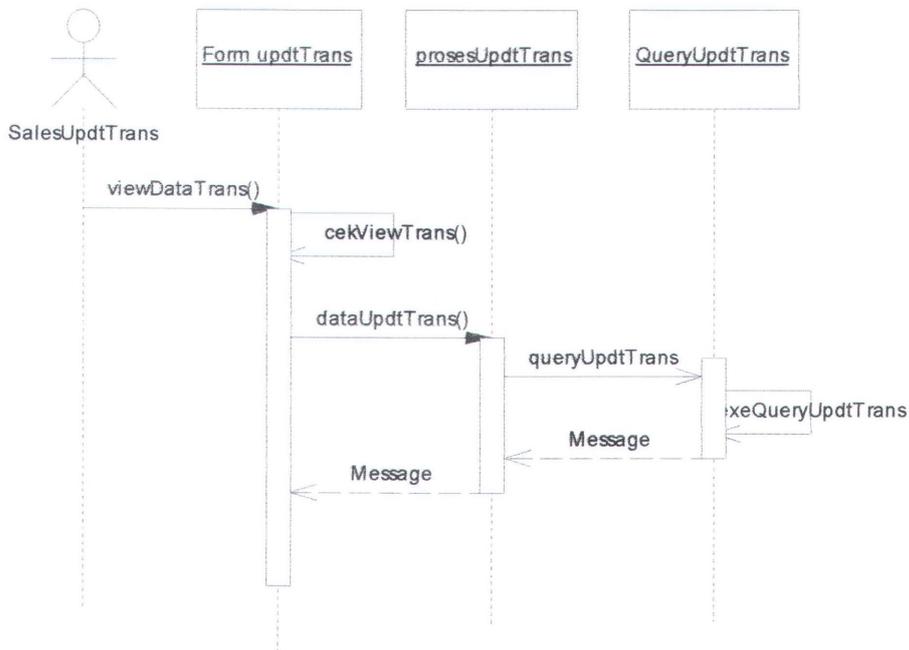
Gambar 4.25 Sequence Diagram lihat data dokter



Gambar 4.26 Sequence diagram insert data transaksi

#### 4.1.3.14 Sequence Diagram update data transaksi

Sequence update data transaksi diawali setelah sales memilih data transaksi yang akan di-update. Proses *cekViewTrans()* memvalidasi data masukan yang dijadikan data pengganti. Dilanjutkan proses *UpdtTrans()* yang menghasilkan query yang dieksekusi oleh *QueryUpdtTrans()* yang akan menampilkan pesan bahwa data telah ter-update. Dapat dilihat pada gambar di halaman berikut



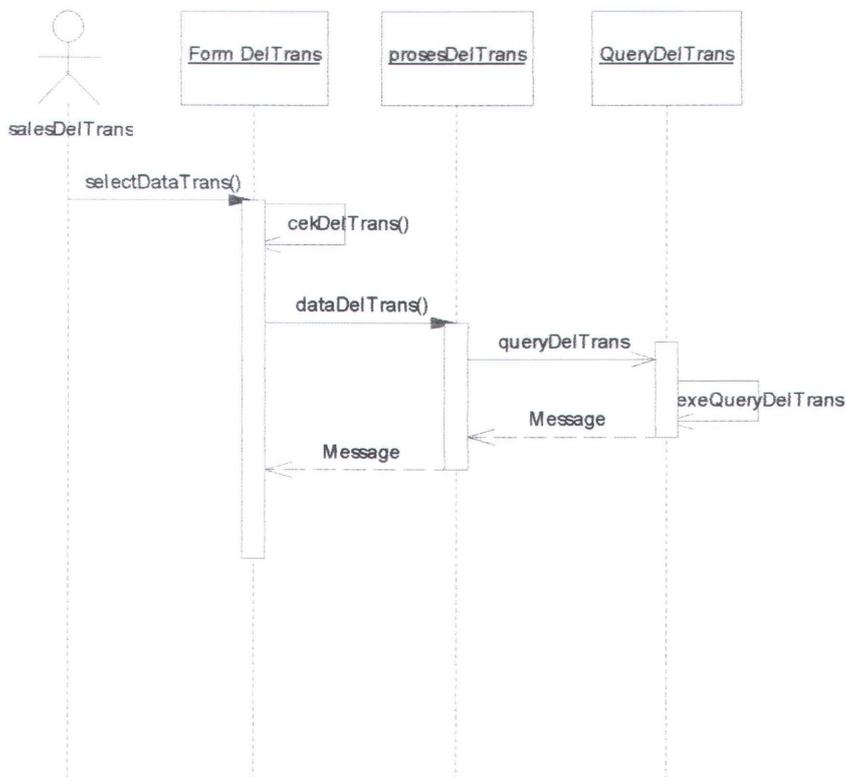
Gambar 4.27 *Sequence diagram update data transaksi*

#### 4.1.3.15 *Sequence Diagram delete data transaksi*

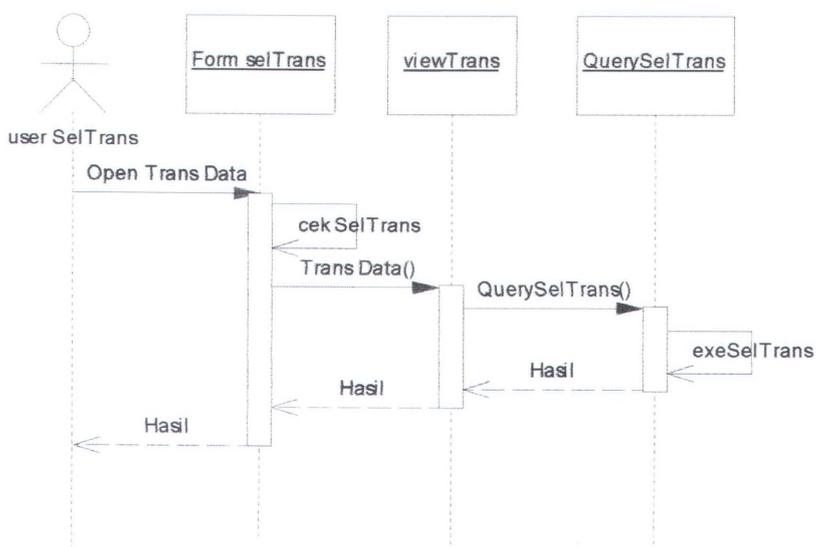
*Sequence Diagram Delete data transaksi* diawali setelah *sales* melihat data yang tersedia dalam *database*. Setelah *sales* memilih data yang akan dihapus, akan diberikan konfirmasi yang menanyakan apakah *sales* benar-benar ingin menghapus data tersebut. Kemudian dijalankan *prosesDelTrans()* yang menghasilkan query delete dan akan dieksekusi oleh *QueryDelTrans()*. Dan menampilkan pesan bahwa data terhapus. *Sequence diagram delete data transaksi* dapat dilihat pada gambar 4.28

#### 4.1.3.16 *Sequence Diagram lihat data transaksi*

*Sequence Diagram lihat data transaksi* diawali ketika *user* yang memiliki hak akses seperti *sales*, *supervisor*, *manajer* atau *medrep* memilih menu manajemen dokter yang menampilkan *form* untuk melihat data transaksi, kemudian dijalankan *Trans Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelTrans()*. *Sequence diagram lihat data transaksi* dapat dilihat pada gambar 4.29.



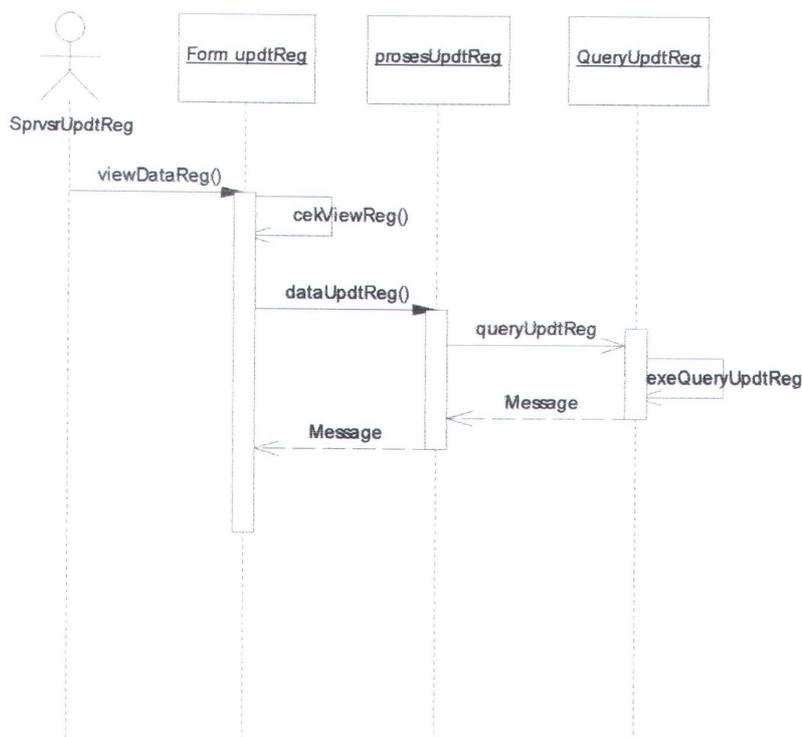
Gambar 4.28 Sequence diagram delete data transaksi



Gambar 4.29 Sequence diagram lihat data transaksi

#### 4.1.3.17 Sequence Diagram update data region

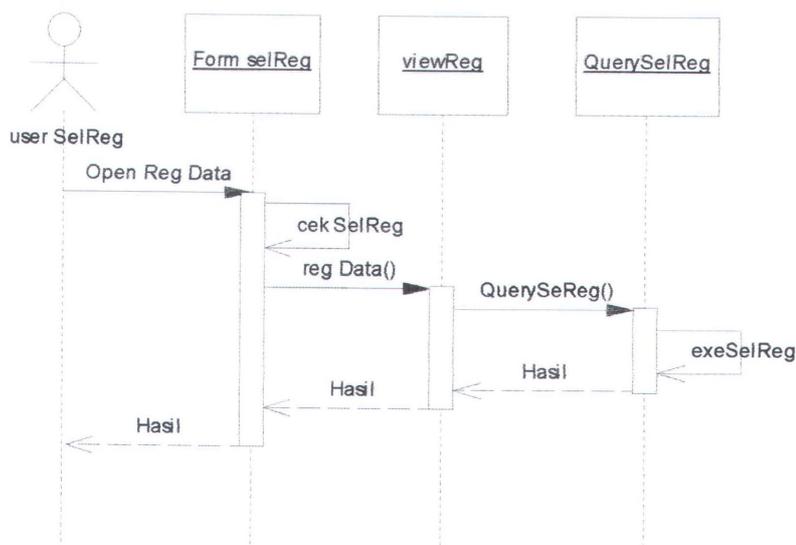
*Sequence* update data region diawali setelah supervisor memilih data region yang akan di-update. Proses *cekViewReg()* dijalankan untuk memvalidasi data masukan yang akan dijadikan data pengganti. Kemudian dijalankan proses *UpdtReg()* yang menghasilkan query yang akan dieksekusi oleh *QueryUpdtReg()* yang akan menampilkan pesan bahwa data telah ter-update. *Sequence Diagram* update data region dapat dilihat pada gambar berikut



Gambar 4.30 *Sequence diagram* update data region

#### 4.1.3.18 Sequence Diagram lihat data region

*Sequence Diagram* lihat data region diawali ketika supervisor memilih menu manajemen region yang menampilkan *form* untuk melihat data region, kemudian dijalankan *reg Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelReg()*. Berikut adalah *sequence diagram* lihat data region.



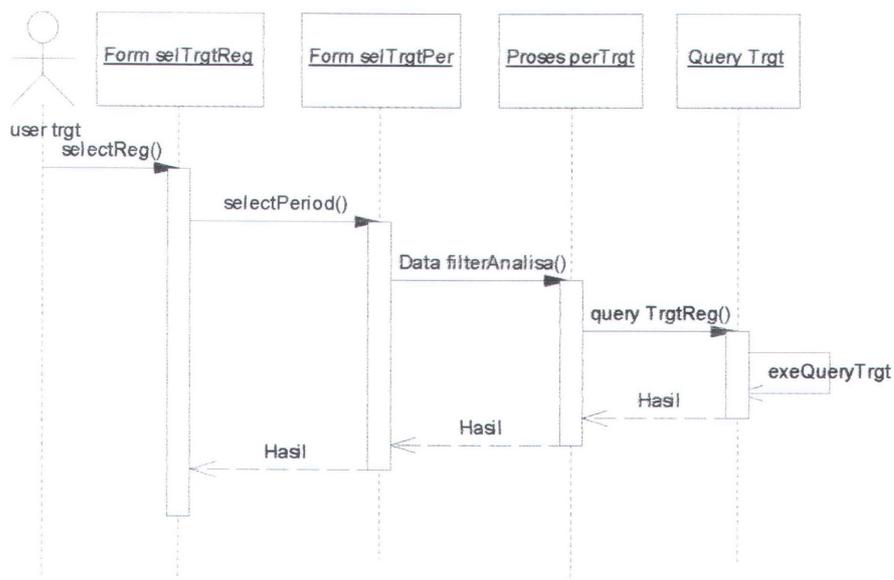
Gambar 4.31 *Sequence diagram* lihat data region

#### 4.1.3.19 *Sequence Diagram* lihat Target Penjualan

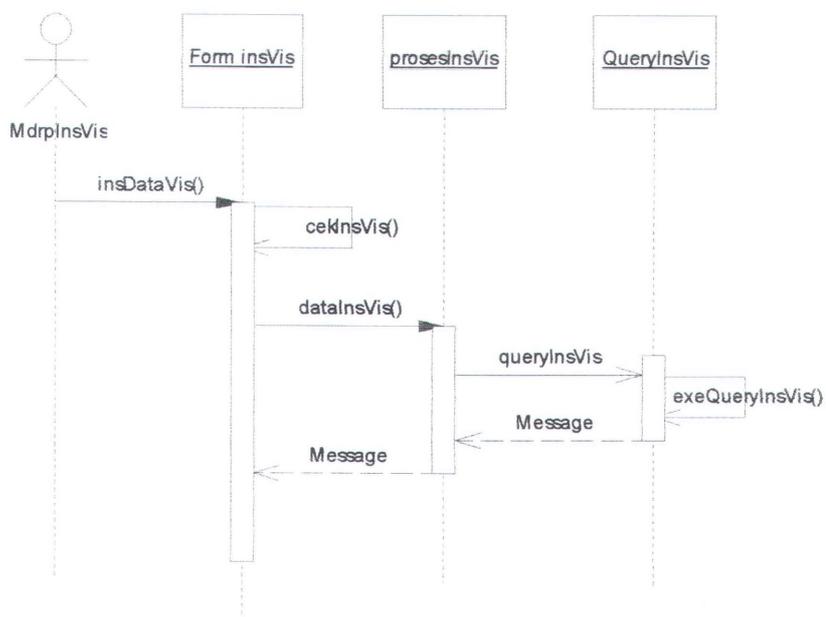
*Sequence Diagram* lihat data region diawali ketika *user* yang memiliki hak akses seperti supervisor, manajer atau *medrep* memilih menu manajemen region atau analisa *medrep* yang menampilkan *form* untuk melihat data region, untuk manajer dilanjutkan dengan proses memilih region, kemudian user memilih periode waktu untuk target, kemudian dijalankan *DataFilterAnalisa()* yang menghasilkan *query select* yang akan dieksekusi oleh *QueryTrgtReg()* yang akan mengembalikan proses dengan menampilkan hasil seperti yang dipilih *user*. *Sequence diagram* lihat target penjualan dapat dilihat pada gambar 4.32

#### 4.1.3.20 *Sequence Diagram* insert data visitasi

Skenario insert data visitasi dimulai saat *medrep* memilih menu manajemen visitasi. Setelah *medrep* memasukkan data avisitasi, proses *cekInsVis()* dijalankan, bila data yang dimasukkan sudah sesuai dilanjutkan dengan proses *InsVis()* yang menghasilkan query dan dieksekusi oleh *QueryInsVis()* yang nantinya akan mengembalikan variabel berhasil atau tidak. *Sequence Diagram* insert data visitasi dapat dilihat pada gambar 4.33



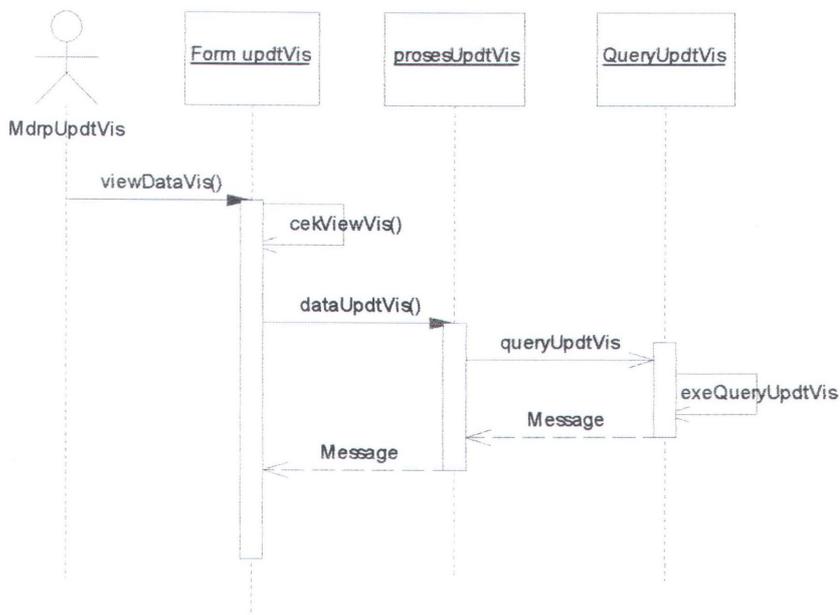
Gambar 4.32 *Sequence diagram* lihat target penjualan



Gambar 4.33 *Sequence diagram* insert data visitasi

#### 4.1.3.21 Sequence Diagram update data visitasi

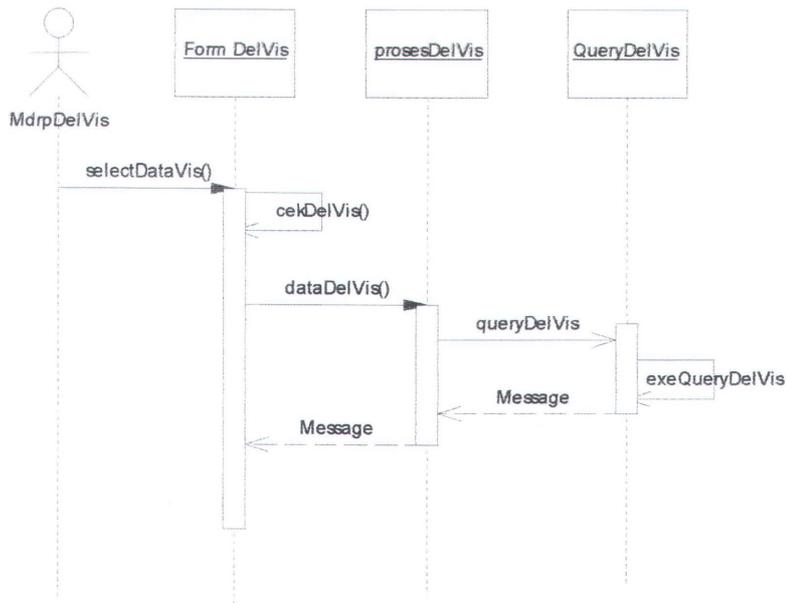
Sequence update data visitasi diawali setelah *medrep* memilih data visitasi yang akan di-*update*. Proses *cekViewVis()* dijalankan untuk memvalidasi data masukan yang akan dijadikan data pengganti. Kemudian dijalankan proses *UpdtVis()* yang menghasilkan query yang akan dieksekusi oleh *QueryUpdtVis()* yang akan menampilkan pesan bahwa data telah ter-*update*. Dapat dilihat pada gambar 4.34



Gambar 4.34 Sequence diagram update data visitasi

#### 4.1.3.22 Sequence Diagram delete data visitasi

Sequence Diagram Delete data visitasi diawali setelah *medrep* melihat data yang tersedia dalam *database*. Setelah *medrep* memilih data yang akan dihapus, akan diberikan konfirmasi yang menanyakan apakah *medrep* benar-benar ingin menghapus data tersebut. Kemudian dijalankan *prosesDelVis()* yang menghasilkan query delete dan akan dieksekusi oleh *QueryDelVis()*. Dan menampilkan pesan bahwa data telah terhapus. Berikut adalah gambar *sequence diagram delete* data visitasi



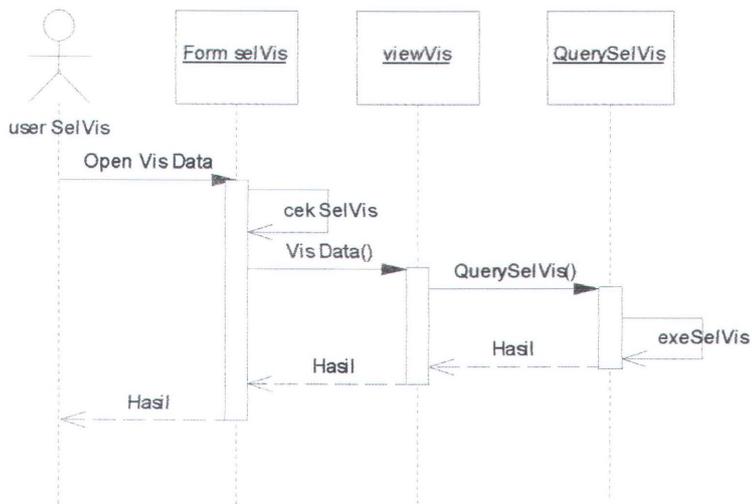
Gambar 4.35 Sequence diagram delete data visitasi

#### 4.1.3.23 Sequence Diagram lihat data visitasi

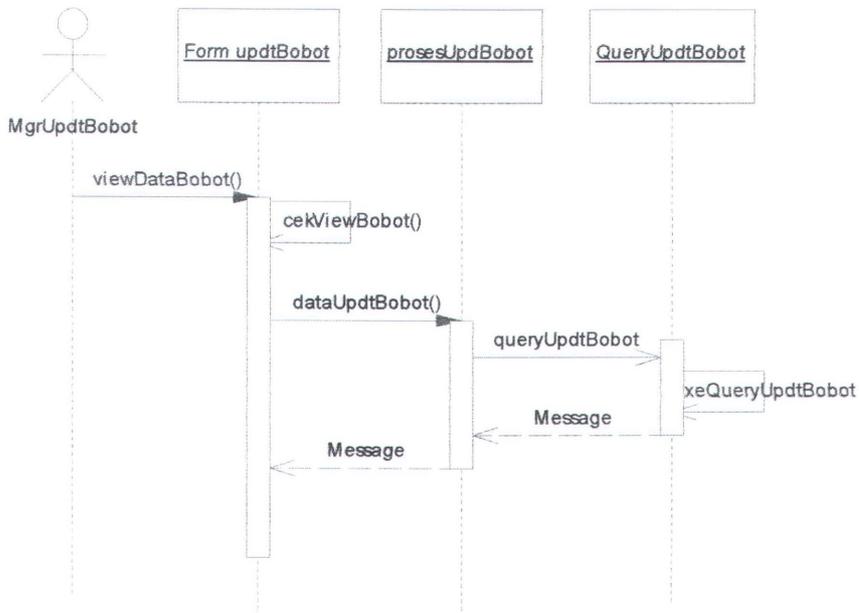
Sequence Diagram lihat data visitasi diawali ketika *user* yang memiliki hak akses seperti supervisor, manajer atau *medrep* memilih memilih menu manajemen visitasi atau analisa kinerja *medrep* yang menampilkan *form* untuk melihat data visitasi, kemudian dijalankan *Vis Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelVis()*. Sequence diagram lihat data visitasi dapat dilihat pada gambar 4.36.

#### 4.1.3.24 Sequence Diagram update bobot perhitungan

Sequence update bobot perhitungan diawali setelah manager memilih bobot perhitungan yang akan di-update. Proses *cekViewBobot()* dijalankan untuk memvalidasi data masukan yang akan dijadikan data pengganti. Kemudian dijalankan proses *UpdtBobot()* yang menghasilkan query yang akan dieksekusi oleh *QueryUpdtBobot()* yang akan menampilkan pesan bahwa data telah ter-update. Sequence Diagram update bobot perhtiungan dapat dilihat pada gambar 4.37



Gambar 4.36 *Sequence diagram* lihat data visitasi



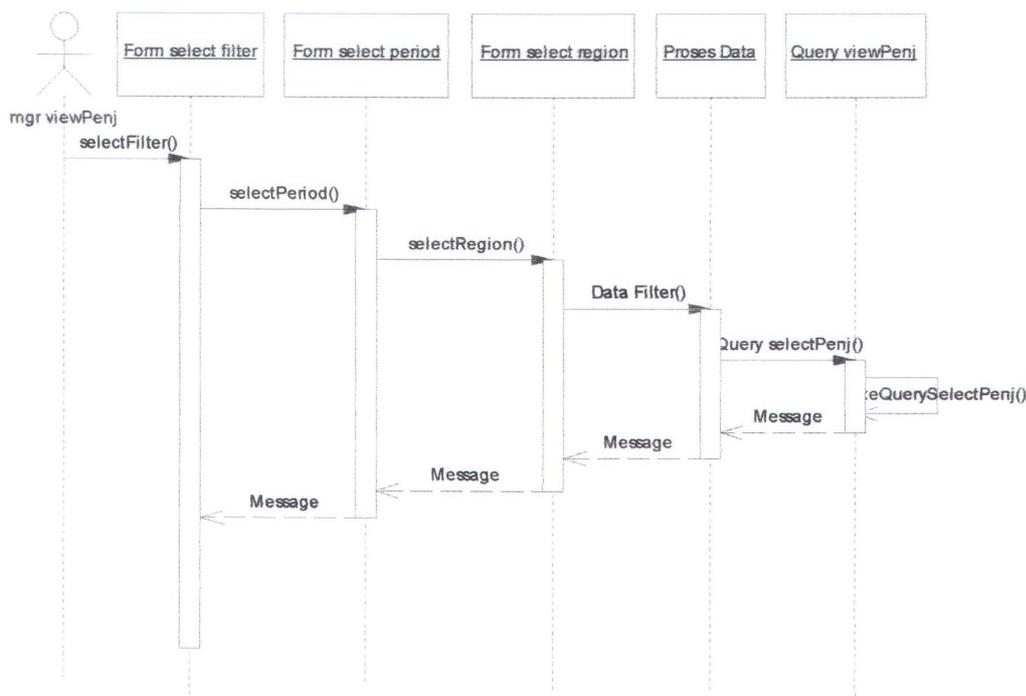
Gambar 4.37 *Sequence diagram* update bobot perhitungan

#### 4.1.3.25 *Sequence Diagram* lihat data penjualan

*Sequence Diagram* lihat data penjualan diawali setelah *user* memilih menu laporan penjualan. Kemudian *user* memilih menu yang ditampilkan dalam *Form selectFilter()* yang berbeda untuk setiap jenis *user*. Untuk *user* yang tergabung dalam grup *sales* dan apotek dapat melihat detail laporan penjualan untuk tiap obat dan apotek, sedangkan *user* yang berada dalam grup *medrep* dan supervisor dapat melihat detail penjualan untuk tiap obat dan dokter, sedangkan manajer dapat melihat keseluruhan dari filter yang tersedia. Kemudian *user* memilih periode waktu laporan penjualan untuk tiap dua minggu dalam *Form selectPeriod()*. Terakhir akan tampil *Form selectReg()* untuk memilih laporan penjualan dari region mana yang akan ditampilkan, kecuali manajer, *user* yang lain hanya dapat melihat laporan penjualan pada regionnya. Setelah itu akan dilanjutkan dengan *Proses Data()* yang akan menghasilkan query yang akan dieksekusi oleh *QueryViewPenj*. Kemudian akan ditampilkan halaman sesuai filter yang dipilih. *Sequence diagram* lihat data penjualan dapat dilihat pada gambar 4.38

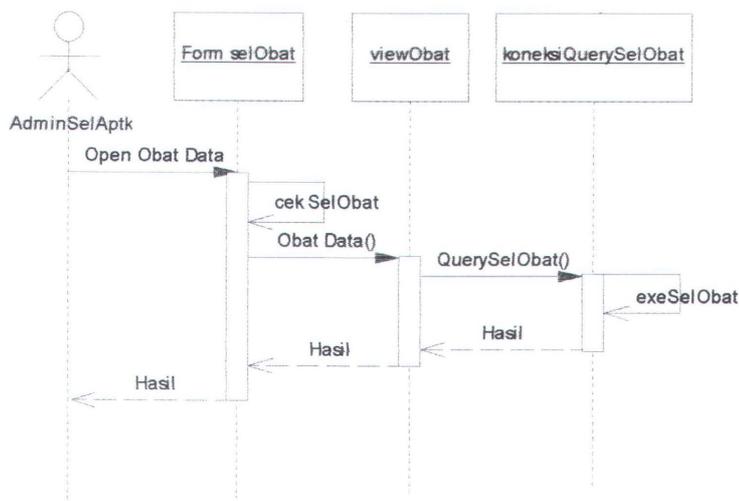
#### 4.1.3.26 *Sequence Diagram* analisa kinerja *medrep*

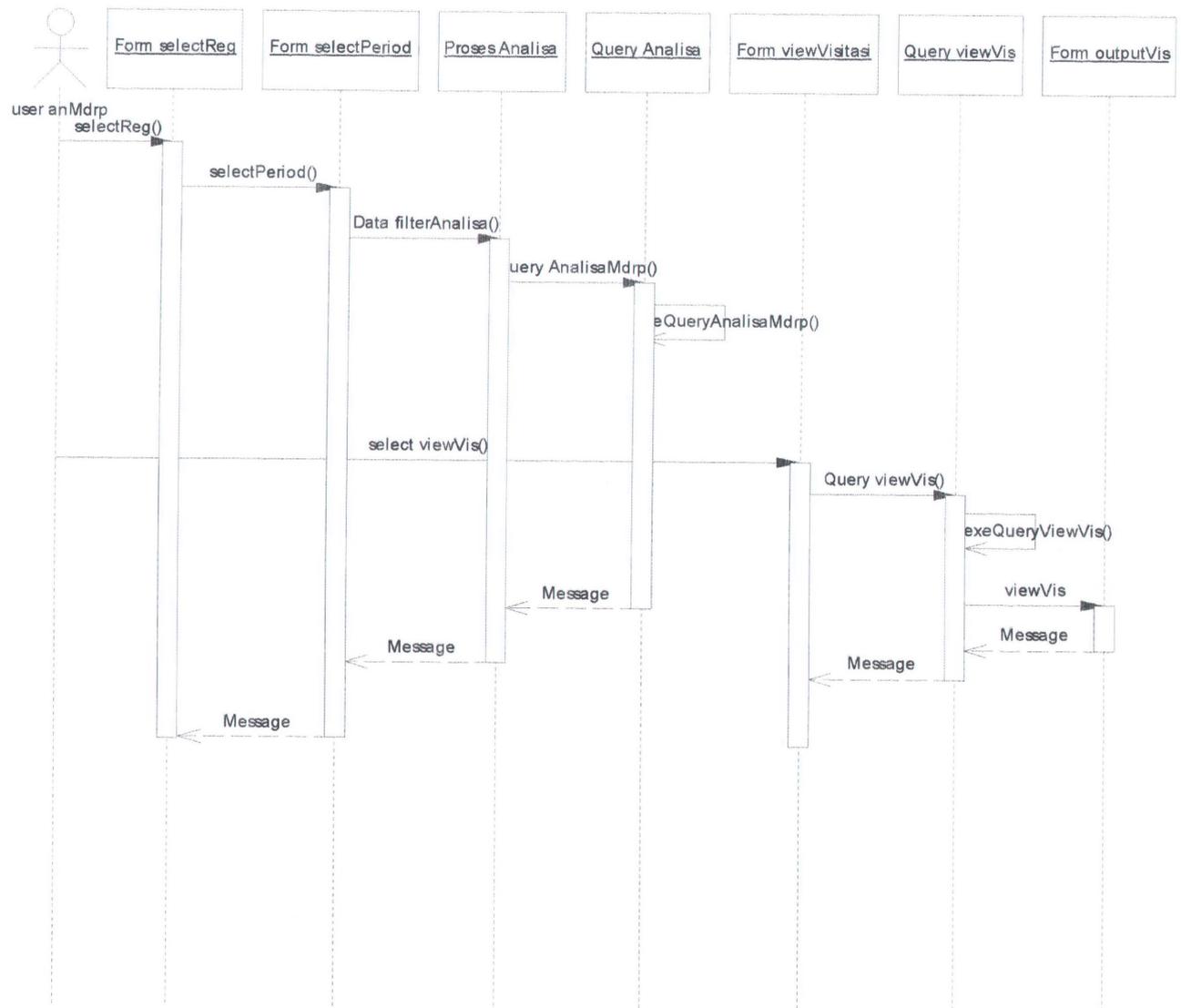
*Sequence Diagram* analisa kinerja *medrep* diawali ketika *user* memilih menu analisa kinerja *medrep*. Menu ini hanya dapat diakses oleh manajer, supervisor dan *medrep* yang bersangkutan, namun hanya manajer yang dapat melihat analisa kinerja *medrep* dari seluruh region seperti yang ditampilkan dalam *Form selectReg()*. Kemudian *user* memilih periode evaluasi yang diinginkan seperti dalam *Form selectPeriod()*. Data dari pilihan *user* akan dilanjutkan dengan proses analisa yang akan diteruskan menuju *QueryAnalisa()*. Kemudian *user* dapat menggunakan link detail visitasi jika ingin melihat detail visitasi dari *medrep* yang bersangkutan. Dari proses tersebut akan dilanjutkan dengan *QueryViewVis()* yang nantinya akan diteruskan dengan ditampilkannya *Form viewVis*. *Sequence diagram* analisa kinerja *medrep* dapat dilihat pada gambar 4.39.

Gambar 4.38 *Sequence diagram* lihat data penjualan

#### 4.1.3.27 *Sequence Diagram* lihat data obat

*Sequence Diagram* lihat data obat diawali ketika *user* yang memiliki hak akses sebagai admin, memilih menu manajemen obat yang menampilkan *form* untuk melihat data obat, kemudian dijalankan *obat Data()* yang menghasilkan *query select* yang akan dieksekusi oleh *QuerySelObat()*. Dapat dilihat pada gambar berikut

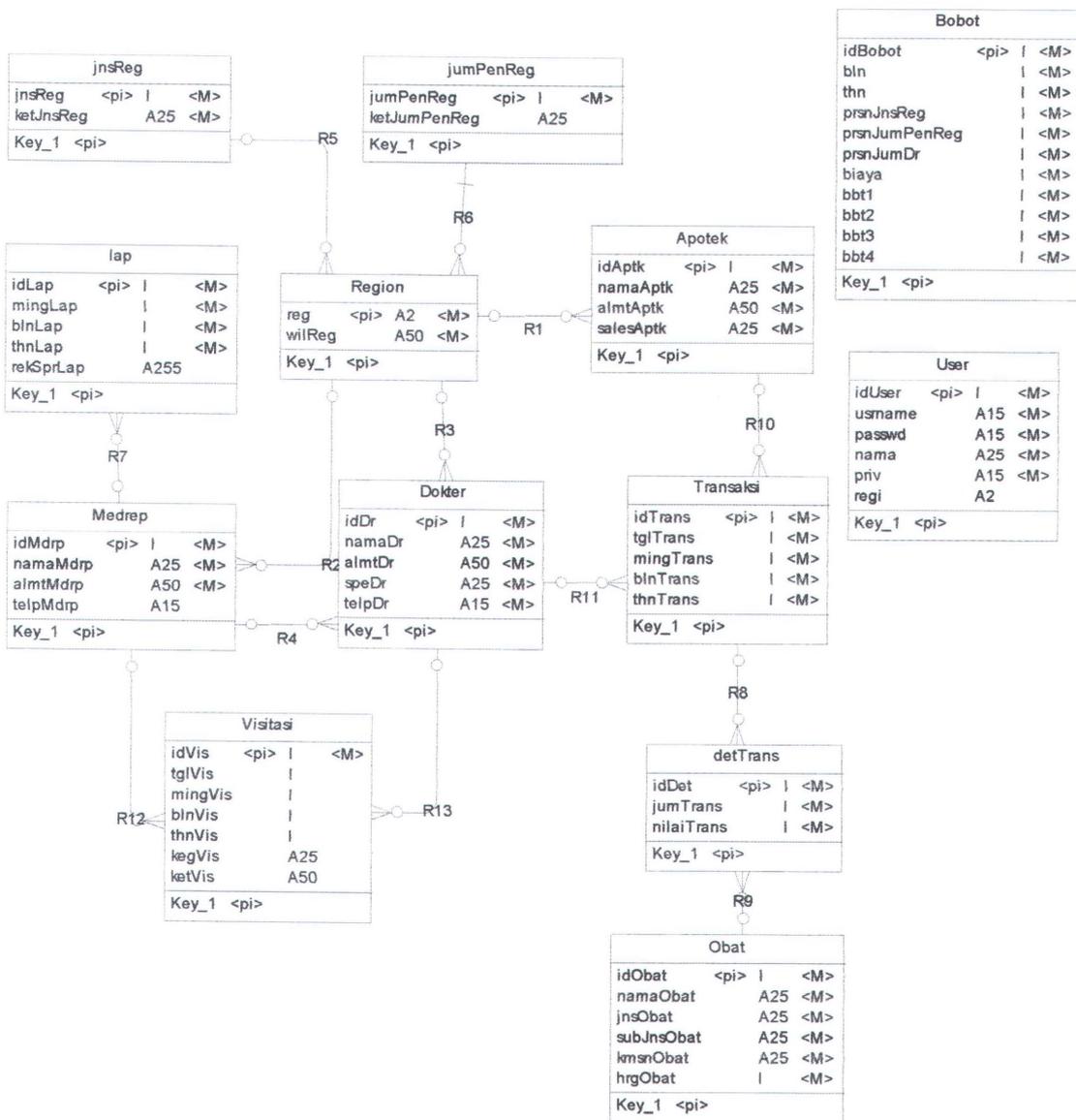
Gambar 4.39 *Sequence diagram* lihat data obat



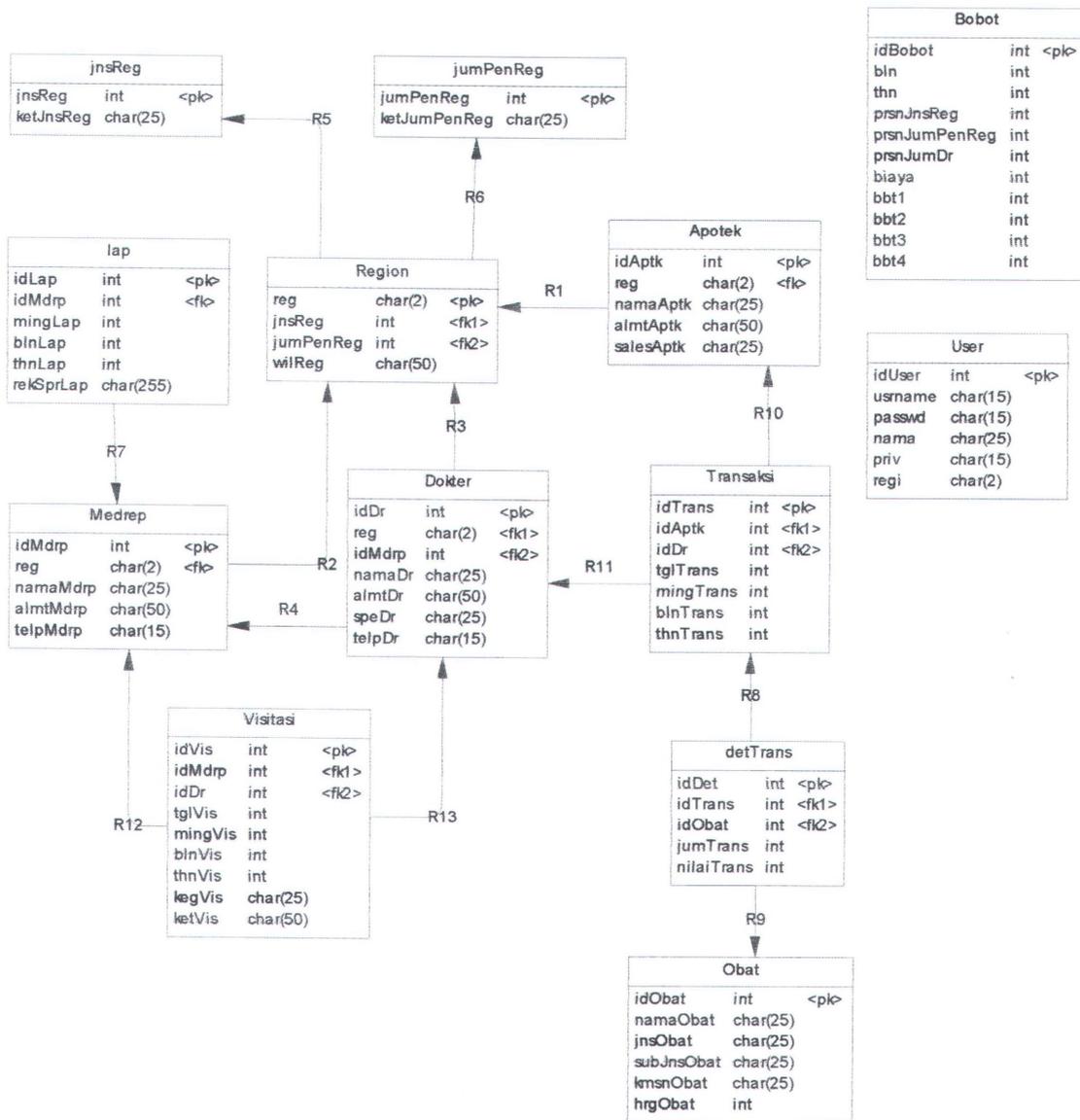
Gambar 4.40 Sequence diagram analisa kinerja medrep

### 4.1.4 Entity Relationship Diagram

Entity Relationship Diagram dibangun untuk menggambarkan database yang akan digunakan sebagai dasar untuk membangun aplikasi. Diagram ERD ini dibangun dengan menggunakan tools Power Designer 9. Berikut adalah diagram dari Conceptual Data Model dan Physical Data Model.



Gambar 4.4 Conceptual Data Model



Gambar 4.42 Physical Data Model

Secara garis besar, hubungan fungsional antara tabel pada diagram diatas akan dijelaskan sebagai berikut

### 1. jnsReg

Adalah tabel yang berisi nilai dari tiap jenis region beserta keterangan yang menjelaskannya. Berhubungan langsung dengan tabel region. Deskripsi tiap *Field* dapat dilihat pada tabel di halaman berikut

Tabel 4.1 Deskripsi Tabel jnsReg

Nama	Jenis	Panjang (max)	Keterangan
JNSREG	Integer	2	(PK), berisi nilai untuk tiap kategori jenis region
KETJNSREG	Char	25	Keterangan untuk tiap kategori

### 2. jumPenReg

Adalah tabel yang berisi nilai dari tiap kategori jumlah penduduk region beserta keterangan yang menjelaskannya. Berhubungan langsung dengan tabel region. Deskripsi tiap *Field* dapat dilihat pada tabel berikut

Tabel 4.2 Deskripsi Tabel jumPenReg

Nama	Jenis	Panjang (max)	Keterangan
JUMPENREG	Integer	2	(PK), berisi nilai untuk tiap kategori jumlah penduduk di region
KETJUMPENREG	Char	25	Keterangan untuk tiap kategori

### 3. Region

Tabel ini berisi data-data untuk tiap region, berhubungan dengan tabel jnsReg dan jumPenReg, tabel ini juga berisi detail nama-nama kecamatan untuk tiap region. keterangan tiap *field* dapat dilihat pada tabel berikut

Tabel 4.3 Deskripsi Tabel Region

Nama	Jenis	Panjang (max)	Keterangan
REG	Char	2	(PK), berisi nama region
JNSREG	Integer	2	(FK), nilai dari jenis region
JUMPENREG	Integer	2	(FK), nilai dari jumlah penduduk region
WILREG	Char	50	Wilayah kerja region

### 4. Obat

Tabel Obat berisi tentang keterangan seperti, nama obat, jenis obat, kemasan, dan harga untuk tiap obat dan tiap kemasan. Tabel ini akan di-*reference* oleh tabel dettrans. keterangan untuk tiap *field* tabel obat dapat dilihat pada halaman berikut

Tabel 4.4 Deskripsi Tabel Obat

Nama	Jenis	Panjang (max)	Keterangan
IDOBAT	Integer	6	(PK), <i>auto increment</i>
NAMAOBAT	Char	25	Nama obat
JNSOBAT	Char	25	Jenis obat (mis antibiotika)
SUBJNSOBAT	Char	25	Sub jenis obat (mis antibiotika syaraf)
KMSNOBAT	Char	25	Kemasan obat
HRGOBAT	Integer	6	Harga untuk masing-masing obat

## 5. Dokter

Sama seperti tabel apotek, tabel dokter berisi data tentang dokter. Deskripsi mengenai tabel dokter dapat dilihat pada deskripsi di bawah.

Tabel 4.5 Deskripsi Tabel Dokter

Nama	Jenis	Panjang (max)	Keterangan
IDDR	Integer	4	(PK), <i>auto increment</i>
REG	Char	2	(FK), Lokasi region dari dokter
IDMDRP	Integer	3	(FK), <i>medrep</i> yang mendidik dokter
NAMADR	Char	25	Nama dokter
ALMTDR	Char	50	Alamat dokter
SPEDR	Char	25	Spesialisasi dokter
TELPDR	Char	15	Nomor telepon dokter

## 6. Medrep

Tabel medrep berisi biodata medrep. Data didapat dari supervisor selaku pimpinan dari Medrep. Dideskripsikan pada tabel berikut

Tabel 4.6 Deskripsi Tabel Medrep

Nama	Jenis	Panjang (max)	Keterangan
IDMDRP	Integer	3	(PK), <i>auto increment</i>
REG	Char	2	(FK), wilayah kerja <i>medrep</i>
NAMAMDRP	Char	25	Nama <i>medrep</i>
ALAMTMDRP	Char	50	Alamat <i>medrep</i>
TELPMDRP	Char	15	Telepon <i>medrep</i>

## 7. Apotek

Tabel ini berisi data-data mengenai apotek yang menjual produk-produk dari Kalbe, keterangan dari tiap *field* dapat dilihat pada tabel berikut

Tabel 4.7 Deskripsi Tabel Apotek

Nama	Jenis	Panjang (max)	Keterangan
IDAPTK	Integer	4	(PK), <i>auto increment</i>
REG	Char	2	(FK), lokasi region apotek
NAMAAPTK	Char	25	Nama apotek
ALMTAPTK	Char	50	Alamat apotek
SALESAPTK	Char	25	Nama <i>sales</i> yang bertanggung jawab

## 8. Lap

Tabel ini berisi rekomendasi yang diberikan supervisor kepada *medrep* tiap dua mingguan. Dijelaskan pada tabel berikut

Tabel 4.8 Deskripsi Tabel Lap

Nama	Jenis	Panjang (max)	Keterangan
IDLAP	Integer	6	(PK), <i>auto increment</i>
IDMDRP	Integr	3	(FK), nama <i>medrep</i> yang dievaluasi
MINGLAP	Integer	2	Minggu laporan dibuat
BLNLAP	Integer	2	Bulan laporan dibuat
THNLAP	Integer	4	Tahun laporan dibuat
REKSPRLAP	Char	255	Isi rekomendasi dari supervisor

## 9. Visitasi

Tabel ini digunakan oleh supervisor untuk mencatat semua kegiatan visitasi yang dilakukan. Dimana pada dasarnya ada 3 jenis kegiatan Medrep yaitu memperkenalkan obat baru, mengingatkan obat lama, dan keiatan kondisional lainnya. *Field* ketVisitasi digunakan untuk menambahkan keterangan dari visitasi yang dilakukan. Dideskripsikan pada tabel di halaman berikut

Tabel 4.9 Deskripsi Tabel Visitasi

Nama	Jenis	Panjang (max)	Keterangan
IDVIS	Integer	7	(PK), <i>auto increment</i>
IDMDRP	Integer	3	(FK), <i>medrep</i> yang melakukan visitasi
IDDR	Integer	4	(FK), dokter yang dikunjungi <i>medrep</i>
TGLVIS	Integer	2	Tanggal visitasi
MINGVIS	Integer	2	Minggu visitasi
BLNVIS	Integer	2	Bulan visitasi
THNVIS	Integer	4	Tahun visitasi
KEGVIS	Char	25	Kegiatan visitasi (mis memperkenalkan obat baru)
KETVIS	Char	50	Keterangan dari kegiatan visitasi (mis memperkenalkan obat baru Bactecyn)

#### 10. Transaksi

Tabel ini digunakan oleh *sales* untuk memasukkan data-data transaksi yang didapat dari apotek, tabel ini merupakan tabel yang sering diakses dan tabel ini akan di-*reference* oleh tabel *dettrans* yang berisi detail dari transaksi tersebut. Dideskripsikan pada tabel berikut

Tabel 4.10 Deskripsi Tabel Transaksi

Nama	Jenis	Panjang (max)	Keterangan
IDTRANS	Integer	11	(PK), <i>auto increment</i>
IDAPTK	Integer	4	(FK), apotek tempat terjadinya transaksi
IDDR	Integer	4	(FK), dokter yang menuliskan resep
TGLTRANS	Integer	2	Tanggal transaksi
MINGTRANS	Integer	2	Minggu transaksi
BLNTRANS	Integer	2	Bulan transaksi
THNTRANS	Integer	4	Tahun transaksi

#### 11. Dettrans

Tabel ini berisi detail penjualan beserta nilai penjualan yang didapat dari sebuah transaksi. Dijelaskan pada tabel di halaman berikut

Tabel 4.11 Deskripsi Tabel Dettrans

Nama	Jenis	Panjang (max)	Keterangan
IDDET	Integer	8	(PK), <i>auto increment</i>
IDTRANS	Integer	7	(FK), nomor transaksi
IDOBAT	Integer	6	(FK), obat transaksi
JUMTRANS	Integer	3	Jumlah obat terjual
NILAITRANS	Integer	7	Nilai obat terjual

## 12. Bobot

Tabel ini berisi bobot yang diputuskan oleh manager setiap bulan. Bobot tersebut berupa persentase yang akan digunakan menghitung target penjualan dari tiap *medrep*. Juga berisi alokasi biaya yang diberikan untuk sekali melakukan kunjungan. Tabel ini tidak berelasi dengan tabel apapun. Dijelaskan pada tabel deskripsi berikut

Tabel 4.12 Deskripsi Tabel Bobot

Nama	Jenis	Panjang (max)	Keterangan
IDBOBOT	Integer	4	(PK), <i>auto increment</i>
BLN	Integer	2	Bulan pembobotan
THN	Integer	4	Tahun pembobotan
PRSNJNSREG	Integer	2	Bobot untuk kategori jenis region
PRSNJUMPENREG	Integer	2	Bobot untuk kategori jumlah penduduk region
PRSNJUMDR	Integer	2	Bobot untuk kategori jumlah dokter di region
BIAYA	Integer	5	Alokasi biaya per visitasi
BBT1	Integer	1	Bobot untuk jumlah kunjungan per hari
BBT2	Integer	1	Bobot untuk nilai penjualan bersih per kunjungan penjualan
BBT3	Integer	1	Bobot untuk besarnya biaya kunjungan
BBT4	Integer	1	Bobot untuk total nilai penjualan

## 13. User

Adalah tabel yang berisi informasi tentang *user* yang menggunakan sistem beserta *privelege*-nya. Dijelaskan pada tabel di halaman berikut

Tabel 4.13 Deskripsi Tabel User

Nama	Jenis	Panjang (max)	Keterangan
IDUSER	Integer	5	(PK), <i>auto increment</i>
USERNAME	Integer	5	<i>Username, unik</i>
PASSWD	Integer	5	<i>Password</i>
NAMA	Integer	5	<i>Nama user</i>
PRIV	Integer	5	<i>Privilege user</i>
REGI	Integer	2	<i>Region user</i>

## 4.2 Implementasi Sistem

Berdasarkan desain aplikasi yang telah didefinisikan sebelumnya, langkah selanjutnya adalah implementasi sistem. Sub bab ini menjelaskan tahapan pembuatan fungsi-fungsi dalam sistem informasi pengukuran kinerja medrep ini. Platform aplikasi yang digunakan adalah JSP dengan editor Macromedia Dreamweaver MX untuk pengkodean serta mySQL-front untuk manajemen database. Desain antar muka halaman utama yang digunakan dalam aplikasi ini dapat dilihat pada Gambar dibawah



Gambar 4.43 Desain Antar Muka Sistem Informasi Kinerja Medrep

Secara umum, modul didalam sistem ini dibedakan berdasar pengguna utama dari sistem ini yaitu *sales*, *apotek*, *medrep*, *ssupervisor*, dan *manajer*. Sistem akan menyediakan layanan yang berbeda untuk setiap tingkatan user.

### 4.2.1 Modul Sales

Setiap user yang login dalam sistem sebagai *sales* akan menuju halaman `salesIndex.jsp`. Layanan yang disediakan sistem dalam modul ini adalah manajemen transaksi, manajemen apotek, dan laporan penjualan.

#### Insert Data Transaksi

Tanggal

AptkTrans:

DrTrans:

[Lihat dan Update Data Transaksi](#)

Gambar 4.44 Halaman manajemen transaksi

Pada halaman manajemen Transaksi ini, pihak *sales* dapat memasukkan input data transaksi. Untuk memudahkan pihak *sales*, input data transaksi dibuat untuk pencatatan data tiap resep. Tampilan bersifat dinamis sehingga *sales* tidak perlu bolak-balik mengklik halaman hanya sekedar untuk menambah obat pada sebuah resep. Selain itu pihak *sales* juga dapat melihat data-data transaksi yang telah dimasukkan pada hari-hari sebelumnya. Gambar 4.42 dan 4.43 menunjukkan proses input dan *view* data transaksi.

Tanggal 1 - 1 - 2004  
 Nama Apotek berkah  
 Nama Dokter Adin, Orthopedi

### Daftar Transaksi

Obat  Sebanyak

Nama Obat Angioten  
 Kemasan tablet 10 mg  
 Harga Rp 5500  
 Jumlah 3

Obat	Kemasan	Harga	Jumlah	Nilai Penjualan	Hapus
Angioten	tablet 10 mg	5500	3	16500	<input type="button" value="Hapus"/>

Total Penjualan Rp 16500.0

Gambar 4.45 Input data transaksi

### Lihat Data Transaksi

Tanggal

### Data Transaksi

Tanggal	Apotek	Dokter	Nilai Penjualan	Detail	Update
9 - 12 - 2005	berkah	Mulyono, Umum	Rp 30000	<a href="#">Detail</a>	<a href="#">Update</a>
9 - 12 - 2005	berkah	Joni, Umum	Rp 23000	<a href="#">Detail</a>	<a href="#">Update</a>
9 - 12 - 2005	Keajaiban	waluyo, Pediatri	Rp 44000	<a href="#">Detail</a>	<a href="#">Update</a>

Gambar 4.46 view data transaksi

Selain manajemen transaksi, untuk *sales* juga tersedia fitur untuk manajemen apotek. Dengan fitur ini, *sales* dapat melakukan manajemen data apotek. *Sales* dapat menambah, mengubah atau menghapus data apotek yang dikelolanya. Setiap data apotek yang dimasukkan seorang *sales*, maka secara otomatis *sales* bertanggung jawab terhadap transaksi di apotek tersebut.

## Daftar Apotek di Region 1

Nama Apotek	Alamat	Sales
anugerah	kuningan	Bob
berkah	Jemursari	Tuan Sales
Keajaiban	Gubeng	Tuan Sales
ragu	keputih	Tuan Sales
Makmur	Krembangan Selatan	Tuan Sales
Sehat	Perak Barat	Tuan Sales

Tambah Apotek

Edit Apotek

Gambar 4.47 Daftar seluruh apotek pada sebuah region

## Update Data Apotek

Nama Apotek	Alamat	Sales	Update	Hapus
berkah	Jemursari	Tuan Sales	Update	Hapus
Keajaiban	Gubeng	Tuan Sales	Update	Hapus
Makmur	Krembangan Selatan	Tuan Sales	Update	Hapus
ragu	keputih	Tuan Sales	Update	Hapus
Sehat	Perak Barat	Tuan Sales	Update	Hapus

Gambar 4.48 update data apotek

Fitur lain yang terdapat pada modul *sales* adalah laporan penjualan yang berisi nilai penjualan untuk tiap obat dan tiap dokter pada region *sales*. Bagian laporan penjualan ini akan ada pada setiap modul dan akan dijelaskan lebih rinci bersama-sama di bagian akhir.

### 4.2.2 Modul Apotek

Modul apotek terdiri dari dua bagian yang keduanya hanya bersifat sebagai pemonitor kegiatan penjualan yang terjadi yaitu penjualan apotek dan penjualan obat. Apotek yang sukses login dalam sistem akan menuju pada halaman `aptk1Index.jsp`.

Bagian penjualan apotek berisi fitur untuk menampilkan data-data transaksi di sebuah apotek dalam periode dua minggu. Dengan fitur ini apotek dapat melakukan *cross check* transaksi yang terjadi di apoteknya meskipun hanya untuk produk dari Kalbe. Berikut adalah gambar halaman penjualan apotek beserta *pseudocode*-nya



**Detail Nilai Penjualan tiap obat  
daerah Pabean Cantikan, Semampir, Krembangan**

Tahun  Bulan  Dwi Minggu

Nama Obat	Kemasan	Harga Satuan	Nilai Transaksi
Bactecyn	viral 10 mg	Rp 7000	Rp 54000.0
Broadced	sirup 220 ml	Rp 6000	Rp 300000.0
Cefspan	tablet 10 mg	Rp 4500	Rp 16000.0
Cravit	tablet 25 mg	Rp 4750	Rp 22500.0

Gambar 4.51 halaman penjualan obat

### 4.2.3 Modul Supervisor

Modul supervisor ini terdiri dari empat bagian, yaitu manajemen region, manajemen *medrep*, manajemen dokter, dan laporan penjualan. Dari Keempat bagian ini, manajemen region, manajemen *medrep*, dan manajemen dokter lebih bersifat sebagai inisiator atau inisialisasi data awal. Fitur tersebut akan sering diakses pada proses awal penggunaan sistem informasi penilaian kinerja *medrep*.

Bagian manajemen region adalah bagian yang digunakan oleh pihak supervisor untuk mengubah data region yang dipimpinnya. Data region digunakan untuk menghitung target region. Data-data region yang digunakan untuk menghitung target region adalah jenis region, jumlah penduduk region, dan jumlah dokter di region tersebut. Data jenis region dan jumlah penduduk region diinisialisasi oleh supervisor berdasar hasil survey dan laporan BPS. Sedangkan jumlah dokter berasal dari database berdasar inputan yang dimasukkan supervisor. Fitur lain yang penting dari bagian ini adalah *view* target penjualan yang didapat dari hasil perhitungan kondisi region dengan bobot yang ditentukan oleh pihak manager. Bobot ini akan diupdate setiap bulan oleh manager. Berikut adalah gambar halaman bobot, *view* target penjualan. Sedangkan *pseudocode* target penjualan dapat dilihat pada gambar 4.51

### Data Region 1

Daerah Pabean Cantikan, Semampir, Krembangan

Jenis Region Derah Perifer

Jumlah Penduduk 50000 - 100000

Jumlah Dokter 4

Update Data Region

Lihat Target Penjualan

Gambar 4.52 manajemen region

### Target Penjualan Region 1

Tahun

Bulan	Jenis Region	Jumlah Penduduk	Jumlah Dokter	Biaya	Target Penjualan
1	30 %	30 %	40 %	Rp 6000	Rp 6000000
2	30 %	30 %	40 %	Rp 6000	Rp 6000000
3	40 %	20 %	40 %	Rp 4000	Rp 6000000
4	30 %	30 %	40 %	Rp 6000	Rp 6000000
5	30 %	30 %	40 %	Rp 6000	Rp 6000000
6	30 %	30 %	40 %	Rp 6000	Rp 6000000
7	30 %	30 %	40 %	Rp 6000	Rp 6000000
8	30 %	30 %	40 %	Rp 6000	Rp 6000000
9	30 %	30 %	40 %	Rp 6000	Rp 6000000
10	30 %	30 %	40 %	Rp 6000	Rp 6000000
11	30 %	30 %	40 %	Rp 6000	Rp 6000000
12	30 %	30 %	40 %	Rp 6000	Rp 6000000

Gambar 4.53 view target penjualan

Fitur berikutnya yang dapat digunakan pihak supervisor adalah manajemen *medrep*. Yang digunakan untuk menambah, mengubah, dan menghapus data *medrep* yang berada di bawah supervisor. Selain itu juga terdapat fitur analisa kinerja *medrep* yang berisi data evaluasi kinerja *medrep* secara dwi mingguan. Seorang supervisor dapat melihat data evaluasi seluruh *medrep* yang ada di region wilayah kerjanya dan memberikan rekomendasi atas kinerja yang dicapai *medrep*-nya selama dua mingguan. Fitur ini akan dijelaskan lebih jelas pada bagian akhir. Gambar halaman manajemen *medrep* dapat dilihat pada gambar 4.52.

```

<table border="1" cellpadding="0" cellspacing="0">
  <% while ((Recordset1_hasData)&&(Repeat1_numRows-- != 0)) { %>
  <tr><%String aJnsReg = Recordset1.getObject("jnsReg").toString();
String aJumPen = Recordset1.getObject("jumPenReg").toString();
String aJumDr = Recordset1.getObject("jumDr").toString();
String bJnsReg = Recordset3.getObject("jnsReg").toString();
String bJumPen = Recordset3.getObject("jumPenReg").toString();
String bJumDr = Recordset2.getObject("count(*)").toString();
  double cJnsReg=Integer.parseInt(aJnsReg);
  double cJumPen=Integer.parseInt(aJumPen);
  double cJumDr=Integer.parseInt(aJumDr);
  double dJnsReg=Integer.parseInt(bJnsReg);
  double dJumPen=Integer.parseInt(bJumPen);
  double dJumDr=Integer.parseInt(bJumDr);
  double eJumDr=0; if (dJumDr<=45){eJumDr=5;}
  else if (45<dJumDr & dJumDr<=90){eJumDr=10;}
  else if (91<dJumDr & dJumDr<=135){eJumDr=15;}
  else {eJumDr=20;} double
  trgt=((cJnsReg*dJnsReg)+(cJumPen*dJumPen)+(cJumDr*eJumDr))/100;
  int target=0; if (trgt<=4){target=4000000;}
  else if (4<trgt & trgt<=8){target=6000000;}
  else if (8<trgt & trgt<=12){target=8000000;}
  else{target=10000000;} %>
  <td><div align="center">
<%=Recordset1.getObject("bln")%></div></td><td>
<div align="center">
<%=Recordset1.getObject("jnsReg")%>&nbsp;%</div></td><td><div
align="center"><%=Recordset1.getObject("jumPenReg")%>&nbsp;%</div>
</td><td><div align="center">
  <%= Recordset1.getObject("jumDr")=%>&nbsp;%</div></td>
  <td>Rp&nbsp;%<%= Recordset1.getObject("biaya")%></td>
</tr>
<td>Rp&nbsp;%<%=target*14%></td>
  <% Repeat1_index++;Recordset1_hasData = Recordset1.next();}%>
</table>

```

Gambar 4.54 pseudocode target penjualan tahun

### Daftar Medical Representative di region 1

Nama Medrep	Alamat	Telepon	Dokter Didikan
Andi	Surabaya 56	5684752	Lihat Dokter
Ardi Bagus	semampir 37	6565268	Lihat Dokter
Tuan Medrep	kreembangan 31	5684525	Lihat Dokter

[Tambah Medrep](#)

[Edit Medrep](#)

[Analisa Kinerja Medrep](#)

Gambar 4.55 halaman manajemen medrep

**Evaluasi**

Jumlah Kunjungan per Hari	E	Bobot	4
Pendapatan per Kunjungan	E	Bobot	1
Biaya Kunjungan Penjualan	A	Bobot	2
Realisasi Nilai Penjualan	E	Bobot	2

Indeks Kinerja **0.89**

Rekomendasi Supervisor

ada rekomendasi

OK

Gambar 4.56 form untuk pemberian rekomendasi

Fitur berikutnya adalah manajemen dokter. Pihak supervisor bertanggung jawab penuh terhadap seluruh data dokter di wilayah regionnya. Salah satu faktor penentu keberhasilan sistem ini adalah manajemen dokter. Jumlah dokter, baik yang berada di region, maupun yang “dididik” masing-masing *medrep* akan berpengaruh pada perhitungan target penjualan maupun proses evaluasi kinerja *medrep*. Dalam fitur ini supervisor dapat menambah, mengubah, dan menghapus data dokter di sebuah region. Setiap penambahan data dokter, harus dibarengi dengan peng-*assign*-an seorang *medrep* yang akan “mendidik” dokter tersebut. Berikut adalah gambar halaman manajemen dokter.

### Daftar Dokter di region 1

Nama Dokter	Spesialisasi	Alamat	Telepon Medrep	Pendidik
Joko	Orthopedi	Malang	5468542	Andi
Anton	Umum	Manyar	-	Ardi Bagus
Mulyono	Umum	Krempangan Timur	8595621	Tuan Medrep
Setiawan	Internist	Krempangan Barat	5465458	Tuan Medrep

Tambah Dokter

Edit Dokter

Gambar 4.57 halaman manajemen dokter

Pada bagian manajemen visitasi, *medrep* wajib melaporkan hasil kegiatan visitasi yang dilakukan. Secara umum ada 3 kegiatan yang dilakukan *medrep* ketika melakukan visitasi yaitu memperkenalkan obat baru, mengingatkan obat lama, dan kegiatan kondisional. *Medrep* harus menyerahkan surat jalan yang telah ditandatangani dokter untuk diberikan kepada supervisor, kemudian supervisorlah yang memasukkan data visitasi tersebut dalam sistem. Disediakan pula *field* yang dapat digunakan untuk menambah keterangan pada visitasi yang dilakukan. Berikut adalah gambar halaman manajemen visitasi.

### Insert Data Visitasi

Medrep

Tanggal  Bulan  Tahun

Dokter

Kegiatan

.....

Gambar 4.58 insert data visitasi

### Lihat Data Visitasi

Tanggal  Bulan  Tahun

### Data Visitasi

Tanggal	Medrep	Dokter	Kegiatan	Update	Hapus
10 - 12 - 2005	Agung	Iman	Memperkenalkan Obat Baru. Bactecyn	Update	Hapus
10 - 12 - 2005	Agung	Ardhika	Memperkenalkan Obat Baru. Bactecyn	Update	Hapus
10 - 12 - 2005	Agung	Rifki	Memperkenalkan Obat Baru. Bactecyn	Update	Hapus
10 - 12 - 2005	Totok	Rudi	Memperkenalkan Obat Baru. Bactecyn	Update	Hapus
10 - 12 - 2005	Totok	Joni	Mengingatkan Obat Lama. Bactecyn	Update	Hapus

Gambar 4.59 view data visitasi

Sebuah fitur lagi yaitu manajemen penjualan yang digunakan untuk monitoring data penjualan di tiap region, akan dijelaskan pada bagian akhir.

#### 4.2.4 Modul *Medrep*

Modul *medrep* terdiri dari 3 bagian yang terbagi menjadi 2 bagian monitoring yaitu bagian analisa kinerja dan laporan penjualan.

Bagian pertama dari modul *medrep* ini adalah kinerja, yang berisi data mengenai evaluasi kinerja *medrep* pada region yang bersangkutan, dan laporan penjualan yang berisi monitoring penjualan pada region. Keduanya akan dijelaskan lebih rinci pada bagian akhir.

#### 4.2.5 Modul Manager

Modul manager terdiri atas empat bagian yaitu manajemen bobot, Region *Overview*, analisa *medrep* dan laporan penjualan. Hampir semua bagian hanya bersifat monitoring kecuali manajemen bobot yang harus diakses minimal sebulan sekali untuk menginisialisasi bobot untuk perhitungan target penjualan bulan berikutnya..

Bagian manajemen bobot digunakan untuk memasukkan atau mengubah bobot untuk perhitungan kinerja. Bobot yang diubah adalah untuk jenis region, jumlah penduduk region, dan jumlah dokter di region yang bobotnya berupa persentase dan jumlah dari ketiga bobot tersebut harus 100%. Selain itu, setiap bulan manager juga harus menginisialisasi biaya visitasi yaitu biaya yang dikeluarkan perusahaan untuk setiap kunjungan penjualan. Biaya visitasi tersebut digunakan untuk proses evaluasi kinerja *medrep*. Gambar halaman manajemen bobot dapat dilihat pada gambar 4.57

Bagian kedua dari modul manager adalah bagian region *Overview*. Pada bagian ini, pihak manager dapat memonitoring secara garis besar kondisi tiap-tiap region di surabaya. Dalam region *Overview* dapat dilihat nama region, jenis region, jumlah penduduk region, supervisor dari setiap region, maupun jumlah *medrep*, jumlah dokter dan jumlah apotek yang masing-masing dapat dilihat detailnya pada 13 region di Surabaya. Dengan adanya region *Overview* ini, bagi pihak manager selain dapat mengontrol dan memonitoring kondisi masing-masing region, juga dapat mengawasi

parameter perhitungan seperti jenis region dan jumlah penduduk tiap region. jika ditemukan kesalahan, pihak manager dapat berkoordinasi dengan supervisor yang mengepalainya. Gambar halaman region *Overview* dapat dilihat pada gambar 4.58.

### Daftar Bobot Perhitungan Target Penjualan

Tahun

Bulan	Jenis Region	Jumlah Penduduk	Jumlah Dokter	Biaya Visitasi	Bobot 1	Bobot 2	Bobot 3	Bobot 4
1	30 %	30 %	40 %	Rp 6000	1	2	2	3
2	30 %	30 %	40 %	Rp 6000	2	3	3	2
3	40 %	20 %	40 %	Rp 4000	3	4	1	2
4	30 %	30 %	40 %	Rp 6000	4	1	2	3
5	30 %	30 %	40 %	Rp 6000	1	2	3	4
6	30 %	30 %	40 %	Rp 6000	2	3	2	2
7	30 %	30 %	40 %	Rp 6000	3	4	3	3
8	30 %	30 %	40 %	Rp 6000	4	1	2	3
9	30 %	30 %	40 %	Rp 6000	1	2	3	2
10	30 %	30 %	40 %	Rp 6000	2	3	2	4
11	30 %	30 %	40 %	Rp 6000	3	4	1	3
12	30 %	30 %	40 %	Rp 6000	4	1	2	2

Gambar 4.60 halaman manajemen bobot

### Region Overview

Reg	Jenis Region	Jumlah Penduduk	Wilayah	Supervisor	Medrep	Dokter	Apotek
1	Daerah Perifer	50000 - 100000	Pabean Cantikan, Semampir, Krembangan	Tuan Supervisor	3 Detail	4 Detail	6 Detail
2	Daerah Perifer	50000 - 100000	Kenjeran, Simokerto	tuan sprvsr2	2 Detail	1 Detail	1 Detail
3	Daerah Perifer	0 - 50000	Asemrowo, Tandes	tuan sprvsr3	1 Detail	1 Detail	1 Detail
4	Daerah Perifer	0 - 50000	Pakal, Benowo	tuan sprvsr4	1 Detail	1 Detail	1 Detail
5	Daerah Pendukung	50000 - 100000	Sambikerep, Lakarsantri	tuan sprvsr5	1 Detail	1 Detail	1 Detail
6	Daerah Pendukung	50000 - 100000	Wiyung, Karangpilang	tuan sprvsr6	1 Detail	1 Detail	1 Detail
7	Tengah Kota	100000 - 150000	Gayungan, Jambangan, Wonocolo	tuan sprvsr7	1 Detail	1 Detail	1 Detail
8	Daerah Pendukung	50000 - 100000	Tenggilis Mejoyo, Rungkut, Gunung Anyar	tuan sprvsr8	4 Detail	2 Detail	1 Detail
9	Tengah Kota	50000 - 100000	Sukolilo, Mulyorejo	tuan sprvsr9	1 Detail	1 Detail	1 Detail
10	Tengah Kota	50000 - 100000	Dukuh Pakis, Sukomanunggal	tuan sprvsr10	1 Detail	1 Detail	1 Detail
11	Tengah Kota	50000 - 100000	Gubeng, Tegalsari, Genteng	tuan sprvsr11	1 Detail	1 Detail	1 Detail
12	Daerah Pendukung	100000 - 150000	Bubutan, Sawahan, Wonokromo	tuan sprvsr12	1 Detail	1 Detail	1 Detail
13	Daerah Pendukung	50000 - 100000	Bulak, Tambaksari	tuan sprvsr13	1 Detail	1 Detail	1 Detail

Gambar 4.61 halaman region *Overview*

Bagian berikutnya adalah analisa kinerja *medrep*. Proses analisa kinerja *medrep* sendiri di-generate setiap dua minggu sekali oleh sistem. Fitur analisa kinerja *medrep* ini juga dimiliki oleh modul *medrep* dan supervisor. Bedanya, jika Manager dapat

mengakses hasil evaluasi seluruh *medrep*, fitur analisa kinerja *medrep* untuk *medrep* dan supervisor hanya dapat menampilkan hasil evaluasi untuk *medrep* pada region yang bersangkutan. Gambar halaman evaluasi kinerja *medrep*.

### Kinerja Medrep Region 8

**Target Penjualan Region** Rp 84,000,000

Medrep	Dokter	Target	Detail
Agung	3	Rp 22,909,090	<a href="#">Detail</a>
Amir	2	Rp 15,272,727	<a href="#">Detail</a>
Budi	2	Rp 15,272,727	<a href="#">Detail</a>
Totok	4	Rp 30,545,454	<a href="#">Detail</a>

**Nilai Penjualan Region** Rp 5,854,250

Medrep	Nilai Transaksi	% Penjualan
Agung	Rp 1,197,750	20.4 %
Amir	Rp 1,878,000	32.0 %
Budi	Rp 969,750	16.5 %
Totok	Rp 1,808,750	30.8 %

Gambar 4.62 evaluasi kinerja *medrep*

Region	8	
Periode	Dwi Minggu 2, 12 2005	
Nama	Totok	
Dokter Didikan	4	<a href="#">Lihat Dokter</a>
(a) Target Penjualan		Rp 30,545,454
(b) Nilai Penjualan		Rp 27,775,000
(c) Persentase Penjualan	90.93%	(b):(a)*100%
(d) Jumlah Kunjungan	3	<a href="#">Detail Visitasi</a>
(e) Rata-rata Kunjungan per Hari	0.21	(d):14
(f) Jumlah Kunjungan Penjualan 2		
(g) Rata-rata Kunjungan Penjualan per Hari	0.14	(f):14
(h) Biaya per Kunjungan		Rp 6,000
(i) Total Biaya Kunjungan Penjualan		Rp 12,000 (f)*(h)
(j) Laba Bersih Penjualan		Rp 27,769,000 (b)-(i)
(k) Pendapatan bersih per Kunjungan Penjualan		Rp 13,884,500 (j):(f)
(l) Pendapatan bersih per Pelanggan		Rp 6,942,250 (j):Jumlah Dokter

Gambar 4.63 halaman detail evaluasi kinerja *medrep*

Untuk mendapatkan penilaian yang lebih obyektif, akan diberikan rekomendasi oleh supervisor pimpinan *medrep*. Pemberian rekomendasi ini didasarkan pada indeks kinerja yang diraih *medrep* selama periode bersangkutan. Rekomendasi supervisor akan menjadi bahan pertimbangan manajer selain data statistik penjualan. Berikut adalah contoh rekomendasi supervisor.

<b>Evaluasi</b>			
Jumlah Kunjungan per Hari	<b>E</b>	Bobot	4
Pendapatan per Kunjungan	<b>A</b>	Bobot	1
Biaya Kunjungan Penjualan	<b>A</b>	Bobot	2
Realisasi Nilai Penjualan	<b>D</b>	Bobot	2
Indeks Kinerja	<b>1.56</b>		
Rekomendasi Supervisor	<div style="border: 1px solid #ccc; padding: 5px; min-height: 60px;">           Medrep ini kurang sekali melakaukan kunjungan         </div>		

Gambar 4.64 contoh rekomendasi supervisor

Sedangkan *Pseudocode* dari halaman analisa *medrep* dapat dilihat pada gambar 4.62 di halaman berikut.

Selain berupa analisa yang bersifat kuantitatif, evaluasi kinerja *medrep* juga memberikan fitur untuk mengevaluasi kinerja *medrep* secara kualitatif melalui detail visitasi. Dalam detail visitasi, manager dapat mengetahui kegiatan apa saja yang dilakukan *medrep* dalam visitasi, lengkap beserta detail tanggal, dokter dan keterangan hasil visitasi. Halaman detail visitasi akan tampak seperti gambar 4.63

```

<table width="570" border="0">
  <tr>
    <td colspan="4"><span class="style1">Report Analisa Kinerja
Medrep </span></td></tr>
  <tr>
    <td colspan="4">&nbsp;</td></tr>
  <tr>
    <td>&nbsp;</td>
    <td>Region</td>
    <td><%=request.getParameter("reon")%></td>
    <td>&nbsp;</td></tr>
  <tr>
    <td>&nbsp;</td>
    <td>Periode</td>
    <td>Dwi Minggu <%=request.getParameter("mgu")%>,
<%=request.getParameter("bul")%> <%=request.getParameter("tah")%>
</td><td>&nbsp;</td></tr>
  <tr>
    <td>&nbsp;</td>
    <td>Nama</td>
    <td><%=request.getParameter("nam")%></td>
    <td>&nbsp;</td></tr>
  <tr>
    <td>&nbsp;</td>
    <td>Dokter Didikan </td>
    <td><%=request.getParameter("dktr")%></td>
    <td align="right"><a
href="lhtDr.jsp?ind=<%=request.getParameter("nam")%>" class="link"
target="_blank">Lihat Dokter</a></td></tr>
  <tr>
    <td colspan="4">&nbsp;</td></tr>
  <tr>
    <td>(a)</td>
    <td>Target Penjualan </td>
    <td align="right">Rp&nbsp;<%=String
bhe=(String)request.getParameter("tgtPen"); int
bhe2=Integer.valueOf(bhe).intValue();%><%=bhe2%>.0</td>
    <td>&nbsp;</td></tr>
  <tr>
    <td>(b)</td><td>Total Penjualan </td>
    <td align="right">Rp&nbsp;<%=String che=((Recordset1_data =
Recordset1.getObject("sum(transaksi.nilaiTransaksi)")==null ||
Recordset1.isNull())?"":Recordset1_data).toString(); double
che2=Double.valueOf(che).doubleValue();%>
    <%=che2%></td><td>&nbsp;</td></tr>
  <tr>
    <td>(c)</td><td>Persentase Penjualan </td>
    <td><%=String aPer=String.valueOf(che2/bhe2);String
bPer=aPer.substring(0,5);%><%=bPer%>&nbsp;<%=>&nbsp;</td>
    <td align="right">(b) : (a)*100%</td></tr>

```

Gambar 4.65 pseudocode analisa kinerja medrep-1

```

<tr>
  <td>(d)</td>
  <td>Jumlah Kunjungan </td>
  <td><%String kunj=((Recordset2_data =
Recordset2.getObject("count(*)")==null ||
Recordset2.isNull())?"":Recordset2_data).toString(); int
kun=Integer.parseInt(kunj);%><%=kun%></td>
  <td align="right"
  <input type="submit" value="Detail Visitasi"
class="box1"></form></td>
</tr>
<tr>
  <td>(e)</td>
  <td>Rata-rata Kunjungan per Hari </td>
  <td><%String aKun=String.valueOf(kun/14.0);String
bKun=aKun.substring(0,4);%><%=bKun%></td>
  <td align="right">(d):14</td>
</tr>
<tr>
  <td>(f)</td>
  <td>Jumlah Kunjungan Penjualan </td>
  <td><%String penj=((Recordset3_data =
Recordset3.getObject("count(*)")==null ||
Recordset3.isNull())?"":Recordset3_data).toString(); int
pen=Integer.parseInt(penj);%><%=pen%></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>(g)</td>
  <td>Rata-rata Kunjungan Penjualan per Hari </td>
  <td><%String aPen=String.valueOf(pen/14.0);String
bPen=aPen.substring(0,4);%><%=bPen%></td>
  <td align="right">(f):14</td>
</tr>
<tr>
  <td>(h)</td>
  <td>Biaya per Kunjungan </td>
  <td align="right">Rp <%String cost=((Recordset4_data =
Recordset4.getObject("biaya")==null ||
Recordset4.isNull())?"":Recordset4_data).toString(); int
bia=Integer.parseInt(cost);%><%=bia%></td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>(i)</td>
  <td>Total Biaya Kunjungan Penjualan </td>
  <td align="right">Rp <%=bia*pen%></td>
  <td align="right">(f) * (h)</td>
</tr>

```

Gambar 4.65 pseudocode analisa kinerja medrep-2



### Detail Visitasi Medrep

Periode Dwi Minggu 1, Bulan 12 Tahun 2005  
 Nama Tuan Medrep  
 Detail Visitasi 2 kali Memperkenalkan obat baru  
 2 kali Mengingatnkan obat lama  
 2 kali kegiatan kondisional

#### Memperkenalkan Obat Baru

Tanggal	Dokter	Keterangan
5 - 12 - 2005	Mulyono	Hexoflam
10 - 12 - 2005	Setiawan	Buinflu

#### Mengingatnkan Obat Lama

Tanggal	Dokter	Keterangan
9 - 12 - 2005	Setiawan	panadol
10 - 12 - 2005	Mulyono	Panadol

#### Kegiatan Kondisional

Tanggal	Dokter	Keterangan
9 - 12 - 2005	Setiawan	Simposium
10 - 12 - 2005	Setiawan	Undangan

Gambar 4.66 halaman detail visitasi

Bagian terakhir dari modul manager adalah laporan penjualan. Dari fitur ini bisa didapat hasil penjualan tiap region setiap 2 minggu sekali. Selain memberikan nilai penjualan secara umum sebuah region, fitur ini juga memberikan detail penjualan untuk tiap dokter, tiap apotek dan tiap obat di setiap region. berikut adalah gambar dari halaman laporan penjualan.

### Laporan Penjualan

Tahun  Bulan  Dwi Minggu

Reg	Nilai Penjualan
I	Rp 7065000.0
II	Rp
III	Rp
IV	Rp
V	Rp
VI	Rp
VII	Rp
VIII	Rp
IX	Rp
X	Rp
XI	Rp
XII	Rp
XIII	Rp

Gambar 4.67 laporan penjualan tiap region

### Detail Nilai Penjualan tiap Obat

Tahun  Bulan  Dwi Minggu   
 Region

Nama Obat	Kemasan	Harga Satuan	Nilai Transaksi
combantrin	sirup 220 ml	15000	Rp 300000.0
zyrex	tablet 25 mg	2500	Rp 22500.0
panadol	virial	4500	Rp 54000.0
inza	tablet 10 mg	2000	Rp 16000.0

Gambar 4.68 detail penjualan tiap obat

### Detail Nilai Penjualan tiap Dokter

Tahun  Bulan  Dwi Minggu   
 Region

Nama Dokter	Spesialis Dokter	Medrep Pendidik	Nilai Transaksi
Mulyono	Umum	Tuan Medrep	Rp 36000.0
Setiawan	Internist	Tuan Medrep	Rp 189000.0
Joko	Orthopedi	Andi	Rp 7500.0
Anton	Umum	Ardi Bagus	Rp 45000.0

Gambar 4.69 detail penjualan tiap dokter

### Detail Nilai Penjualan tiap Apotek

Tahun  Bulan  Dwi Minggu   
 Region

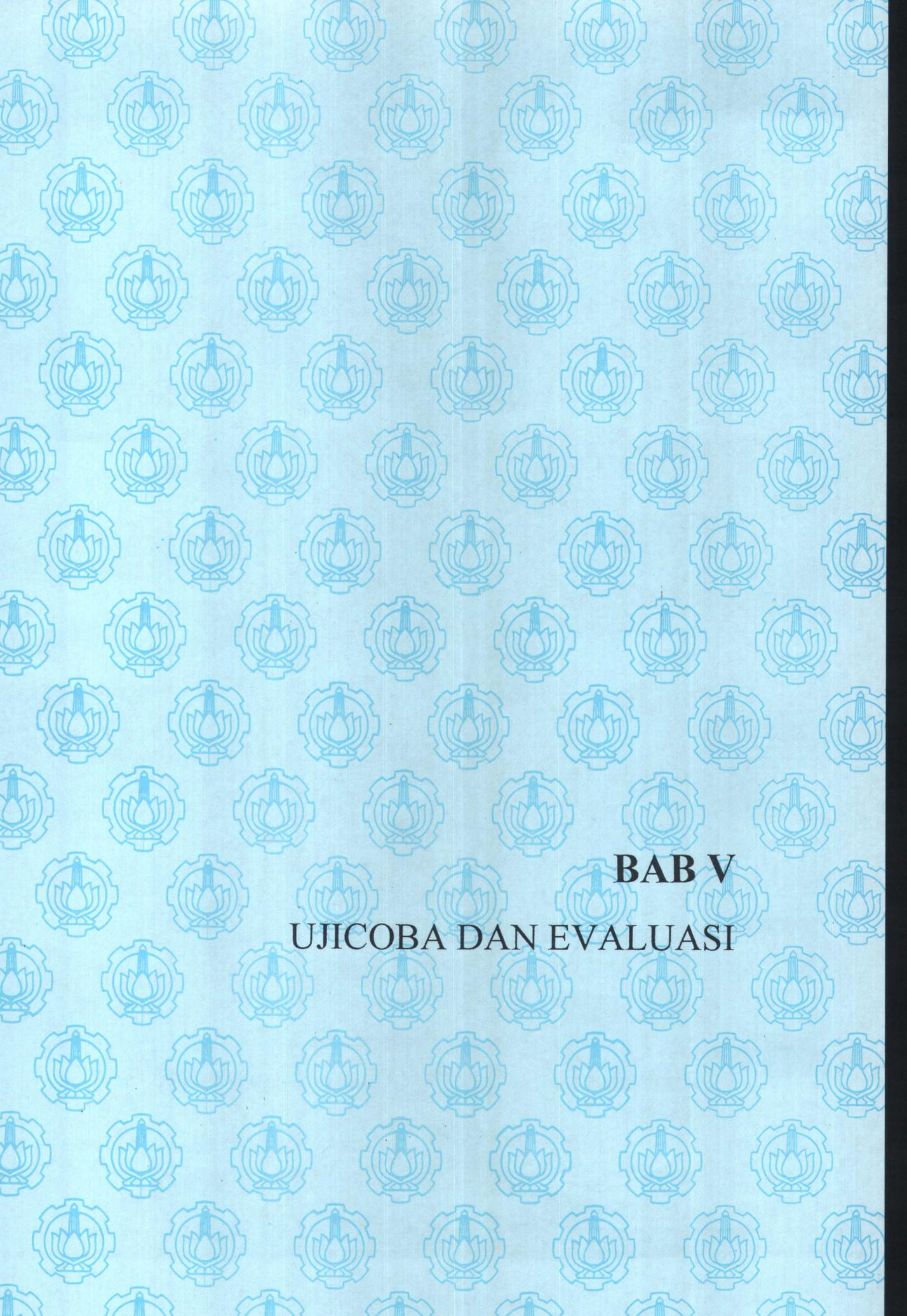
Nama Apotek	Alamat Apotek	Nilai Transaksi
berkah	Jemursari	Rp 128000.0
Keajaiban	Gubeng	Rp 251000.0
ragu	keputih	Rp 7500.0
Makmur	Kremlangan Selatan	Rp 6000.0

Gambar 4.70 detail penjualan tiap apotek

## 4.2.6 Modul Administrator

Modul Admin adalah modul pendukung dari modul-modul lainnya. Modul Admin terdiri dari dua bagian yaitu manajemen user dan manajemen obat.

Bagian manajemen user digunakan untuk menambah, mengubah atau menghapus data user yang memiliki hak akses menggunakan sistem. Admin bertanggung jawab penuh terhadap data user. Berikut adalah gambar manajemen user.



**BAB V**  
**UJICOBAN DAN EVALUASI**

## BAB V

### UJI COBA DAN EVALUASI

Dalam bab ini dijelaskan mengenai uji coba dan evaluasi terhadap sistem yang telah dibuat. Hasil dari uji coba ini nantinya dapat digunakan sebagai masukan dalam pengembangan aplikasi di masa mendatang. Secara umum, pengujian dilakukan dengan variasi skenario.

#### 5.1 Lingkungan Uji Coba

Dalam melakukan evaluasi ini, uji coba secara teknis perlu dilakukan. Pada pelaksanaan uji coba masih dilakukan di lingkungan komputer lokal dengan asumsi server dan client menjadi satu. Dengan perincian sebagai berikut :

- *Processor* pentium 4 – 2.66 GHz
- *Memory* 512 MB
- *Kapasitas Hard Disk* 80 GB
- *Sistem Operasi* Microsoft Windows XP Professional.
- *Server database* MySQL 3.23.42-nt
- *Server web* Apache Tomcat/5.0.18

#### 5.2 Skenario dan Proses Uji Coba

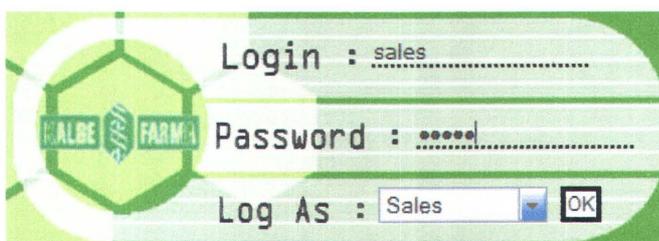
Berdasarkan lingkungan yang telah didefinisikan sebelumnya, proses uji coba dilakukan dengan beberapa skenario pengujian. Skenario ini untuk menunjukkan proses yang ada dalam sistem informasi pengukuran kinerja *medrep*. Skenario uji coba yang dilakukan antara lain:

- A. Uji coba validasi, untuk menguji fungsi pemberian hak akses sistem terhadap *user* yang mengakses aplikasi, untuk menghindari pengaksesan data oleh *user* yang tidak berhak.

- B. Uji coba masukan data *user*, untuk menguji penanganan terhadap data yang dimasukkan dalam *form-form* di sistem informasi pengukuran kinerja *medrep* ini. Uji coba ini merupakan *sample* untuk fungsi-fungsi yang lain, seperti manajemen apotek, manajemen transaksi, manajemen visitasi, manajemen region, manajemen *medrep*, manajemen dokter, dan manajemen bobot.
- C. Uji coba masukan untuk target dalam analisa *medrep*, untuk menguji penanganan integritas data untuk target yang didefinisikan.
- D. Uji coba validitas data, untuk menguji kebenaran data analisa yang dihasilkan, dengan cara dibandingkan dengan aplikasi *spreadsheet*.
- E. Uji coba evaluasi kinerja *medrep*, dalam evaluasi ini akan diuji coba proses penilaian kinerja *medrep* dengan segala aspeknya. Juga akan diuji coba bentuk laporan yang dihasilkan oleh sistem.

### 5.2.1 Uji coba validasi

Proses yang diujicobakan dalam uji coba validasi adalah penanganan terhadap akses *user* kepada aplikasi. Dalam aplikasi ini, *user* hanya bisa mengakses fungsi-fungsi yang secara proses bisnis memang ditangani olehnya. Ujicoba ini dimulai dari memasukkan data *user* yang tidak ada dalam *database*.

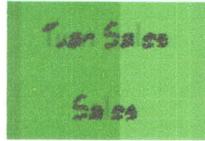


Gambar 5.1 contoh login

Selain memasukkan username dan password, *user* yang akan masuk dalam sistem juga harus memilih *privilege*/jabatan yang sesuai untuknya. Sistem informasi kinerja *medrep* sendiri memiliki enam jenis *privilege* dengan jenis informasi yang berbeda untuk setiap *privilege*. Jika seorang *user* sukses melakukan login, ia akan



masuk pada halaman yang sesuai dengan *privilegenya*. Berikut adalah contoh login yang sukses.



Gambar 5.2 contoh login *sales* yang sukses

Baris diatas adalah nama *user* seperti yang terdaftar dalam *database user*. Dan baris di bawahnya menunjukkan *privilege* yang diberikan sistem kepada *user*. Baik nama *user* maupun *privilege*-nya, keduanya didapat dengan menggunakan fasilitas *session* yang terdapat dalam JSP.

Jika *user* salah memasukkan username, password, maupun *privilege* sistem akan menolak *user* masuk dan akan mempersilahkan *user* untuk login kembali. Berikut adalah contoh *user* yang gagal login.

Maaf Username, Password, atau Privilage anda salah  
Silahkan Login kembali

Gambar 5.3 contoh login yang gagal

### 5.2.2 Uji coba masukan data *user*

Uji coba ini digunakan untuk menguji *field* validasi dasar seperti form yang tidak boleh kosong, konfirmasi atas perubahan atau penghapusan data, dll. Contoh yang digunakan berikut adalah salah satu contoh dari *form-form* lain dalam sistem informasi kinerja *medrep*.

#### Insert Data Medrep

Region	8
Nama Medrep	Toni
Alamat	Surabaya
Telepon	5468542

OK Cancel

Gambar 5.4 *form insert medrep* baru

Pada contoh penambahan *medrep* baru seperti tampak pada gambar di halaman sebelumnya, ketika supervisor menekan tombol "OK", maka akan muncul pesan konfirmasi yang menanyakan apakah supervisor benar-benar ingin menambah *medrep* baru, gambar pesan konfirmasi tersebut akan muncul seperti gambar di bawah.



Gambar 5.5 contoh pesan konfirmasi

### 5.2.3 Uji coba integritas data

Uji coba integritas data dilakukan dengan cara membandingkan perhitungan yang dilakukan sistem untuk penanganan *form* yang berbeda. Uji coba akan dikatakan sukses jika sistem berhasil menampilkan data yang valid dan konsisten pada *form* yang berbeda.

Uji coba yang pertama adalah membandingkan nilai antara bobot perhitungan untuk target penjualan maupun indeks kinerja yang berada pada modul manajer, dengan nilai bobot yang akan muncul pada halaman evaluasi kinerja *medrep*. Dengan uji coba ini dapat dilihat apakah bobot tersebut konsisten untuk berbagai *form*.

## Daftar Bobot Perhitungan Target Penjualan

Tahun

2005

Tampilkan

Bulan	Jenis Region	Jumlah Penduduk	Jumlah Dokter	Biaya Visitasi	Bobot 1	Bobot 2	Bobot 3	Bobot 4
1	30 %	30 %	40 %	Rp 6000	1	2	2	3
2	30 %	30 %	40 %	Rp 6000	2	3	3	2
3	40 %	20 %	40 %	Rp 4000	3	4	1	2
4	30 %	30 %	40 %	Rp 6000	4	1	2	3
5	30 %	30 %	40 %	Rp 6000	1	2	3	4
6	30 %	30 %	40 %	Rp 6000	2	3	2	2
7	30 %	30 %	40 %	Rp 6000	3	4	3	3
8	30 %	30 %	40 %	Rp 6000	4	1	2	3
9	30 %	30 %	40 %	Rp 6000	1	2	3	2
10	30 %	30 %	40 %	Rp 6000	2	3	2	4
11	30 %	30 %	40 %	Rp 6000	3	4	1	3
12	30 %	30 %	40 %	Rp 6000	4	1	2	2

Gambar 5.6 detail bobot pada modul manajer

## Report Analisa Kinerja Medrep

Region 8  
 Periode Dwi Minggu 2, 12 2005  
 Nama Totok  
 Dokter Didikan 4

### Evaluasi

Jumlah Kunjungan per Hari	<b>E</b>	Bobot	4
Pendapatan per Kunjungan	<b>A</b>	Bobot	1
Biaya Kunjungan Penjualan	<b>A</b>	Bobot	2
Realisasi Nilai Penjualan	<b>D</b>	Bobot	2

Gambar 5.7 form evaluasi kinerja medrep

Dari gambar diatas bisa kita lihat dari *form* detail bobot pada modul manajer, bobot untuk setiap bulan pada tahun 2005. Pada bulan Desember 2005 bobot untuk bobot 1 adalah 4, kemudian bobot 2, 3 dan 4 masing-masing adalah 1, 2, 2. Pada halaman evaluasi kinerja *medrep* dapat dilihat bahwa bobot tersebut konsisten. Karena bobot untuk jumlah kunjungan perhari (bobot 1) adalah 4, bobot untuk pendaptan per kunjungan (bobot 2) adalah 1, dan seterusnya.

Uji coba berikutnya adalah membandingkan hasil perhitungan target penjualan region yang ada pada bagian target penjualan dengan yang ada pada *form* analisa

kinerja *medrep*. Dengan data yang konsisten, akan memudahkan *medrep* untuk memonitor target penjualan yang harus dicapainya.

### Target Penjualan Region 8

Tahun

Bulan	Jenis Region	Jumlah Penduduk	Jumlah Dokter	Biaya	Target Penjualan
1	30 %	30 %	40 %	Rp 6000	Rp 84000000
2	30 %	30 %	40 %	Rp 6000	Rp 84000000
3	40 %	20 %	40 %	Rp 4000	Rp 84000000
4	30 %	30 %	40 %	Rp 6000	Rp 84000000
5	30 %	30 %	40 %	Rp 6000	Rp 84000000
6	30 %	30 %	40 %	Rp 6000	Rp 84000000
7	30 %	30 %	40 %	Rp 6000	Rp 84000000
8	30 %	30 %	40 %	Rp 6000	Rp 84000000
9	30 %	30 %	40 %	Rp 6000	Rp 84000000
10	30 %	30 %	40 %	Rp 6000	Rp 84000000
11	30 %	30 %	40 %	Rp 6000	Rp 84000000
12	30 %	30 %	40 %	Rp 6000	Rp 84000000

Gambar 5.8 target penjualan pada *form* manajemen region

### Analisa Kinerja Medical Representative

Tahun  Bulan  Dwi Minggu

#### Kinerja Medrep Region 8

Target Penjualan Region Rp 84000000

Nama Medrep	Dokter Didikan	Target Penjualan	Detail
Agung	3	Rp 25200000	<input type="button" value="Detail"/>
Amir	1	Rp 8400000	<input type="button" value="Detail"/>
Budi	1	Rp 8400000	<input type="button" value="Detail"/>
Totok	4	Rp 33600000	<input type="button" value="Detail"/>

Gambar 5.9 target penjualan pada *form* analisa kinerja *medrep*

#### 5.2.4 Uji coba validitas data

Uji coba kevalidan data adalah uji coba dengan cara membandingkan data, hasil dari pengolahan sistem dengan data asli yang dikerjakan manual dengan *spreadsheet* yang lain.

Dalam sistem penilaian kinerja sebelumnya, laporan penjualan per hari didapat dari DPU yang dihasilkan Enseval. Isi dari DPU tersebut adalah detail penjualan obat di sejumlah apotek yang dikelola masing-masing oleh seorang *sales*. Uji coba yang dilakukan adalah membandingkan DPU seperti yang terdapat pada Lampiran C. Dengan penghitungan oleh sistem pada region 8 dengan periode waktu dwi minggu pertama Desember 2005. Diasumsikan data dari DPU yang bertanggal 14 Desember 2005 dari seorang *sales*, Agus Santoso, adalah total transaksi selama periode dwi minggu pertama 2005 untuk seluruh apotek dari semua *sales* di region 8. data hanya terbatas pada 12 obat *Ethycal* utama. Berikut adalah gambar dari hasil perhitungan sistem informasi kinerja *medrep*.

### Laporan Penjualan

Tahun  Bulan  Dwi Minggu

Reg	Nilai Penjualan
I	Rp 392500.0
II	Rp
III	Rp
IV	Rp
V	Rp
VI	Rp
VII	Rp
VIII	Rp 4387250.0
IX	Rp
X	Rp
XI	Rp
XII	Rp
XIII	Rp

Gambar 5.10 nilai penjualan sebuah region sebuah periode

#### 5.2.5 Uji coba evaluasi kinerja *medrep*

Skenario uji coba kali ini adalah mencoba melakukan proses evaluasi kinerja *medrep* Totok pada periode dua mingguan kedua Desember 2005. Totok adalah seorang *medrep* di region 8. sedangkan region 8 sendiri memiliki karakteristik seperti gambar di halaman berikut ini

## Data Region 8

Daerah	Tenggilis Mejoyo, Rungkut, Gunung Anyar
Jenis Region	Derah Perifer
Jumlah Penduduk	0 - 100000
Jumlah Dokter	11

Gambar 5.11 Data region 8

Sedangkan bobot untuk periode Desember 2005 seperti yang telah didefinisikan sebelumnya oleh pihak manajer adalah sebagai berikut

## Daftar Bobot Perhitungan Target Penjualan

Tahun

Bulan	Jenis Region	Jumlah Penduduk	Jumlah Dokter	Biaya Visitasi	Bobot 1	Bobot 2	Bobot 3	Bobot 4
1	30	30	40	Rp 6000	1	2	2	3
2	30	30	40	Rp 6000	2	3	3	2
3	40	20	40	Rp 4000	3	4	1	2
4	30	30	40	Rp 6000	4	1	2	3
5	30	30	40	Rp 6000	1	2	3	4
6	30	30	40	Rp 6000	2	3	2	2
7	30	30	40	Rp 6000	3	4	3	3
8	30	30	40	Rp 6000	4	1	2	3
9	30	30	40	Rp 6000	1	2	3	2
10	30	30	40	Rp 6000	2	3	2	4
11	30	30	40	Rp 6000	3	4	1	3
12	30	30	40	Rp 6000	4	1	2	2

Gambar 5.12 bobot untuk periode tahun 2005

Dari gambar diatas dapat dilihat bahwa untuk bulan Desember 2005, bobot untuk target penjualan masing-masing adalah 30% untuk jenis region, 30% untuk jumlah penduduk, 40% untuk jumlah dokter di region dan Rp 6000 adalah alokasi biaya yang disediakan untuk setiap melakukan kunjungan. Sedangkan bobot untuk indeks kinerja *medrep* masing-masing adalah 4 untuk bobot 1 (bobot untuk jumlah rata-rata kunjungan per hari), 1 untuk bobot 2 (bobot untuk besarnya nilai penjualan bersih per pelanggan), 2 untuk bobot 3 (bobot untuk besarnya biaya kunjungan) dan 2 untuk bobot 4 (bobot untuk nilai total penjualan). Disini dapat dilihat bahwa pihak manajemen menginginkan *medrep* untuk mempebanyak melakukan kunjungan. Dari berbagai macam bobot di atas bisa dihitung besarnya target penjualan dan indeks

kinerja *medrep* dengan menggunakan formulasi seperti yang disebutkan pada bab 3. gambar berikut menggambarkan hasil perhitungan yang dilakukan sistem.

### Analisa Kinerja Medical Representative

Tahun  Bulan  Dwi Minggu

#### Kinerja Medrep Region 8

Target Penjualan Region Rp **84,000,000**

Medrep	Dokter	Target	Detail
Agung	3	Rp 22,909,090	<input type="button" value="Detail"/>
Amir	2	Rp 15,272,727	<input type="button" value="Detail"/>
Budi	2	Rp 15,272,727	<input type="button" value="Detail"/>
Totok	4	Rp 30,545,454	<input type="button" value="Detail"/>

Gambar 5.13 target penjualan *medrep* Totok dwi minggu kedua Desember 2005

Dari Gambar diatas dapat dilihat bahwa target penjualan untuk region 8 pada periode dwiminggu kedua Desember 2005 adalah Rp 84.000.000. Sedangkan target penjualan untuk *medrep* Totok sendiri adalah Rp 30.545.454. Target penjualan *medrep* Totok didapat dari persentase antara dokter yang dididik *medrep* Totok dengan jumlah seluruh dokter di region 8. sedangkan dokter yang dididik *medrep* Totok dapat dilihat seperti pada gambar dibawah.

#### Dokter Didikan Medrep Totok

Nama Dokter	Spesialisasi	Alamat	Telepon
Joni	Umum	Rungkut Timur	8568422
Rudi	Umum	Rungkut barat	5854587
Adin	Orthopedi	rungkut barata	589645
Vendry	Pediatri	rungkut selatan	8545545

Gambar 5.14 dokter didikan *medrep* Totok

Sedangkan detail evaluasi dari *medrep* Totok dapat dilihat setelah *user* mengklik tombol detail yang akan menampilkan detail kinerja *medrep* Totok selama periode dwiminggu kedua Desember 2005. Pada gambar halaman detail juga akan ditampilkan perhitungan indeks kinerja *medrep* dan rekomendasi supervisor yang digunakan untuk menjaga keobyektifitasan proses penilaian.

Region	8		
Periode	Dwi Minggu 2, 12 2005		
Nama	Totok		
Dokter Didikan	4		Lihat Dokter
(a) Target Penjualan		Rp 30,545,454	
(b) Nilai Penjualan		Rp 27,775,000	
(c) Persentase Penjualan	90.93%		(b):(a)*100%
(d) Jumlah Kunjungan	3		<a href="#">Detail Visitasi</a>
(e) Rata-rata Kunjungan per Hari	0.21		(d):14
(f) Jumlah Kunjungan Penjualan 2			
(g) Rata-rata Kunjungan Penjualan per Hari	0.14		(f):14
(h) Biaya per Kunjungan		Rp 6,000	
(i) Total Biaya Kunjungan Penjualan		Rp 12,000	(f)*(h)
(j) Laba Bersih Penjualan		Rp 27,769,000	(b)-(i)
(k) Pendapatan bersih per Kunjungan Penjualan		Rp 13,884,500	(j):(f)
(l) Pendapatan bersih per Pelanggan		Rp 6,942,250	(j):Jumlah Dokter

Gambar 5.15 Form Penilaian Kinerja Medrep Totok

Untuk melihat detail visitasi yang dilakukan *medrep* dapat dilakukan dengan mengklik tombol *Detail Visitasi*. Seperti tampak pada gambar di atas bahwa selama periode dwiminggu kedua Desember 2005, Totok hanya melakukan 3 kali kunjungan dengan 2 di antaranya adalah kunjungan penjualan. Pada halaman berikut akan ditampilkan halaman detail visitasi *medrep* Totok

### Detail Visitasi Medrep

Periode	Dwi Minggu 2, Bulan 12 Tahun 2005
Nama	Totok
Detail Visitasi	1 kali Memperkenalkan obat baru 1 kali Mengingatkan obat lama 1 kali kegiatan kondisional

#### Memperkenalkan Obat Baru

Tanggal	Dokter	Keterangan
16 - 12 - 2005	Rudi	Bactecyn

#### Mengingatkan Obat Lama

Tanggal	Dokter	Keterangan
18 - 12 - 2005	Rudi	Angioten

#### Kegiatan Kondisional

Tanggal	Dokter	Keterangan
16 - 12 - 2005	Rudi	Seminar

Gambar 5.16 Detail Visitasi *Medrep* Totok

Pada bagian bawah halaman evaluasi kinerja *medrep* akan ditampilkan indeks kinerja *medrep* pada periode tertentu lengkap beserta rekomendasi supervisor. Diharapkan setelah mengevaluasi bagian ini manajer mendapatkan gambaran yang lebih baik atas kinerja *medrep*-nya. Gambar berikut adalah indeks kinerja *medrep* Totok untuk periode dwiminggu kedua Desember 2005.

#### Evaluasi

Jumlah Kunjungan per Hari	<b>E</b>	Bobot	4
Pendapatan per Kunjungan	<b>A</b>	Bobot	1
Biaya Kunjungan Penjualan	<b>A</b>	Bobot	2
Realisasi Nilai Penjualan	<b>E</b>	Bobot	2

Indeks Kinerja **1.33**

Rekomendasi Supervisor

Medrep ini harus lebih meningkatkan kinerjanya

Gambar 5.17 indeks kinerja *Medrep* Totok

Setelah itu, manajer dapat mengklik tombol *Grafik Kinerja* yang ada di bawah *form*. Setelah itu, manajer akan diberikan pilihan untuk periode yang akan ditampilkan. Periode yang disediakan adalah untuk tiga bulanan, enam bulanan (semesteran), dan tahunan. Grafik yang akan muncul adalah grafik dari *medrep* dengan periode tahun yang sama seperti *form* sebelumnya. Gambar 5.19 akan menunjukkan form pilihan periode grafik, dan gambar 5.20 akan menunjukkan salah satu contoh dari grafik yang ditampilkan sistem.

### Grafik Kinerja Medrep 2005

Nama Medrep		Totok	
<b>Periode Tiga Bulanan</b>		<b>Periode Semesteran</b>	
<input type="radio"/> Pertama	<input type="radio"/> Ketiga	<input type="radio"/> Pertama	<input checked="" type="radio"/> Tahunan
<input type="radio"/> Kedua	<input type="radio"/> Keempat	<input type="radio"/> Kedua	
<input type="button" value="OK"/>			

Gambar 5.18 *Form* Pemilihan Periode Grafik



Gambar 5.19 Grafik Tahunan nilai penjualan seorang *Medrep*

## BAB VI

### PENUTUP

Bab ini berisi kesimpulan yang dapat diambil berdasarkan hasil uji coba yang telah dilakukan. Selanjutnya, diberikan beberapa saran yang mungkin dapat digunakan untuk mengembangkan hasil yang telah diperoleh pada tugas akhir ini.

#### 6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil analisa kebutuhan informasi evaluasi kinerja *medrep* dari berbagai sudut pandang pengguna dan dari hasil implementasi perangkat lunak yang telah dilakukan adalah sebagai berikut:

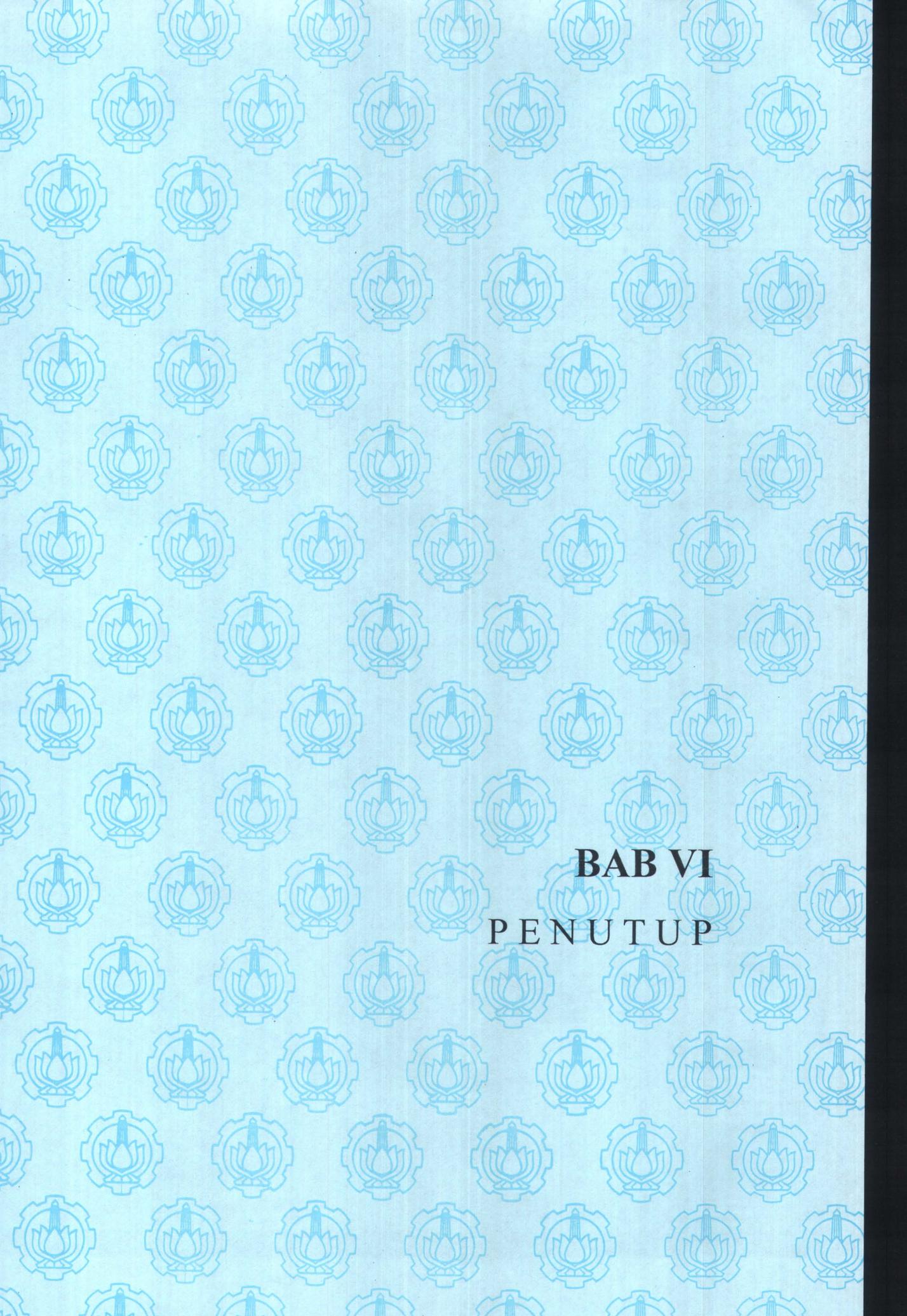
1. Sistem informasi pelaporan penjualan *medrep* ini memiliki fungsi utama evaluasi kinerja *medrep* dengan fitur tambahan seperti monitoring penjualan pada tiap region, manajemen dokter, manajemen obat, dan manajemen apotek yang diharapkan dapat memudahkan pihak manajer dalam proses pencarian informasi dan pembuatan keputusan pemasaran mengingat segala informasi yang dibutuhkan telah berada dalam sebuah sistem yang terintegrasi.
2. Berdasar data pada saat pengumpulan informasi, parameter yang digunakan baik untuk menghitung target penjualan maupun indeks kinerja *medrep* sudah ditentukan. Sehingga desain sistem yang dibuat berdasarkan parameter yang sudah ditentukan dan tidak mendukung adanya fleksibilitas dalam penambahan parameter baru.
3. Berdasar desain yang dibuat, dapat mengimplementasikan sistem yang menghasilkan penilaian kinerja *medrep* yang lebih detail, lebih fleksibel dalam perubahan bobot parameter, dan mencetak laporan dengan format isi yang dinamis.

4. Berdasar hasil uji coba, didapatkan bahwa beberapa fungsi untuk mengevaluasi kinerja *medrep* memiliki kesesuaian dengan proses manual, bahkan dapat mengurangi resiko kesalahan, dan memberikan gambaran yang lebih menyeluruh terhadap kondisi pemasaran perusahaan.

## 6.1 Saran

Beberapa hal yang diharapkan dapat dikembangkan di masa mendatang adalah sebagai berikut:

1. Perlunya *handler* khusus yang menangani proses keamanan pemasukan data transaksi, karena sistem ini bersifat *online* dan berhubungan langsung dengan basis data perusahaan.
2. Perlu integrasi dengan sistem distribusi dan penjualan Enseval, sehingga nantinya diharapkan proses pengukuran kinerja penjualan dan pemasaran dapat dilakukan bersama-sama.



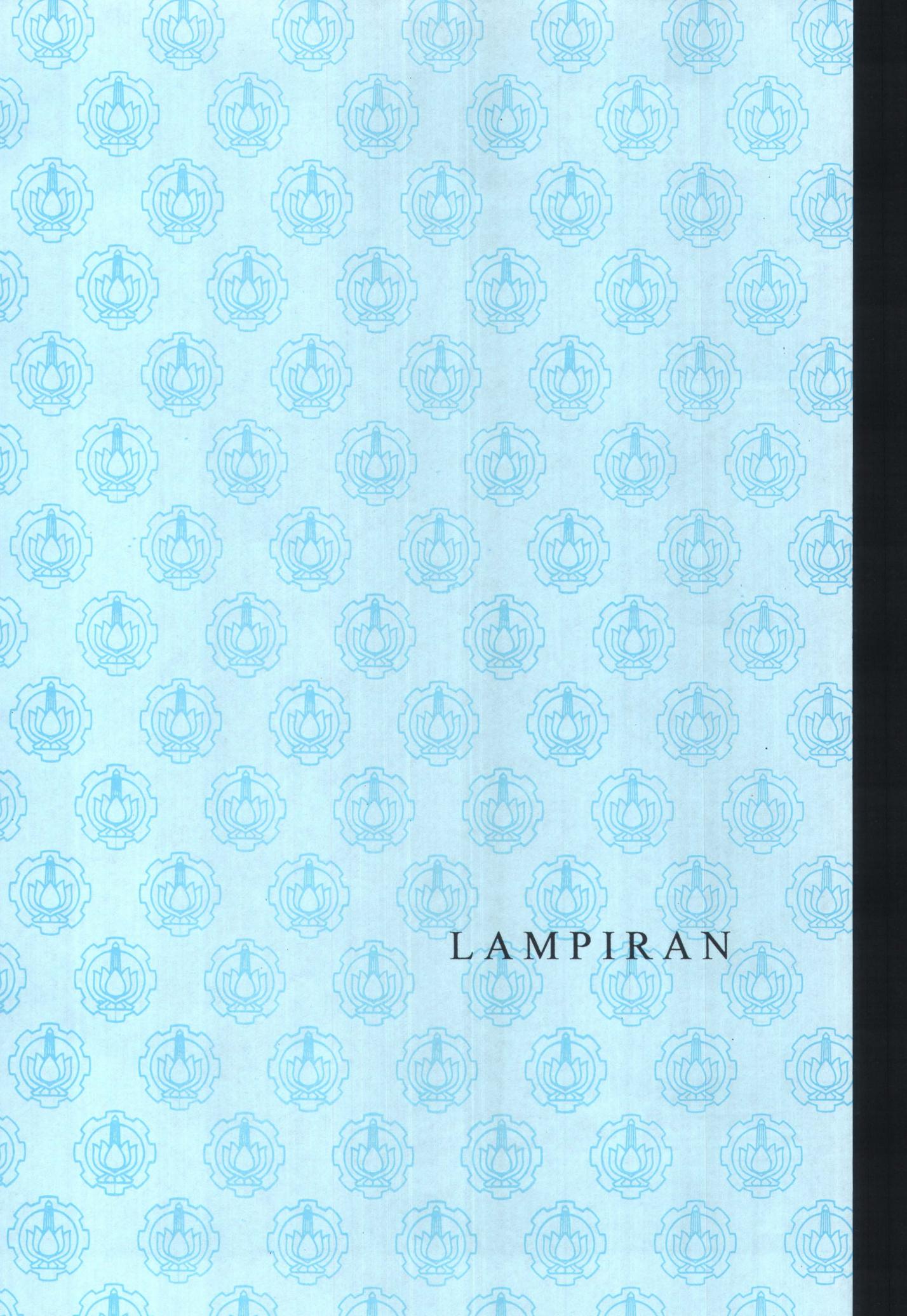
**BAB VI**  
**PENUTUP**



DAFTAR PUSTAKA

## DAFTAR PUSTAKA

- [ASH-00] Ashill, Nicholas J. and Jobber, David; *Marketing Information System (MkIS) Design Characteristics: Preliminary Findings and Future Research Directions*, 2000, ANZMAC.
- [BEN-04] Bentley, Lonnie D. and Dittman, Kevin C. and Whitten, Jeffery L.; *System Analysis and Design Method*, 2004, The McGraw-Hill Companies Inc.
- [DHA-03] Dharwiyanti, Sri; Wahono, Romi Satria. 2003, *Pengantar Unified Modelling Language (UML)*, IlmuKomputer.Com.
- [DYC-02] Dyche, Jill; *The CRM Handbook*, 2002, Addison-Wesley.
- [KOT-00] Kotler, Philip; *Marketing Management*, 2000, Prentice Hall.
- [MCL-01] McLeod Jr, Raymond; *Sistem Informasi Manajemen*, 2001, Prenhalindo.
- [PEN-02] Pender, Thomas A; *UML Weekend Crash Course*, 2002, Wiley Publishing Inc.
- [SCH-02] Schwalbe, Katty; *Information Technology Project Management*, 2002, Thomson Course Technology.



LAMPIRAN

## **Lampiran A**

### **Narasi Use Case**

## Use Case Mengelola Data Penjualan

<b>Nama Use Case</b>	Mengelola Data Penjualan	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Sales	
<b>Pelaku Partisipasi Lain</b>	Apotek (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Enseval - Pemilik data penjualan asli	
<b>Deskripsi</b>	Use Case ini mendeskripsikan sebuah kejadian dimana seorang Sales melakukan input data penjualan dari tiap apotek tiap hari	
<b>Pra Kondisi</b>	Sales yang melakukan input data harus terdaftar dan hanya memiliki hak akses pada apotek tertentu	
<b>Sasaran</b>	Use Case ini diawali setelah login sales di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Sales meminta sistem untuk menampilkan form untuk melakukan manajemen data transaksi</p> <p><b>Langkah 3:</b> Sales memilih apotek yang akan diinputkan data transaksinya</p> <p><b>Langkah 5:</b> Sales memasukkan data transaksi sesuai form yang disediakan sistem</p> <p><b>Langkah 7:</b> Sales memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan menampilkan form untuk melakukan manajemen data transaksi penjualan</p> <p><b>Langkah 4:</b> Sistem memanggil Use Case Mengelola Data Apotek</p> <p><b>Langkah 6:</b> Sekali Sales melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 8:</b> Sistem menyimpan data apotek dalam Database</p> <p><b>Langkah 9:</b> Sistem menampilkan pesan bahwa data apotek telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 3:</b> Jika Sales tidak menemukan data apotek dalam pilihan, maka Sales harus melakukan pengelolaan kembali Use Case manajemen apotek</p> <p><b>Alt-Langkah 6:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 7:</b> Jika sales melihat ada kesalahan dalam menginputkan, sales menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data transaksi	
<b>Post Kondisi</b>	Data transaksi penjualan tiap hari telah tersimpan dalam Database	
<b>Aturan Bisnis</b>	<ol style="list-style-type: none"> <li>1. Tiap Sales hanya dapat melakukan input data transaksi pada apotek yang berada di wilayah lingkungan kerjanya.</li> <li>2. Apotek dapat melakukan entry data transaksi sendiri setelah melakukan koordinasi dengan pihak Sales</li> </ol>	
<b>Batasan dan Spesifikasi Implementasi</b>	<ol style="list-style-type: none"> <li>1. Form input data transaksi untuk setiap Sales yang memiliki hak melakukan input data</li> <li>2. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</li> <li>3. Frekuensi - diestimasikan Use Case ini akan dieksekusi 400 kali per hari. Use Case ini seharusnya mendukung sampai 25 Sales secara bersamaan. Puncak aktifitas Use Case ini diperkirakan jatuh pada hari Senin setiap minggunya</li> </ol>	
<b>Asumsi</b>	Proses koordinasi entry data transaksi antara pihak sales dengan apotek dilakukan secara terpisah (tidak termasuk dalam Use Case)	
<b>Masalah Terbuka</b>	Tidak Ada	

## Use Case Mengelola Data Apotek

<b>Nama Use Case</b>	Mengelola Data Apotek	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Sales	
<b>Pelaku Partisipan Lain</b>	Apotek (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Enseval - Yang tertarik untuk mengetahui proses distribusi dan penjualan dari tiap apotek	
<b>Deskripsi</b>	Use Case ini mendeskripsikan sebuah kejadian dimana seorang Sales melakukan input data mengenai sebuah apotek	
<b>Pra Kondisi</b>	Sales yang melakukan input data harus terdaftar dan hanya memiliki hak akses untuk mengelola data apotek.	
<b>Sasaran</b>	Use Case ini diawali setelah login sales di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Sales meminta sistem menampilkan form untuk manajemen data apotek</p> <p><b>Langkah 3:</b> Sales menginputkan data apotek sesuai dengan form yang ditampilkan oleh sistem</p> <p><b>Langkah 5:</b> Sales memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data apotek</p> <p><b>Langkah 4:</b> Sekali Sales melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan data apotek dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa data apotek telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika sales melihat ada kesalahan dalam menginputkan, sales menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data apotek	
<b>Post Kondisi</b>	Data apotek telah tersimpan dalam database	
<b>Aturan Bisnis</b>	1. Tiap Sales hanya dapat mengelola data apotek yang berada dalam wilayah lingkungan kerjanya	
<b>Batasan dan Spesifikasi Implementasi</b>	<p>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</p> <p>2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inisialisasi input data apotek. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data apotek</p>	
<b>Asumsi</b>	1. Data sebuah apotek akan dikelola oleh seorang sales	
<b>Masalah Terbuka</b>	Tidak ada	

## Use Case Mengelola Data Dokter

<b>Nama Use Case</b>	Mengelola Data Dokter	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Supervisor	
<b>Pelaku Partisipan Lain</b>	Dokter (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Manajer - untuk pengelolaan data target region	
<b>Deskripsi</b>	Use Case ini mendeskripsikan sebuah kejadian ketika seorang Supervisor melakukan manajemen data dokter yang berada di region lingkungan kerjanya	
<b>Pra Kondisi</b>	Supervisor yang melakukan input data harus terdaftar dan hanya memiliki hak akses pada dokter di dalam region	
<b>Sasaran</b>	Use Case ini diawali setelah login supervisor di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Supervisor meminta sistem menampilkan form untuk manajemen data dokter</p> <p><b>Langkah 3:</b> Supervisor menginputkan data dokter sesuai dengan form yang disediakan oleh sistem</p> <p><b>Langkah 5:</b> Supervisor harus meng-assign seorang medrep yang bertanggung jawab memasarkan</p> <p><b>Langkah 8:</b> Supervisor memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data dokter</p> <p><b>Langkah 4:</b> Sekali Supervisor melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem akan memanggil Use Case Manajemen medrep</p> <p><b>Langkah 7:</b> Sistem akan menampilkan pesan konfirmasi</p> <p><b>Langkah 9:</b> Sistem menyimpan data dokter dalam Database</p> <p><b>Langkah 10:</b> Sistem menampilkan pesan data dokter telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika Supervisor tidak menemukan data medrep, maka Supervisor harus melakukan pengelolaan kembali Use Case manajemen medrep</p> <p><b>Alt-Langkah 8:</b> Jika supervisor melihat ada kesalahan dalam menginputkan, Supervisor menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data dokter	
<b>Post Kondisi</b>	Data dokter tersimpan dalam Database	
<b>Aturan Bisnis</b>	1. Tiap Supervisor hanya dapat melakukan input data pada dokter yang berada di wilayah lingkungan kerjanya.	
<b>Batasan dan Spesifikasi Implementasi</b>	2. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja 2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inialisasi input data dokter. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data dokter	
<b>Asumsi</b>	1. Data seorang dokter akan dikelola oleh seorang supervisor	
<b>Masalah Terbuka</b>	1. Bagaimana jika seorang dokter dari sebuah region pindah lokasi ke region lainnya?	

## Use Case Mengelola Data Medrep

<b>Nama Use Case</b>	Mengelola Data Medrep	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Supervisor	
<b>Pelaku Partisipan Lain</b>	Medrep (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Manager - selaku pimpinan dan pemilik data medrep	
<b>Deskripsi</b>	Use Case ini digunakan untuk mendeskripsikan sebuah kejadian dimana Supervisor melakukan manajemen data medrep yang ada di bawahnya	
<b>Pra Kondisi</b>	Supervisor yang melakukan input data harus terdaftar dan memiliki hak akses untuk mengelola data medrep	
<b>Sasaran</b>	Use Case ini diawali setelah login Supervisor di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Supervisor meminta sistem menampilkan form untuk manajemen data Medrep</p> <p><b>Langkah 3:</b> Supervisor menginputkan data medrep sesuai dengan form yang ditampilkan oleh sistem</p> <p><b>Langkah 5:</b> Supervisor memeriksa inputan kemudian memberikan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data medrep</p> <p><b>Langkah 4:</b> Sekali Supervisor melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan data medrep dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa data medrep telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika supervisor melihat ada kesalahan dalam menginputkan, Supervisor menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data medrep	
<b>Post Kondisi</b>	Data medrep telah tersimpan dalam database	
<b>Aturan Bisnis</b>	1. Tiap Supervisor hanya dapat mengelola data medrep yang berada dibawahnya	
<b>Batasan dan Spesifikasi Implementasi</b>	<p>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</p> <p>2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inialisasi input data medrep. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data medrep</p>	
<b>Asumsi</b>	1. Data seorang medrep akan dikelola oleh seorang Supervisor	
<b>Masalah Terbuka</b>	Tidak ada	

## Use Case Mengelola Data Visitasi

<b>Nama Use Case</b>	Mengelola Data Visitasi	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Medrep	
<b>Pelaku Partisipan Lain</b>	Dokter (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Supervisor - Monitoring dokter di region	
<b>Deskripsi</b>	Use Case ini digunakan oleh supervisor untuk melaporkan hasil dan kegiatan yang dilakukan medrep ketika melakukan kunjungan ke dokter	
<b>Pra Kondisi</b>	Supervisor yang melakukan input data harus terdaftar dan memiliki hak akses untuk manajemen data visitasi dari medrep di bawahnya	
<b>Sasaran</b>	Use Case ini diawali setelah login supervisor di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Supervisor meminta sistem menampilkan form manajemen data visitasi</p> <p><b>Langkah 3:</b> Supervisor memilih medrep yang melakukan visitasi</p> <p><b>Langkah 5:</b> Supervisor memilih dokter yang dikunjungi</p> <p><b>Langkah 7:</b> supervisor memasukkan data visitasi sesuai form dalam sistem</p> <p><b>Langkah 9:</b> Supervisor memeriksa inputan dan melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data visitasi</p> <p><b>Langkah 4:</b> Sistem memanggil Use Case Mengelola Data Medrep</p> <p><b>Langkah 6:</b> Sistem memanggil Use Case Mengelola Data Dokter</p> <p><b>Langkah 8:</b> Sekali supervisor melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 10:</b> Sistem menyimpan data visitasi dalam Database</p> <p><b>Langkah 11:</b> Sistem menampilkan pesan bahwa data visitasi telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 3:</b> Jika Supervisor tidak menemukan data medrep dalam pilihan, maka Supervisor harus terlebih dahulu melakukan Use Case Mengelola Data Medrep</p> <p><b>Alt-Langkah 5:</b> Jika Supervisor tidak menemukan data dokter dalam pilihan, maka Supervisor harus terlebih dahulu melakukan Use Case Mengelola Data Dokter</p> <p><b>Alt-Langkah 8:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan pesan dan meminta user mengisi form kembali</p> <p><b>Alt-Langkah 9:</b> Jika Supervisor melihat ada kesalahan dalam menginputkan, Supervisor menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data visitasi	
<b>Post Kondisi</b>	Data visitasi medrep telah tersimpan dalam database	
<b>Aturan Bisnis</b>	1. Jenis kegiatan yang dilaporkan medrep dalam manajemen visitasi terdiri dari Memperkenalkan produk baru, mengingatkan produk lama, dan kegiatan lain yang bersifat kondisional	
<b>Batasan dan Spesifikasi Implementasi</b>	1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja 2. Frekuensi - Use Case ini diestimasikan akan dieksekusi 150 kali sehari. dengan prediksi akan ada 20 Medrep yang mengakses secara bersamaan	
<b>Asumsi</b>	1. Data visitasi dari seorang dokter akan dikelola oleh seorang Supervisor	
<b>Masalah Terbuka</b>	Tidak ada	

## Use Case Mengelola Bobot

<b>Nama Use Case</b>	Mengelola Bobot	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Manajer	
<b>Pelaku Partisipan Lain</b>	Tidak ada	
<b>Stakeholder Lain yang Berminat</b>	Supervisor - bersama-sama mendiskusikan bobot yang sesuai dengan kondisi di lapangan	
<b>Deskripsi</b>	Use Case ini mendeskripsikan sebuah kejadian dimana seorang Manajer manajemen data bobot untuk perhitungan target penjualan tiap region	
<b>Pra Kondisi</b>	Manajer yang melakukan input data harus terdaftar dan hanya memiliki hak akses untuk melakukan manajemen bobot	
<b>Sasaran</b>	Use Case ini diawali setelah login manajer di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Manager meminta sistem menampilkan form untuk mengupdate bobot</p> <p><b>Langkah 3:</b> Manajer mengupdate bobot dengan bobot perhitungan baru</p> <p><b>Langkah 5:</b> Manajer memeriksa hasil update dan memberikan jawaban atas konfirmasi sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman untuk mengupdate bobot perhitungan</p> <p><b>Langkah 4:</b> Sekali Manajer melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan nilai bobot dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa bobot yang baru telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan pesan dan meminta user mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika sales melihat ada kesalahan dalam menginputkan, manajer menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini digunakan untuk mengupdate bobot perhitungan	
<b>Post Kondisi</b>	Nilai bobot yang baru telah tersimpan dalam database	
<b>Aturan Bisnis</b>	<ol style="list-style-type: none"> <li>Hanya pihak manajer yang dapat melakukan update bobot perhitungan</li> <li>Bobot yang dikelola masing-masing adalah bobot untuk jenis region, jumlah dokter, jumlah penduduk dari sebuah region, biaya alokasi per kunjungan, bobot untuk kunjungan per hari, bobot untuk pendapatan per kunjungan, bobot untuk jumlah biaya kunjungan, dan bobot untuk realisasi penjualan</li> </ol>	
<b>Batasan dan Spesifikasi Implementasi</b>	<ol style="list-style-type: none"> <li>Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</li> <li>Frekuensi - Use Case ini akan banyak dieksekusi pada proses inisialisasi bobot perhitungan. Selanjutnya, Use Case ini akan dieksekusi setiap manajer merasa perlu untuk merevisi bobot</li> </ol>	
<b>Asumsi</b>	1. Hanya user yang berada dalam grup manajer saja yang bisa melakukan update bobot	
<b>Masalah Terbuka</b>	1. Bagaimana proses penilaian kinerja medrep ketika parameter perhitungan berubah?	

## Use Case Mengelola Data User

<b>Nama Use Case</b>	Mengelola Data User	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Administrator	
<b>Pelaku Partisipan Lain</b>	Tidak ada	
<b>Stakeholder Lain yang Berminat</b>	Semua aktor yang terlibat dalam sistem	
<b>Deskripsi</b>	Use Case ini digunakan oleh Administrator untuk mengelola user beserta privilege-privilegenya	
<b>Pra Kondisi</b>	Administrator adalah pihak yang melakukan manajemen seluruh sistem dan telah memiliki daftar calon user	
<b>Sasaran</b>	Use Case ini diawali setelah login admin di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Admin meminta sistem menampilkan form untuk manajemen data user</p> <p><b>Langkah 3:</b> Admin melakukan manajemen data baik untuk insert, update, maupun delete data user pada form yang disediakan sistem</p> <p><b>Langkah 5:</b> Admin memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form manajemen data user</p> <p><b>Langkah 4:</b> Sekali Admin melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan data user dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa data user telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 6:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan pesan dan meminta user mengisi form kembali</p> <p><b>Alt-Langkah 7:</b> Jika admin melihat ada kesalahan dalam menginputkan, admin menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini digunakan untuk mengelola user yang terlibat dalam sistem	
<b>Post Kondisi</b>	Data user telah tersimpan dalam database	
<b>Aturan Bisnis</b>	1. Secara struktural, tugas admin hanya mengelola data user dan obat. Namun secara fungsional, admin bertanggung jawab penuh atas berjalannya sistem dengan baik	
<b>Batasan dan Spesifikasi Implementasi</b>	<p>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</p> <p>2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inialisasi input data user. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data user</p>	
<b>Asumsi</b>	<p>1. Seluruh data user akan dikelola oleh seorang administrator</p> <p>2. Administrator diharapkan dapat menangani setiap permasalahan baik yang berhubungan dengan berjalannya sistem, manajemen data, dan lain-lain</p>	
<b>Masalah Terbuka</b>	<p>1. Berapakah jumlah login admin yang ideal?</p> <p>2. Jika hanya satu, siapakah yang bertanggung jawab ketika sistem bermasalah sedangkan saat itu admin tidak bertugas?</p>	

## Use Case Mengelola Data Region

<b>Nama Use Case</b>	Mengelola Data Region	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Supervisor	
<b>Pelaku Partisipan Lain</b>	Medrep (Server Eksternal)	
<b>Stakeholder Lain yang Berminat</b>	Manajer - monitoring dan evaluasi data region	
<b>Deskripsi</b>	Use Case ini digunakan oleh supervisor untuk memasukkan data region yang dipimpinnya untuk penentuan target penjualan region	
<b>Pra Kondisi</b>	Supervisor yang melakukan input data harus terdaftar dan memiliki hak akses untuk manajemen data pada region yang dipimpinnya	
<b>Sasaran</b>	Use Case ini diawali setelah login supervisor di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Supervisor meminta sistem menampilkan form untuk manajemen data region</p> <p><b>Langkah 3:</b> Supervisor mengisi/mengupdate data jenis region dan jumlah penduduk region disesuaikan setiap parameter dan bobot</p> <p><b>Langkah 5:</b> Supervisor memeriksa inputan kemudian melakukan jawaban atas konfirmasi sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data region</p> <p><b>Langkah 4:</b> Sistem melakukan validasi dan menampilkan pesan konfirmasi jika ditemukan kesalahan</p> <p><b>Langkah 6:</b> Sistem menyimpan data region dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa data region telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 3:</b> Jika data bobot tidak ada atau tidak valid, supervisor harus berkoordinasi dengan pihak manajer yang mengelola manajemen bobot</p> <p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika supervisor melihat ada kesalahan dalam menginputkan, supervisor menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data region	
<b>Post Kondisi</b>	Data region telah tersimpan dalam database	
<b>Aturan Bisnis</b>	<ol style="list-style-type: none"> <li>1. Tiap supervisor hanya dapat mengelola data pada region yang dipimpinnya</li> <li>2. Aspek yang digunakan untuk menentukan target penjualan tiap region adalah (1) jenis region dan (2) jumlah penduduk yang nilainya diinputkan oleh supervisor dan (3) jumlah dokter yang nilainya didapat dari proses database</li> </ol>	
<b>Batasan dan Spesifikasi Implementasi</b>	<ol style="list-style-type: none"> <li>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</li> <li>2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inisialisasi input data region. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data region</li> </ol>	
<b>Asumsi</b>	1. Data sebuah region akan dikelola oleh medrep yang dipimpinnya	
<b>Masalah Terbuka</b>	Tidak ada	

## Use Case Analisa

<b>Nama Use Case</b>	Analisa	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Supervisor, manajer,	
<b>Pelaku Partisipan Lain</b>	Semua aktor pengguna sistem	
<b>Stakeholder Lain yang Berminat</b>	Medrep - monitoring hasil analisa	
<b>Deskripsi</b>	Use Case ini digunakan oleh supervisor untuk memberikan rekomendasi atas kinerja medrep di bawahnya dan oleh pihak manajer untuk menganalisa kinerja secara keseluruhan	
<b>Pra Kondisi</b>	Supervisor dan manajer yang melakukan analisa medrep secara keseluruhan harus terdaftar dalam sistem	
<b>Sasaran</b>	Use Case ini diawali setelah login manajer atau supervisor di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Supervisor meminta sistem menampilkan form untuk memberikan rekomendasi kinerja medrep</p> <p><b>Langkah 3:</b> Supervisor mengisi form rekomendasi kinerja medrep</p> <p><b>Langkah 5:</b> Supervisor memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p> <p><b>Langkah 8:</b> Manajer meminta sistem menampilkan form hasil analisa penilaian kinerja medrep</p> <p><b>Langkah 10:</b> Manajer meminta hasil analisa pemasaran lainnya</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk pemberian rekomendasi atas kinerja medrep</p> <p><b>Langkah 4:</b> Sistem melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan rekomendasi supervisor dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa rekomendasi supervisor telah tersimpan dalam database</p> <p><b>Langkah 9:</b> Sistem menampilkan halaman analisa kinerja medrep yang telah mendapatkan rekomendasi sebelumnya oleh supervisor</p> <p><b>Langkah 11:</b> Sistem menampilkan halaman analisa yang diminta oleh pihak manajer</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan peringatan dan meminta user untuk mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika supervisor melihat ada kesalahan dalam menginputkan, supervisor menekan cancel dan melakukan input data kembali</p> <p><b>Alt-Langkah 9:</b> Jika halaman kinerja medrep belum direkomendasi, manajer harus berkoordinasi dengan supervisor yang bersangkutan</p>	
<b>Kesimpulan</b>	Use Case ini digunakan pihak manajer untuk menganalisa kinerja medrep yang telah direkomendasi supervisornya dan hasil-hasil pemasaran lainnya	
<b>Post Kondisi</b>	Manajer dapat melakukan berbagai analisa untuk kepentingan perusahaan	
<b>Aturan Bisnis</b>	<ol style="list-style-type: none"> <li>1. Tiap supervisor hanya dapat memberikan rekomendasi kepada medrep di bawahnya</li> <li>2. Hasil analisa lain yang disediakan sistem adalah (1) analisa penjualan tiap apotek, (2) analisa penjualan tiap dokter, dan (3) analisa penjualan tiap region</li> <li>3. Hasil analisa berupa tabel dan grafik</li> <li>4. Hasil analisa akan digenerate sistem tiap dua minggu sekali</li> </ol>	

<b>Batasan dan Spesifikasi Implementasi</b>	<ol style="list-style-type: none"><li>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</li><li>2. Frekuensi - Use Case ini akan banyak dieksekusi oleh pihak supervisor untuk memberikan rekomendasi kepada medrepnya. Manajer akan mengeksekusi Use Case ini dua minggu sekali untuk melihat hasil analisa</li></ol>
<b>Asumsi</b>	<ol style="list-style-type: none"><li>1. Data rekomendasi seorang medrep akan dikelola oleh seorang supervisor</li></ol>
<b>Masalah Terbuka</b>	Diperlukan tenggat waktu untuk "memaksa" pihak supervisor memberikan rekomendasi sebelum pihak manajer melakukan analisa setiap dua minggu sekali

## Use Case Validasi

<b>Nama Use Case</b>	Validasi	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Administrator	
<b>Pelaku Partisipasi Lain</b>	Tidak ada	
<b>Stakeholder Lain yang Berminat</b>	Semua pihak yang berinteraksi dengan sistem	
<b>Deskripsi</b>	Use Case ini adalah Use Case abstrak yang digunakan untuk memvalidasi semua user yang menggunakan sistem	
<b>Pra Kondisi</b>	Admin telah melakukan manajemen data user	
<b>Sasaran</b>	Use Case ini diawali setelah user mengakses halaman login	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> User meminta sistem menampilkan halaman login</p> <p><b>Langkah 3:</b> User memasukkan user name, password, dan login sebagai, kemudian menekan "OK"</p>	<p><b>Langkah 2:</b> Sistem menampilkan halaman yang menyediakan form login untuk user</p> <p><b>Langkah 4:</b> Sistem memverifikasi form login</p> <p><b>Langkah 5:</b> Sistem membandingkan form user name, password dan log as dari inputan user dengan data yang ada dalam database</p> <p><b>Langkah 6:</b> Sistem memvalidasi user dan menampilkan halaman yang sesuai loginnya</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 3:</b> Jika user menemukan kesalahan dalam melakukan input, user menekan cancel untuk mengulang proses login</p> <p><b>Alt-Langkah 4:</b> Jika sistem menemukan bahwa field user name dan atau password belum terisi, sistem akan menampilkan form konfirmasi</p> <p><b>Alt-Langkah 5:</b> Jika entry login yang dimasukkan user tidak sesuai atau tidak ada dalam database, sistem mempersilahkan user untuk melakukan login kembali</p> <p><b>Alt-Langkah 6:</b></p> <ul style="list-style-type: none"> <li>* Jika login sebagai sales, sistem akan menampilkan halaman sales.html</li> <li>* Jika login sebagai apotek, sistem akan menampilkan halaman apotek.html</li> <li>* Jika login sebagai medrep, sistem akan menampilkan halaman medrep.html</li> <li>* Jika login sebagai supervisor, sistem akan menampilkan halaman sprvsr.html</li> <li>* Jika login sebagai manager, sistem akan menampilkan halaman manager.html</li> <li>* Jika login sebagai administrator, sistem akan menampilkan halaman admin.html</li> </ul>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap login user	
<b>Post Kondisi</b>	User telah masuk dalam sistem	
<b>Aturan Bisnis</b>	1. Untuk pengelolaan hak login beserta akses yang didapat dalam sistem, user harus berkoordinasi dengan administrator sebagai pengelola manajemen user	
<b>Batasan dan Spesifikasi Implementasi</b>	1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja 2. Frekuensi - Use Case ini diestimasikan akan dieksekusi 600-700 kali sehari. Dengan prediksi akan ada 100 user yang mengakses secara bersamaan	
<b>Asumsi</b>	1. Satu orang hanya akan mendapatkan satu login	
<b>Masalah Terbuka</b>	Tidak ada	

## Use Case Mengelola Data Obat

<b>Nama Use Case</b>	Mengelola Data Obat	
<b>Prioritas</b>	Tinggi	
<b>Pelaku Bisnis Utama</b>	Administrator	
<b>Pelaku Partisipan Lain</b>	Tidak ada	
<b>Stakeholder Lain yang Berminat</b>	Semua pihak yang menggunakan sistem	
<b>Deskripsi</b>	Use Case ini mendeskripsikan sebuah kejadian dimana seorang administrator melakukan input data obat	
<b>Pra Kondisi</b>	Admin yang melakukan input data harus terdaftar dan hanya memiliki hak akses untuk mengelola data obat.	
<b>Sasaran</b>	Use Case ini diawali setelah login admin di-autentifikasi sistem	
<b>Bidang Khas Suatu Event</b>	<b>Kegiatan Pelaku</b>	<b>Respon Sistem</b>
	<p><b>Langkah 1:</b> Admin meminta sistem menampilkan form untuk manajemen data obat</p> <p><b>Langkah 3:</b> Admin menginputkan data obat sesuai dengan form yang ditampilkan oleh sistem</p> <p><b>Langkah 5:</b> Admin memeriksa inputan kemudian melakukan jawaban atas konfirmasi dari sistem</p>	<p><b>Langkah 2:</b> Sistem merespon dengan cara menampilkan halaman berisi form untuk manajemen data obat</p> <p><b>Langkah 4:</b> Sekali Admin melengkapi form yang diminta, sistem segera melakukan validasi dan menampilkan pesan konfirmasi</p> <p><b>Langkah 6:</b> Sistem menyimpan data obat dalam Database</p> <p><b>Langkah 7:</b> Sistem menampilkan pesan bahwa data obat telah tersimpan dalam database</p>
<b>Bidang Alternatif</b>	<p><b>Alt-Langkah 4:</b> Jika sistem menemukan kesalahan atau data yang tidak sesuai, sistem menampilkan pesan dan meminta user mengisi form kembali</p> <p><b>Alt-Langkah 5:</b> Jika admin melihat ada kesalahan dalam menginputkan, sales menekan cancel dan melakukan input data kembali</p>	
<b>Kesimpulan</b>	Use Case ini melakukan pengecekan awal terhadap entry data obat	
<b>Post Kondisi</b>	Data obat telah tersimpan dalam database	
<b>Aturan Bisnis</b>	1. Hanya pihak administrator yang dapat melakukan manajemen data obat	
<b>Batasan dan Spesifikasi Implementasi</b>	<p>1. Use Case ini akan tersedia dari jam 08.00 sampai jam 18.00 pada hari kerja</p> <p>2. Frekuensi - Use Case ini akan banyak dieksekusi pada proses inisialisasi input data obat. Selanjutnya, Use Case ini akan dieksekusi setiap ada perubahan baik insert, update maupun delete data obat</p>	
<b>Asumsi</b>	1. Data obat sudah disediakan perusahaan	
<b>Masalah Terbuka</b>	Tidak ada	

# Lampiran B

## Data Gathering



Narasumber : Bpk Alan	
Posisi : Manajer Pemasaran bagian <i>Ethycal</i>	
Hasil :	
1	Informasi penjualan yang dibutuhkan manajer adalah - Data penjualan tiap dokter - Data penjualan tiap apotek - Data penjualan tiap obat Dimana laporan harus tersedia setiap dua mingguan
2	Target penjualan didapat dari hasil perhitungan antara kondisi di region dengan bobot-bobot perhitungan yang telah ditentukan oleh pihak manajer dengan berkoordinasi dengan supervisor yang mengetahui kondisi region.
3	Pembobotan dan pengalokasian biaya untuk visitasi dibuat setiap bulan dengan keputusan mutlak berada di tangan manajer. Aspek-aspek pembobotan region adalah jenis region, jumlah penduduk region, dan jumlah dokter di region tersebut.
5	Kalbe Surabaya memiliki 13 region dengan rata-rata berisi 2 sampai 3 kecamatan. Pembagian kecamatan berdasarkan kedekatan lokasi dan luas geografis region. Manajer harus dapat mengetahui kondisi secara umum dari tiap region.
6	Parameter yang dievaluasi dari kinerja <i>medrep</i> adalah (1) jumlah kunjungan yang dilakukan <i>medrep</i> per hari (2) besarnya pendapatan yang diperoleh tiap kunjungan penjualan (3) nilai biaya kunjungan yang dibutuhkan <i>medrep</i> (4) Nilai total penjualan yang diperoleh <i>medrep</i> selama dua minggu.

Narasumber : Bpk Heru	
Posisi : Supervisor	
Hasil :	
1	Informasi penjualan yang dibutuhkan supervisor adalah - Data penjualan tiap dokter - Data penjualan tiap obat
2	Informasi yang diharapkan supervisor tentang data dokter adalah nama, alamat, nomor telepon, dan spesialisasi dokter untuk mempermudah proses pemberian undangan seminar atau simposium. Supervisor juga bertanggung jawab pada data semua dokter di region
3	Informasi yang diharapkan supervisor tentang data <i>medrep</i> adalah nama, alamat, dan nomor telepon. Supervisor akan membawahi antara 3-4 <i>medrep</i> . Peng-assign-an seorang <i>medrep</i> untuk mendidik seorang dokter didasarkan pada jumlah dokter didikan dan lokasi dokter. Sehingga diharapkan akan ada keseimbangan target penjualan antara tiap <i>medrep</i> .
4	Data jenis region dan jumlah dokter didapat dari hasil survey supervisor di lokasi, sedangkan data jumlah penduduk didapat dari laporan BPS.
5	Pada proses evaluasi kinerja <i>medrep</i> , supervisor mengevaluasi tiap parameter yang telah ditentukan, membandingkan dengan kinerja <i>medrep</i> lainnya pada periode yang sama dan memberikan rekomendasinya atas kinerja <i>medrep</i> yang bersangkutan.

Narasumber	: Bpk Khoirul
Posisi	: <i>Medical Representative</i>
Hasil	:
1	Informasi penjualan yang dibutuhkan <i>medrep</i> adalah <ul style="list-style-type: none"> <li>- Data penjualan tiap dokter</li> <li>- Data penjualan tiap obat</li> </ul>
2	Kunjungan yang dilakukan <i>medrep</i> ke tiap dokter pada dasarnya terdiri atas 3 kategori yaitu memperkenalkan obat baru, mengingatkan obat lama, dan terakhir kegiatan kondisional yang tidak berhubungan dengan proses pemasaran seperti memberikan undangan simposium, seminar dll. Setiap melakukan visitasi <i>medrep</i> harus mendapatkan tanda tangan dokter sebagai bukti kunjungannya.
3	<i>Medrep</i> mengharapkan adanya proses pengevaluasian kinerja yang <i>fair</i> dari supervisor. Dengan adanya transparansi pada detail penjualan yang dicapai <i>medrep</i> .

Narasumber	: Bpk Budi
Posisi	: <i>Sales penjualan</i>
Hasil	:
1	Informasi penjualan yang dibutuhkan <i>salesr</i> adalah <ul style="list-style-type: none"> <li>- Data penjualan tiap apotek</li> <li>- Data penjualan tiap obat</li> </ul>
2	<i>Sales</i> mengelola (memasarkan, menjual, dan merekapitulasi data penjualan) dari 30-40 apotek di sebuah region.
3	Proses pencatatan data transaksi didasarkan pada resep yang didapat dari apotek. <i>Sales</i> tidak setiap hari datang di semua apotek karena itu proses pencatatan transaksi tidak dilakukan tiap hari
4	Setiap harinya <i>sales</i> rata-rata hanya mengunjungi 10 apotek. Biasanya <i>sales</i> beroperasi pada jam 9 pagi sampai jam 3 sore sebelum kembali ke kantor untuk merekapitulasi data transaksi yang didapat.

Narasumber	: Ibu Hendrianah
Posisi	: Kepala apotek RS Haji Surabaya
Hasil	:
1	Informasi penjualan yang dibutuhkan apotek adalah <ul style="list-style-type: none"> <li>- Data penjualan apoteknya</li> <li>- Data penjualan tiap obat</li> </ul>
2	Apotek melakukan rekapitulasi hasil penjualan pada malam hari atau setelah apotek tutup disebabkan banyaknya obat baik dari jenis obat, kemasan sampai produsen. Karena itu apotek baru bisa memberikan data penjualan hari ini di keesokan harinya
3	Apotek menyambut baik sistem ini, namun untuk pengaplikasiannya masih terganjal pada masalah penyediaan komputer dan jaringan internet.

## **Lampiran C**

### **Daftar Penjualan Unit**

# DAFTAR PENJUALAN UNIT

TANGGAL : 14 - 12 - 2005

REGION : 8

DETAILER : Agus Santoso

---

**ABADI**

AB2503	BROADCED	7	Rp	42000
AB2508	CRAVIT	11	Rp	52250
AB2527	BACTECYN	9	Rp	63000
AB2569	CEFSPAN	10	Rp	45000
AB2604	CEFOTAXIME	5	Rp	40000
CV1127	ANGIOTEN	8	Rp	44000
NR2432	NEUROTAM	8	Rp	60000
NR2467	NEUROCET	8	Rp	42000
NR2470	NEURALGIN	7	Rp	17500
RH0756	KALTROFEN	4	Rp	27000
OC1630	FARMORUBICIN	7	Rp	66500
HP3782	HEPASIL	5	Rp	41250

Rp 540500

**CEMPAKA**

AB2503	BROADCED	6	Rp	36000
AB2508	CRAVIT	11	Rp	52250
AB2527	BACTECYN	9	Rp	63000
AB2569	CEFSPAN	15	Rp	67500
AB2604	CEFOTAXIME	4	Rp	32000
CV1127	ANGIOTEN	8	Rp	44000
NR2432	NEUROTAM	16	Rp	120000
NR2467	NEUROCET	12	Rp	63000
NR2470	NEURALGIN	10	Rp	25000
RH0756	KALTROFEN	9	Rp	60750
OC1630	FARMORUBICIN	7	Rp	66500
HP3782	HEPASIL	5	Rp	41250

Rp 671250

**CERDAS**

AB2503	BROADCED	3	Rp	18000
AB2508	CRAVIT	5	Rp	23750
AB2527	BACTECYN	15	Rp	105000
AB2569	CEFSPAN	7	Rp	31500
AB2604	CEFOTAXIME	4	Rp	32000
CV1127	ANGIOTEN	8	Rp	44000
NR2432	NEUROTAM	11	Rp	82500
NR2467	NEUROCET	6	Rp	31500

NR2470	NEURALGIN	3	Rp	7500	
RH0756	KALTROFEN	8	Rp	54000	
OC1630	FARMORUBICIN	0	Rp	0	
HP3782	HEPASIL	5	Rp	41250	
					Rp 471000

## DEWATA

AB2503	BROADCED	10	Rp	60000	
AB2508	CRAVIT	5	Rp	23750	
AB2527	BACTECYN	7	Rp	49000	
AB2569	CEFSPAN	4	Rp	18000	
AB2604	CEFOTAXIME	4	Rp	32000	
CV1127	ANGIOTEN	8	Rp	44000	
NR2432	NEUROTAM	2	Rp	15000	
NR2467	NEUROCET	6	Rp	31500	
NR2470	NEURALGIN	6	Rp	15000	
RH0756	KALTROFEN	8	Rp	54000	
OC1630	FARMORUBICIN	3	Rp	28500	
HP3782	HEPASIL	5	Rp	41250	
					Rp 412000

## FARIA

AB2503	BROADCED	2	Rp	12000	
AB2508	CRAVIT	5	Rp	23750	
AB2527	BACTECYN	3	Rp	21000	
AB2569	CEFSPAN	0	Rp	0	
AB2604	CEFOTAXIME	1	Rp	8000	
CV1127	ANGIOTEN	4	Rp	22000	
NR2432	NEUROTAM	3	Rp	22500	
NR2467	NEUROCET	6	Rp	31500	
NR2470	NEURALGIN	7	Rp	17500	
RH0756	KALTROFEN	4	Rp	27000	
OC1630	FARMORUBICIN	3	Rp	28500	
HP3782	HEPASIL	0	Rp	0	
					Rp 213750

## KENCANA

AB2503	BROADCED	9	Rp	54000	
AB2508	CRAVIT	5	Rp	23750	
AB2527	BACTECYN	3	Rp	21000	
AB2569	CEFSPAN	11	Rp	49500	
AB2604	CEFOTAXIME	2	Rp	16000	
CV1127	ANGIOTEN	1	Rp	5500	
NR2432	NEUROTAM	3	Rp	22500	
NR2467	NEUROCET	0	Rp	0	
NR2470	NEURALGIN	7	Rp	17500	
RH0756	KALTROFEN	4	Rp	27000	
OC1630	FARMORUBICIN	3	Rp	28500	

HP3782	HEPASIL	6	Rp	49500	
PAHALA					Rp 314750
AB2503	BROADCED	3	Rp	18000	
AB2508	CRAVIT	5	Rp	23750	
AB2527	BACTECYN	2	Rp	14000	
AB2569	CEFSPAN	7	Rp	31500	
AB2604	CEFOTAXIME	4	Rp	32000	
CV1127	ANGIOTEN	8	Rp	44000	
NR2432	NEUROTAM	4	Rp	30000	
NR2467	NEUROCET	6	Rp	31500	
NR2470	NEURALGIN	7	Rp	17500	
PH0756	KALTROFEN	5	Rp	33750	
OC1630	FARMORUBICIN	4	Rp	38000	
HP3782	HEPASIL	5	Rp	41250	Rp 355250
RAJA					
AB2503	BROADCED	4	Rp	24000	
AB2508	CRAVIT	5	Rp	23750	
AB2527	BACTECYN	3	Rp	21000	
AB2569	CEFSPAN	4	Rp	18000	
AB2604	CEFOTAXIME	4	Rp	32000	
CV1127	ANGIOTEN	2	Rp	11000	
NR2432	NEUROTAM	0	Rp	0	
NR2467	NEUROCET	4	Rp	21000	
NR2470	NEURALGIN	6	Rp	15000	
RH0756	KALTROFEN	0	Rp	0	
OC1630	FARMORUBICIN	7	Rp	66500	
HP3782	HEPASIL	5	Rp	41250	Rp 273500
SAHABAT					
AB2503	BROADCED	3	Rp	18000	
AB2508	CRAVIT	0	Rp	0	
AB2527	BACTECYN	2	Rp	14000	
AB2569	CEFSPAN	3	Rp	13500	
AB2604	CEFOTAXIME	4	Rp	32000	
CV1127	ANGIOTEN	0	Rp	0	
NR2432	NEUROTAM	4	Rp	30000	
NR2467	NEUROCET	6	Rp	31500	
NR2470	NEURALGIN	4	Rp	10000	
RH0756	KALTROFEN	6	Rp	40500	
OC1630	FARMORUBICIN	4	Rp	38000	
HP3782	HEPASIL	5	Rp	41250	Rp 268750

AB2503	BROADCED	4	Rp	24000
AB2508	CRAVIT	2	Rp	9500
AB2527	BACTECYN	7	Rp	49000
AB2569	CEFSPAN	2	Rp	9000
AB2604	CEFOTAXIME	1	Rp	8000
CV1127	ANGIOTEN	0	Rp	0
NR2432	NEUROTAM	3	Rp	22500
NR2467	NEUROCET	4	Rp	21000
NR2470	NEURALGIN	7	Rp	17500
RH0756	KALTROFEN	2	Rp	13500
OC1630	FARMORUBICIN	4	Rp	38000
HP3782	HEPASIL	1	Rp	8250

Rp 220250

## WARAS

AB2503	BROADCED	2	Rp	12000
AB2508	CRAVIT	3	Rp	14250
AB2527	BACTECYN	7	Rp	49000
AB2569	CEFSPAN	4	Rp	18000
AB2604	CEFOTAXIME	4	Rp	32000
CV1127	ANGIOTEN	8	Rp	44000
NR2432	NEUROTAM	5	Rp	37500
NR2467	NEUROCET	2	Rp	10500
NR2470	NEURALGIN	6	Rp	15000
RH0756	KALTROFEN	8	Rp	54000
OC1630	FARMORUBICIN	3	Rp	28500
HP3782	HEPASIL	4	Rp	33000

Rp 347750

## YAHYA

AB2503	BROADCED	3	Rp	18000
AB2508	CRAVIT	5	Rp	23750
AB2527	BACTECYN	3	Rp	21000
AB2569	CEFSPAN	7	Rp	31500
AB2604	CEFOTAXIME	4	Rp	32000
CV1127	ANGIOTEN	8	Rp	44000
NR2432	NEUROTAM	4	Rp	30000
NR2467	NEUROCET	0	Rp	0
NR2470	NEURALGIN	4	Rp	10000
RH0756	KALTROFEN	5	Rp	33750
OC1630	FARMORUBICIN	4	Rp	38000
HP3782	HEPASIL	2	Rp	16500

Rp 298500

Rp 4387250

## **Lampiran D**

### **Tabel Deskripsi Kebutuhan**

si	Rangkuman Jawaban	Kebutuhan Sistem		
		Fungsional	Batasan	
	Pemberian bobot terhadap tiap parameter perlu dilakukan untuk mengantisipasi kondisi pemasaran yang berubah-ubah. Pemberian bobot dilakukan setiap bulan	Mengubah bobot, baik bobot untuk perhitungan target penjualan maupun bobot yang digunakan untuk mengevaluasi kinerja medrep.		
	Manajer membutuhkan sebuah alat untuk memonitor seluruh region, lengkap beserta detail region, nilai penjualan, maupun medrep, dokter maupun apotek yang terintegrasi	Melihat detail seluruh region, baik maupun supervisornya, jenis region, jumlah penduduk, detail setiap dokter, medrep, dan apotek yang ada di setiap region.		
	Dasar yang digunakan dalam mengevaluasi kinerja <i>medrep</i> adalah indeks kinerja dan rekomendasi supervisor	Melihat detail penilaian kinerja seluruh <i>medrep</i> beserta laporan penjualannya setiap dua minggu		Sistem tidak dapat melakukan perhitungan jika manajer tidak mengisi bobot
	Detail Informasi penjualan yang dibutuhkan manajer adalah tiap apotek, jenis obat dan tiap dokter	Melihat detail nilai penjualan untuk setiap dokter, obat, dan apotek, di seluruh region.		

si	Rangkuman Jawaban	Kebutuhan Sistem	
		Fungsional	Batasan
	<i>Medrep</i> membutuhkan sebuah sarana yang dapat digunakan untuk mengukur kinerjanya secara fair, tanpa ada unsur subyektifitas supervisor	Melihat evaluasi kinerja <i>medrep</i> dan rekomendasi yang diberikan supervisor untuk region dimana medrep berada	Diasumsikan lokasi <i>medrep</i> tetap, tidak berpindah region.
	Detail Informasi penjualan yang dibutuhkan <i>medrep</i> adalah tiap jenis obat dan tiap dokter	Melihat detail nilai penjualan untuk setiap dokter dan obat di regionnya	

isi	Rangkuman Jawaban	Kebutuhan Sistem	
		Fungsional	Batasan
	Supervisor perlu untuk melakukan manajemen region, sehingga setiap perubahan pada region dapat selalu dimonitor	Mengubah detail region, yaitu jenis region dan jumlah penduduk di region, sedangkan untuk jumlah dokter di region dihitung dari database.	Sistem tidak menangani kevalidan jenis dan jumlah penduduk region, isi detail region menjadi tanggung jawab supervisor
	Pengawasan dan manajemen <i>medrep</i> sangat dibutuhkan pihak supervisor. Supervisor perlu mengetahui nama, alamat dan no telp <i>medrep</i>	Melihat, menambah, mengubah dan menghapus data <i>medrep</i> yang menjadi bawahannya. Supervisor juga dapat memberikan rekomendasinya atas kinerja <i>medrep</i> setiap dua mingguan.	
	Supervisor perlu memberikan rekomendasinya atas kinerja <i>medrep</i> sebelum dievaluasi oleh manajer		
	Pengawasan dan manajemen dokter sangat dibutuhkan pihak supervisor. Supervisor perlu mengetahui nama, alamat dan no telp dokter	Melihat, menambah, mengubah dan menghapus data dokter yang ada di region. selain itu supervisor juga dapat meng-assign seorang <i>medrep</i> yang ditugaskan untuk mendidik dokter.	Diasumsikan lokasi dokter tetap, tidak berpindah region
	Pengelolaan data visitasi dilakukan oleh supervisor setelah <i>medrep</i> menyerahkan surat jalan yang berisi detail kegiatan kunjungan yang dilakukan <i>medrep</i>	Melihat, menambah, mengubah dan menghapus data visitasi yang telah dilakukan <i>medrep</i>	
	Detail Informasi penjualan yang dibutuhkan supervisor adalah tiap jenis obat dan tiap dokter	Melihat detail nilai penjualan untuk setiap dokter, obat, di regionnya.	

isi	Rangkuman Jawaban	Kebutuhan Sistem	
		Fungsional	Batasan
	Pengawasan dan manajemen apotek sangat dibutuhkan pihak sales. Sales perlu mengetahui nama, alamat dan no telp apotek	Melihat, menambah, mengubah dan menghapus data apotek yang berada di region tempat sales berada	Sistem tidak dapat menangani pembelian obat tanpa resep
	Proses pemasukan data transaksi dilakukan setiap hari oleh pihak sales. Data transaksi yang dimasukkan berasal dari resep yang didapat dari apotek.	Melihat, menambah, mengubah dan menghapus data transaksi yang diperoleh melalui resep yang didapat dari apotek	Diasumsikan semua resep memiliki data yang dibutuhkan sistem, yaitu tanggal, nama apotek, nama dokter, dan detail transaksi
	Detail Informasi penjualan yang dibutuhkan sales adalah tiap jenis obat dan tiap apotek	Melihat detail nilai penjualan untuk setiap obat, dan apotek, di regionnya.	

isi	Rangkuman Jawaban	Kebutuhan Sistem	
		Fungsional	Batasan
	Apotek membutuhkan sebuah <i>record</i> transaksi yang mencatat seluruh detail transaksi yang dilakukan	Melihat detail transaksi pada apoteknya untuk periode kapan pun	Monitoring hanya dapat dilakukan untuk obat-obatan dari Kalbe
	Apotek memerlukan data penjualan obat yang terjadi di daerahnya	Monitoring detail penjualan obat di region apotek, sebagai bahan pertimbangan penjualan	

isi	Rangkuman Jawaban	Kebutuhan Sistem	
		Fungsional	Batasan
	Admin adalah pihak yang bertanggung jawab atas jalannya sistem. Admin banyak menghabiskan waktunya di kantor untuk mengelola data-data administratif	Manajemen Data <i>User</i> Manajemen DataObat	Data obat beserta detailnya sudah ada