

TEKNOLOGI

LAPORAN PENELITIAN HIBAH KOMPETENSI



PERANCANGAN DAN PEMBUATAN  
SISTEM APLIKASI DATA WAREHOUSE REKAM  
MEDIK PENUNJANG KEPUTUSAN STRATEGIS  
RSU DR. SOETOMO

Oviliani Yenty Yuliana, S.T., MSCIS

Silvia Rostianingsih, S.Kom, M.MT

Karjono, SKM, SIP, MARS

UNIVERSITAS KRISTEN PETRA

DESEMBER 2009

## HALAMAN PENGESAHAN LAPORAN AKHIR

---

1. Judul Penelitian : Perancangan dan Pembuatan Sistem Aplikasi Data Warehouse Rekam Medik Penunjang Keputusan Strategis RSU Dr. Soetomo
2. Ketua Peneliti
- a. Nama : Oviliani Yenty Yuliana, ST., MSCIS
- b. Jenis Kelamin : Perempuan
- c. NIP : 94-014
- d. Pangkat/Golongan : Lektor Kepala/IVC
- e. Jabatan Struktural : Koordinator Bidang Studi Sistem Informasi
- f. Bidang Keahlian : Database, Data Warehouse, dan Data Mining
- g. Fakultas/Jurusan : Teknologi Industri/Teknik Informatika
- h. Perguruan Tinggi : Universitas Kristen Petra
- i. Tim Peneliti

No	Nama dan Gelar Akademik	Bidang Keahlian	Fakultas/Jurusan	Perguruan Tinggi
1.	Silvia Rostianingsih, M.MT	Database, Sistem Informasi Geografis	Teknologi Industri/Teknik Informatika	Universitas Kristen Petra
2.	Karjono, SKM, SIP, MARS	Rekam Medik dan Sistem Informasi Manajemen Rumah Sakit	POLTEKKES dan FKM	Universitas Airlangga

3. Pendanaan dan jangka waktu penelitian
- a. Jangka Waktu Penelitian : 3 tahun
- b. Biaya total yang diusulkan : Rp. 132.730.000,00
- c. Biaya yang disetujui tahun ke-1 : Rp. 42.500.000,00

Surabaya, 10 Desember 2009

Mengetahui

Dekan Fakultas Teknologi Industri,

Ketua Tim Peneliti,

(Ir. Djoni Haryadi Setiabudi, M.Eng)  
NIP. 85-009

Oviliani Yenty Yuliana, ST., MSCIS  
NIP. 94-014

Menyetujui,  
Kepala Lembaga Penelitian dan Pengabdian pada Masyarakat

(Prof. Ir. Lilianny Sigit A., M.Sc., Ph.D)  
NIP. 84-011

## A. LAPORAN HASIL PENELITIAN

### RINGKASAN DAN SUMMARY

Penelitian Hibah Bersaing ini diajukan kepada DIKTI guna perancangan dan pembuatan Sistem Aplikasi Data Warehouse Rekam Medik Penunjang Keputusan Strategis RSUD Dr. Soetomo. Penelitian ini diajukan untuk memenuhi permintaan RSUD Dr. Soetomo yang membutuhkan suatu informasi berbasis pengetahuan diagnosa penyakit dalam bentuk tabel multi dimensi dan grafik Sistem Informasi Geografis (SIG). Basis pengetahuan tersebut digunakan oleh *knowledge workers* untuk membuat keputusan strategis dalam melakukan tindakan preventif atau kuratif. Secara tidak langsung dapat menunjang tujuan Pemerintah Republik Indonesia dalam mewujudkan Indonesia Sehat 2010. Penelitian ini merupakan kerja sama antara Dosen Peneliti dari Jurusan Teknik Informatika Universitas Kristen Petra dengan Rekam Medik RSUD Dr. Soetomo Surabaya (sebagai rumah sakit pemerintah yang terbesar di Kawasan Indonesia Bagian Timur).

Untuk dapat menyajikan informasi berbasis pengetahuan, perlu dirancang *Star Schema*. Berdasarkan *Star Schema* tersebut dibentuk database baru (*Data Warehouse*), kemudian dibuat program transformasi untuk mengisi *data warehouse* berdasarkan data dalam Database SIRS. Setelah itu dirancang *On-Line Analytical Processing (OLAP)* tool sebagai sarana bagi *knowledge workers* guna menganalisa data-data historikal *track record* penyakit yang tersimpan dan juga data-data lainnya. OLAP tool ini juga direncanakan untuk diberi kapabilitas/kemampuan lebih dalam menganalisa *track record* penyakit dengan menambahkan metode-metode data mining yang ada. Tujuannya untuk lebih meningkatkan mutu informasi berbasis pengetahuan yang dapat disajikan pada *knowledge workers*. Selanjutnya dibuat grafik SIG untuk memetakan data-data penyakit, epidemi dan pandemi yang telah terjadi pada sebuah peta geografis.

Penelitian telah berlangsung selama 1 tahun dari tiga tahun yang direncanakan. Sampai saat ini telah berhasil dibuat dua buah prototipe *data warehouse* dan *OLAP tool* untuk data historikal *track record* penyakit dan data

historikal manajemen RSUD Dr. Soetomo Surabaya. Selain itu telah berhasil dirancang dan dikembangkan sistem aplikasi SIG penyakit dan juga sistem aplikasi *sequential mining* untuk menganalisa *track record* penyakit. Tujuan dari sistem aplikasi *sequential mining* untuk mendapatkan pola sekuensial tentang terjangkit penyakit pada suatu daerah yang disebabkan oleh penyakit lain yang telah diderita sebelumnya. Publikasi dari hasil penelitian telah dilakukan dua kali yaitu:

1. Budhi Gregorius Satia, Yuliana Oviliani Yenty, Rostianingsih Silvia, Koncoro Sunny, “Data Warehousing Disease Track Record Historical RSUD Dr. Soetomo For Supporting Decision Making”, Proceeding of International Industrial Informatics Seminar 2009, UIN Sunan Kalijaga, Hotel University Yogyakarta, Indonesia, August 15, 2009.
2. Yuliana Oviliani Yenty, Rostianingsih Silvia, dan Budhi Gregorius Satia, “Discovering Sequential Disease Patterns in Medical Databases Using FreeSpan Mining Approach”, International Conference on Advanced Computer Science and Information System, Universitas Indonesia yang disponsori oleh IEEE sub capter Indonesia, Depok, Indonesia, December 7-8, 2009.

Penelitian yang dilakukan melibatkan enam mahasiswa S1 untuk mengembangkan program aplikasi hasil penelitian. Keterlibatan mahasiswa tersebut dalam proyek tugas akhir maupun kerja praktek.

## **PRAKATA**

Penelitian Hibah Bersaing dengan judul "Perancangan dan Pembuatan Sistem Aplikasi Data Warehouse Rekam Medik Penunjang Keputusan Strategis RSUD Dr. Soetomo", merupakan suatu proyek penelitian dimulai pada tahun 2009 dengan perencanaan alokasi waktu selama tiga tahun, sehingga akan berakhir pada akhir 2011. Hasil penelitian diharapkan dapat memberikan berguna yang sangat penting terhadap pemanfaatan teknologi informasi dalam menunjang sektor kesehatan. Oleh sebab itu pada kesempatan ini kami mengucapkan terima kasih atas dukungan dari beberapa pihak sebagai berikut:

- DIKTI selaku penyandang dana dalam proyek penelitian ini.
- Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM), Universitas Kristen Petra yang telah membantu memfasilitasi dan mengadministrasi mulai dari pengajuan proposal sampai dengan penyelesaian proyek penelitian.
- Jurusan Teknik Informatika, Universitas Kristen Petra sebagai tempat (sarana dan prasarana) pembuatan/pelaksanaan penelitian.
- RSUD Dr. Soetomo sebagai tempat pengimplementasian dari hasil penelitian yang dibuat.

Akhir kata, kami berharap hasil penelitian ini dapat diselesaikan sesuai dengan hasil dan target waktu yang direncanakan.

Hormat kami,  
Ketua Tim Peneliti

Oviliani Yenty Yuliana, ST., MSCIS.

## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
<b>A. LAPORAN HASIL PENELITIAN</b>	
RINGKASAN DAN SUMMARY.....	iii
PRAKATA .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	vii
DAFTAR TABEL.....	xi
BAB 1. PENDAHULUAN.....	1
BAB 2. TINJAUAN PUSTAKA .....	3
2.1. Sistem Informasi Rumah Sakit (SIRS).....	3
2.2. <i>Data Warehouse</i> .....	4
2.3. Model Data Multi Dimensi .....	5
2.4. Star Schema .....	7
2.5. Bagan Hirarki.....	8
2.6. Operasi-Operasi OLAP Dalam Model Data Multi Dimensi.....	9
2.7. Sistem Informasi Geografik.....	11
2.8. Data Mining .....	13
2.9. K-Nearest Neighbors .....	14
2.10. <i>Sequential Pattern Mining</i> .....	20
2.11. Algoritma <i>FreeSpan</i> .....	20
BAB 3. TUJUAN DAN MANFAAT PENELITIAN .....	24
BAB 4. METODE PENELITIAN .....	25
BAB 5. HASIL DAN PEMBAHASAN.....	28
5.1. Prototipe data warehouse dan OLAP Tool Track Record Penyakit di RSU Dr. Soetomo Surabaya. ....	28
5.2. Prototipe Data Warehouse Dan OLAP Tool Manajemen Data RSU Dr. Soetomo Surabaya. ....	53
5.3. Sub Sistem Pemetaan SIG Penyebaran Wabah Penyakit .....	77
5.4. Sub Sistem <i>Sequential Mining</i> Untuk Mengenali Pola Sebab-Akibat Terjangkitnya Penyakit Pada Pasien.....	86
BAB 6. KESIMPULAN DAN SARAN.....	105
DAFTAR PUSTAKA .....	106
LAMPIRAN .....	108
<b>B. DRAF ARTIKEL ILMIAH</b>	
<b>C. SINOPSIS PENELITIAN LANJUTAN</b>	

## DAFTAR GAMBAR

Gambar 2.1. Hubungan Antara SIRS Dengan Aplikasi Data Mining dan Usulan Aplikasi <i>Data warehouse</i> .....	3
Gambar 2.2. Data Cube 3-D yang Menunjukkan Rekapitulasi Diagnosa Penyakit Pasien pada Tabel 2.2 .....	6
Gambar 2.3. Data Cube 4-D yang Menunjukkan Rekapitulasi Diagnosa Penyakit Pasien.....	7
Gambar 2.4. Star Schema dari <i>Data Warehouse</i> Diagnosa penyakit .....	8
Gambar 2.5. Bagan Hirarki untuk Dimensi Lokasi .....	9
Gambar 2.6. Contoh Operasi OLAP Khusus pada Data Multidimensial .....	10
Gambar 2.7. Representasi SIG terhadap dunia nyata .....	12
Gambar 2.8. Peta penyebaran jumlah vaksin flu.....	12
Gambar 2.9. Tahap-tahapan dalam data mining (telah diolah kembali).....	13
Gambar 2.10. Contoh pasien berdasarkan umur dan rasio Na/K .....	15
Gambar 2.11. Pembesaran gambar New Patient 2 dari Gambar 2.10 .....	16
Gambar 2.12. Pembesaran gambar New Patient 3 dari Gambar 2.10 .....	17
Gambar 2.13. Construct a frequent item matrix.....	21
Gambar 2.14. Generate length-2 sequential patterns.....	22
Gambar 2.15. Generate item-repeating patterns.....	22
Gambar 2.16. Generate projected DB.....	23
Gambar 2.17. Four projected DB and their sequential patterns .....	23
Gambar 4.1. Kerangka Konseptual Penelitian .....	26
Gambar 5.1. Desain Keseluruhan Star Schema (Conceptual Design).....	29
Gambar 5.2. Desain Keseluruhan Star Schema (Physical Design) .....	29
Gambar 5.3. Context Diagram OLAP Track Record Penyakit .....	35
Gambar 5.4. DFD Level 0 OLAP Track Record Penyakit .....	36
Gambar 5.5. DFD Level 1 OLAP Track Record Penyakit .....	37
Gambar 5.6. Struktur Menu OLAP Track Record Penyakit .....	37
Gambar 5.7. Form Main Menu.....	38
Gambar 5.8. Form Login.....	38

Gambar 5.9. Menu Tranformasi .....	40
Gambar 5.10. Tampilan Awal Menu Analisis.....	41
Gambar 5.11. Form Filter Menu Analisis Perawatan .....	42
Gambar 5.12. Tampilan Filter .....	43
Gambar 5.13. Form Tabel Menu Analisis Perawatan.....	43
Gambar 5.14. Tampilan untuk Menyimpan Pivot Table .....	44
Gambar 5.15. Tampilan untuk Menampilkan Pivot Table yang telah disimpan..	44
Gambar 5.16. Form Grafik Batang.....	45
Gambar 5.17. Form Grafik Garis .....	46
Gambar 5.18. Hasil pivot table pada Sistem Aplikasi.....	46
Gambar 5.19. Tampilan jika field kolom yang Dipilih Lebih dari Satu.....	47
Gambar 5.20. Hasil Analisis jika field kolom yang Dipilih Lebih dari Satu.....	47
Gambar 5.21. Hasil Analisis jika di Roll Up .....	48
Gambar 5.22. Hasil Zoom Up .....	48
Gambar 5.23. Tampilan Grafik Baris jika field kolom lebih dari Satu .....	49
Gambar 5.24. Tampilan Grafik Garis jika field kolom lebih dari Satu .....	49
Gambar 5.25. Hasil pivot table pada Sistem Aplikasi .....	50
Gambar 5.26. Hasil pivot table pada Sistem Aplikasi setelah di Roll Up .....	50
Gambar 5.27. Hasil Analisis jika field kolom dan baris yang Dipilih Lebih dari Satu	51
Gambar 5.28. Hasil Analisis jika field Baris dan Kolom yang Dipilih Lebih dari Satu dan di Roll Up.....	51
Gambar 5.29. Hasil pivot table pada Sistem Aplikasi setelah di Roll Up .....	52
Gambar 5.30. Form View Table Fakta Perawatan .....	53
Gambar 5.31. Desain Keseluruhan Star Schema (Conceptual Design).....	54
Gambar 5.32. Desain Keseluruhan Star Schema (Physical Design) .....	54
Gambar 5.33. Context Diagram OLAP Manajemen Data .....	60
Gambar 5.34. DFD Level 0 OLAP Manajemen Data .....	61
Gambar 5.35. DFD Level 1 OLAP Manajemen Data .....	61
Gambar 5.36. Struktur Menu OLAP Manajemen Data .....	62
Gambar 5.37. Tampilan Menu MainPivot .....	63
Gambar 5.38. Tampilan saat Memilih Value Pivot Table .....	64

Gambar 5.39. Tampilan saat Memilih Field Value .....	64
Gambar 5.40. Tampilan saat Setting Value Field.....	65
Gambar 5.41. Tampilan saat Setting Value Field.....	65
Gambar 5.42. Tampilan Saat Memilih Tipe Data .....	66
Gambar 5.43. Tampilan Hasil Pivot Table dengan Tipe Data ‘Count’ .....	67
Gambar 5.44. Tampilan Hasil Pivot Table dengan Tipe Data ‘Sum’ .....	67
Gambar 5.45. Tampilan Hasil Pivot Table dengan Tipe Data ‘Average’ .....	68
Gambar 5.46. Tampilan Hasil Pivot Table dengan Tipe Data ‘Minimum’ .....	68
Gambar 5.47. Tampilan Hasil Pivot Table dengan Tipe Data ‘Maximum’ .....	69
Gambar 5.48. Tampilan Setting Date Filter .....	69
Gambar 5.49. Tampilan Hasil Pivot Tabel dengan Filter Tanggal.....	70
Gambar 5.50. Tampilan Zoom Out pada Pivot Tabel .....	71
Gambar 5.51. Tampilan Pivot Tabel dengan 2 Data pada Baris .....	72
Gambar 5.52. Tampilan Pivot Tabel dengan 2 Data pada Kolom.....	72
Gambar 5.53. Tampilan Pivot Tabel dengan Data hanya pada Baris.....	73
Gambar 5.54. Tampilan Pivot Tabel dengan Data hanya pada Kolom .....	73
Gambar 5.55. Tampilan Pivot Tabel dengan 2 Data pada Baris dan Kolom.....	74
Gambar 5.56. Tampilan Hasil Analisa Grafik dalam Bentuk Barchart.....	74
Gambar 5.57. Tampilan Hasil Analisis Grafik dalam Bentuk LineChart.....	75
Gambar 5.58. Tampilan Hasil Scroll Grafik .....	76
Gambar 5.59. Conceptual Model.....	77
Gambar 5.60. Flowchart Utama .....	78
Gambar 5.61. Flowchart Prosedur Min-Max Normalization .....	79
Gambar 5.62. Flowchart Prosedur Euclidean Distance .....	80
Gambar 5.63. Flowchart Prosedur Weighted Voting .....	81
Gambar 5.64. Pemilihan Data Penyebaran Wabah Penyakit .....	82
Gambar 5.65. Pemilihan Data yang Ditampilkan pada Penyebaran Wabah Penyakit	82
Gambar 5.66. Hasil Normalisasi pada Penyebaran Wabah Penyakit.....	83
Gambar 5.67. Hasil <i>Euclidean Distance</i> pada Penyebaran Wabah Penyakit .....	84
Gambar 5.68. Hasil <i>Weighted Voting</i> pada Penyebaran Wabah Penyakit.....	84
Gambar 5.69. Hirarki Hasil Voting pada Penyebaran Wabah Penyakit.....	85

Gambar 5.70. Hirarki Hasil Voting pada Penyebaran Wabah Penyakit.....	85
Gambar 5.71. Conceptual Model Pola Penyakit .....	86
Gambar 5.72. Flowchart prosedur utama.....	87
Gambar 5.73. Flowchart pemrosesan data awal.....	88
Gambar 5.74. Flowchart pemberian nomor <i>sequence</i> .....	89
Gambar 5.75. Flowchart Frequent Item Matrix .....	90
Gambar 5.76. Flowchart Generate Length-2 <i>Sequential Patterns</i> .....	91
Gambar 5.77. Repeating Items .....	92
Gambar 5.78. Generate annotations on projected DB .....	93
Gambar 5.79. Generate sequential patterns.....	94
Gambar 5.80. Struktur Menu Penentuan Pola Penyakit .....	95
Gambar 5.81. Inputan awal user.....	95
Gambar 5.82. Data awal dan data yang diolah.....	96
Gambar 5.83. Item pattern dan frequent item matrix .....	96
Gambar 5.84. Length-2 sequential patterns .....	96
Gambar 5.85. Repeating items .....	97
Gambar 5.86. Annotation projected DB .....	97
Gambar 5.87. Projected DB .....	97
Gambar 5.88. Sequential Patterns dalam bentuk angka.....	98
Gambar 5.89. Sequential Patterns dalam bentuk Kode Penyakit.....	98
Gambar 5.90. Rule.....	99
Gambar 5.91. Graph Sequential Patterns .....	100
Gambar 5.92. Input data awal .....	100
Gambar 5.93. Item Patterns dan Frequent Item Matrix .....	101
Gambar 5.94. Length-2 Sequential Patterns .....	101
Gambar 5.95. <i>Repeating Items</i> .....	102
Gambar 5.96. <i>Annotation Projected DB</i> .....	102
Gambar 5.97. <i>Annotation Projected DB</i> .....	103
Gambar 5.98. <i>Sequential Patterns</i> .....	103
Gambar 5.99. <i>Sequential Patterns</i> dalam bentuk Kode Penyakit .....	104
Gambar 5.100. <i>Rule</i> .....	104
Gambar 5.101. <i>Graph Sequential Patterns</i> .....	104

## DAFTAR TABEL

Tabel 2.1.	Rekapitulasi Diagnosa Penyakit Pasien 2-D .....	5
Tabel 2.2.	Rekapitulasi Diagnosa Penyakit Pasien 3-D .....	6
Tabel 2.3.	Nilai umur dan rasio Na/K untuk Gambar 2.11 .....	19
Tabel 4.1.	Capaian Target .....	27
Tabel 5.1.	Tabel DIMENSI_PASIEN .....	30
Tabel 5.2.	Tabel DIMENSI_PENJAGA .....	31
Tabel 5.3.	Tabel DIMENSI_WAKTU .....	31
Tabel 5.4.	Tabel DIMENSI_TEMPAT .....	32
Tabel 5.5.	Tabel DIMENSI_DOKTER .....	32
Tabel 5.6.	Tabel DIMENSI_DIAGNOSA .....	33
Tabel 5.7.	Tabel BENTUK_DIAGNOSA .....	33
Tabel 5.8.	Tabel DIMENSI_JENAZAH .....	33
Tabel 5.9.	Tabel FAKTA_PERAWATAN .....	34
Tabel 5.10.	Tabel FAKTA_KEMATIAN .....	35
Tabel 5.11.	Hasil pivot table pada Microsoft Excel 2003 .....	46
Tabel 5.12.	Hasil pivot table pada Microsoft Excel 2003 .....	50
Tabel 5.13.	Hasil pivot table pada Microsoft Excel 2003 .....	52

## BAB 1. PENDAHULUAN

Selama lebih dari tiga dasawarsa, Indonesia telah melaksanakan berbagai upaya dalam rangka meningkatkan kesehatan dan kesejahteraan masyarakat. Departemen Kesehatan telah menyelenggarakan serangkaian reformasi di bidang kesehatan guna meningkatkan pelayanan kesehatan dan menjadikannya lebih efisien, efektif serta terjangkau oleh masyarakat. Berbagai model pembiayaan kesehatan, sejumlah program intervensi teknis bidang kesehatan, serta perbaikan organisasi dan manajemen telah diperkenalkan.

Namun sebagian besar masyarakat Indonesia masih sulit mendapatkan pelayanan kesehatan. Untuk mengatasi masalah-masalah tersebut, ditetapkan Keputusan Menteri Kesehatan Republik Indonesia Nomor 574/Menkes/SK/IV/2000. Dalam keputusan tersebut dinyatakan Visi, Misi, dan Strategi Pembangunan Kesehatan. Untuk menunjang keberhasilan visi tersebut ditetapkan Keputusan Menteri Kesehatan Nomor 1410/Menkes/SK/X/2003 tentang Sistem Informasi Rumah Sakit (SIRS).

SIRS merupakan pedoman untuk pengisian formulir, laporan yang harus dihasilkan, dan prosedur pelaporan. Data-data *track record* pasien yang disimpan oleh rumah sakit berdasarkan SIRS terdiri lebih dari 20 item, diantaranya adalah Tanggal Masuk Rumah Sakit, Tanggal Operasi, Tanggal Keluar Rumah Sakit, Jenis Pelayanan Rawat Inap, Kelas, Prosedur Masuk Rumah Sakit, Cara Masuk Rumah Sakit, Tempat Tinggal, Tanggal Lahir, Pendidikan yang Diselesaikan, Pekerjaan/Jabatan, Status Kawin dan lain-lain. Luaran yang dihasilkan oleh sistem masih sebatas pada kebutuhan administrasi dan pelaporan rutin, seperti: kegiatan rumah sakit, keadaan morbiditas pasien rawat inap dan rawat jalan, keadaan penyakit khusus pasien rawat inap dan rawat jalan rumah sakit, status imunisasi, dasar rumah sakit, dan sebagainya. Berdasarkan survei awal, RSUD Dr. Soetomo sudah memiliki dan mengoperasikan SIRS yang tidak dikembangkan oleh *programmer* internal. Semua format formulir dan laporan yang dihasilkan sudah sesuai dengan standar yang ditetapkan.

RSUD Dr. Soetomo Surabaya adalah Rumah Sakit (RS) Pemerintah kelas A, RS rujukan dari RS daerah, dan merupakan RS pendidikan. RSUD Dr. Soetomo

mengemban Visi dan Misi Indonesia Sehat 2010 pada umumnya dan Indonesia Bagian Timur pada khususnya. Dalam upaya menyediakan data dan informasi rutin untuk membuat keputusan terstruktur RSUD Dr. Soetomo telah mengkomputerisasi SIRS sejak tahun 2002.

Dengan semakin kompleksnya permasalahan yang dihadapi oleh pihak manajerial RSUD Dr. Soetomo, maka pihak manajerial membutuhkan informasi dari *data warehouse* (surat permohonan terlampir). Pihak manajerial RSUD Dr. Soetomo membutuhkan informasi yang ditampilkan secara visual berupa tabel multi dimensi dan atau grafik SIG untuk membantu pengambilan keputusan strategis. Sistem aplikasi yang dimiliki oleh RSUD Dr. Soetomo selama ini masih belum dapat menyajikan informasi tersebut.

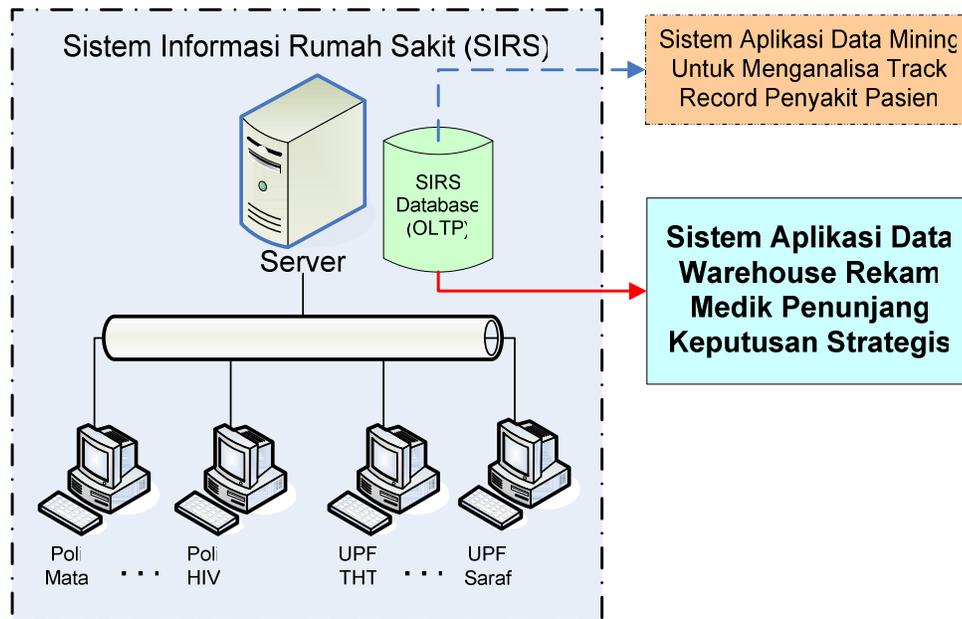
Dengan menggunakan *Data Warehouse dan OLAP Tool-nya*, diharapkan data-data dalam SIRS, yang telah ditransformasikan ke dalam *data warehouse*, dapat lebih mudah untuk digali dan dianalisa kembali. Dengan bantuan sebuah sistem OLAP yang dapat melakukan proses *slicing, dicing, pivoting, roll-up* dan *drill-down* terhadap data yang ada, serta menampilkan hasilnya dalam bentuk tabel maupun grafik, para *knowledge workers* akan lebih cepat dan mudah dalam menganalisa data dan mengambil keputusan strategis yang diperlukan.

Aplikasi Sistem Informasi Geografis (SIG) penyakit yang telah dibangun dapat memudahkan para *knowledge workers* untuk mendeteksi kemungkinan terjadinya wabah, epidemi dan atau pandemi yang dapat terjadi di lokasi, kota dan propinsi lain, bila wabah lokal telah terjadi pada sebuah lokasi tertentu. SIG yang dibangun ini dapat menunjukkan pola penyebaran wabah penyakit dengan cara menggali data-data historikal yang ada menggunakan metode data mining *k-nearest neighbour* dan *statistik*. Aplikasi Sequential pattern dapat digunakan oleh para *knowledge workers* untuk mengetahui pola urutan terjangkitnya suatu penyakit. Dengan demikian dapat membantu para medis dalam melakukan tindakan preventif maupun kuratif.

## BAB 2. TINJAUAN PUSTAKA

### 2.1. Sistem Informasi Rumah Sakit (SIRS)

RSU Dr. Soetomo Surabaya sebagai institusi yang terlibat dan menjadi objek dalam penelitian ini sudah mengoperasikan SIRS sesuai dengan Keputusan Menteri Kesehatan Nomor 1410/Menkes/SK/ X/2003 sejak tahun 2002. SIRS digunakan agar RSU Dr. Soetomo dapat memberikan data dan informasi secara rutin kepada Departemen Kesehatan dan Departemen lain. SIRS RSU Dr. Soetomo dirancang berupa aplikasi *client server* menggunakan Oracle Database 9i dan Oracle Developer. Salah satu fungsi SIRS adalah untuk mencatat rekam medik secara elektronik Pasien Rawat Inap dan Pasien Rawat Jalan. Sebagai gambaran dari SIRS yang dioperasikan oleh RSU Dr. Soetomo dapat dilihat pada Gambar 2.1.



Gambar 2.1. Hubungan Antara SIRS Dengan Aplikasi Data Mining dan Usulan Aplikasi *Data warehouse*

Peneliti terdahulu (Intan dkk, 2008) pada tahun 2007 memenangkan Penelitian Hibah Bersaing untuk Merancang dan Mengembangkan Sistem

Aplikasi Data *Mining* Untuk Menganalisa *Track Record* Penyakit Pasien di RSUD Dr. Soetomo. Saat ini penelitian tersebut sudah masuk ke tahun ketiga, yakni: pembuatan dan pengujian sistem aplikasi. Pihak RSUD Dr. Soetomo merasa puas dan terbantu dengan adanya sistem aplikasi tersebut. Untuk itu, tahun ini RSUD Dr. Soetomo memohon kepada Jurusan Teknik Informatika Universitas Kristen Petra untuk merancang dan mengembangkan sistem aplikasi *Data Warehouse* rekam medik penunjang keputusan strategis. Keterkaitan antara SIRS dengan aplikasi *Data Mining* yang sedang dikembangkan dan *Data Warehouse* yang diusulkan dapat dilihat pada Gambar 2.1.

## **2.2. *Data Warehouse***

Han dan Micheline (2001) menyatakan bahwa *Data Warehouse* adalah data yang berorientasi persoalan (*subject-oriented*), terintegrasi (*integrated*), perbedaan waktu (*time-variant*), dan koleksi data yang tidak berubah (*nonvolatile*) dalam mendukung proses pembuatan keputusan strategis manajemen. Hal tersebut yang membedakan *Data Warehouse* yang diusulkan dalam penelitian ini dengan sistem *database relational* SIRS di RSUD Dr. Soetomo.

Tugas utama dari sistem *database relational* adalah untuk pemrosesan transaksi dan mencari data secara on line. Sistem ini disebut dengan *On-Line Transaction Processing (OLTP)*. Sistem tersebut mencakup hampir keseluruhan operasi sehari-hari di rumah sakit, seperti: registrasi pasien, rekam medik, pembayaran biaya perawatan, dan lain sebagainya. Dalam kasus penelitian yang diusulkan RSUD Dr. Soetomo, SIRS merupakan sistem OLTP.

Sedangkan *Data Warehouse* digunakan untuk melayani *knowledge workers* dalam menganalisa data dengan tujuan untuk pembuatan keputusan strategis. Sistem ini dikenal dengan *On-Line Analytical Processing (OLAP)*. Dalam penelitian ini, sistem OLAP yang akan dirancang dan dibuat. Sistem OLTP dapat menyajikan informasi dari data masa lampau dalam berbagai format untuk memenuhi kebutuhan pemakai yang berbeda.

### 2.3. Model Data Multi Dimensi

*Data Warehouse* dan OLAP *tool* berdasarkan pada Model Data Multi Dimensi. Model ini memandang data dalam bentuk *Data Cube* yang mengijinkan data dimodelkan dan dipandang dalam berbagai dimensi. *Data Cube* terbentuk dari beberapa Dimensi dan *Fact*. Dimensi adalah entiti-entiti atau perspektif-perspektif yang berkenaan dengan keinginan suatu organisasi untuk menganalisa data. Sebagai contoh, *Data Warehouse* diagnosa penyakit pasien memiliki dimensi penyakit, waktu, dan lokasi. Dimensi tersebut menyimpan diagnosa pasien per bulan, nama penyakit, dan tempat tinggal penderita penyakit tersebut.

Untuk memperoleh pemahaman *Data Cube* dan Model Data Multi Dimensi dimulai dengan *Data Cube* dua dimensi (2-D). Sebagai contoh rekapitulasi diagnosa penyakit per tiga bulanan di lokasi Kota Sidoarjo ditampilkan pada Tabel 2.1. *Data Cube* 2-D diagnosa penyakit untuk lokasi Kota Sidoarjo ditampilkan dalam dimensi waktu (terorganisasi dalam tiga bulanan) dan dimensi penyakit (terorganisasi berdasarkan nama penyakit yang menjadi fokus). Tampilan *Fact* berupa jumlah penderita penyakit (dalam ribuan).

Tabel 2.1. Rekapitulasi Diagnosa Penyakit Pasien 2-D

Lokasi="Sidoarjo"				
Waktu	Nama Penyakit			
	Typhus	Malaria	TBC	DB
Triwulan I	605	825	14	400
Triwulan II	680	952	31	512
Triwulan III	812	1023	30	501
Triwulan IV	927	1038	38	580

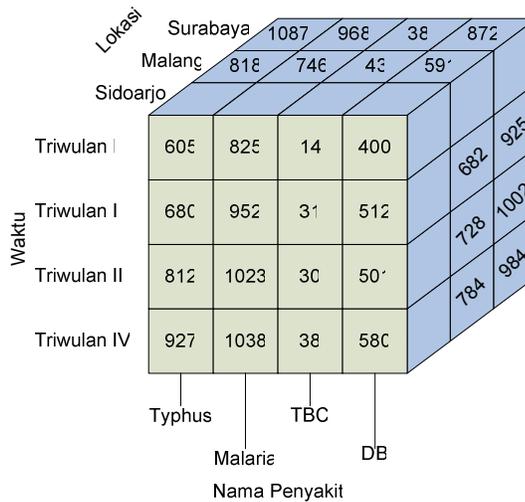
Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, *Data Mining: Concepts and Techniques*, 2001, p.45

Rekapitulasi diagnosa penyakit dalam 3-D berdasarkan waktu (per triwulanan), penyakit (Typhus, Malaria, TBC, DB), dan lokasi (Surabaya, Malang dan Sidoarjo) ditampilkan pada Tabel 2.2. Rekapitulasi diagnosa penyakit 3-D pada Tabel 2.2 ditampilkan sebagai sekumpulan tabel-tabel 2-D. Secara konsep, penggambaran rekapitulasi diagnosa penyakit pasien yang sama dalam bentuk data *Cube* 3-D tampak pada Gambar 2.2.

Tabel 2.2. Rekapitulasi Diagnosa Penyakit Pasien 3-D

Waktu	Lokasi="Surabaya"				Lokasi="Malang"				Lokasi="Sidoarjo"			
	Nama Penyakit				Nama Penyakit				Nama Penyakit			
	Typhus	Malaria	TBC	DB	Typhus	Malaria	TBC	DB	Typhus	Malaria	TBC	DB
Triwulan I	1087	968	38	872	818	746	43	591	605	825	14	400
Triwulan II	1130	1024	41	925	894	769	52	682	680	952	31	512
Triwulan III	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Triwulan IV	1142	1091	54	984	978	864	59	784	927	1038	38	580

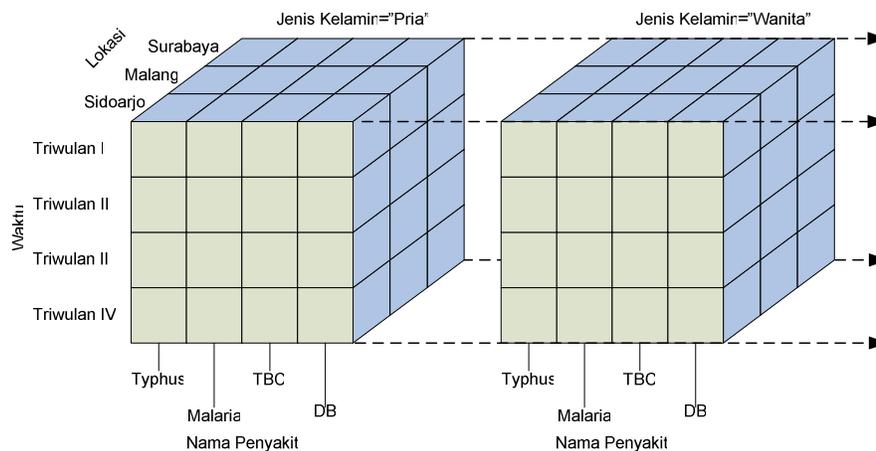
Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, Data Mining: Concepts and Techniques, 2001, p.46



Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, Data Mining: Concepts and Techniques, 2001, p.46

Gambar 2.2. Data Cube 3-D yang Menunjukkan Rekapitulasi Diagnosa Penyakit Pasien pada Tabel 2.2

Andaikan rekapitulasi diagnosa penyakit pasien ingin ditampilkan dalam bentuk 4-D, dengan tambahan satu dimensi misalnya jenis kelamin. Bentuk 4-D sesungguhnya bukan berupa Cube 4-D. Cube 4-D dapat dianggap sebagai sekumpulan Cube 3-D, seperti yang ditampilkan pada Gambar 2.3. Dengan cara yang sama, dimungkinkan untuk menampilkan data dalam n-D, sebagai sekumpulan dari Cube (n-1)-D.



Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, *Data Mining: Concepts and Techniques*, 2001, p.47

Gambar 2.3. Data Cube 4-D yang Menunjukkan Rekapitulasi Diagnosa Penyakit Pasien

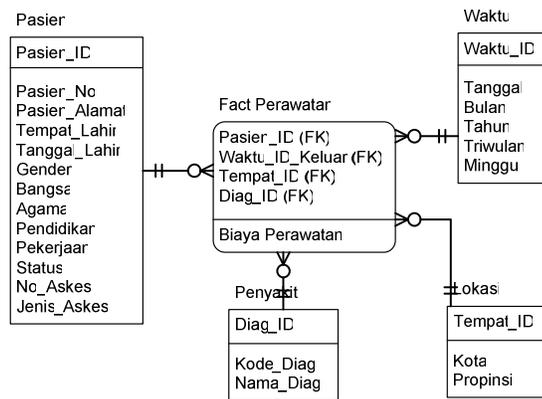
Secara umum proses dari data *mining* dari suatu *database* yang sudah dinormalisasikan adalah dengan mengkombinasikan semua *table* yang diperlukan, sehingga didapat suatu *table* denormalisasi. Proses untuk menggabungkan atau mengkombinasikan *table-table* yang telah dinormalisasikan menjadi suatu *table* baru inilah yang disebut dengan proses denormalisasi data (*Join Table*). (Intan, 2007)

Dalam proses *denormalisasi* data, sangat tidak dianjurkan apabila saat menggabungkan tabel-tabel tersebut dilakukan pada semua *field* yang ada. Namun sebaiknya saat dilakukan proses penggabungan dilakukan pada *field-field* yang dibutuhkan saja pada proses *data mining*. Proses *denormalisasi* data dapat dilakukan berdasarkan dua jenis relasi data, yaitu:

#### 2.4. Star Schema

*Data Warehouse* menghasilkan suatu laporan berupa ringkasan dan menggunakan skema berorientasi persoalan yang memfasilitasi analisis data on-line. Data model yang paling populer untuk *Data Warehouse* adalah model data multi dimensi. Model tersebut dapat berupa *Star Schema*, *Snowflake Schema*, atau *Fact Constellation Schema*. Dalam penelitian ini adalah model data *Star Schema*.

*Star Schema* pada *data warehouse* berisi: (1) suatu Tabel *Fact* dengan data yang jumlahnya sangat banyak dan tanpa duplikasi dan (2) sejumlah Tabel Dimensi. *Star Schema* menggambarkan tabel-tabel dimensi yang mengelilingi Tabel *Fact*. Contoh dari *Star Schema* diagnosa penyakit pasien ditampilkan pada Gambar 2.4. Diagnosa penyakit pasien dalam 4-D, yaitu waktu, penyakit, lokasi, dan jenis kelamin. *Schema* tersebut berisi suatu Tabel *Fact* diagnosa penyakit pasien yang berisi *key* untuk keempat dimensi tersebut dengan 1 pengukuran biaya. Dalam penelitian ini akan dirancang *Star Schema* berdasarkan hasil analisis kebutuhan. Dengan segala kemungkinan dimensi seperti yang sudah disebutkan dalam Sub Bab 1.2.

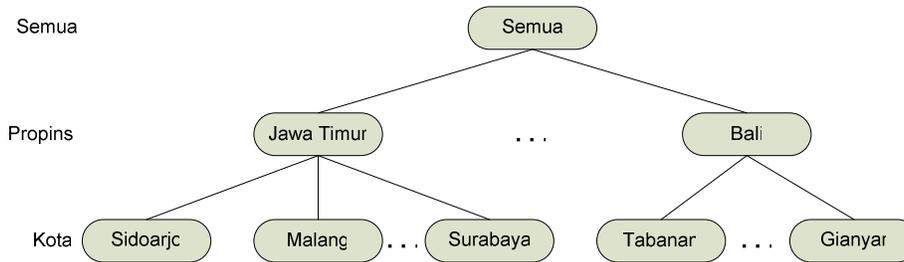


Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, *Data Mining: Concepts and Techniques*, 2001, p.49

Gambar 2.4. *Star Schema* dari *Data Warehouse* Diagnosa penyakit

## 2.5. Bagan Hirarki

Suatu bagan hirarki menetapkan urutan pemetaan dari bagan tingkat rendah sampai ke tingkat yang lebih tinggi, bagan yang lebih umum. Bagan hirarki terendah untuk dimensi lokasi adalah kota. Setiap kota dipetakan ke bagan hirarki yang lebih tinggi, yakni: propinsi. Sebagai contoh, Kota Sidoarjo, Malang, Surabaya dan lain sebagainya dipetakan ke bagan hirarki yang lebih tinggi, yaitu: Propinsi Jawa Timur. Contoh lain, kota Tabanan, Gianyar dan lain sebagainya dipetakan ke hirarki yang lebih tinggi, yaitu: Propinsi Bali. Bagan hirarki untuk contoh tersebut tampak pada Gambar 2.5.



Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, *Data Mining: Concepts and Techniques*, 2001, p.56

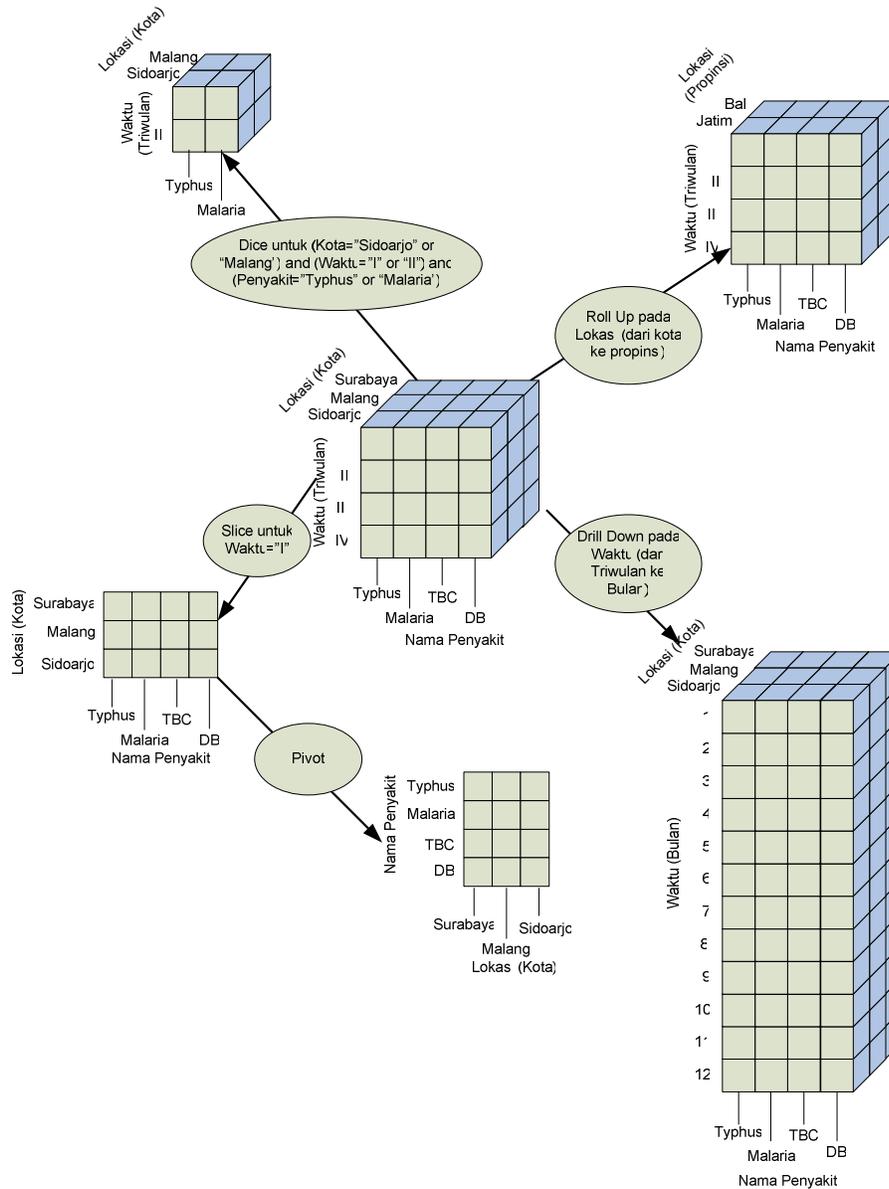
Gambar 2.5. Bagan Hirarki untuk Dimensi Lokasi

## 2.6. Operasi-Operasi OLAP Dalam Model Data Multi Dimensi

Dalam model multi dimensi terorganisasi ke dalam banyak dimensi, dan setiap dimensi berisi beberapa tingkat abstraksi yang didefinisikan oleh bagan hirarki. Pengorganisasian tersebut menyediakan keflexibelan pemakai untuk melihat data dari beberapa perspektif yang berbeda. Sejumlah operasi-operasi *Data Cube* OLAP mewujudkan tampilan-tampilan yang berbeda, memungkinkan pencarian secara interaktif dan menganalisa data yang ada. Untuk itu OLAP menyediakan suatu fasilitas yang *user-friendly* untuk analisis data secara interaktif. Contoh operasi OLAP khusus untuk Data Multi dimensi ditampilkan pada Gambar 2.6. Pusat dari gambar tersebut adalah *data cube* diagnosa penyakit. *Cube* tersebut berisi dimensi lokasi, waktu, dan nama penyakit.

Operasi yang dapat dilakukan pada *Data Cube* adalah *Drill-Up/Roll-Up*, *Drill-Down*, Operasi *Slice*, dan *Pivot*. Operasi *Roll-Up* melakukan penjumlahan pada suatu data *Cube* dengan meningkatkan hirarki suatu dimensi, misalnya membuat rekapitulasi diagnosa penyakit dari dimensi lokasi kota ke propinsi. Operasi *Drill-Down* mengubah tampilan menjadi lebih detail dengan menurunkan hirarki suatu dimensi atau dengan penambahan dimensi, misalnya menampilkan rekapitulasi diagnosa penyakit dari triwulanan ke bulanan. Operasi *Slice* melakukan pemilihan pada satu Dimensi *Cube* yang diberikan, menghasilkan pada suatu sub*Cube*, misalnya menampilkan rekapitulasi penyakit pada lokasi Sidoarjo,

Malang, Surabaya untuk penyakit Typhus, Malaria, TBC, dan DB pada Triwulan I. Operasi Pivot/ rotate adalah suatu operasi visualisasi yang memutar tampilan data axes untuk menyediakan alternatif penyajian data.



Sumber: Modifikasi dari Jiawei Han & Micheline Kamber, Data Mining: Concepts and Techniques, 2001, p.59

Gambar 2.6. Contoh Operasi OLAP Khusus pada Data Multidimensial

## 2.7. Sistem Informasi Geografik

SIG (Aronoff, 1989) adalah sistem informasi berbasis komputer yang digunakan untuk memetakan dan menganalisis segala sesuatu yang ada dan terjadi di bumi. Sistem informasi ini mengintegrasikan *query database* dan analisis data dengan visualisasi yang menarik serta analisis geografik yang lebih baik. Kemampuan inilah yang membedakan SIG dari sistem yang lain dan membuatnya berguna untuk banyak kepentingan seperti pengambilan keputusan strategis yang lebih efektif tindakan preventif atau kuratif terhadap suatu penyakit.

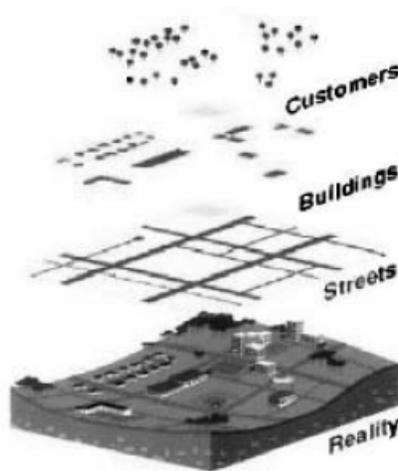
Hasil dari *data warehouse* tersebut ditampilkan dengan bantuan peta dua dimensi, agar pihak rumah sakit lebih mudah melakukan analisis. Informasi yang disajikan meliputi:

**Memetakan letak:** data realita pada pulau Jawa dipetakan ke dalam layer kabupaten dan layer propinsi. Contoh dapat dilihat seperti pada Gambar 2.7. Setiap data pada suatu layer dapat dicari, seperti halnya melakukan *query* terhadap *database*, untuk kemudian dilihat letaknya dalam keseluruhan peta.

**Memetakan kuantitas:** informasi jumlah penderita penyakit tertentu di daerah tertentu dengan kriteria jenis kelamin dan umur. Pemetaan kuantitas adalah memetakan sesuatu data yang berhubungan dengan jumlah, seperti dimana yang paling banyak atau dimana yang paling sedikit. Dengan melihat penyebaran kuantitas tersebut, dapat dicari tempat-tempat yang sesuai dengan kriteria yang diinginkan dan dapat digunakan untuk pengambilan keputusan strategis. Contoh SIG yang ada pada Gambar 2.8 adalah memetakan jumlah vaksin flu di suatu daerah.

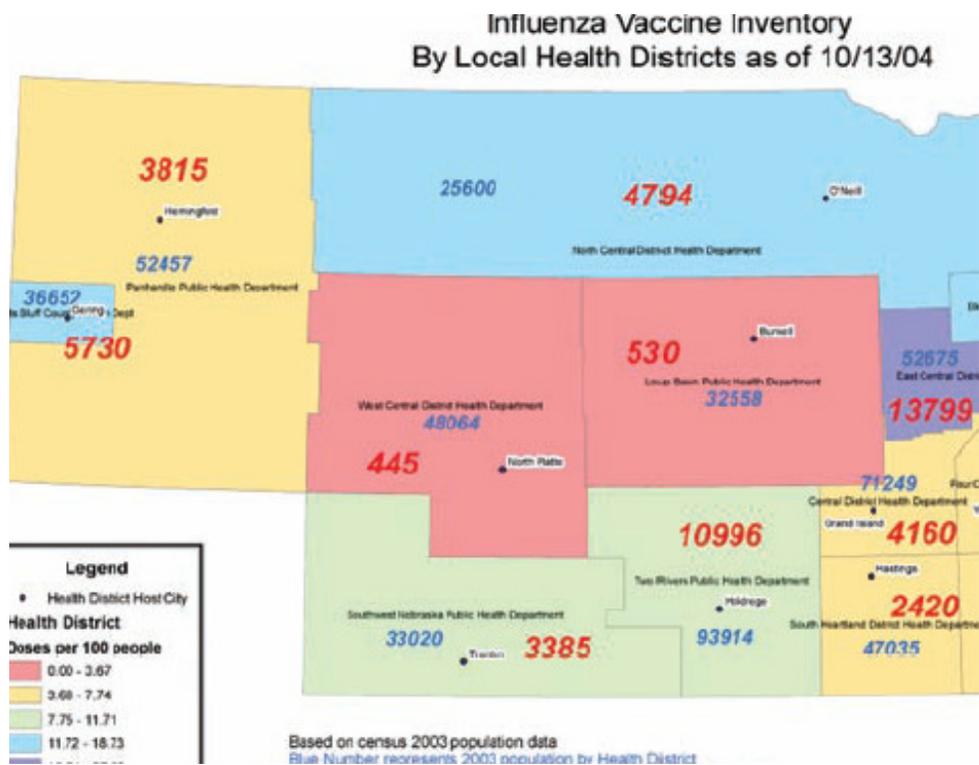
**Memetakan kerapatan (*densities*):** analisis daerah yang rawan terhadap penyakit tertentu berdasarkan penyebaran penyakit tertentu. Sewaktu orang melihat konsentrasi penyebaran lokasi dari fitur-fitur, di wilayah yang mengandung banyak fitur mungkin akan mendapat kesulitan untuk melihat wilayah mana yang mempunyai konsentrasi yang lebih tinggi dari wilayah lainnya. Peta kerapatan dapat mengubah bentuk konsentrasi ke dalam unit-unit

yang lebih mudah untuk dipahami dan memiliki keseragaman, misal membagi dalam kotak-kotak selebar 10 km<sup>2</sup>, dengan menggunakan perbedaan warna untuk menandai tiap-tiap kelas kerapatan.



Sumber: GIS Info, 2008

Gambar 2.7. Representasi SIG terhadap dunia nyata



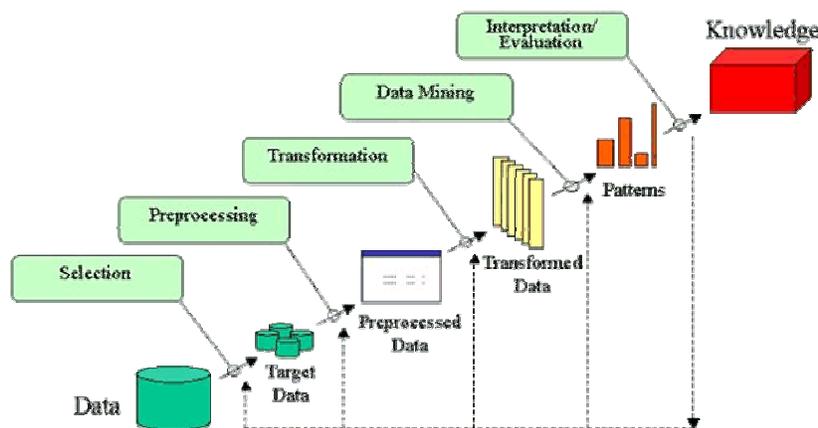
Sumber: Health, 2008

Gambar 2.8. Peta penyebaran jumlah vaksin flu

## 2.8. Data Mining

Secara sederhana, *data mining* adalah suatu proses untuk menemukan *interesting knowledge* dari sejumlah besar data yang disimpan dalam *database*, *data warehouse* atau media penyimpanan lainnya (Han, Kamber, 2001). Dengan melakukan *data mining* terhadap sekumpulan data, bisa didapatkan *interesting pattern* yang akan disimpan sebagai *knowledge* yang baru. *Pattern* yang didapat digunakan untuk melakukan evaluasi terhadap data tersebut untuk mendapatkan informasi yang berguna dalam mendukung proses pengambilan keputusan.

Diagram pada Gambar 2.9 merupakan ringkasan dari tahapan-tahapan proses yang dilakukan pada saat melakukan *data mining* dan proses untuk menemukan *knowledge*. Tahap-tahapnya dimulai dari pemrosesan *raw data* atau data mentah sampai pada penyaringan hingga ditemukannya *knowledge*, dapat dijabarkan sebagai berikut:



Sumber: Data Mining: Concepts and Techniques, Jiawei Han

Gambar 2.9. Tahap-tahapan dalam data mining (telah diolah kembali)

1. *Selection*, yaitu proses memilih dan memisahkan data berdasarkan beberapa kriteria, misalnya berdasarkan kota tempat tinggal konsumen.

2. *Pre-processing*, yaitu mempersiapkan data dengan cara membersihkan data, informasi atau *field* yang tidak dibutuhkan. Jika data tersebut dibiarkan hanya akan memperlambat proses *query*, misalnya nama pelanggan dapat dihapus jika sudah diketahui kode pelanggannya. Selain itu, pada tahap ini juga dilakukan penyeragaman format terhadap data yang tidak konsisten, misalnya pada suatu *field* dari suatu tabel, data jenis kelamin diinputkan dengan “L” atau “M”. Sedangkan pada tabel yang lain data tersebut diinputkan sebagai “P” atau “W”.
3. *Transformation*, data yang telah melalui proses *select* dan *pre-processing* tidak dapat langsung digunakan. Akan tetapi data tersebut ditransformasikan terlebih dahulu ke bentuk yang lebih *navigable* dan *useable*, misalnya dengan menambahkan *field-field* tertentu yang bersifat demografi, seperti propinsi, kota, atau informasi apapun yang biasanya digunakan pada identitas pasien.
4. *Data mining*, tahap ini dipusatkan untuk mendapatkan pola dari data (*extraction of data*).
5. *Interpretation and evaluation*, dalam proses ini *pattern* atau pola-pola yang telah diidentifikasi oleh sistem kemudian diterjemahkan/diintepretasikan ke bentuk *knowledge* yang lebih mudah dimengerti oleh *user* untuk membantu pengambilan keputusan, misalnya menunjukkan data yang saling berasosiasi melalui grafik atau bentuk lain yang lebih mudah dimengerti.

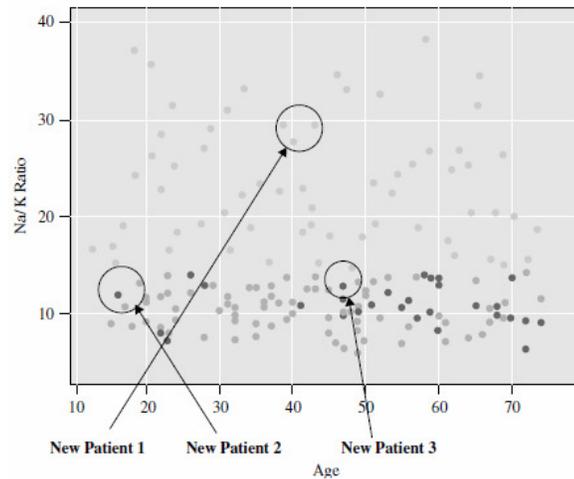
## 2.9. K-Nearest Neighbors

Metode *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut. Metode ini merupakan metode yang paling umum digunakan untuk estimasi dan prediksi. (T. Larose, 2005, p. 94)

Keunggulan dari metode KNN ini adalah relative tidak terpengaruh dari error dari data dan juga dapat digunakan dengan kumpulan data dengan jumlah besar. Namun kekurangan metode ini adalah proses pelaksanaannya lambat.

Algoritma KNN dapat dilihat pada contoh berikut. Klasifikasi tipe obat untuk resep seorang pasien berdasarkan karakteristik pasien seperti umur dan

rasio *sodium/potassium* (Na/K). Pada Gambar 2.10 merepresentasikan plot dari umur pasien dan rasio Na/K. Resep yang mengandung obat tertentu dibedakan berdasarkan warna lingkarannya. Titik abu-abu muda melambangkan obat Y. Titik abu-abu normal melambangkan obat A dan X. Titik abu-abu gelap melambangkan obat B dan C.

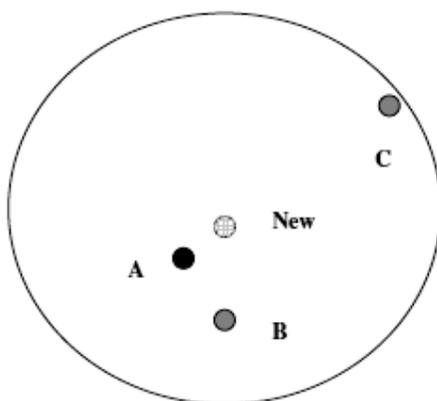


Sumber: T. Larose (2005, p. 97)

Gambar 2.10. Contoh pasien berdasarkan umur dan rasio Na/K

Misalkan saat ini kita memiliki data pasien baru, tanpa klasifikasi obat, dan ingin mengklasifikasi resep untuk pasien tersebut berdasarkan pasien lain yang memiliki atribut yang sama. Identifikasi pasien 1 berumur 40 dan rasio Na/K 29. Berdasarkan Gambar 2.10 pasien 1 berada di posisi dalam lingkaran New Patient 1. Dari lingkaran tersebut kita mengklasifikasi resep untuk pasien 1 adalah obat Y karena titik di sekitar pasien 1 semuanya diberikan resep yang sama.

Berikutnya pasien 2 berumur 17 tahun dengan rasio Na/K 12,5. Gambar 2.11 memperlihatkan pembesaran dari lingkaran New Patient 2 di Gambar 2.10 beserta dengan letak deskripsi pasien 2. Misalkan kita menentukan bahwa  $k=1$  untuk algoritma KNN, jadi pasien 2 akan diklasifikasikan berdasarkan satu titik yang berada paling dekat dengan titik pasien 2 yaitu titik A dengan resep obat B dan C.

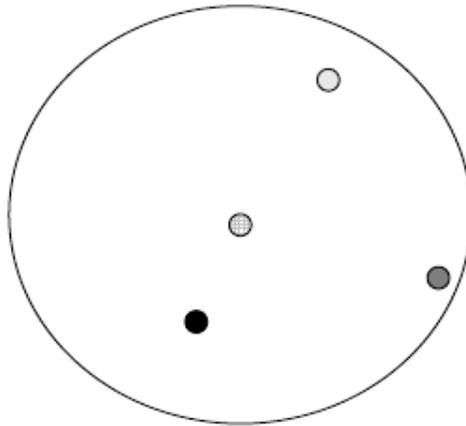


Sumber: T. Larose (2005, p. 97)

Gambar 2.11. Pembesaran gambar New Patient 2 dari Gambar 2.10

Namun, apabila menentukan  $k=2$  maka pasien 2 akan diklasifikasikan berdasarkan 2 titik yang berada paling dekat, tetapi tidak terdapat hasil yang ditemukan karena titik A dan titik B merupakan 2 titik yang berbeda. Namun, jika  $k=3$  maka pasien 2 akan diklasifikasi berdasarkan 3 titik yang paling dekat. Berdasarkan Gambar 2.11 ada 1 titik abu-abu gelap dan 2 titik abu-abu normal. Maka pasien 2 akan diklasifikasi sama dengan titik abu-abu normal karena jumlahnya lebih banyak jadi resep obat A dan X.

Terakhir, pasien 3 berumur 47 tahun dan rasio Na/K 13,5. Gambar 2.12 memperlihatkan pembesaran dari lingkaran New Patient 3 di Gambar 2.10 beserta dengan letak deskripsi pasien 3. Untuk nilai  $k=1$  maka pasien 3 akan diklasifikasi berdasarkan titik abu-abu gelap maka resep obat B dan C. Untuk  $k=2$  tidak ditemukan hasil karena 2 titik berbeda, demikian juga dengan  $k=3$  karena 3 titik tersebut berbeda satu sama lain.



Sumber: T. Larose (2005, p. 98)

Gambar 2.12. Pembesaran gambar New Patient 3 dari Gambar 2.10

Berdasarkan contoh kasus di atas KNN mengklasifikasi data baru berdasarkan jarak titik yang paling dekat dengan posisi titik yang baru. Namun bagaimana kita mengukur jarak kedekatan antara data yang baru dengan data-data yang lama tersebut? Untuk masalah itu digunakan rumus *distance* agar diketahui jarak kedekatan antar data tersebut. Rumus *distance* yang paling umum dipakai adalah *Euclidean Distance* :

$$d_{\text{Euclidean}}(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (1)$$

di mana  $x=x_1, x_2, \dots, x_m$  dan  $y=y_1, y_2, \dots, y_m$ .  $x$  dan  $y$  adalah nilai data yang ingin dihitung kedekatan jaraknya dan  $m$  adalah jumlah atribut data yang ada. Seperti contoh, misal A adalah  $x_1=20$  tahun dan rasio Na/K  $x_2=12$ , sedangkan B adalah  $y_1=30$  tahun dan rasio Na/K  $y_2=8$ . Maka *Euclidean Distance* antara 2 data tersebut adalah

$$\begin{aligned} d_{\text{Euclidean}}(x,y) &= \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{(20 - 30)^2 + (12 - 8)^2} \\ &= \sqrt{100 + 16} = 10.77 \end{aligned}$$

Namun pada saat melakukan pengukuran *distance*, kadang kala atribut tertentu dari sebuah data memiliki nilai yang sangat besar sehingga mengakibatkan nilai dari atribut lain menjadi tidak berpengaruh karena menggunakan skala kecil. Untuk mengatasi hal tersebut maka perlu melakukan *normalize* pada nilai atribut tersebut. Rumus yang sering digunakan untuk *normalize* nilai atribut antara lain

*Min-max normalization:*

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (2)$$

*X-score standardization:*

$$X^* = \frac{X - \text{mean}(X)}{SD(X)} \quad (3)$$

Untuk mengidentifikasi suatu data baru berdasarkan beberapa macam data seperti contoh di atas, maka diperlukan *combination function* untuk menentukan keputusan mengklasifikasi data baru tersebut. Menurut Larose (2005), *Combination function* tersebut dibagi menjadi 2 yaitu:

### 2.9.2. *Simple Unweighted Voting*

1. Sebelum melakukan algoritma KNN, ditentukan dahulu nilai k yang akan dipakai
2. Kemudian melakukan metode KNN, hitung jarak kedekatan data menggunakan *Euclidean distance*. Data yang terdekat yang dipilih
3. Setelah mendapatkan data yang akan digunakan sebagai acuan sejumlah nilai k, jarak kedekatan data baru terhadap data lama tidak berguna lagi. Kemudian dilakukan penghitungan voting dari data yang sudah didapatkan sebelumnya

Contoh penggunaan *Simple Unweighted Voting* ini seperti yang terlihat dari contoh kasus New Patient 2 dengan menggunakan nilai k=3.

### 2.9.3. *Weighted Voting*

Pada *weighted voting*, data yang jarak kedekatannya dengan data baru lebih jauh tidak memiliki nilai voting yang sama dengan data yang jarak kedekatannya lebih dekat. Data yang jarak kedekatannya lebih dekat memiliki nilai voting yang lebih besar.

Tabel 2.3. Nilai umur dan rasio Na/K untuk Gambar 2.11

Record	Age	Na/K	Age <sub>MMN</sub>	Na/K <sub>MMN</sub>
New	17	12.5	0.05	0.25
A (dark gray)	16.8	12.4	0.0467	0.2471
B (medium gray)	17.2	10.5	0.0533	0.1912
C (medium gray)	19.5	13.5	0.0917	0.2794

Sumber: T. Larose (2005, p. 102)

Sebagai contoh New Patient 2 dengan  $k=3$  dari Gambar 2.11. Diasumsikan nilai dari contoh di atas seperti terlihat pada Tabel 2.3 beserta dengan hasil *Min-Max normalizations*. Kemudian ditemukan nilai *distance*-nya sebagai berikut

$$d(new, A) = \sqrt{(0.05 - 0.0467)^2 + (0.25 - 0.2471)^2} = 0.004393$$

$$d(new, B) = \sqrt{(0.05 - 0.0533)^2 + (0.25 - 0.1912)^2} = 0.58893$$

$$d(new, C) = \sqrt{(0.05 - 0.0917)^2 + (0.25 - 0.2794)^2} = 0.051022$$

Data A memberi voting untuk data baru sebagai abu-abu gelap (obat B dan C) jadi perhitungan *weighted voting*-nya

$$votes(dark\ gray) = \frac{1}{d(new, A)^2} = \frac{1}{0.004393^2} \cong 51,818$$

Untuk data B dan C member voting untuk data baru sebagai abu-abu normal (obat A dan X) jadi perhitungan *weighted voting*-nya

$$votes(medium\ gray) = \frac{1}{d(new, B)^2} + \frac{1}{d(new, C)^2} = \frac{1}{0.58893^2} + \frac{1}{0.051022^2} \cong 672$$

Jadi berdasarkan nilai *weighted voting* maka data baru termasuk sebagai abu-abu gelap (obat B dan C) karena nilainya votingnya lebih besar. Hal ini berlawanan dengan hasil klasifikasi dengan menggunakan *simple unweighted voting*. Namun dengan adanya penambahan perhitungan *weighted voting* maka proses komputer akan berjalan lambat karena perlu dilakukan perhitungan ulang terhadap jarak kedekatan dan *weighted voting*-nya setiap kali ingin mengklasifikasi data yang baru.

## **2.10. Sequential Pattern Mining**

Merupakan metode untuk mencari dan menemukan hubungan antar item yang ada pada suatu *dataset*. Program *sequential pattern mining* bertujuan untuk menemukan informasi *item-item* yang saling berhubungan dalam bentuk *rule*. Dengan demikian *sequential pattern* tepat untuk diterapkan pada *mining sequential pattern*. Masalah yang ada pada *data mining* adalah menemukan pola *sequential*. Inputan datanya merupakan sekumpulan *sequence* yang disebut *data-sequences*. Setiap data *sequential* merupakan suatu daftar dari transaksi-transaksi, dimana setiap transaksi merupakan sekumpulan *item*. Suatu *sequential-pattern* juga terdiri daftar dari sekumpulan *item*.

## **2.11. Algoritma FreeSpan**

*Frequent pattern-projected sequential pattern mining* disingkat dengan FreeSpan, yang menggunakan *item frequent* untuk memproyeksikan *sequence database* dalam *database* proyeksi. Setiap *database* proyeksi akan diproyeksikan lebih lanjut secara rekursif. Ukuran *database* proyeksi biasanya lebih kecil dan lebih mudah untuk bekerja dengan *database* ini. Metode ini secara signifikan lebih cepat dari metode yang berbasis apriori. Permasalahan pada metode ini adalah sekuen yang sama dapat diduplikasi dalam banyak *database* proyeksi (Jiawei Han, 2000).

Berikut ini adalah algoritma *FreeSpan*:

1. *Scan DB, find frequent items, and sort into f\_list (frequent item list).*

2. (1) Construct a frequent item matrix by scanning DB once.
- (2) Generate length-2 sequential patterns.
- (3) Generate annotations on item-repeating patterns.
- (4) Annotations on projected DB.
- (5) Scan DB to generate item-repeating patterns and projected DB.

Proses *FreeSpan* dapat dilihat pada Gambar 2.13 sampai dengan Gambar 2.17.

(1) construct a frequent item matrix

Sequence_id	Sequence	Item pattern
10	<(bd) cd (ac)>	{a,b,c,d}
20	<(bf) (ce) b (fg)>	{b,c,e,f,g}
30	<(ah) (bf) abf>	{a,b,f,h}
40	<(be) (ce) d>	{b,c,d,e}
50	<a (bd) bcb (ade)>	{a,b,c,d,e}

```

let min. support threshold: 2
f_list(frequent item list):
<b:5, c:4, a:3, d:3, e:3, f:2>
frequent item matrix:
b | 4
c | (4,3,0) 1
a | (3,2,0) (2,1,1) 2
d | (2,2,2) (2,2,0) (1,2,1) 1
e | (3,1,1) (1,1,2) (1,0,1) (1,1,1) 1
f | (2,2,2) (1,1,0) (1,1,0) (0,c,0) (1,1,0) 2
  |-----|-----|-----|-----|-----|-----|
  | b      c      a      d      e      f

```

Sumber: Jiawei Han. (2000)

Gambar 2.13. Construct a frequent item matrix

(2) Generate length-2 sequential patterns

frequent item matrix:

b	4					
c	(4,3,0)	1				
a	(3,2,0)	(2,1,1)	2			
d	(2,2,2)	(2,2,0)	(1,2,1)	1		
e	(3,1,1)	(1,1,2)	(1,0,1)	(1,1,1)	1	
f	(2,2,2)	(1,1,0)	(1,1,0)	(0,0,0)	(1,1,0)	2
	b	c	a	d	e	f

$\langle bf \rangle$ ,  $\langle fb \rangle$ ,  $\langle (bf) \rangle$ ,  $\langle ff \rangle$

Item	Length-2 sequential patterns
f	$\langle bf \rangle:2$ , $\langle fb \rangle:2$ , $\langle (bf) \rangle:2$ , $\langle ff \rangle:2$
e	$\langle be \rangle:3$ , $\langle (ce) \rangle:2$
d	$\langle bd \rangle:2$ , $\langle db \rangle:2$ , $\langle (bd) \rangle:2$ , $\langle cd \rangle:2$ , $\langle dc \rangle:2$ , $\langle da \rangle:2$
a	$\langle ba \rangle:3$ , $\langle ab \rangle:2$ , $\langle ca \rangle:2$ , $\langle aa \rangle:2$
c	$\langle bc \rangle:4$ , $\langle cb \rangle:3$
b	$\langle bb \rangle:4$

Sumber: Jiawei Han. (2000)

Gambar 2.14. Generate length-2 sequential patterns

(3) Generate item-repeating patterns

frequent item matrix:

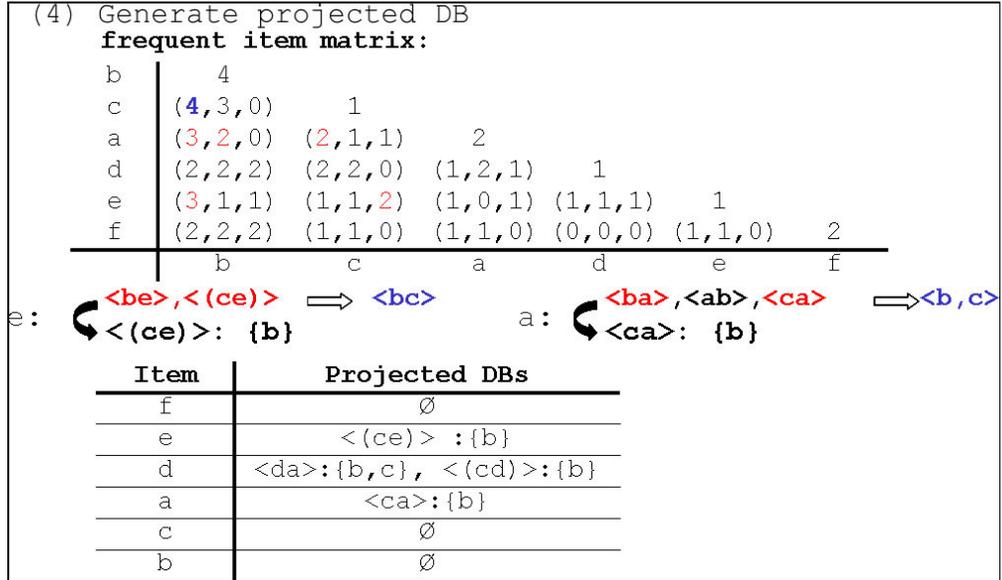
b	4					
c	(4,3,0)	1				
a	(3,2,0)	(2,1,1)	2			
d	(2,2,2)	(2,2,0)	(1,2,1)	1		
e	(3,1,1)	(1,1,2)	(1,0,1)	(1,1,1)	1	
f	(2,2,2)	(1,1,0)	(1,1,0)	(0,0,0)	(1,1,0)	2
	b	c	a	d	e	f

$\{b^+ f^+\}$

Item	Repeating items
f	$\{b^+ f^+\}$
e	$\langle b^+ e \rangle$
d	$\{b^+ d\}$ , $\langle da^+ \rangle$
a	$\langle aa^+ \rangle$ , $\{a^+ b^+\}$ , $\langle ca^+ \rangle$
c	$\{b^+ c\}$
b	$\langle bb^+ \rangle$

Sumber: Jiawei Han. (2000)

Gambar 2.15. Generate item-repeating patterns



Sumber: Jiawei Han. (2000)

Gambar 2.16. Generate projected DB

(5) Four projected DB and their sequential patterns

<b>annotation</b>	$\langle (ce) \rangle : \{b\}$	$\langle da \rangle : \{b, c\}$
<b>projected DB</b>	$\langle b(ce)b \rangle, \langle b(ce) \rangle$	$\langle (bd)cb(ac) \rangle, \langle (bd)bcba \rangle$
<b>sequential patterns</b>	$\langle b(ce) \rangle : 2$	$\langle (bd)a \rangle : 2, \langle dca \rangle : 2, \langle dba \rangle : 2, \langle (bd)ca \rangle : 2, \langle (bd)ba \rangle : 2, \langle dcba \rangle : 2, \langle (bd)cba \rangle : 2$
	$\langle cd \rangle : \{b\}$	$\langle ca \rangle : \{b\}$
	$\langle (bd)cbc \rangle, \langle bcd \rangle, \langle (bd)bcbd \rangle$	$\langle bcba \rangle, \langle bbcba \rangle$
	$\langle bcd \rangle : 2, \langle (bd)c \rangle : 2, \langle dcb \rangle : 2, \langle (bd)cd \rangle : 2, \langle (bd)bc \rangle : 2$	$\langle bca \rangle : 2, \langle cba \rangle : 2, \langle bcba \rangle : 2$

Sumber: Jiawei Han. (2000)

Gambar 2.17. Four projected DB and their sequential patterns

### BAB 3. TUJUAN DAN MANFAAT PENELITIAN

Tujuan yang ingin dicapai dari penelitian ini adalah untuk menciptakan suatu sistem aplikasi yang dapat menghasilkan basis pengetahuan berupa:

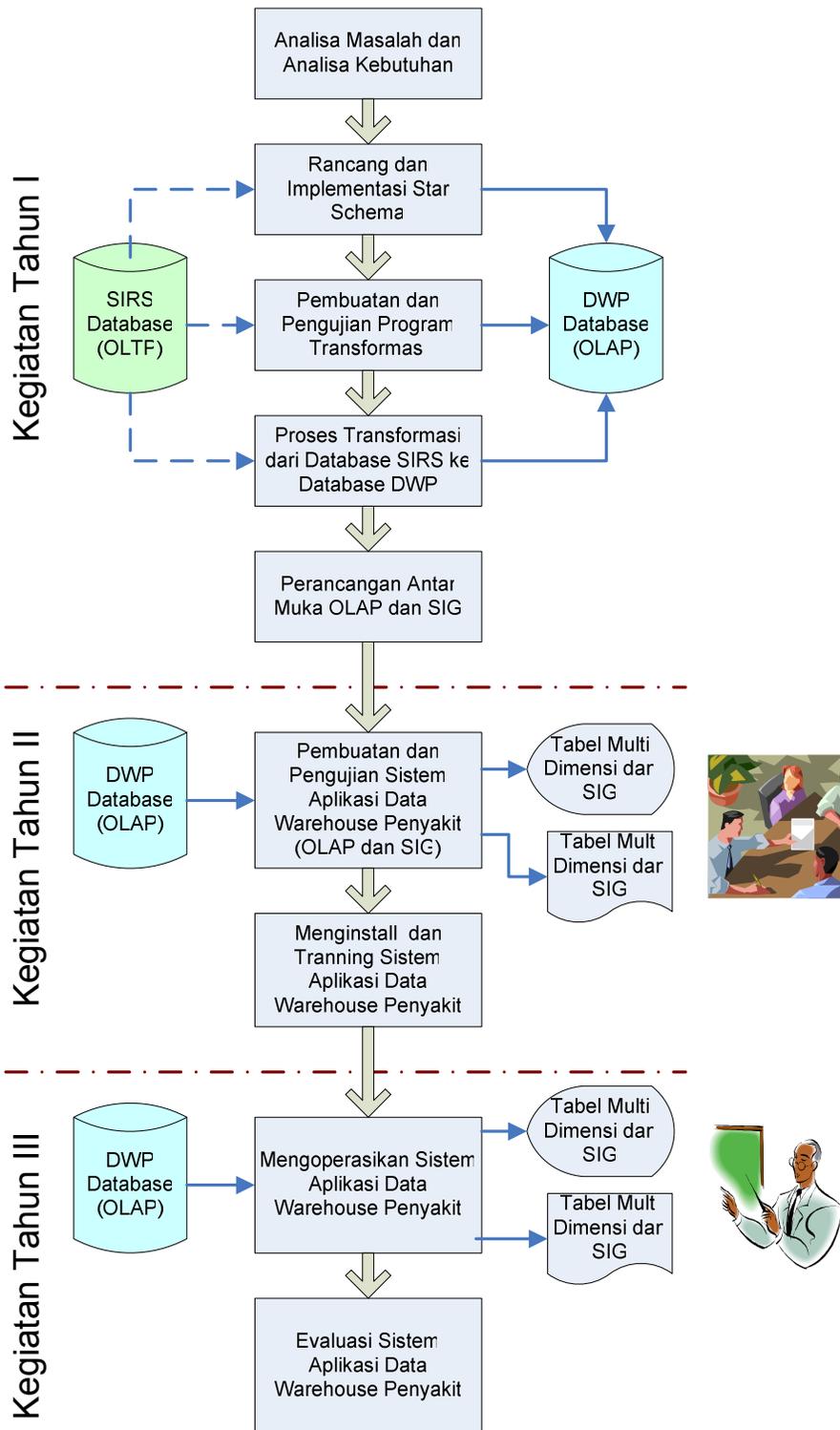
- Tabel rekapitulasi diagnosa penyakit pasien dalam multi dimensi.
- *Knowledge workers* dapat memilih dimensi-dimensi untuk disajikan dalam suatu tabel secara mudah, cepat, dan fleksibel sesuai dengan masalah dan keputusan strategis tidak terstruktur yang harus diambil.
- Dengan metode *sequential mining* akan dihasilkan pola sekuensial tentang terjangkit penyakit pada suatu daerah yang disebabkan oleh penyakit lain yang telah diderita sebelumnya. Informasi ini akan memudahkan *knowledge workers* untuk mengambil tindakan yang dibutuhkan.
- Tampilan grafik SIG untuk memvisualisasikan/ memetakan rekapitulasi penyebaran wabah penyakit tertentu pada kota-kota di suatu propinsi. Dengan visualisasi SIG, *knowledge workers* dapat memprediksi arah penyebaran wabah dan mengambil tindakan preventif yang diperlukan.

Luaran sistem aplikasi yang dihasilkan dalam penelitian ini diharapkan dapat memberikan informasi yang lebih kepada *knowledge workers* untuk menunjang pengambilan keputusan strategis di RSUD Dr. Soetomo dan instansi kesehatan yang terkait. Dengan demikian diharapkan dapat menentukan tindakan-tindakan yang tepat dalam pemberantasan penyakit dan pencegahan penyakit tersebut.

## BAB 4. METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini mengikuti sistem *development life cycle* yang diajukan oleh Witten et al (2004). Dokumentasi sistem dilakukan pada setiap langkah/ aktifitas ke dalam *log book*. Pertemuan rutin peneliti dilakukan minimal 2 kali dalam satu bulan. Langkah-langkah yang akan dilakukan untuk menyelesaikan penelitian dapat dilihat pada Gambar 4.1, sebagai berikut:

1. Mengadakan survei/ wawancara/ observasi ke RSUD Dr. Soetomo untuk mengetahui lebih lanjut permasalahan dan kebutuhan sistem yang perlu dirancang dan dibuat. Alat yang digunakan untuk mengumpulkan masalah dan kebutuhan sistem adalah daftar pertanyaan. Dalam kegiatan ini juga dikumpulkan dokumen-dokumen SIRS.
2. Berdasarkan daftar pertanyaan dan dokumen SIRS yang terkumpul dilakukan analisis masalah dan analisis kebutuhan. Dilanjutkan dengan mempresentasikan hasil analisis dan memperbaiki terhadap analisis jika diperlukan.
3. Berdasarkan analisis kebutuhan dirancang *Star Schema* seperti yang sudah dibahas pada Sub Bab 2.3. Kemudian mengimplementasikan *Star Schema* tersebut ke dalam skema *Database Data Warehouse* Diagnosa Penyakit (DWP).
4. Dibuat program untuk mentransformasikan *Database* SIRS ke dalam *Database* DWP berdasarkan *Star Schema* yang dibuat. Setelah itu program tersebut diuji coba sampai terbebas dari error.
5. Dilakukan proses transformasi dari *Database* SIRS ke dalam *Database* DWP. Hal tersebut dilakukan untuk menyiapkan *database* untuk sistem aplikasi *data warehouse* pasien yang akan dirancang.
6. Melakukan perancangan terhadap antar muka sistem aplikasi *data warehouse* pasien, dalam hal ini OLAP dan SIG menggunakan konsep yang dibahas pada Bab 2.



Gambar 4.1. Kerangka Konseptual Penelitian

7. Membuat dan menguji coba sistem aplikasi *data warehouse* diagnosa penyakit sampai terbebas dari kesalahan program. Mulai dari pertengahan sampai dengan akhir tahap ini didemokan sistem aplikasi kepada *knowledge users* secara rutin. Tujuannya agar aplikasi yang dibuat benar-benar sesuai dengan kebutuhan RSUD Dr. Soetomo.
8. Menginstall sistem aplikasi *data warehouse* diagnosa penyakit di RSUD Dr. Soetomo. Memberikan training kepada *knowledge workers*.
9. Mengoperasikan sistem aplikasi *data warehouse* penyakit. Agar sistem aplikasi dapat benar-benar digunakan oleh pihak RSUD Dr. Soetomo, maka akan dilakukan pendampingan dan evaluasi terhadap sistem aplikasi selama 6 bulan. Selama periode ini akan dipublikasikan hasil penelitian dalam seminar, baik di tingkat nasional maupun internasional, atau jurnal terakreditasi.

Dari langkah-langkah tersebut di atas diusulkan untuk dikerjakan dalam waktu 3 tahun dengan pencapaian target yang dapat dilihat pada Tabel 4.1.

Tabel 4.1. Capaian Target

<b>Tahun</b>	<b>Luaran Yang Dihasilkan</b>	<b>Target</b>
I	Star Schema	1 schema
	Program Pentransfer <i>Database</i> SIRS ke <i>Database</i> DWP	1 aplikasi
	<i>Database</i> DWP	1 <i>database</i>
	Antar Muka OLAP dan SIG	1 rancangan
	Seminar/ Publikasi Ilmiah	2 artikel
II	Modul Program OLAP	1 aplikasi
	Modul Program SIG	1 aplikasi
	Buku Pemakaian Sistem Aplikasi	1 buku
	Knowledge workers yang dapat menggunakan sistem aplikasi	4 orang
	Seminar/ Publikasi Ilmiah	1 artikel
III	Sistem Aplikasi <i>Data Warehouse</i> Pasien	1 aplikasi
	Evaluasi Sistem untuk pemeliharaan mengembangkan sistem.	6 bulan
	Hak Cipta	1 Hak Cipta

## BAB 5. HASIL DAN PEMBAHASAN

Pada laporan penelitian ini, ada beberapa hasil capaian yang dibahas yaitu:

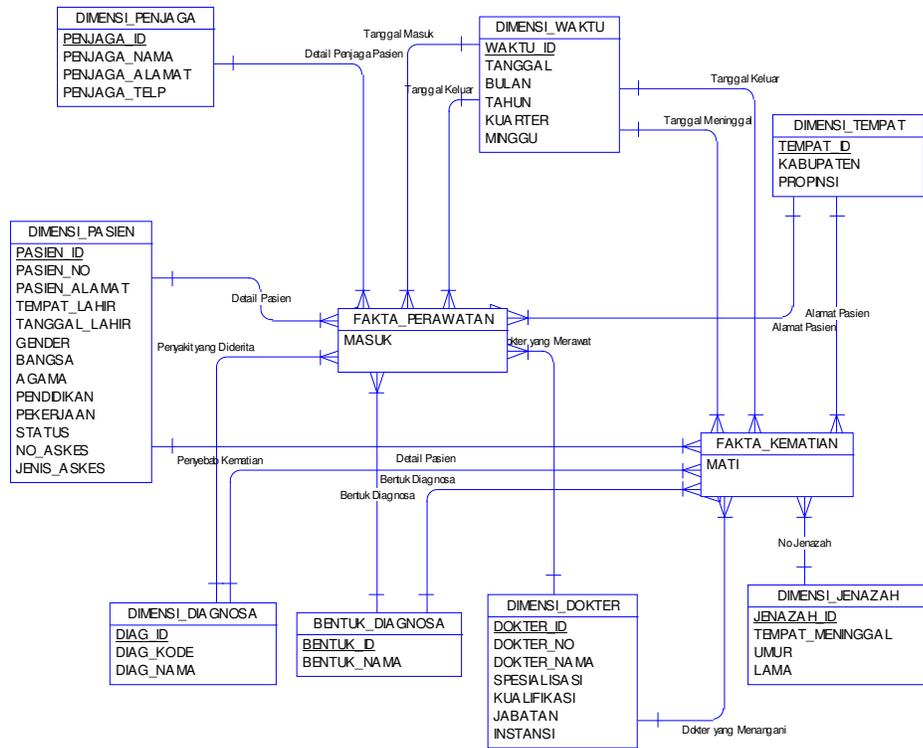
1. Prototipe *data warehouse* dan OLAP Tool Track Record Penyakit di RSUD Dr. Soetomo Surabaya.
2. Prototipe *data warehouse* dan OLAP Tool Manajemen Data RSUD Dr. Soetomo Surabaya.
3. Sistem aplikasi SIG penyebaran wabah penyakit.
4. Sistem aplikasi *sequential mining* untuk mengenali pola sebab-akibat terjangkitnya penyakit pada pasien.

### **5.1. Prototipe data warehouse dan OLAP Tool Track Record Penyakit di RSUD Dr. Soetomo Surabaya.**

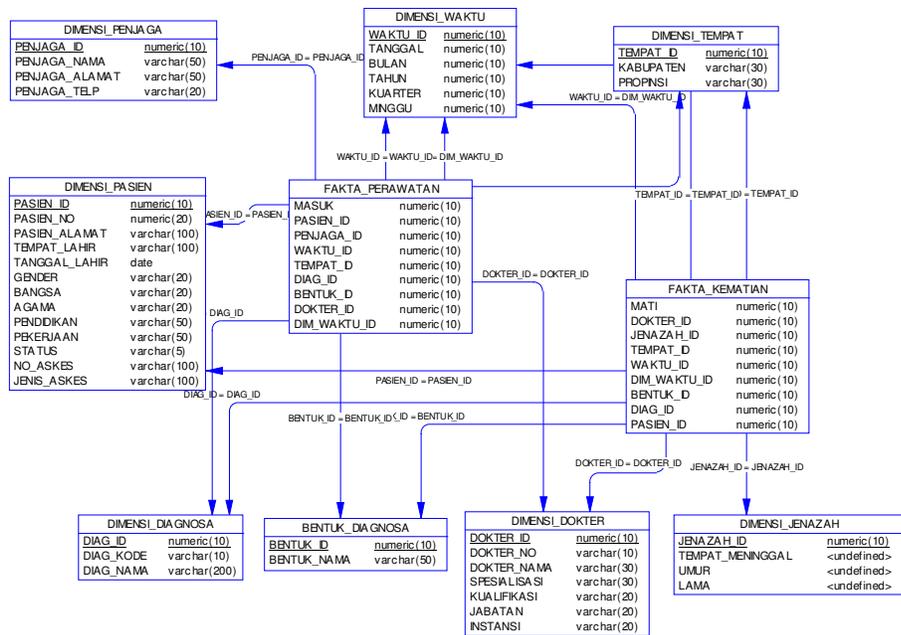
#### 5.1.1. Star Schema

*Database star schema* yang akan dibuat berbeda dengan *database* yang ada pada RSUD Dr. Soetomo. Perbedaannya terletak pada struktur *database*. *Database star schema* dirancang khusus dengan tujuan mempermudah proses *query* dan membantu menyimpan data dalam *database* khusus yang hanya dapat diimplementasikan dengan *OLAP Tools*.

Semua data yang terdapat pada *database* perusahaan telah melewati proses *cleansing* dan proses transformasi ke dalam *database star schema*. Proses *cleansing* ini berguna untuk menghilangkan data-data yang kurang *valid* dan membuat data-data yang ditransformasi menjadi lebih konsisten. Gambar 5.1 dan Gambar 5.2 adalah hasil perancangan *star schema* secara keseluruhan baik itu secara *conceptual design* maupun *physical design*.



Gambar 5.1. Desain Keseluruhan Star Schema (Conceptual Design)



Gambar 5.2. Desain Keseluruhan Star Schema (Physical Design)

Berikut ini adalah keterangan dari masing-masing tabel yang terdapat dalam *database star schema*:

#### 5.1.1.1 Tabel DIMENSI\_PASIEN

Tabel ini berisi data tentang detail pasien yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.1. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.1. Tabel DIMENSI\_PASIEN

Nama Field	Tipe Data	Ukuran	Keterangan
<u>PASIEN_ID</u> *	NUMBER	10	Menyimpan id pasien dari pasien
PASIEN_NO	VARCHAR2	10	Menyimpan nomor identitas pasien
PASIEN_ALAMAT	VARCHAR2	100	Menyimpan alamat pasien
TEMPAT_LAHIR	VARCHAR2	100	Menyimpan kota kelahiran pasien
TANGGAL_LAHIR	DATE		Menyimpan tanggal lahir pasien.
GENDER	VARCHAR2	2	Menyimpan gender pasien
BANGSA	VARCHAR2	50	Menyimpan bangsa pasien
AGAMA	VARCHAR2	50	Menyimpan agama pasien
PENDIDIKAN	VARCHAR2	50	Menyimpan pendidikan pasien
PEKERJAAN	VARCHAR2	50	Menyimpan pekerjaan pasien
STATUS	VARCHAR2	50	Menyimpan status nikah pasien
NO_ASKES	VARCHAR2	50	Menyimpan nomor ASKES pasien
JENIS_ASKES	VARCHAR2	50	Menyimpan jenis ASKES pasien

#### 5.1.1.2 Tabel DIMENSI\_PENJAGA

Tabel ini berisi data tentang detail penjaga pasien yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.2. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.2. Tabel DIMENSI\_PENJAGA

Nama Field	Tipe Data	Ukuran	Keterangan
<u>PENJAGA_ID</u> *	NUMBER	10	Menyimpan id penjaga pasien
PENJAGA_NAMA	VARCHAR2	50	Menyimpan nama penjaga
PENJAGA_ALAMAT	VARCHAR2	50	Menyimpan alamat penjaga
PENJAGA_TELP	VARCHAR2	20	Menyimpan nomor telepon penjaga

## 5.1.1.3 Tabel DIMENSI\_WAKTU

Tabel ini berisi data tentang detail waktu masuk dan keluarnya pasien yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN\_IK. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.3. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.3. Tabel DIMENSI\_WAKTU

Nama Field	Tipe Data	Ukuran	Keterangan
<u>WAKTU_ID</u> *	NUMBER	10	Menyimpan id waktu
TANGGAL	NUMBER	10	Menyimpan tanggal
BULAN	NUMBER	10	Menyimpan bulan
TAHUN	NUMBER	10	Menyimpan tahun
KUARTER	NUMBER	10	Menyimpan kuartar
MINGGU	NUMBER	10	Menyimpan minggu

## 5.1.1.4 Tabel DIMENSI\_TEMPAT

Tabel ini berisi data tentang detail tempat seperti nama kabupaten dan nama propinsi yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.4. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.4. Tabel DIMENSI\_TEMPAT

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>TEMPAT_ID</u> *	NUMBER	10	Menyimpan id tempat
KABUPATEN	VARCHAR2	30	Menyimpan nama kabupaten
PROPINSI	VARCHAR2	30	Menyimpan alamat propinsi

## 5.1.1.5 Tabel DIMENSI\_DOKTER

Tabel ini berisi data tentang detil dokter yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari database RSDBA, yaitu dari tabel DOKTER. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.5. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.5. Tabel DIMENSI\_DOKTER

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>DOKTER_ID</u> *	NUMBER	10	Menyimpan id dokter
DOKTER_NO	VARCHAR2	10	Menyimpan nomor identitas dokter
DOKTER_NAMA	VARCHAR2	30	Menyimpan nama dokter
SPELIALISASI	VARCHAR2	30	Menyimpan spesialisasi dokter
KUALIFIKASI	VARCHAR2	20	Menyimpan kualifikasi dokter
JABATAN	VARCHAR2	20	Menyimpan jabatan dokter
INSTANSI	VARCHAR2	20	Menyimpan instansi dokter

## 5.1.1.6 Tabel DIMENSI\_DIAGNOSA

Tabel ini berisi data tentang diagnosa yang dilakukan oleh dokter yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN\_IK. *Field-field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.6. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.6. Tabel DIMENSI\_DIAGNOSA

Nama Field	Tipe Data	Ukuran	Keterangan
<u>DIAG_ID</u> *	NUMBER	10	Menyimpan id diagnosa
DIAG_KODE	VARCHAR2	10	Menyimpan kode diagnosa
DIAG_NAMA	VARCHAR2	200	Menyimpan nama diagnosa

## 5.1.1.7 Tabel BENTUK\_DIAGNOSA

Tabel ini berisi data tentang bentuk diagnosa yang dilakukan oleh dokter yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel PASIEN\_IK. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.7. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.7. Tabel BENTUK\_DIAGNOSA

Nama Field	Tipe Data	Ukuran	Keterangan
<u>BENTUK_ID</u> *	NUMBER	10	Menyimpan id diagnosa
BENTUK_NAMA	VARCHAR2	50	Menyimpan nama bentuk diagnosa

## 5.1.1.8 Tabel DIMENSI\_JENAZAH

Tabel ini berisi data tentang detail jenazah yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari tabel JENAZAH. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.8. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.8. Tabel DIMENSI\_JENAZAH

Nama Field	Tipe Data	Ukuran	Keterangan
<u>JENAZAH_ID</u> *	NUMBER	10	Menyimpan id jenazah
TEMPAT_MENINGGAL	VARCHAR2	50	Menyimpan tempat pasien meninggal
UMUR	NUMBER	10	Menyimpan umur
LAMA	NUMBER	10	Menyimpan lama inap

#### 5.1.1.9 Tabel FAKTA\_PERAWATAN

Tabel ini berisi data tentang fakta perawatan yang terdapat dalam RSUD Dr. Soetomo Surabaya. Tabel ini berelasi dengan tabel DIMENSI\_PASIEN, tabel DIMENSI\_PENJAGA, tabel DIMENSI\_DOKTER, tabel DIMENSI\_WAKTU, tabel DIMENSI\_TEMPAT, tabel DIMENSI\_DIAGNOSA, dan tabel BENTUK\_DIAGNOSA. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.9. Nama *field* yang bertanda # memiliki arti *foreign key*.

Tabel 5.9. Tabel FAKTA\_PERAWATAN

Nama Field	Tipe Data	Ukuran	Keterangan
PASIEN_ID #	NUMBER	10	Menyimpan id pasien
DOKTER_ID #	NUMBER	10	Menyimpan id dokter
WAKTU_ID_MASUK #	NUMBER	10	Menyimpan id waktu masuk
WAKTU_ID_KELUAR #	NUMBER	10	Menyimpan id waktu keluar
TEMPAT_ID #	NUMBER	10	Menyimpan id tempat
PENJAGA_ID #	NUMBER	10	Menyimpan id penjaga
DIAG_ID #	NUMBER	10	Menyimpan id diagnosa
BENTUK_ID #	NUMBER	10	Menyimpan id bentuk diagnosa
MASUK	NUMBER	10	Menyimpan status masuk rumah sakit

#### 5.1.1.10 Tabel FAKTA\_KEMATIAN

Tabel ini berisi data tentang fakta kematian yang terdapat dalam RSUD Dr. Soetomo Surabaya. Tabel ini berelasi dengan tabel DIMENSI\_PASIEN, tabel DIMENSI\_JENAZAH, tabel DIMENSI\_DOKTER, tabel DIMENSI\_WAKTU, tabel DIMENSI\_TEMPAT, tabel DIMENSI\_DIAGNOSA, dan tabel BENTUK\_DIAGNOSA. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.10. Nama *field* yang bertanda # memiliki arti *foreign key*.

Tabel 5.10. Tabel FAKTA\_KEMATIAN

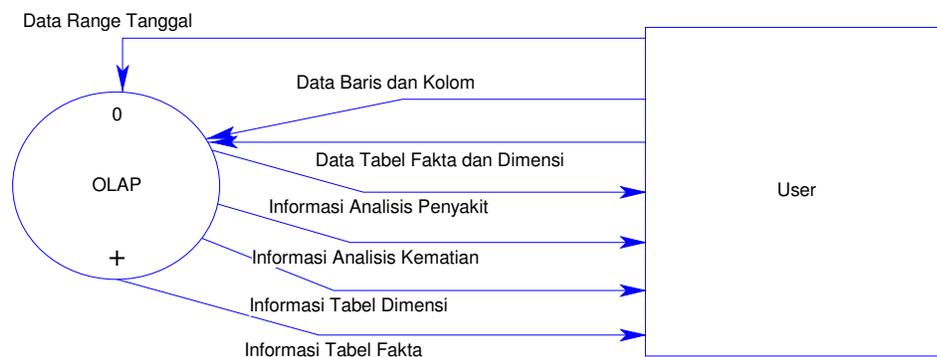
Nama Field	Type Data	Ukuran	Keterangan
JENAZAH_ID #	NUMBER	10	Menyimpan id jenazah
PASIEN_ID #	NUMBER	10	Menyimpan id pasien
DOKTER_ID #	NUMBER	10	Menyimpan id dokter
WAKTU_ID_MAS UK #	NUMBER	10	Menyimpan id waktu masuk
WAKTU_ID_KEL UAR #	NUMBER	10	Menyimpan id waktu keluar
TEMPAT_ID #	NUMBER	10	Menyimpan id tempat
DIAG_ID #	NUMBER	10	Menyimpan id diagnosa
BENTUK_ID #	NUMBER	10	Menyimpan id bentuk diagnosa
MATI	NUMBER	10	Menyimpan pasien meninggal

### 5.1.2. Data Flow Diagram OLAP

Penjelasan mengenai *data flow diagram* (DFD) dari prototipe OLAP untuk *data warehouse track record* penyakit adalah sebagai berikut:

#### 5.1.2.1 Context Diagram

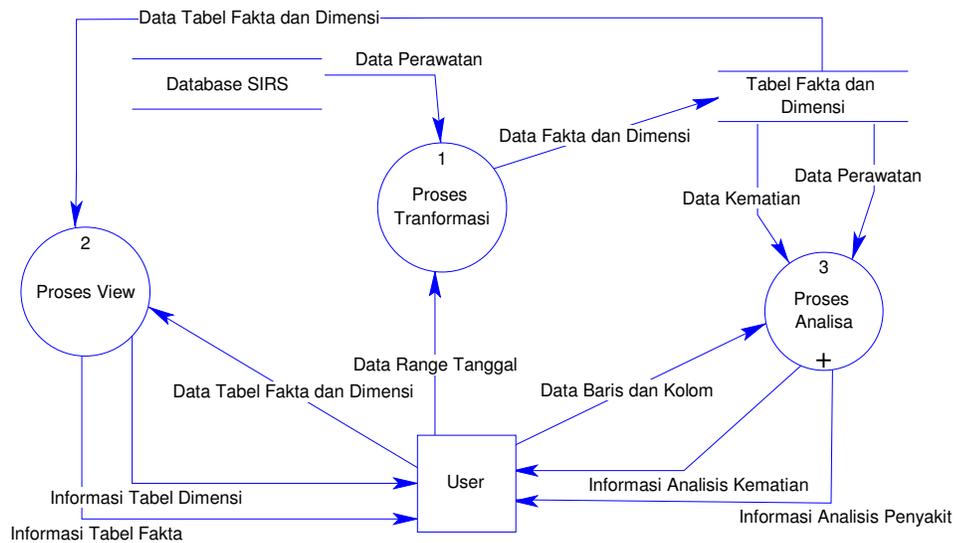
Gambar 5.3 adalah *Context Diagram*, dimana terdapat suatu proses yaitu OLAP pada rumah sakit. Dalam diagram terdapat dua *entity* yaitu *user* dan Sistem Rumah Sakit yang merupakan sumber data.



Gambar 5.3. Context Diagram OLAP Track Record Penyakit

### 5.1.2.2 DFD Level 0

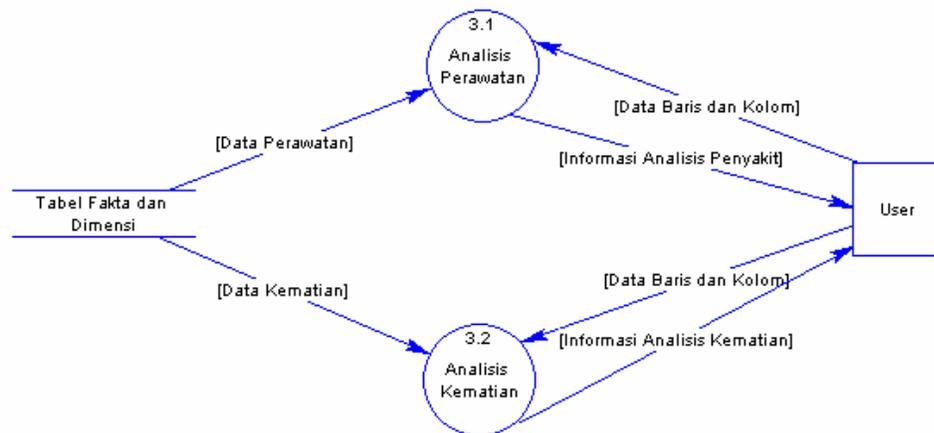
Gambar 5.4 adalah DFD *Level 0*. Pada *level* ini terdapat proses *meta table*, transformasi, analisis dan *view table*. Sumber data perawatan dan kematian berasal dari *database* operasional milik SIRS. Setelah data tersebut melalui proses transformasi, maka data tersebut akan dimasukkan dalam *table fakta* dan *tabel dimensi*.



Gambar 5.4. DFD Level 0 OLAP Track Record Penyakit

### 5.1.2.3 DFD Level 1

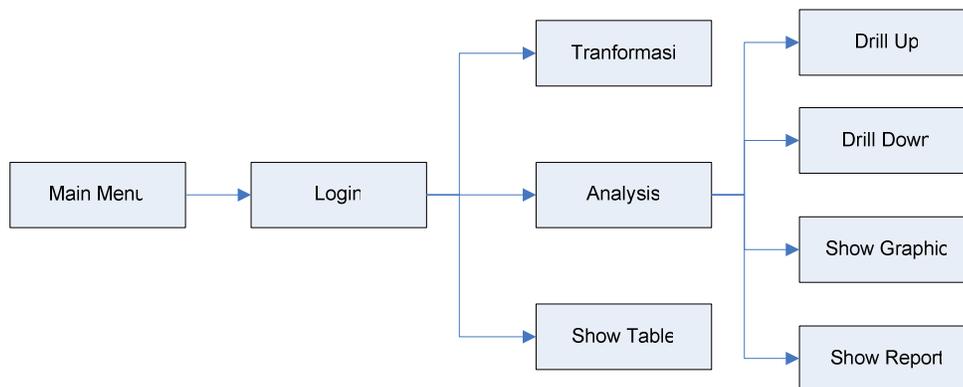
Gambar 5.5 adalah gambar DFD *level 1*. Pada setiap proses analisis akan menggunakan data dari *table fakta* dan *tabel dimensi* untuk menghasilkan laporan untuk *user*.



Gambar 5.5. DFD Level 1 OLAP Track Record Penyakit

### 5.1.3. Struktur Menu Prototipe OLAP Track Record Penyakit

Pada Gambar 5.6 dapat dilihat struktur menu dari prototipe aplikasi *data warehouse* dan OLAP untuk Track Record penyakit di RSUD Dr. Soetomo Surabaya.

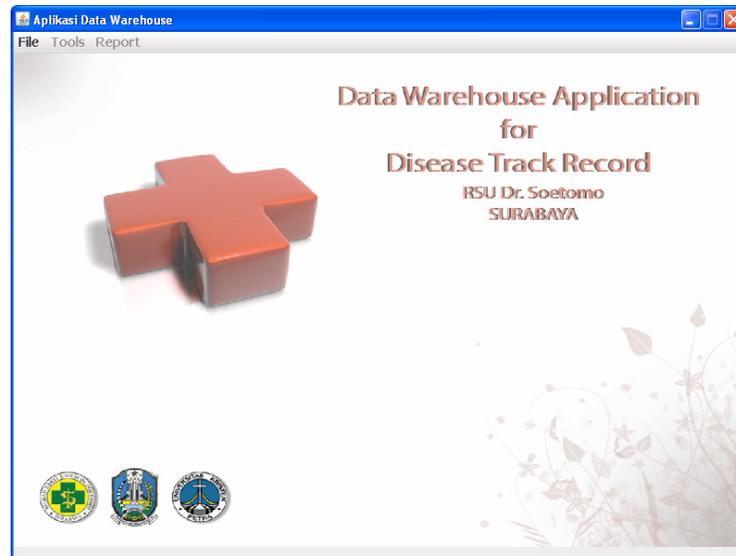


Gambar 5.6. Struktur Menu OLAP Track Record Penyakit

### 5.1.4. Pengujian Sistem Prototipe Data Warehouse dan OLAP untuk Track Record Penyakit Pasien di RSUD Dr. Soetomo

*Form* Main Menu adalah *form* yang pertama kali muncul saat program dijalankan. Dalam *form* ini, *user* harus melakukan *login* terlebih dahulu sebelum

dapat melakukan aktivitas yang lain. Gambar 5.7 adalah tampilan *form* Main Menu.



Gambar 5.7. Form Main Menu

*Form* Login adalah menu yang harus dijalankan agar dapat menjalankan menu yang lain. Dalam *form* ini, *user* melakukan *input* hak akses *user* berupa *username* dan *password* terlebih dahulu. Gambar 5.8 adalah tampilan dari *form* Login.

Gambar 5.8. Form Login

Menu Transformation adalah menu yang harus dijalankan untuk memindahkan data dari *database* Sistem Informasi Rumah Sakit (SIRS) ke *database star schema* yang diberi nama Data\_Warehouse. Data yang dipindahkan dari *database* SIRS ke dalam *database* Data\_warehouse, hanya data-data yang akan dianalisis saja.

Dalam proses transformasi, data dibagi ke dalam tabel fakta dan tabel dimensi yang terdapat pada *database* Data\_Warehouse. Data yang merupakan dimensi akan diperiksa terhadap atribut selain *primary key* pada tabel dimensi. Jika belum terdapat *record* yang sama pada tabel dimensi, maka dibuatkan *ID* baru pada tabel dimensi tersebut. Sedangkan untuk proses transformasi pada tabel fakta, dilakukan pemeriksaan terlebih dahulu terhadap kombinasi *foreign key* yang terdapat dalam tabel fakta. Jika *record* tersebut tidak ada dalam tabel fakta, maka *record* tersebut dimasukkan dalam tabel fakta tersebut. Jika *record* telah ada dalam tabel fakta, maka tidak terjadi proses *input*.

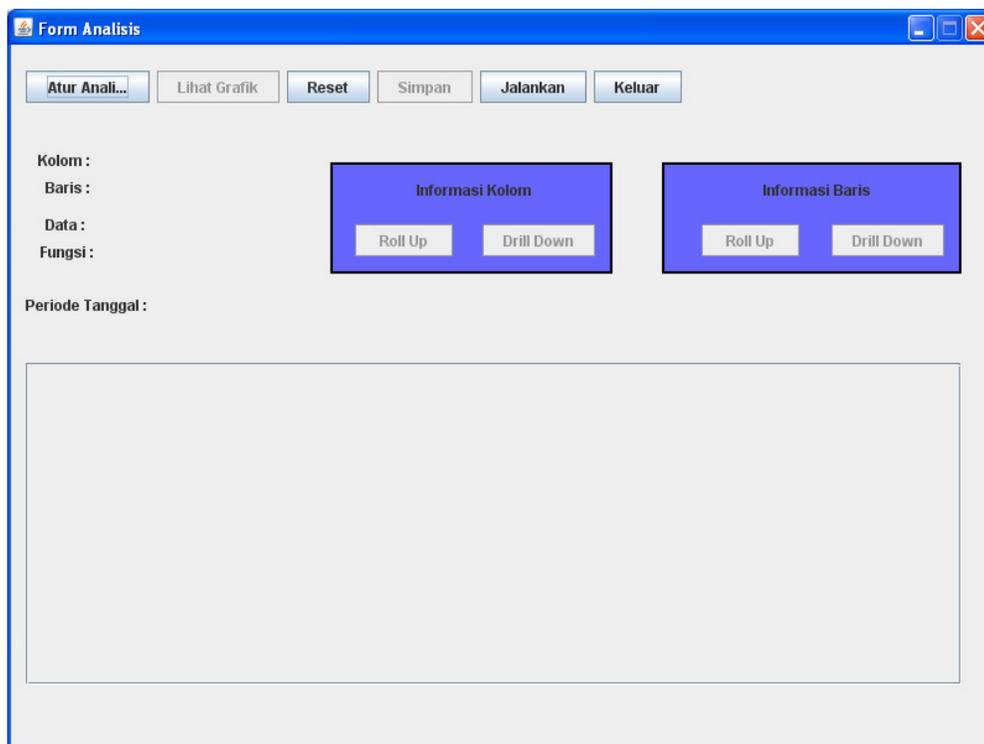
Sebelum proses transformasi dilakukan, setiap tabel fakta dan dimensi dalam *database* Data\_Warehouse tidak memiliki data sama sekali. Namun setelah dilakukan proses transformasi, maka setiap tabel yang terkait dengan fakta yang dianalisis tersebut akan terisi dengan data. Gambar 5.9 menunjukkan menu Transformation, dimana *user* ingin melakukan analisis terhadap fakta perawatan atau fakta kematian pada tanggal 1 Januari 2003 hingga 2 Januari 2003.



Gambar 5.9. Menu Tranformasi

Dalam menu Transformation disediakan pilihan waktu yang diproses. Jika *user* memilih "All", maka semua data yang ada pada *database* rumah sakit lama akan ditransformasikan ke dalam *database* Data\_Warehouse. Tetapi jika *user* memilih "Range", maka *user* harus memilih tanggal awal dan tanggal akhir pada "Start Date" dan "End Date". Jika *user* telah memilih waktu yang diinginkan, maka *user* harus menekan tombol "Process" untuk memulai proses transformasi.

Menu Analisis digunakan untuk melakukan analisis terhadap data yang telah ditransformasi. Analisis yang dapat dilakukan antara lain analisis terhadap perawatan dan kematian. Gambar 5.10 adalah tampilan awal menu Analysis. Tombol *Set Value* digunakan untuk mengeset *filter* yang diinginkan *user* untuk ditampilkan di tabel.



Gambar 5.10. Tampilan Awal Menu Analisis

Gambar 5.11 adalah tampilan *form* yang berfungsi untuk mengatur tampilan tabel analisis sesuai sudut pandang dan *item* setiap dimensi yang ingin ditampilkan. Pada bagian "Analisis", *user* dapat memilih analisis yang akan dilakukan, seperti analisis perawatan. Setelah memilih jenis analisis, *user* dapat meletakkan dimensi pada sisi kolom atau baris pada tabel. Analisis perawatan dapat dilihat dari:

*Filter* dimensi waktu diatur dalam "Tanggal Awal" dan "Tanggal Akhir" dimana *user* dapat menentukan *range* tanggal dari data yang akan dianalisis dengan memilih tanggal yang diinginkan.

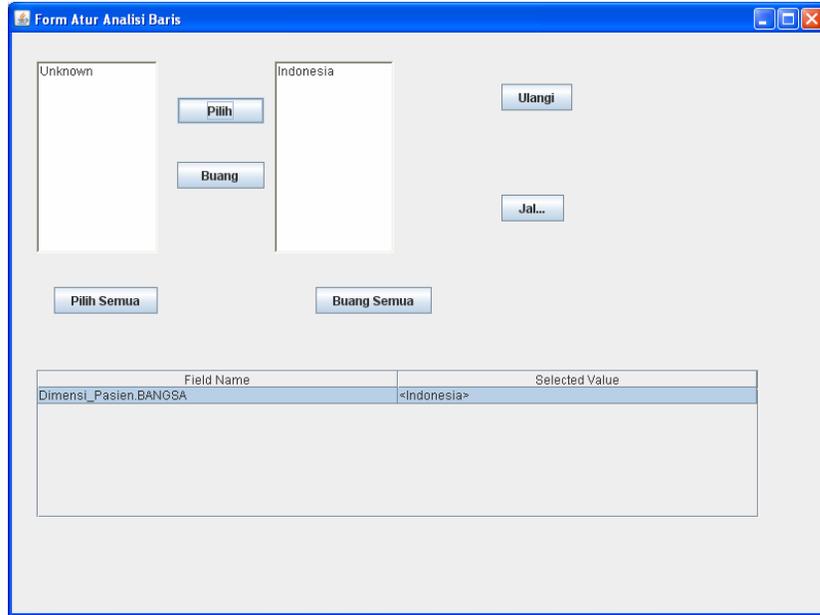
*Filter* dimensi fakta perawatan diatur dalam "Dasar Analisis" dimana *user* memilih kolom atau baris yang akan ditampilkan, misalnya nama penyakit sebagai baris dan jenis kelamin sebagai kolom. Setelah memilih kolom atau baris yang akan dianalisis maka isi dari kolom atau baris tersebut akan muncul dalam *listbox* di sebelah bawah.

Pada Gambar 5.12, *user* dapat memilih semua *item* dengan memilih "select all" atau memilih beberapa *item* dari *listbox* di sebelah kiri. *Item* yang dipilih akan dipindahkan ke dalam *listbox* di sebelah kanan. Hal yang sama dapat dilakukan pada filter kolom.

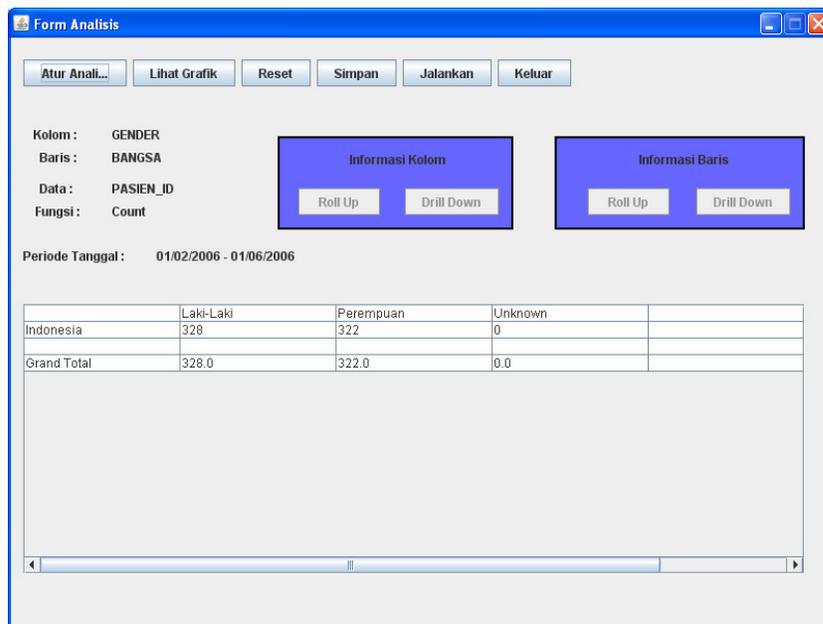
Hasil analisis ditampilkan dalam bentuk tabel seperti ditunjukkan pada Gambar 5.13, dimana analisis yang dilakukan dipilih pada tanggal 2 Januari 2005 sampai 5 Januari 2005, dengan memilih *field* *Diag\_nama* sebagai baris dan *field* *Gender* sebagai kolom. *Field* *Pasien\_id* dipilih sebagai data.

Tanggal Awal	Tanggal Akhir
01/02/2006	01/06/2006

Gambar 5.11. Form Filter Menu Analisis Perawatan

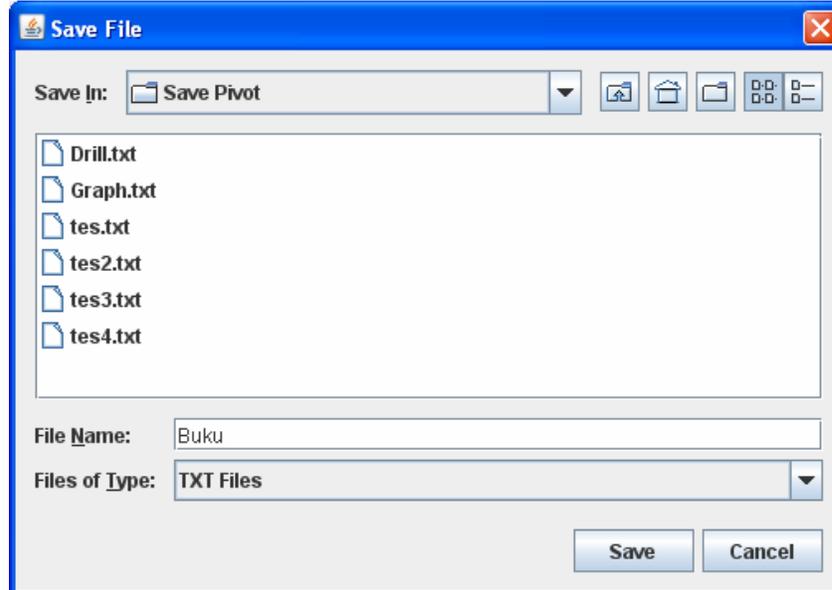


Gambar 5.12. Tampilan Filter

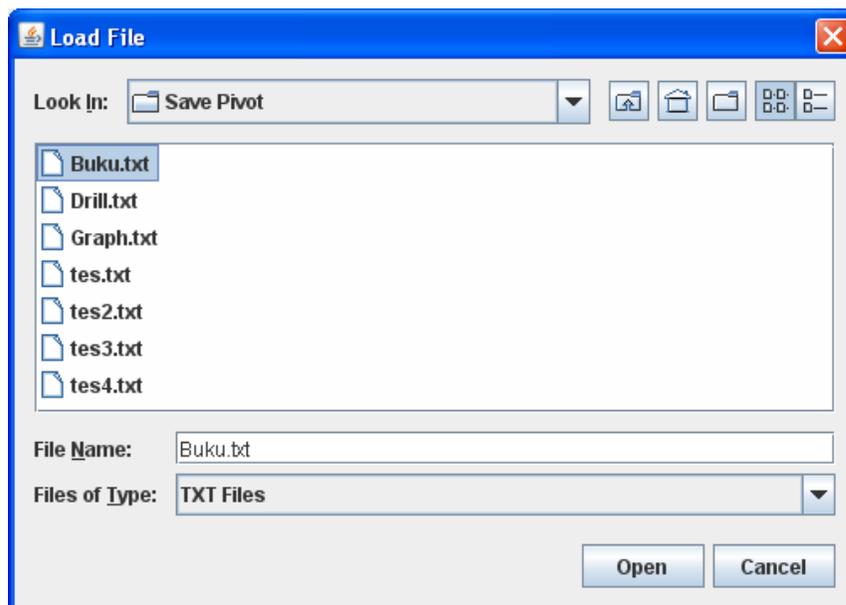


Gambar 5.13. Form Tabel Menu Analisis Perawatan

*User* dapat menyimpan hasil analisis yang telah dilakukan dengan menekan tombol Save. Gambar 5.14 adalah tampilan yang muncul jika ingin menyimpan hasil analisis. Sedangkan Gambar 5.15 adalah tampilan yang muncul jika ingin menampilkan hasil analisis yang telah disimpan.



Gambar 5.14. Tampilan untuk Menyimpan Pivot Table

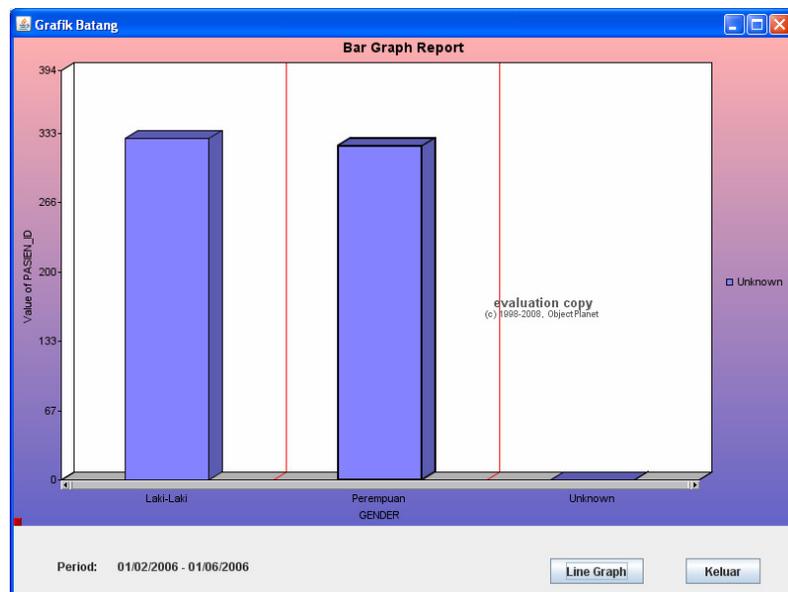


Gambar 5.15. Tampilan untuk Menampilkan Pivot Table yang telah disimpan

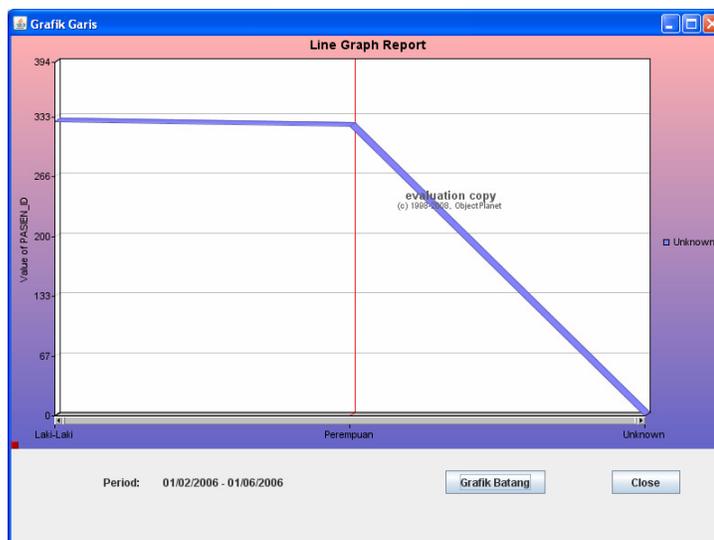
User dapat melihat data dalam bentuk grafik dengan menekan tombol View Graphic. Form grafik akan tampak seperti Gambar 5.16 dan Gambar 5.17. Pada sumbu X akan ditampilkan sesuai dengan *field* yang dipilih pada kolom,

sedangkan pada sumbu Y akan ditampilkan jumlah data. Legenda di sebelah kanan akan ditampilkan *field* yang dipilih pada baris.

Untuk pengujian hasil dari pivot table dibandingkan dengan pivot table pada Microsoft Excel 2003. Hal ini bertujuan untuk menguji keakuratan data hasil analisis. Sebelum melakukan pengujian terlebih dahulu harus men-setting data yang akan dibuat pivot table pada Excel 2003. Tabel 5.11 adalah hasil pivot table pada Microsoft Excell 2003. Gambar 5.18 adalah hasil pivot table pada sistem aplikasi Data Warehouse.



Gambar 5.16. Form Grafik Batang



Gambar 5.17. Form Grafik Garis

Tabel 5.11. Hasil pivot table pada Microsoft Excel 2003

Count(PASIEN_ID)	GENDER		
	DIAG_NAMA	L	P
Anaemia, unspecified	5	11	16
Anal abscess	1	1	2
Anaphylactic shock, unspecified		1	1
Grand Total	6	13	19

	L	P	U	
Anaemia, unspecified	5	11	0	
Anal abscess	1	1	0	
Anaphylactic shock, unsp...	0	1	0	
Grand Total	6	13	0	

Gambar 5.18. Hasil pivot table pada Sistem Aplikasi

User juga dapat memilih *field* lebih dari satu terhadap baris maupun kolom. Gambar 5.19 hingga Gambar 5.20 adalah tampilan dan hasil analisis jika *user* memilih dua *field* sebagai kolom, yaitu *field* Gender dan *field* Bangsa. Sedangkan pada baris dipilih *field* pendidikan. *Field* Pasien\_id dijadikan sebagai data. Tabel 5.12 dan Tabel 5.13 adalah hasil dari pivot table pada Excel 2003.

Gambar 5.19. Tampilan jika field kolom yang Dipilih Lebih dari Satu

	L Indonesia	L Unknown	L null	P Indonesia
Belum/Tidak Tamat SD	43	0	0	27
Tamat D3	2	0	0	6
Tamat S1	11	0	0	11
Tamat S2	0	0	0	0
Tamat S3	0	0	0	0
Tamat SD	31	0	0	48
Tamat SMTA	59	0	0	59
Tamat SMTP	38	0	0	32
Tidak Sekolah	42	0	0	25
Tidak tahu	15	0	0	16
Unknown	0	0	0	0
Grand Total	241	0	0	224

Gambar 5.20. Hasil Analisis jika field kolom yang Dipilih Lebih dari Satu

Analysis Form

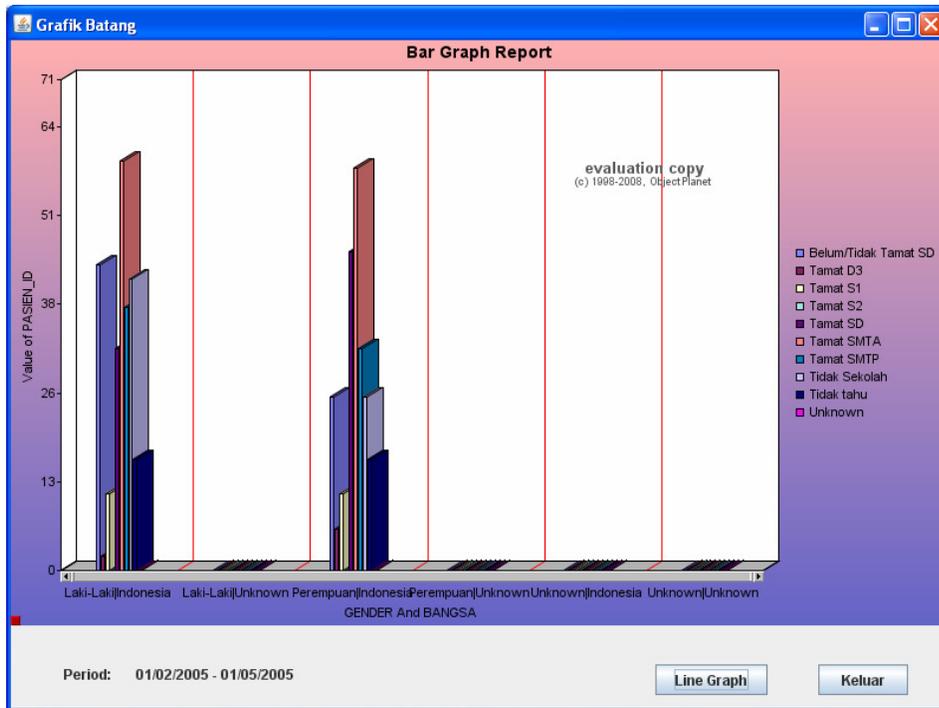
**Start Date :** 01/02/2005  
**End Date :** 01/05/2005  
**Column :** GENDER|BANGSA  
**Row :** PENDIDIKAN  
**Data :** PASIEN\_ID

	L	P	U	null
Belum/Tidak Tamat SD	43	27	0	0
Tamat D3	2	6	0	0
Tamat S1	11	11	0	0
Tamat S2	0	0	0	0
Tamat S3	0	0	0	0
Tamat SD	31	48	0	0
Tamat SMTA	59	59	0	0
Tamat SMTP	38	32	0	0
Tidak Sekolah	42	25	0	0
Tidak tahu	15	16	0	0
Unknown	0	0	0	0
Grand Total	241	224	0	0

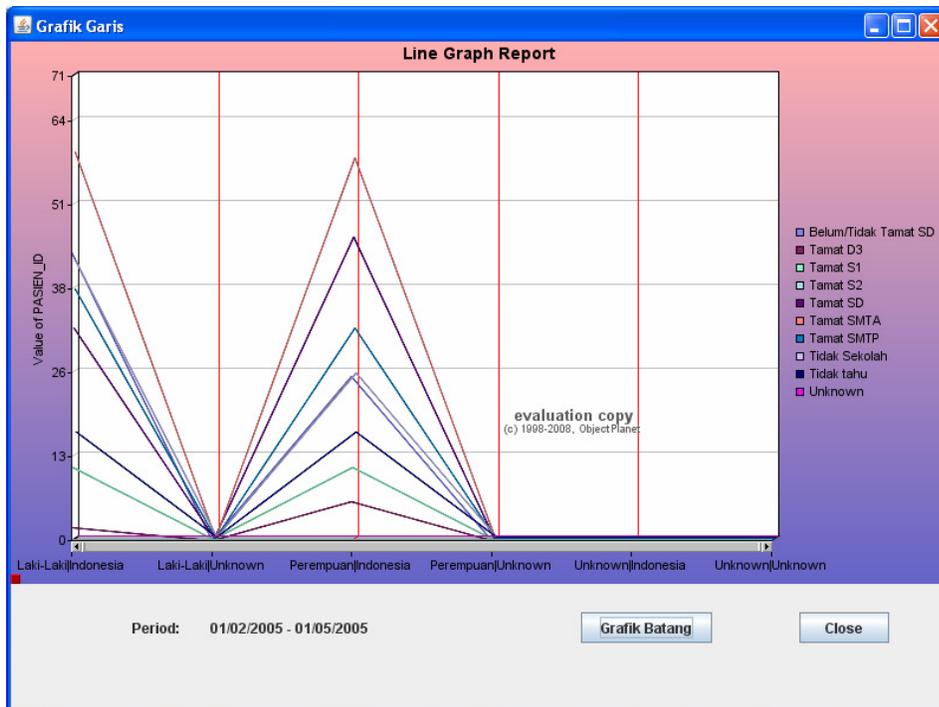
Gambar 5.21. Hasil Analisis jika di Roll Up

PASIEN_ID	PASIEN_NO	PASIEN_AL...	TEMPAT L...	TANGGAL
16	10295124	KEBONSA...	LAMONGAN	1996-06-3
23	10427471	WIGUNA T...		2003-06-2
38	10439800	GUBENG ...		2005-01-0
83	10383195	KRESIKAN ...		2000-01-0
199	10437826	PLOSO TIM...		2003-08-1
537	10437287	BANDUNG...		
540	10437328	NGAGEL R...		2004-03-0
642	10021263	TAMBAK M...		1915-06-3
647	10437542	BRATANG ...		
651	10422375	MANYAR S...		
653	10437530	BREBEK 3...		
657	10437572	BANYUATES		
802	10436257	PACAR KE...		2000-09-0
851	10419734	ASEM 4B/24		2003-12-0
937	10430697	WOTAN SU...		1996-04-2

Gambar 5.22. Hasil Zoom Up



Gambar 5.23. Tampilan Grafik Baris jika field kolom lebih dari Satu



Gambar 5.24. Tampilan Grafik Garis jika field kolom lebih dari Satu

Tabel 5.12. Hasil pivot table pada Microsoft Excel 2003

Count(PASIEN_ID)	GENDER		BANGSA		Grand Total
	L	L Total	P	P Total	
PENDIDIKAN	Indonesia		Indonesia		
Belum/Tidak Tamat SD	43	43	27	27	70
Tamat D3	2	2	6	6	8
Tamat S1	11	11	11	11	22
Tamat SD	31	31	48	48	79
Tamat SMTA	59	59	59	59	118
Tamat SMTP	38	38	32	32	70
Tidak Sekolah	42	42	25	25	67
Tidak tahu	15	15	16	16	31
Grand Total	241	241	224	224	465

	L Indonesia	L Unknown	L null	P Indonesia
Belum/Tidak Tamat SD	43	0	0	27
Tamat D3	2	0	0	6
Tamat S1	11	0	0	11
Tamat S2	0	0	0	0
Tamat S3	0	0	0	0
Tamat SD	31	0	0	48
Tamat SMTA	59	0	0	59
Tamat SMTP	38	0	0	32
Tidak Sekolah	42	0	0	25
Tidak tahu	15	0	0	16
Unknown	0	0	0	0
Grand Total	241	0	0	224

Gambar 5.25. Hasil pivot table pada Sistem Aplikasi

	L	P	U	null
Belum/Tidak Tamat SD	43	27	0	0
Tamat D3	2	6	0	0
Tamat S1	11	11	0	0
Tamat S2	0	0	0	0
Tamat S3	0	0	0	0
Tamat SD	31	48	0	0
Tamat SMTA	59	59	0	0
Tamat SMTP	38	32	0	0
Tidak Sekolah	42	25	0	0
Tidak tahu	15	16	0	0
Unknown	0	0	0	0
Grand Total	241	224	0	0

Gambar 5.26. Hasil pivot table pada Sistem Aplikasi setelah di Roll Up

Gambar 5.25 sampai dengan Gambar 5.29 adalah tampilan dan hasil analisis jika *user* memilih dua *field* sebagai kolom, yaitu *field* Gender dan *field* Agama. Sedangkan pada baris dipilih *field* Status dan *field* Pekerjaan. *Field* Jumlah\_Hari dijadikan sebagai data dan digunakan metode *sum* terhadap perhitungan data.

Analysis Form

Set Value Create Report View Graphic Reset Close Save Load

Drill Up Drill Down

Start Date : 01/02/2005

End Date : 01/05/2005

Column : GENDER|AGAMA

Row : STATUS|PEKERJAAN

Data : JUMLAH\_HARI

	L Budha	L Hindu	L Islam	L Katolik
B Dibawah Umur	0	0	1186	19
B Ibu Rumah Tangga	0	0	0	0
B Mahasiswa	0	0	0	0
B Nelayan	0	0	0	0
B Pedagang	0	0	0	0
B Peg Harian	0	0	0	0
B Pegawai Negeri	0	0	0	0
B Pegawai Swasta	0	0	113	0
B Pekerja Lepas	0	0	0	0
B Pelajar	0	0	345	0
B Pensiunan	0	0	3	0
B Petani	0	0	0	0
B Polri	0	0	0	0
B TNI	0	0	0	0
B Tidak Bekerja	0	0	145	0

Gambar 5.27. Hasil Analisis jika field kolom dan baris yang Dipilih Lebih dari Satu

Analysis Form

Set Value Create Report View Graphic Reset Close Save Load

Drill Up Drill Down

Start Date : 01/02/2005

End Date : 01/05/2005

Column : GENDER|AGAMA

Row : STATUS|PEKERJAAN

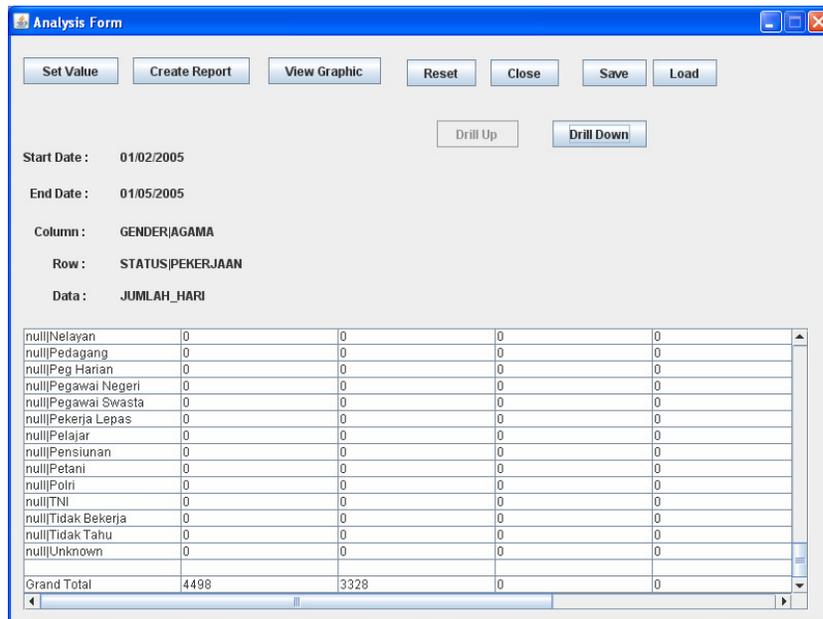
Data : JUMLAH\_HARI

	L	P	U	null
B Dibawah Umur	1253	630	0	0
B Ibu Rumah Tangga	0	0	0	0
B Mahasiswa	24	0	0	0
B Nelayan	0	0	0	0
B Pedagang	0	0	0	0
B Peg Harian	0	0	0	0
B Pegawai Negeri	0	0	0	0
B Pegawai Swasta	113	110	0	0
B Pekerja Lepas	0	9	0	0
B Pelajar	345	71	0	0
B Pensiunan	3	0	0	0
B Petani	0	0	0	0
B Polri	0	0	0	0
B TNI	0	0	0	0
B Tidak Bekerja	145	17	0	0

Gambar 5.28. Hasil Analisis jika field Baris dan Kolom yang Dipilih Lebih dari Satu dan di Roll Up

Tabel 5.13. Hasil pivot table pada Microsoft Excel 2003

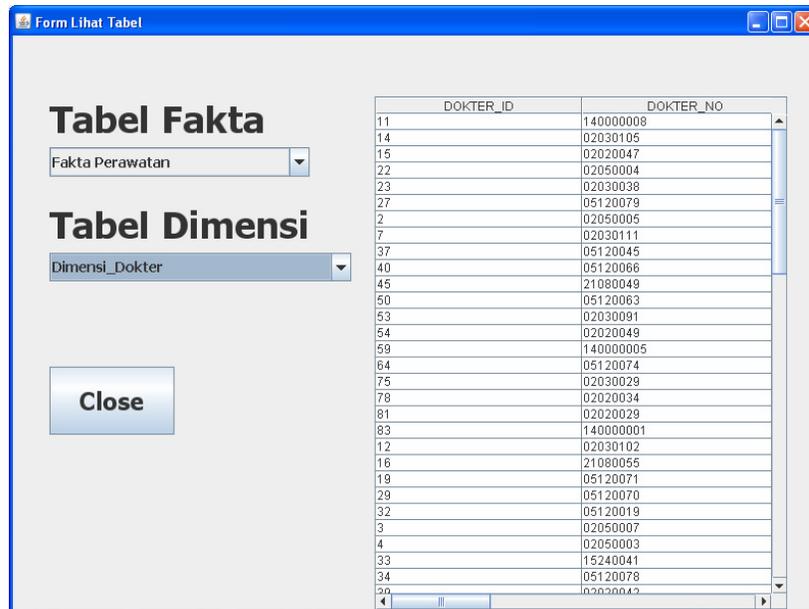
Sum of Sum(JUMLAH_HARI)		GENDER \ AGAMA				L Total		P Total		Grand Total
STATUS	PEKERJAAN	Islam	Katolik	Kristen	(blank)	Islam	Katolik	Kristen		
B	Dibawah Umur	1186	19	36	12	1253	621	9	630	1883
	Mahasiswa			24		24				24
	Pegawai Swasta	113				113	98	12	110	223
	Pekerja Lepas						9		9	9
	Pelajar	345				345	71		71	416
	Pensiunan	3				3				3
	Tidak Bekerja	145				145	17		17	162
Tidak Tahu	71				71	15		15	86	
<b>B Total</b>		<b>1863</b>	<b>19</b>	<b>60</b>	<b>12</b>	<b>1954</b>	<b>831</b>	<b>21</b>	<b>852</b>	<b>2606</b>
D	Pensiunan	70				70				70
	Petani	216				216				216
	Tidak Bekerja	6				6				6
	Tidak Tahu	20				20				20
<b>D Total</b>		<b>312</b>								<b>312</b>
J	Ibu Rumah Tangga						56		56	56
	Tidak Bekerja							28	28	28
<b>J Total</b>							<b>56</b>	<b>28</b>	<b>84</b>	<b>84</b>
M	Dibawah Umur						8		8	8
	Ibu Rumah Tangga	111				111	1787	8	1796	1909
	Mahasiswa						8		8	8
	Pegawai Negeri	332		14		346	194	32	226	572
	Pegawai Swasta	1012	22	24		1058	122	25	1177	1205
	Pekerja Lepas	23				23				23
	Pelajar	2				2				2
	Pensiunan	277		7		284	87		87	371
	Petani						4		4	4
Tidak Bekerja	334				334	90		90	424	
Tidak Tahu	74				74	24		24	98	
<b>M Total</b>		<b>2165</b>	<b>22</b>	<b>45</b>	<b>12</b>	<b>2232</b>	<b>2324</b>	<b>8</b>	<b>60</b>	<b>2392</b>
<b>Grand Total</b>		<b>4340</b>	<b>41</b>	<b>105</b>	<b>12</b>	<b>4498</b>	<b>3211</b>	<b>8</b>	<b>109</b>	<b>3328</b>



Gambar 5.29. Hasil pivot table pada Sistem Aplikasi setelah di Roll Up

Menu View Table digunakan untuk menampilkan data dalam tabel-tabel yang terdapat pada *database* Data\_Warehouse. Gambar 5.30 adalah tampilan *form* jika *user* menekan tombol “Show Table” untuk menampilkan isi tabel Fakta Perawatan . Pada *combo box* tabel dimensi, tabel dimensi yang muncul adalah

dimensi yang terkait dengan fakta yang dipilih sesuai dengan desain *setar schema* yang telah dibuat.



The screenshot shows a software window titled "Form Lihat Tabel". On the left side, there are two dropdown menus: "Fakta Perawatan" and "Dimensi Dokter". Below these is a "Close" button. On the right side, there is a table with two columns: "DOKTER\_ID" and "DOKTER\_NO". The table contains 30 rows of data.

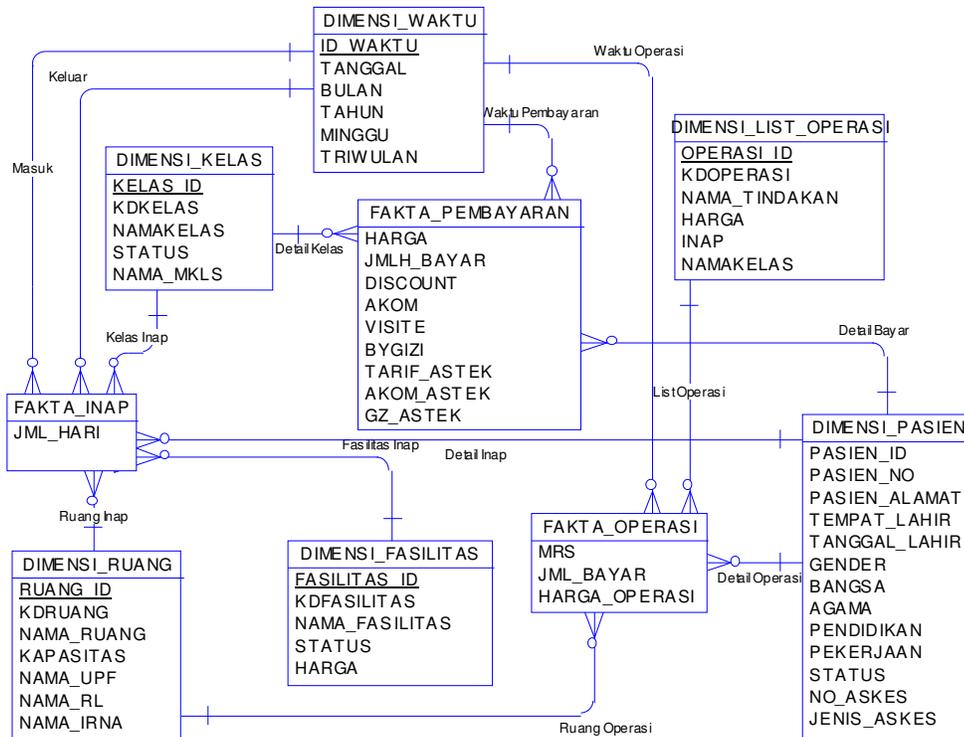
DOKTER_ID	DOKTER_NO
11	140000008
14	02030105
15	02020047
22	02050004
23	02030038
27	05120079
2	02050005
7	02030111
37	05120045
40	05120066
45	21080049
50	05120063
53	02030091
54	02020049
59	140000005
64	05120074
75	02030029
78	02020034
81	02020029
83	140000001
12	02030102
16	21080055
19	05120071
29	05120070
32	05120019
3	02050007
4	02050003
33	15240041
34	05120078
36	02020042

Gambar 5.30. Form View Table Fakta Perawatan

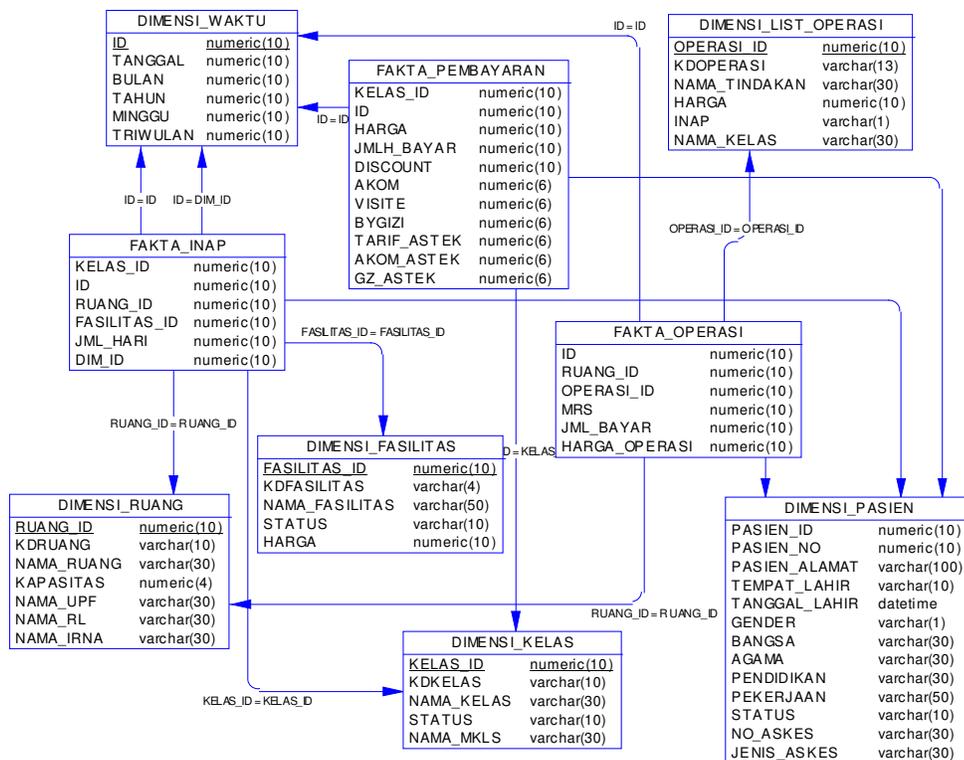
## 5.2. Prototype Data Warehouse Dan OLAP Tool Manajemen Data RSU Dr. Soetomo Surabaya.

### 5.2.1. Star Schema

Proses yang dilakukan sama dengan pembuatan star schema sebelumnya. Semua data yang terdapat pada *database* perusahaan telah melewati proses *cleansing* dan proses transformasi ke dalam *database star schema*. Proses *cleansing* ini berguna untuk menghilangkan data-data yang kurang *valid* dan membuat data-data yang ditransformasi menjadi lebih konsisten. Gambar 5.31 dan Gambar 5.32 adalah hasil perancangan *star schema* secara keseluruhan baik itu secara *conceptual design* maupun *physical design*.



Gambar 5.31. Desain Keseluruhan Star Schema (Conceptual Design)



Gambar 5.32. Desain Keseluruhan Star Schema (Physical Design)

Berikut ini adalah keterangan dari masing-masing tabel yang terdapat dalam *database star schema*:

#### 5.2.1.2 Tabel DIMENSI\_PASIEN

Tabel ini berisi data tentang detil pasien yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari *database* RSDBA, yaitu dari tabel PASIEN. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.14. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.14. Tabel DIMENSI\_PASIEN

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>PASIEN_ID</u> *	NUMBER	10	Menyimpan id pasien dari pasien
PASIEN_NO	VARCHAR2	10	Menyimpan nomor identitas pasien
PASIEN_ALAMAT	VARCHAR2	100	Menyimpan alamat pasien
TEMPAT_LAHIR	VARCHAR2	100	Menyimpan kota kelahiran pasien
TANGGAL_LAHIR	DATE		Menyimpan tanggal lahir pasien
GENDER	VARCHAR2	2	Menyimpan gender pasien
BANGSA	VARCHAR2	50	Menyimpan bangsa pasien
AGAMA	VARCHAR2	50	Menyimpan agama pasien
PENDIDIKAN	VARCHAR2	50	Menyimpan pendidikan pasien
PEKERJAAN	VARCHAR2	50	Menyimpan pekerjaan pasien
STATUS	VARCHAR2	50	Menyimpan status nikah pasien
NO_ASKES	VARCHAR2	50	Menyimpan nomor ASKES pasien
JENIS_ASKES	VARCHAR2	50	Menyimpan jenis ASKES pasien

#### 5.2.1.3 Tabel DIMENSI\_WAKTU

Tabel ini berisi data tentang detil waktu masuk dan keluarnya pasien yang terdapat dalam RSUD Dr. Soetomo Surabaya. Data yang didapatkan dalam tabel ini berasal dari *database* RSDBA, yaitu dari tabel PASIEN\_IK. *Field* dari tabel ini

beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.15. Nama *field* yang bertanda \* memiliki arti primary key.

Tabel 5.15. Tabel DIMENSI\_WAKTU

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>WAKTU_ID</u> *	NUMBER	10	Menyimpan id waktu
TANGGAL	NUMBER	10	Menyimpan tanggal
BULAN	NUMBER	10	Menyimpan bulan
TAHUN	NUMBER	10	Menyimpan tahun
KUARTER	NUMBER	10	Menyimpan kuartar
MINGGU	NUMBER	10	Menyimpan minggu

#### 5.2.1.4 Tabel DIMENSI\_FASILITAS

Tabel ini berisi data tentang fasilitas yang tersedia. Data yang didapatkan dalam tabel ini berasal dari tabel FAST. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.16. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.16. Tabel DIMENSI\_FASILITAS

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>FASILITAS_ID</u> *	NUMERIC	10	Menyimpan id fasilitas (primary key)
KDFASILITAS	VARCHAR	4	Menyimpan kode fasilitas
NAMA_FASILITAS	VARCHAR	50	Menyimpan nama fasilitas
STATUS	VARCHAR	10	Menyimpan status fasilitas
HARGA	NUMERIC	10	Menyimpan harga fasilitas

#### 5.2.1.5 Tabel DIMENSI\_RUANG

Tabel ini berisi data tentang jenis ruang yang ada. Data yang didapatkan dalam tabel ini berasal dari tabel RUANG, RL, UPFA, IRNA. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.17. Nama *field* yang bertanda \* memiliki arti *primary key*.

Tabel 5.17. Tabel DIMENSI\_RUANG

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>RUANG_ID*</u>	NUMERIC	10	Menyimpan id ruang (primary key)
KDRUANG	VARCHAR	10	Menyimpan kode ruang
NAMA_RUANG	VARCHAR	30	Menyimpan nama ruang
KAPASITAS	NUMERIC	4	Menyimpan kapasitas ruang
NAMA_UPF	VARCHAR	30	Menyimpan nama upf
NAMA_RL	VARCHAR	30	Menyimpan nama rl
NAMA_IRNA	VARCHAR	30	Menyimpan nama irna

## 5.2.1.6 Tabel DIMENSI\_KELAS

Tabel ini berisi data tentang jenis kelas yang ada. Data yang didapatkan dalam tabel ini berasal dari tabel KELAS dan MKLS. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.18. Nama *field* yang bertanda \* memiliki arti primary key.

Tabel 5.18. Tabel DIMENSI\_KELAS

<b>Nama Field</b>	<b>Tipe Data</b>	<b>Ukuran</b>	<b>Keterangan</b>
<u>KELAS_ID*</u>	NUMERIC	10	Menyimpan id kelas
KDKELAS	VARCHAR	10	Menyimpan kode kelas
NAMA_KELAS	VARCHAR	30	Menyimpan nama kelas
STATUS	VARCHAR	10	Menyimpan status kelas
NAMA_MKLS	VARCHAR	30	Menyimpan nama mkls dari kelas

## 5.2.1.7 Tabel DIMENSI\_LIST\_OPERASI

Tabel ini berisi data tentang data daftar tindakan operasi yang ada. Data yang didapatkan dalam tabel ini berasal dari tabel LISTTIND dan RECOPR. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.19. Nama *field* yang bertanda \* memiliki arti primary key.

Tabel 5.19. Tabel DIMENSI\_LIST\_OPERASI

Nama Field	Type Data	Ukuran	Keterangan
OPERASI_ID	NUMERIC	10	Menyimpan id operasi (primary key)
KDOPERASI	VARCHAR	13	Menyimpan kode operasi
NAMA_TINDAK AKAN	VARCHAR	30	Menyimpan nama tindakan operasi
HARGA	NUMERIC	10	Menyimpan harga operasi
INAP	VARCHAR	1	Menyimpan status inap/tidak
NAMA_KELAS	VARCHAR	30	Menyimpan nama kelas untuk operasi

## 5.2.1.8 Tabel FAKTA\_OPERASI

Tabel ini berisi data tentang fakta tentang fakta tindakan operasi pada pasien di rumah sakit. Tabel ini berelasi dengan tabel DIMENSI\_PASIEN, tabel DIMENSI\_WAKTU, tabel DIMENSI\_LIST\_OPERASI, tabel DIMENSI\_RUANG. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.20. Nama *field* yang bertanda # memiliki arti *foreign key*.

Tabel 5.20. Tabel FAKTA\_LIST\_OPERASI

Nama Field	Type Data	Ukuran	Keterangan
ID_WAKTU#	NUMERIC	10	Menyimpan id waktu terjadinya proses operasi
OPERASI_ID#	NUMERIC	10	Menyimpan id jenis operasi yang digunakan pada proses operasi
PASIEN_ID#	NUMERIC	10	Menyimpan id pasien yang di operasi
RUANG_ID#	NUMERIC	10	Menyimpan id ruang yang dipakai dalam proses operasi
MRS	NUMERIC	10	Menyimpan status mrs pada saat proses operasi
JML_BAYAR	NUMERIC	10	Menyimpan jumlah harga yang dibayar
HARGA_OPERASI	NUMERIC	10	Menyimpan harga dari tindakan operasi yang dilakukan

## 5.2.1.9 Tabel FAKTA\_INAP

Tabel ini berisi data tentang fakta tentang fakta inap di rumah sakit. Tabel ini berelasi dengan tabel DIMENSI\_PASIEN, tabel DIMENSI\_WAKTU, tabel

DIMENSI\_KELAS, tabel DIMENSI\_FASILITAS. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.21. Nama *field* yang bertanda # memiliki arti *foreign key*.

Tabel 5.21. Tabel FAKTA\_INAP

Nama Field	Tipe Data	Ukuran	Keterangan
PASIEN_ID#	NUMERIC	10	Menyimpan id pasien yang menginap di rumah sakit
KELAS_ID#	NUMERIC	10	Menyimpan id kelas yang dipakai menginap
ID_WAKTU_MS K#	NUMERIC	10	Menyimpan id waktu saat pasien masuk rumah sakit
RUANG_ID#	NUMERIC	10	Menyimpan id ruang yang dipakai menginap
FASILITAS_ID#	NUMERIC	10	Menyimpan id fasilitas yang dipakai saat menginap
ID_WAKTU_KLR #	NUMERIC	10	Menyimpan id waktu saat pasien keluar rumah sakit
JML_HARI	NUMERIC	10	Menyimpan jumlah hari pasien menginap di rumah sakit

#### 5.2.1.10 Tabel FAKTA\_PEMBAYARAN

Tabel ini berisi data tentang fakta tentang fakta transaksi di rumah sakit. Tabel ini berelasi dengan tabel DIMENSI\_PASIEN, tabel DIMENSI\_WAKTU, dan tabel DIMENSI\_KELAS. *Field* dari tabel ini beserta tipe data, ukuran, dan keterangannya dapat dilihat pada Tabel 5.22. Nama *field* yang bertanda # memiliki arti *foreign key*.

Tabel 5.22. Tabel FAKTA\_PEMBAYARAN

Nama Field	Tipe Data	Ukuran	Keterangan
KELAS_ID#	NUMERIC	10	Menyimpan id kelas yang dipakai pasien
ID_WAKTU#	NUMERIC	10	Menyimpan id waktu terjadinya proses pembayaran oleh pasien
PASIEN_ID#	NUMERIC	10	Menyimpan id pasien yang melakukan pembayaran
LOKET_ID	NUMERIC	10	Menyimpan id loket tempat membayar
HARGA	NUMERIC	10	Menyimpan harga dibayar
JMLH_BAYAR	NUMERIC	10	Menyimpan jumlah harga yang dibayar pasien

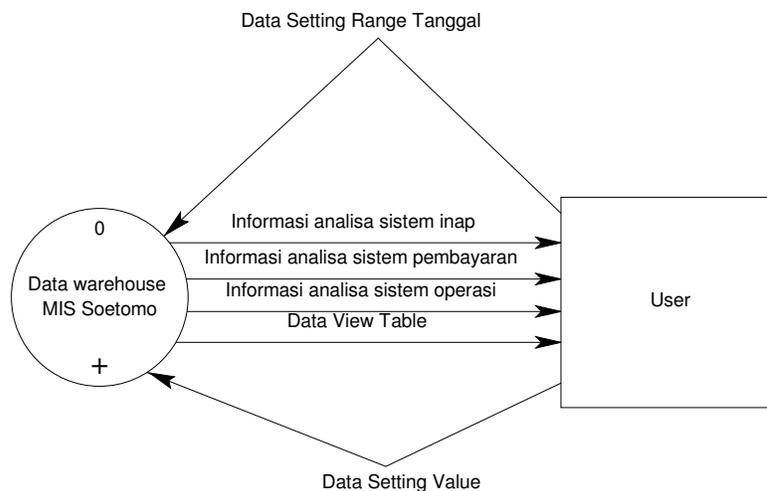
DISCOUNT	NUMERIC	10	Menyimpan jumlah diskon yang didapat pasien sesuai askes
AKOM	NUMERIC	6	Menyimpan biaya akomodasi
VISITE	NUMERIC	6	Menyimpan biaya visite
BYGIZI	NUMERIC	6	Menyimpan biaya bygizi
TARIF_ASTEK	NUMERIC	6	Menyimpan biaya astek
AKOM_ASTEK	NUMERIC	6	Menyimpan biaya akomodasi astek
GZ_ASTEK	NUMERIC	6	Menyimpan biaya gizi astek

### 5.2.2. Data Flow Diagram OLAP

Penjelasan mengenai *data flow diagram* (DFD) dari prototipe OLAP untuk *data warehouse* manajemen data RSU Dr. Soetomo adalah sebagai berikut:

#### 5.2.2.1 Context Diagram

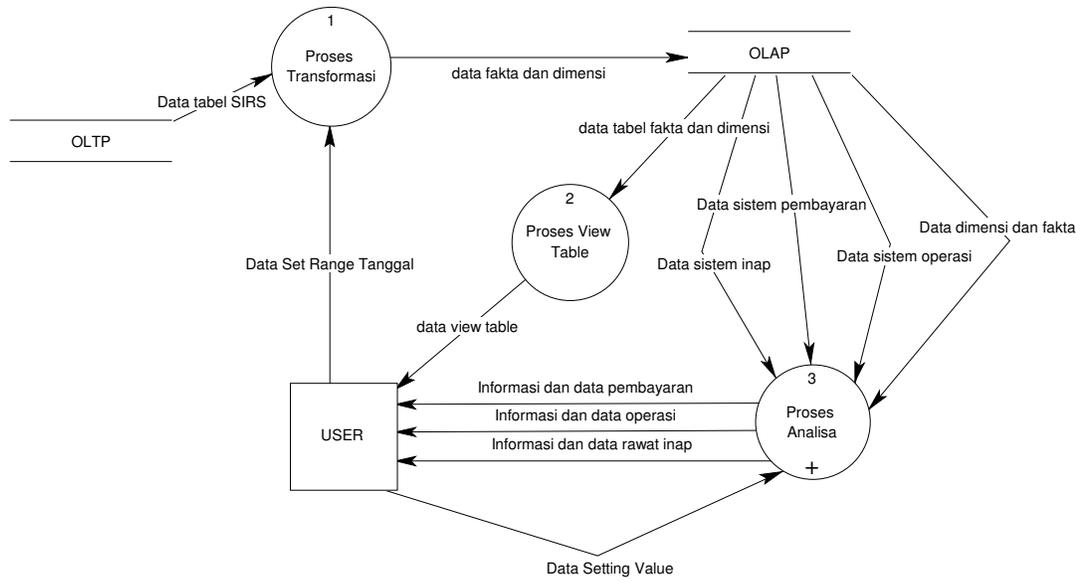
Gambar 5.33 adalah *Context Diagram*, dimana terdapat suatu proses yaitu OLAP pada rumah sakit. Dalam diagram terdapat *entity user* yang merupakan sumber data.



Gambar 5.33. Context Diagram OLAP Manajemen Data

#### 5.2.2.2 DFD Level 0

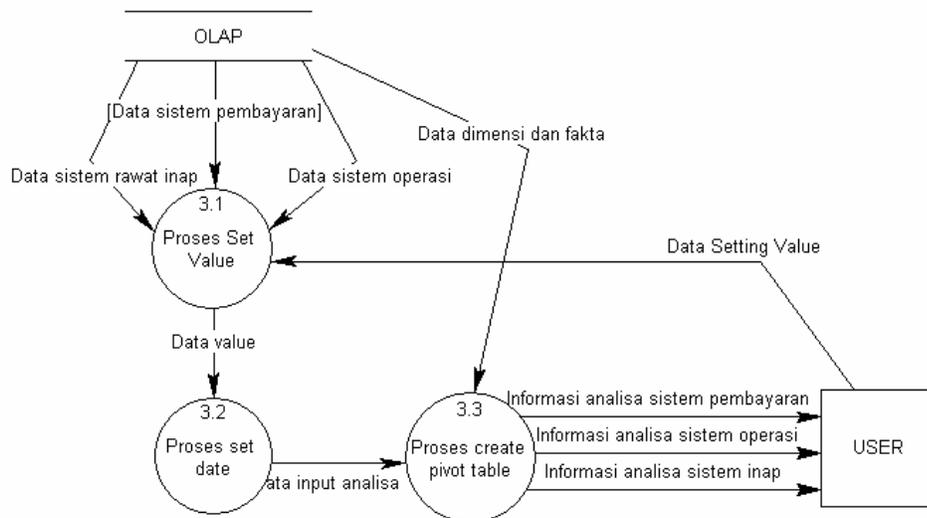
Gambar 5.34 adalah DFD *Level 0*. Pada *level* ini terdapat proses transformasi, analisis dan *view table*.



Gambar 5.34. DFD Level 0 OLAP Manajemen Data

### 5.2.2.3 DFD Level 1

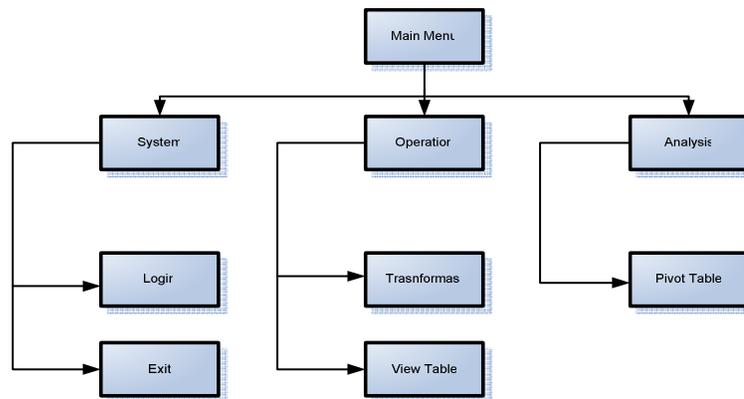
Gambar 5.35 adalah gambar DFD *level 1*. Pada setiap proses analisis digunakan data dari tabel fakta dan tabel dimensi untuk menghasilkan laporan untuk *user*.



Gambar 5.35. DFD Level 1 OLAP Manajemen Data

### 5.2.3. Struktur Menu Prototipe OLAP Manajemen Data

Pada Gambar 5.36 dapat dilihat struktur menu dari prototipe aplikasi *data warehouse* dan OLAP untuk Manajemen Data di RSUD Dr. Soetomo Surabaya.

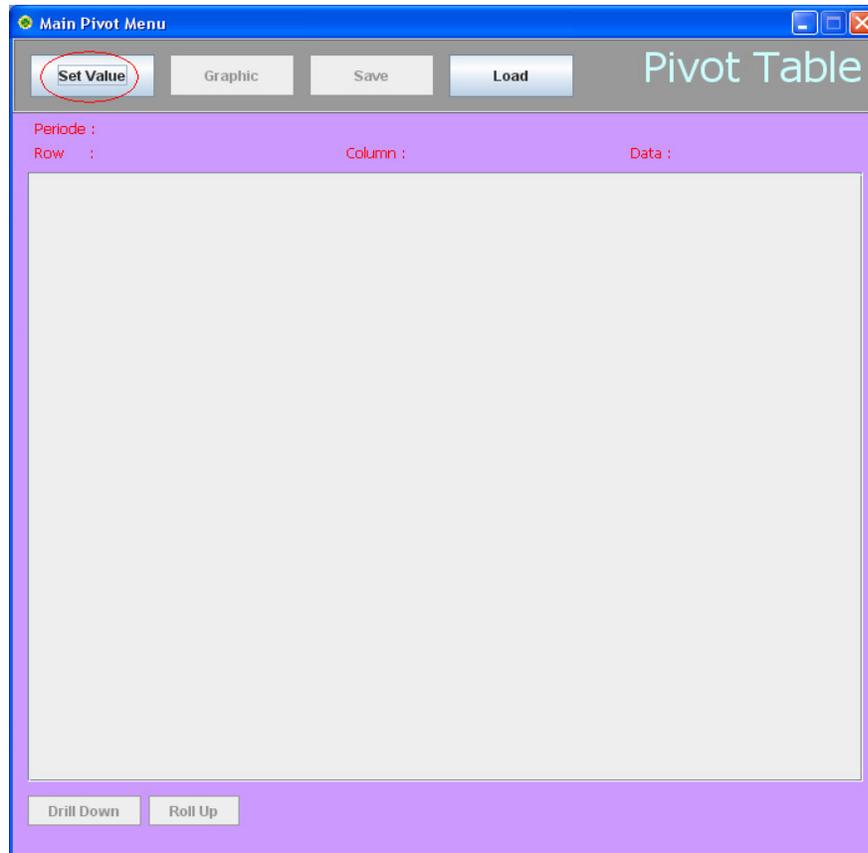


Gambar 5.36. Struktur Menu OLAP Manajemen Data

### 5.2.4. Pengujian Sistem Prototipe Data Warehouse dan OLAP untuk Manajemen Data di RSUD Dr. Soetomo

Menu Login dan menu Transformation sama dengan menu pada sub bab 5.1.4. Menu Pivot Table digunakan untuk melakukan analisis dari hasil transformasi data. Langkah pertama adalah mengatur *value* yang dianalisis dengan memilih tombol 'Set Value'.

Gambar 5.37 merupakan form utama yang muncul saat melakukan analisis data. *Setting value* dilakukan pada form Pivot yang ditunjukkan pada Gambar 5.38. Setting dilakukan dengan memilih fakta yang dianalisis dan kemudian *field* apa saja yang dijadikan filter-nya. Untuk memilih *setting value* ditunjukkan pada Gambar 5.39 hingga Gambar 5.42.



Gambar 5.37. Tampilan Menu MainPivot

Gambar 5.38 dimisalkan bahwa data yang dianalisis adalah *field* pendidikan dari Dimensi\_Pasien. Untuk menempatkan *field* tersebut sebagai baris atau kolom pada saat membuat *pivot* tabel adalah dengan memilih tombol Row, Column atau Data. Pada Gambar 5.39 ditunjukkan saat *value* dari baris, kolom dan data sudah terisi.



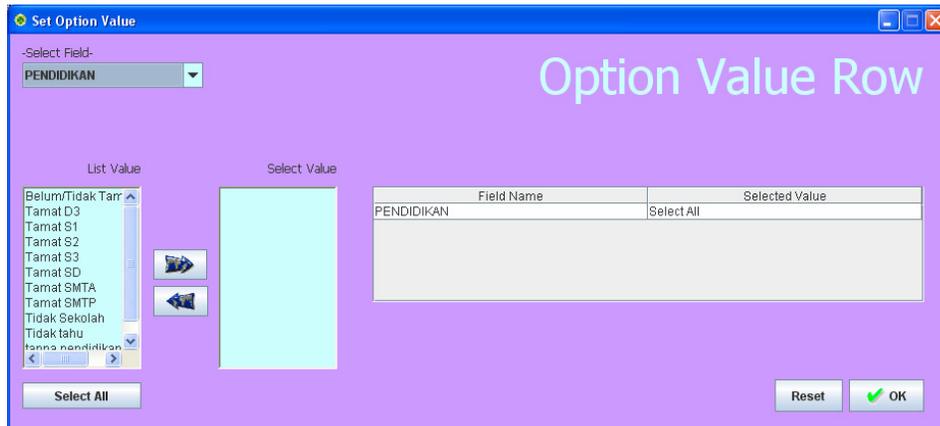
Gambar 5.38. Tampilan saat Memilih Value Pivot Table



Gambar 5.39. Tampilan saat Memilih Field Value

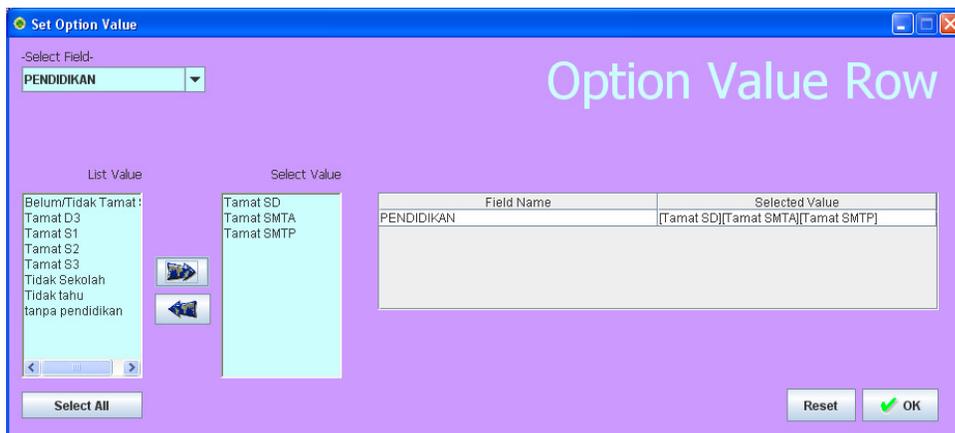
Sebelum memproses *pivot* tabel, *value* dari *field* baris dan kolom perlu di *setting* terlebih dahulu isi dari *field* yang akan dianalisa. *Setting* dilakukan dengan menekan tombol 'Set value'. Untuk memilih isi dari *field* adalah dengan memilih dahulu nama *field* pada JComboBox dan memilih isi *field* yang ditampilkan pada List seperti pada Gambar 5.40.

Pada Gambar 5.40 ditunjukkan bahwa *default* untuk *field* baris adalah 'Select All'. Data yang dipilih akan dimasukkan ke dalam List 'Select Value' pada tabel seperti pada Gambar 5.41.



Gambar 5.40. Tampilan saat Setting Value Field

Gambar 5.41 adalah tampilan *setting field* pendidikan dimana *value* yang dipilih adalah Tamat SD, Tamat SMPTA dan Tamat SMTP. Untuk *field* kolom memilih *value* dengan cara yang sama yang dimisalkan dipilih *value* untuk ruang Anak Klas 1, Anak Klas 2, dan Anak Klas 3 .



Gambar 5.41. Tampilan saat Setting Value Field

Untuk memilih tipe data yang akan dianalisa, yaitu: *Sum*, *Count*, *Average*, *Maximum*, atau *Minimum* dapat memilih pada JComboBox diatas tombol ‘Create’ seperti ditunjukkan pada Gambar 5.42. Untuk *sum*, *average*, *maximum*, dan *minimum*, hanya dapat dijalankan untuk tipe data *number*.



Gambar 5.42. Tampilan Saat Memilih Tipe Data

Gambar 5.43 menunjukkan tipe data yang dipilih adalah ‘Count’ yang berarti analisa data yang dilakukan adalah mencari *count* dari jumlah hari yang terdapat pada baris dan kolom. Untuk *filter* tanggal, *default* adalah menampilkan hasil analisa dari semua periode. Untuk memulai proses pembuatan *pivot* tabel dengan menekan tombol ‘Create’. Hasil analisis dari data yang sudah di-*setting*, ditampilkan pada *form* MainPivot seperti pada Gambar 5.43. Sedangkan untuk tipe data yang lain, akan ditunjukkan pada Gambar 5.44 hingga Gambar 5.47.

Gambar 5.43 adalah hasil analisis *pivot* tabel dengan *filter field* pendidikan pada baris, nama\_ruang pada kolom dan jumlah hari sebagai data dengan tipe data *count*.

Main Pivot Menu  
 Set Value    Graphic    Save    Load    Pivot Table

Periode : ALL PERIOD    Count of  
 Row : PENDIDIKAN    Column : NAMA\_RUANG    Data : JUMLAH\_HARI

	Anak Klas 1	Anak Klas 2	Anak Klas 3	
Tamat SD	3	4	21	
Tamat SMTA	0	2	10	
Tamat SMTP	2	0	6	
Grand Total	5.0	6.0	37.0	

Gambar 5.43. Tampilan Hasil Pivot Table dengan Tipe Data ‘Count’

Gambar 5.44 adalah hasil analisa *pivot* tabel dengan *filter field* pendidikan pada baris, nama\_ruang pada kolom dan jumlah hari sebagai data dengan tipe data *sum*.

Main Pivot Menu  
 Set Value    Graphic    Save    Load    Pivot Table

Periode : ALL PERIOD    Sum of  
 Row : PENDIDIKAN    Column : NAMA\_RUANG    Data : JUMLAH\_HARI

	Anak Klas 1	Anak Klas 2	Anak Klas 3	
Tamat SD	2	2	206	
Tamat SMTA		4	24	
Tamat SMTP	8		14	
Grand Total	10.0	6.0	244.0	

Gambar 5.44. Tampilan Hasil Pivot Table dengan Tipe Data ‘Sum’

Gambar 5.45 adalah hasil analisis *pivot* tabel dengan *filter field* pendidikan pada baris, nama\_ruang pada kolom dan jumlah hari sebagai data dengan tipe data *average*.

Main Pivot Menu  
 Set Value    Graphic    Save    Load    Pivot Table

Periode : ALL PERIOD    Average of  
 Row : PENDIDIKAN    Column : NAMA\_RUANG    Data : JUMLAH\_HARI

	Anak Klas 1	Anak Klas 2	Anak Klas 3	
Tamat SD	0.6666667	0.5	9.809524	
Tamat SMTA	0.0	2.0	2.4	
Tamat SMTP	4.0	0.0	2.3333333	
Grand Total	4.6666667	2.5	14.5428573	

Gambar 5.45. Tampilan Hasil Pivot Table dengan Tipe Data ‘Average’

Gambar 5.46 adalah hasil analisis *pivot* tabel dengan *filter field* pendidikan pada baris, nama\_ruang pada kolom dan jumlah hari sebagai data dengan tipe data *minimum*.

Main Pivot Menu  
 Set Value    Graphic    Save    Load    Pivot Table

Periode : ALL PERIOD    Minimum of  
 Row : PENDIDIKAN    Column : NAMA\_RUANG    Data : JUMLAH\_HARI

	Anak Klas 1	Anak Klas 2	Anak Klas 3	
Tamat SD	0	0	0	
Tamat SMTA		2	0	
Tamat SMTP	4		1	
Grand Total	4.0	2.0	1.0	

Gambar 5.46. Tampilan Hasil Pivot Table dengan Tipe Data ‘Minimum’

Gambar 5.47 adalah hasil analisis *pivot* tabel dengan *filter field* pendidikan pada baris, nama\_ruang pada kolom dan jumlah hari sebagai data dengan tipe data *maximum*.

**Main Pivot Menu**

Buttons: Set Value, Graphic, Save, Load

Pivot Table

Periode : ALL PERIOD Maximum of  
 Row : PENDIDIKAN Column : NAMA\_RUANG Data : JUMLAH\_HARI

	Anak Klas 1	Anak Klas 2	Anak Klas 3	
Tamat SD	1	1	62	
Tamat SMTA		2	9	
Tamat SMTP	4		5	
Grand Total	5.0	3.0	76.0	

Gambar 5.47. Tampilan Hasil Pivot Table dengan Tipe Data ‘Maximum’

Untuk memilih periode, dapat dipilih pada panel ‘Date Value’ yang terdapat pada *form* Pivot. *Filter* tanggal dipilih melalui dua komponen KzaaoCalendar sebagai *range*, setelah menentukan *range* tanggal maka untuk memasukkan dalam proses dengan menekan tombol ‘Set Date’ dimana *range* tanggal akan ditampilkan dalam JTable. Untuk mengaktifkan *filter* tanggal dilakukan dengan memilih CheckBox ‘Page Select’. Detilnya pada Gambar 5.47.

Gambar 5.48 menunjukkan *range* tanggal yang dipilih adalah tanggal 02/01/2005 hingga 05/01/2005.

**Set Pivot Table**

**Date Value**

Range Awal: 02/01/2005 s/d Range Akhir: 05/01/2005

Buttons: Set Date, Reset Date

Page Select

**Field Value**

Row: Dimensi\_Pasien.PENDIDIKAN  
 Column: Dimensi\_Ruang.NAMA\_RUANG  
 Data: Fakta\_Inap.JUMLAH\_HARI

Buttons: Set value, Set value, Data Type: Count

Buttons: Reset, Create

Message: press 'Set Value' first to modify your selected field !!!

Gambar 5.48. Tampilan Setting Date Filter

*Filter* tanggal mempengaruhi hasil pada *pivot* tabel awal seperti ditunjukkan pada Gambar 5.49. Aplikasi ini menyediakan fasilitas *Zoom Out* yang digunakan untuk mengetahui isi dari hasil *pivot* tabel pada tiap sel. *Zoom Out* dapat dilakukan dengan men-*click* pada sel yang diinginkan seperti ditunjukkan pada Gambar 5.50.

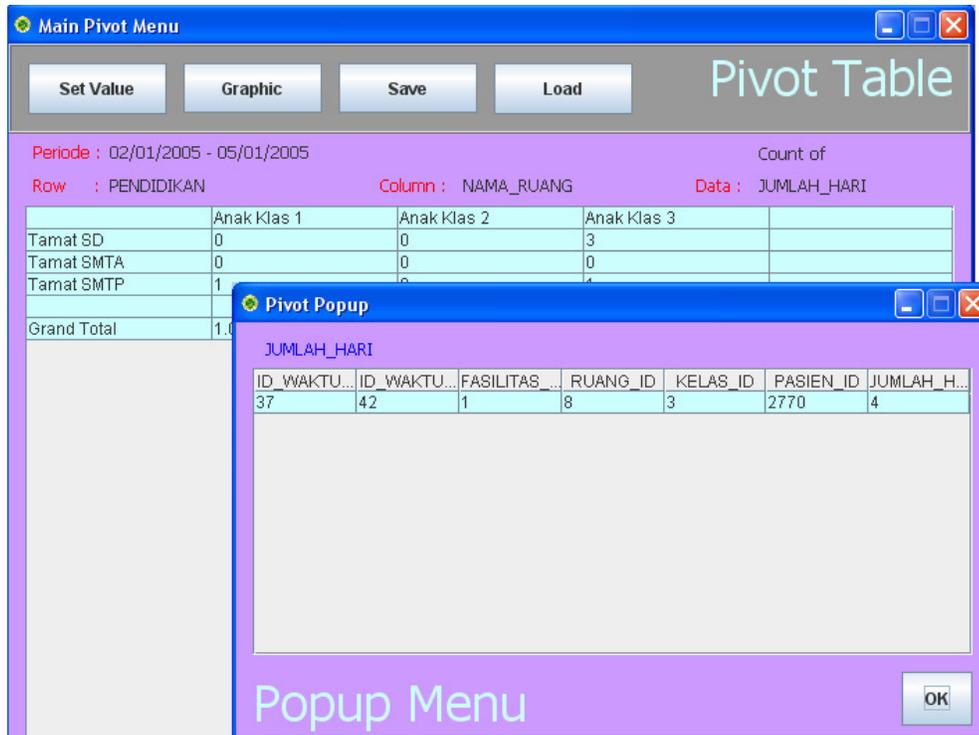
Hasil dari *pivot* tabel dapat disimpan dalam *format* .txt. Untuk menyimpan hasil analisa *pivot* table dilakukan dengan menekan tombol ‘Save’ pada panel atas *form* MainPivot. Kemudian dimasukkan nama untuk menyimpan *pivot* dalam bentuk .txt. Hal ini ditujukan apabila suatu saat akan menganalisa data yang sama maka tidak perlu melakukan *setting* data kembali tetapi tinggal menekan tombol ‘Load’ dan memilih hasil analisis *pivot* yang telah disimpan.

The screenshot shows a software window titled "Main Pivot Menu" with a "Pivot Table" section. The filters are set to "Periode : 02/01/2005 - 05/01/2005" and "Row : PENDIDIKAN". The table displays data for "Anak Klas 1", "Anak Klas 2", and "Anak Klas 3" across categories "Tamat SD", "Tamat SMTA", "Tamat SMTP", and "Grand Total".

	Anak Klas 1	Anak Klas 2	Anak Klas 3	Count of
Tamat SD	0	0	3	
Tamat SMTA	0	0	0	
Tamat SMTP	1	0	1	
Grand Total	1.0	0.0	4.0	

Gambar 5.49. Tampilan Hasil Pivot Tabel dengan Filter Tanggal

Gambar 5.50 menunjukkan hasil *Zoom Out* pada saat menekan sel pada baris 4 kolom 2 yang bernilai 1. Hasil pada menu *Zoom Out* sesuai dengan hasil yang terdapat pada sel.



Gambar 5.50. Tampilan Zoom Out pada Pivot Tabel

Analisis data dapat dilakukan secara lebih bervariasi dimana *field* pada baris dan kolom dapat diisi lebih dari satu data atau data hanya diisi pada baris saja atau kolom saja. Berikut adalah tampilan dari analisis datanya.

Gambar 5.51 menunjukkan hasil dari *pivot* tabel dimana data pada *field* baris ditambahkan satu data lagi, yaitu Nama\_Fasilitas. Pada *pivot* tabel, jika data berisi lebih dari satu, maka akan dipisahkan dengan tanda 'I'.

Gambar 5.52 menunjukkan hasil dari *pivot* tabel dimana data ditambahkan pada *field* kolom, yaitu Nama\_Fasilitas.



**Main Pivot Menu**

Buttons: Set Value, Graphic, Save, Load

**Pivot Table**

Periode : ALL PERIOD  
 Row : PENDIDIKAN  
 Column :  
 Data : JUMLAH\_HARI

			Grand Total
Belum/Tidak Tamat SD	1185		1185.0
Tamat D3	139		139.0
Tamat S1	242		242.0
Tamat S2	11		11.0
Tamat S3	4		4.0
Tamat SD	1221		1221.0
Tamat SMTA	2376		2376.0
Tamat SMTP	1907		1907.0
Tidak Sekolah	1206		1206.0
Tidak tahu	662		662.0
tanpa pendidikan	0		0.0
Grand Total	8953.0		

Gambar 5.53. Tampilan Pivot Tabel dengan Data hanya pada Baris

Gambar 5.54 menunjukkan hasil *pivot* tabel dengan data yang diisikan hanya pada kolom, yaitu pendidikan dan pada data, yaitu jumlah\_hari dengan tipe data *count*.

**Main Pivot Menu**

Buttons: Set Value, Graphic, Save, Load

**Pivot Table**

Periode : ALL PERIOD  
 Row :  
 Column : PENDIDIKAN  
 Data : JUMLAH\_HARI

	Belum/Tidak Tamat SD	Tamat D3	Tamat S1	Tamat S2
	1185	139	242	11
Grand Total	1185.0	139.0	242.0	11.0

Gambar 5.54. Tampilan Pivot Tabel dengan Data hanya pada Kolom

Gambar 5.55 menunjukkan hasil *pivot* tabel dengan 2 data pada baris, yaitu: pendidikan dan nama\_fasilitas dan pada baris, yaitu: nama\_ruang dan nama\_kelas, serta pada data, yaitu jumlah\_hari dengan tipe data *count*.

Main Pivot Menu

Pivot Table

Set Value    Graphic    Save    Load

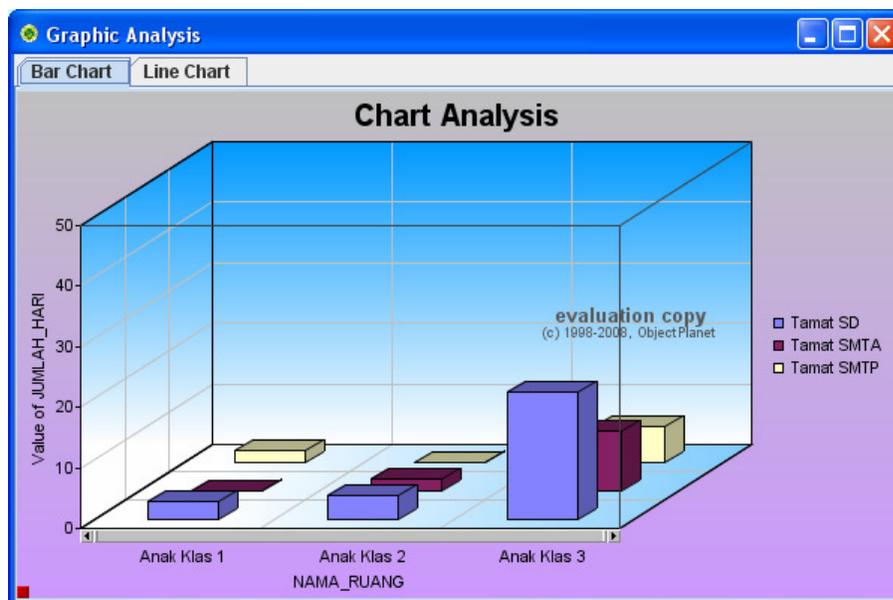
Periode : ALL PERIOD    Count of

Row : PENDIDIKAN and NAMA\_FASI...    Column : NAMA\_RUANG and NAMA\_KELAS    Data : JUMLAH\_HARI

	Anak Kelas 1 Kelas 1	Anak Kelas 1 Kelas 2	Anak Kelas 1 Kelas 3A	Anak Kelas 2 Kelas 1
Tamat SD Askeskin	0	0	0	0
Tamat SD Astek	0	0	0	0
Tamat SD GRIU	0	0	0	0
Tamat SD RSUD Dr. ...	0	0	0	0
Tamat SD Umum	3	0	0	0
Tamat SMTA Askeskin	0	0	0	0
Tamat SMTA Astek	0	0	0	0
Tamat SMTA GRIU	0	0	0	0
Tamat SMTA RSUD D...	0	0	0	0
Tamat SMTA Umum	0	0	0	0
Tamat SMTPI Askeskin	0	0	0	0
Tamat SMTPI Astek	0	0	0	0
Tamat SMTPI GRIU	0	0	0	0
Tamat SMTPI RSUD D...	0	0	0	0
Tamat SMTPI Umum	2	0	0	0
Grand Total	0.0	5.0	0.0	0.0

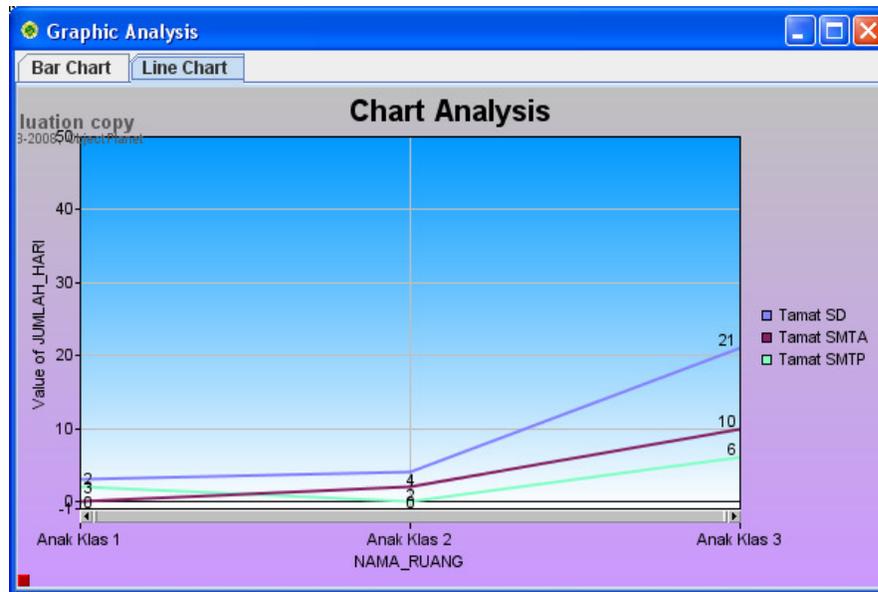
Gambar 5.55. Tampilan Pivot Tabel dengan 2 Data pada Baris dan Kolom

Gambar 5.56 adalah tampilan analisa grafik dalam bentuk *Barchart* 3D. Grafik menggambarkan *field* kolom sebagai sumbu X dan *field* baris digambarkan dengan warna balok yang berbeda dimana warna tersebut dijelaskan pada Legenda di sisi kanan grafik. Untuk memilih grafik dalam bentuk *Linechart* dengan memilih menu 'Line Chart' di bagian atas.



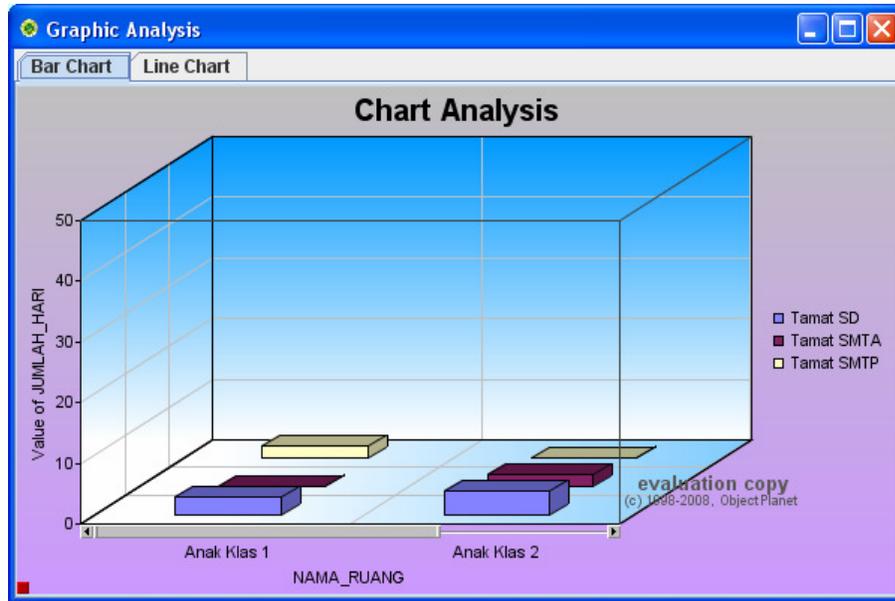
Gambar 5.56. Tampilan Hasil Analisa Grafik dalam Bentuk Barchart

Pada Gambar 5.57 grafik ditampilkan dalam bentuk *Linechart* dengan *format* seperti pada Barchart, yaitu sumbu X adalah *field* kolom dan warna garis sebagai *field* baris dengan keterangan dijelaskan pada Legenda.



Gambar 5.57. Tampilan Hasil Analisis Grafik dalam Bentuk LineChart

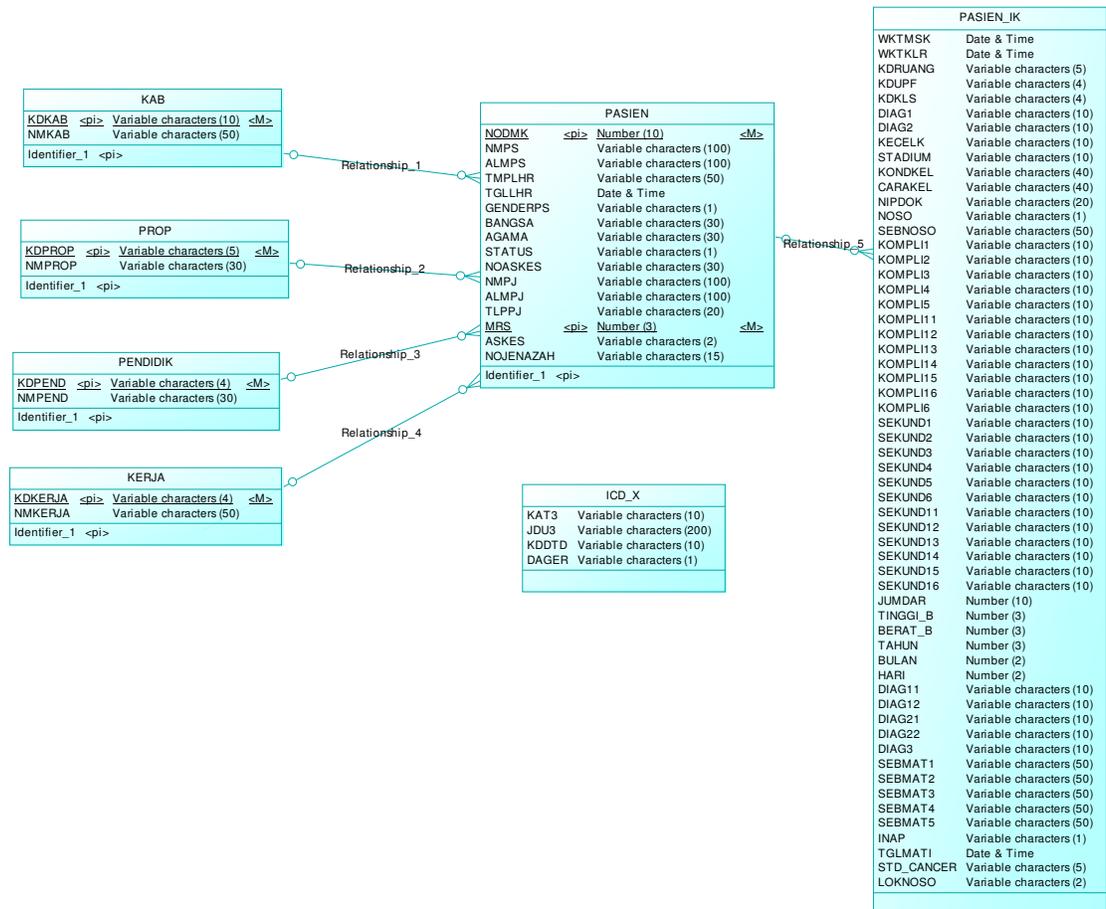
Grafik Gambar 5.58 menyediakan aplikasi untuk memudahkan *user* dapat menganalisa data dengan melakukan *scroll* pada bagian bawah grafik. Dengan *scroll*, *user* dapat melakukan analisa terhadap data yang diinginkan.



Gambar 5.58. Tampilan Hasil Scroll Grafik

### 5.3. Sub Sistem Pemetaan SIG Penyebaran Wabah Penyakit

#### 5.3.1. Conceptual Model



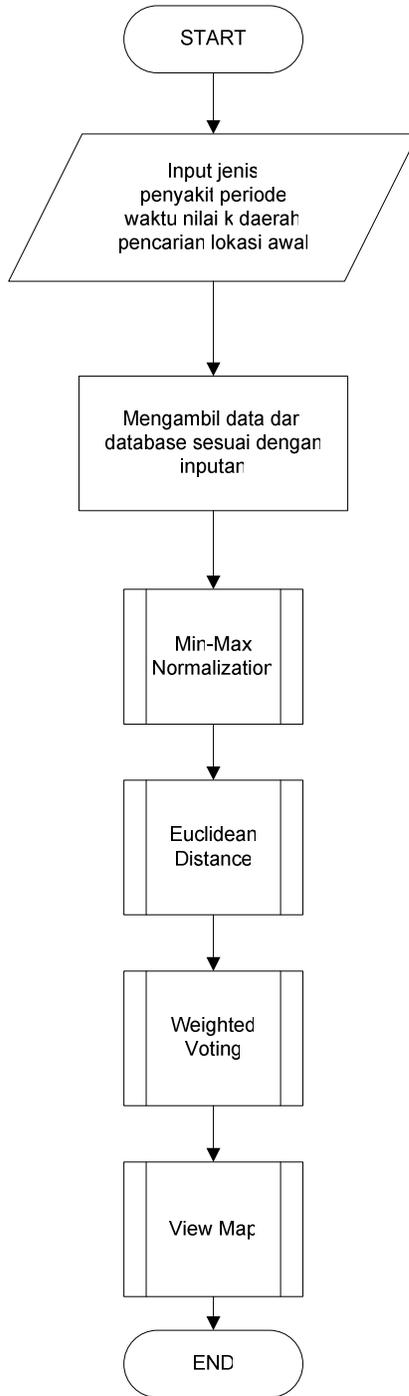
Gambar 5.59. Conceptual Model

#### 5.3.2. Algoritma *k-nearest neighbors*

Berikut ini adalah algoritma *k-nearest neighbors* yang digambarkan dalam bentuk *flowchart*.

a. Flowchart untuk prosedur utama

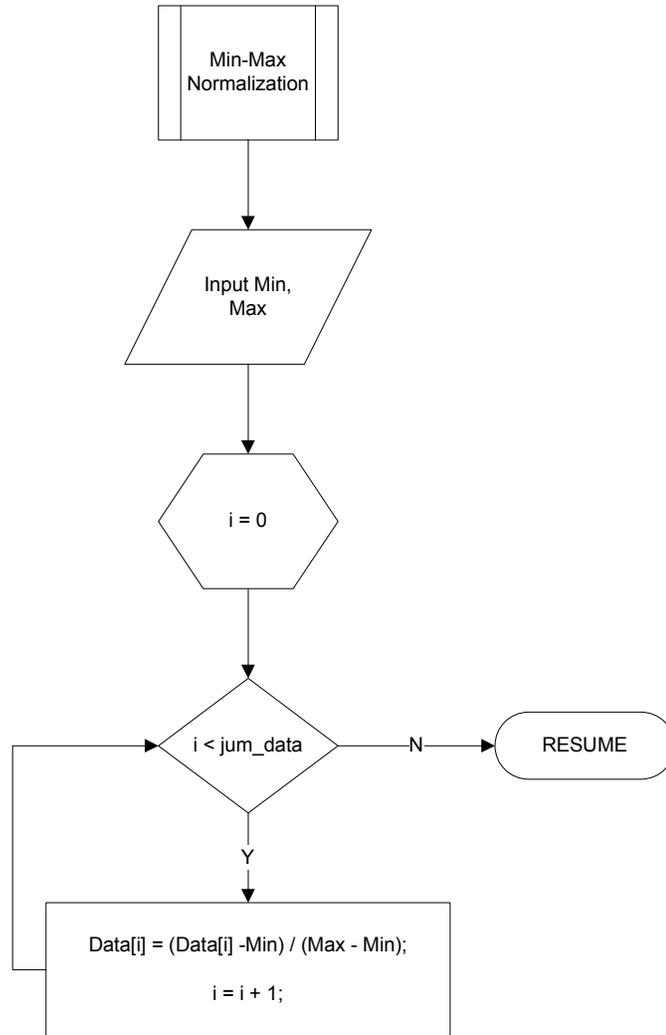
Prosedur utama pada program adalah untuk menentukan data mana yang memiliki nilai *voting* kedekatan paling tinggi dari data awal yang dipilih (Gambar 5.60). Selanjutnya, tersebut akan dipetakan pada peta *digital*.



Gambar 5.60. Flowchart Utama

b. Flowchart *Min-Max Normalization*

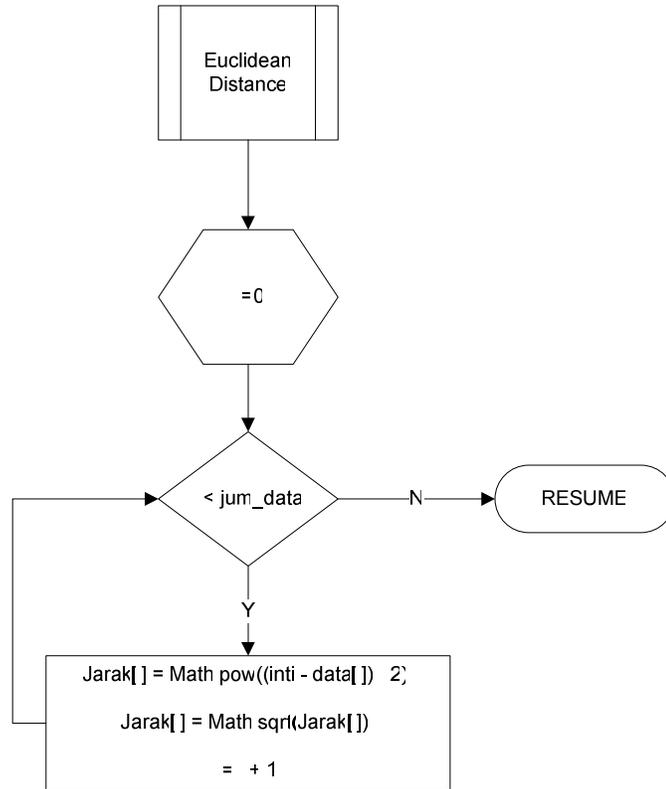
Prosedur ini digunakan untuk menormalisasi data-data yang diklasifikasi untuk digunakan dalam proses berikutnya (Gambar 5.61).



Gambar 5.61. Flowchart Prosedur Min-Max Normalization

c. Flowchart *Euclidean Distance*

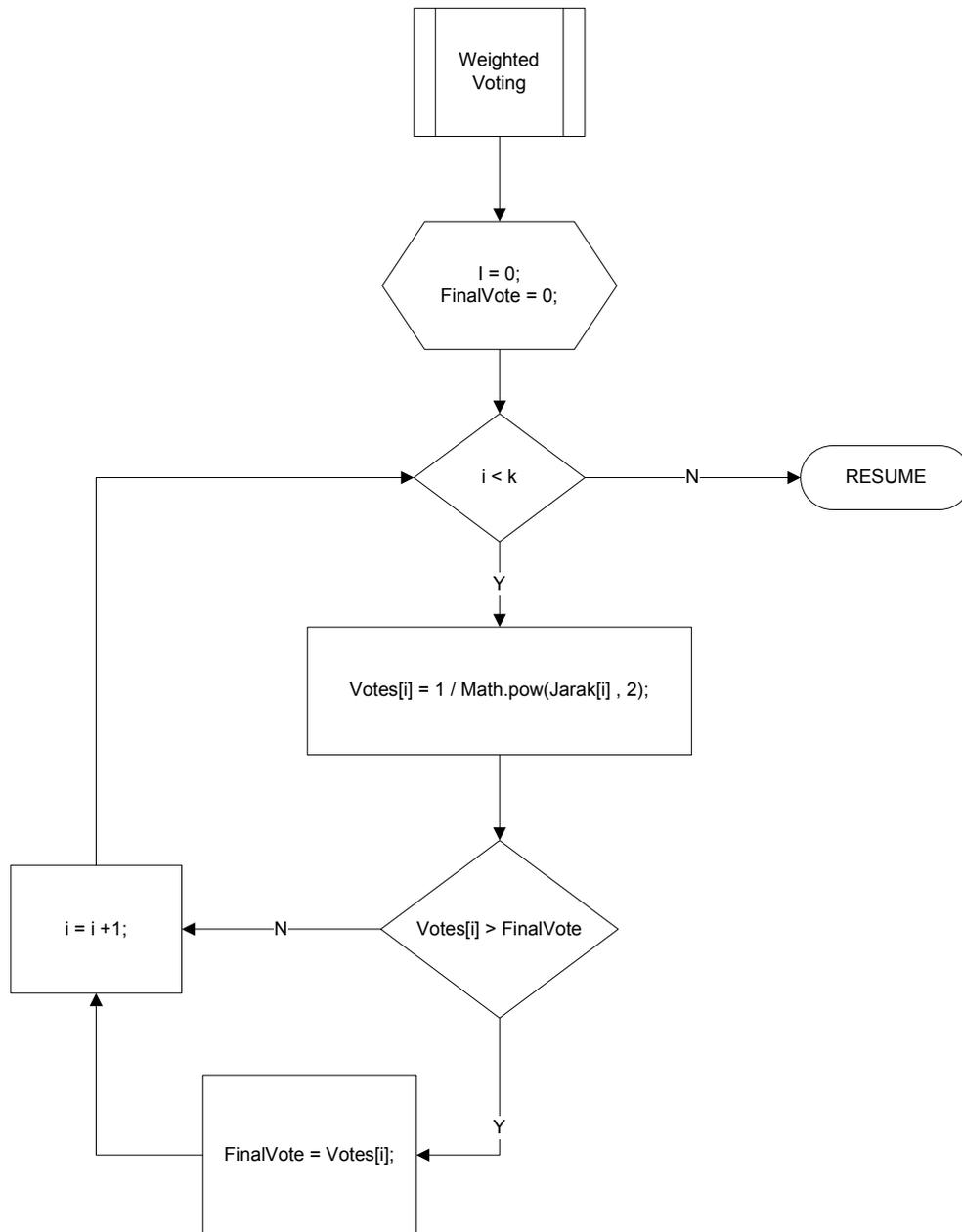
Prosedur ini untuk mendapatkan jarak antara data awal yang dipilih dengan data-data yang digunakan untuk klasifikasi (Gambar 5.62).



Gambar 5.62. Flowchart Prosedur Euclidean Distance

d. Flowchart *Weighted Voting*

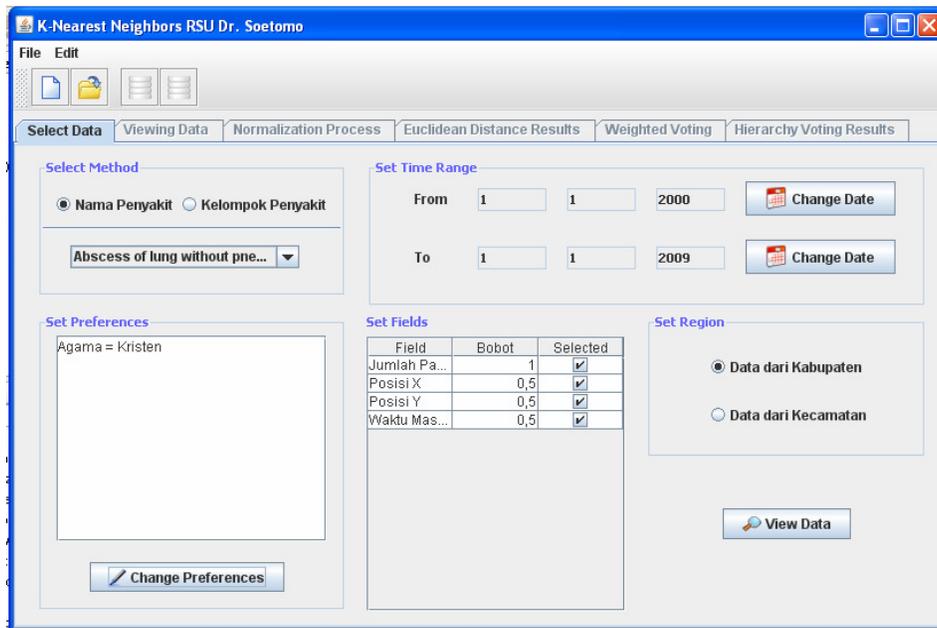
Prosedur ini digunakan untuk menghitung nilai *voting* dari jarak data sehingga dapat ditentukan data mana yang memiliki kedekatan paling besar dengan data awal yang dipilih (Gambar 5.63).



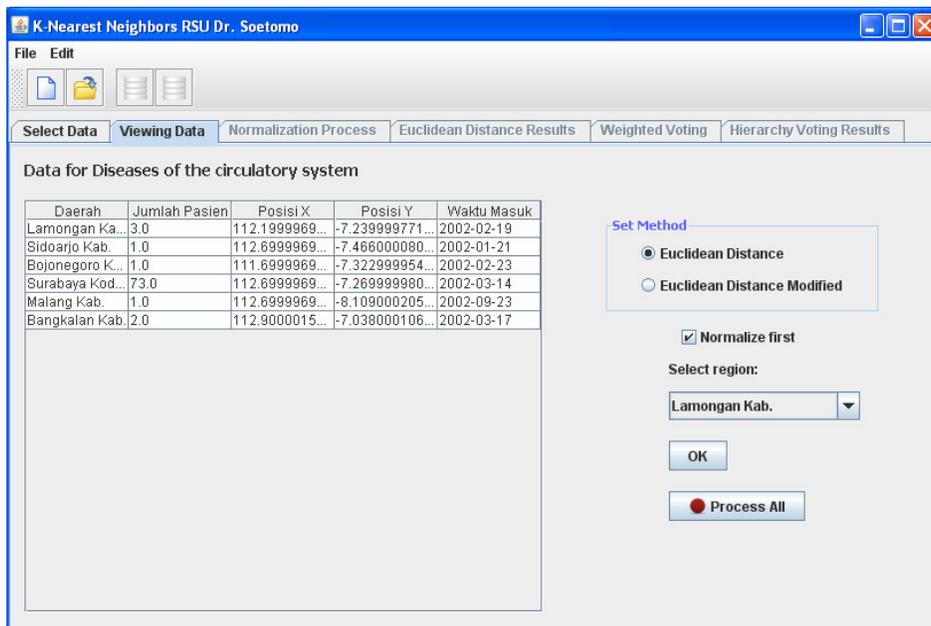
Gambar 5.63. Flowchart Prosedur Weighted Voting

### 5.3.3. Pengujian Sistem Prototipe Pemetaan SIG Penyebaran Wabah Penyakit

Sebelum melakukan pemetaan penyebaran wabah penyakit, *user* diminta untuk memilih data yang diinginkan. Sebagai contoh, pada Gambar 5.64 dipilih data berdasarkan nama penyakit abscess lung, untuk agama Kristen, pada tanggal 1 Januari 2000 – 1 Januari 2009.

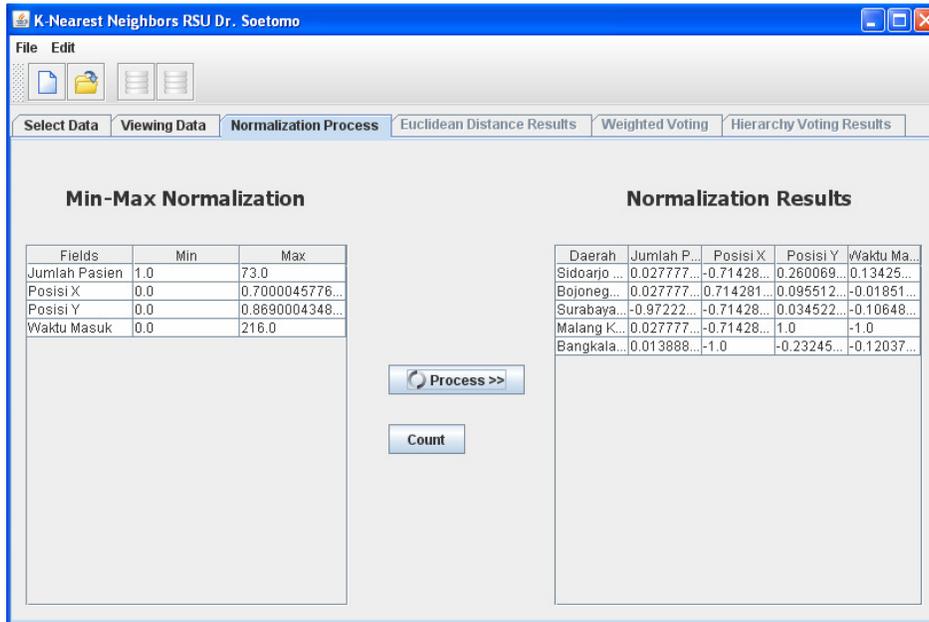


Gambar 5.64. Pemilihan Data Penyebaran Wabah Penyakit



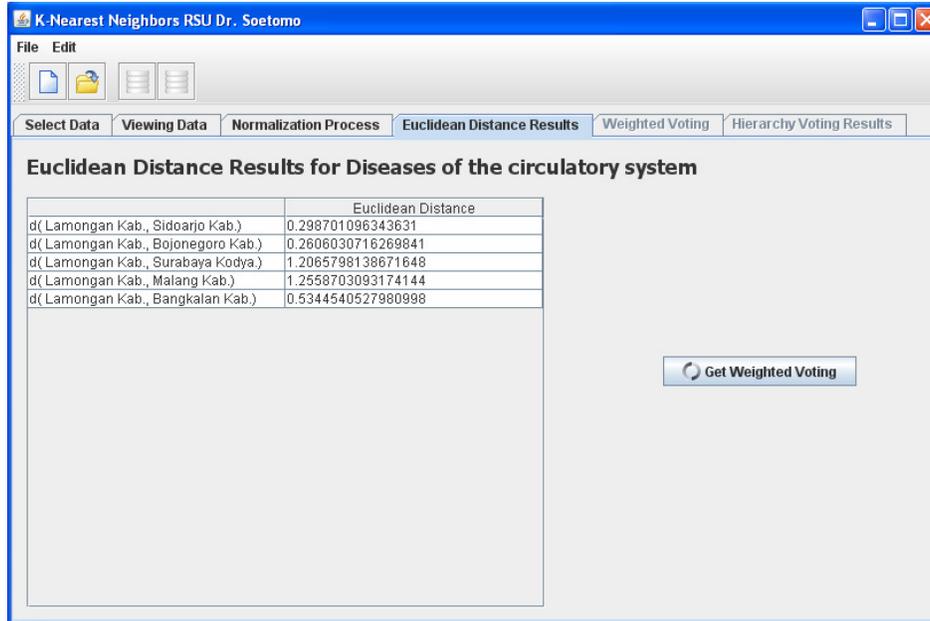
Gambar 5.65. Pemilihan Data yang Ditampilkan pada Penyebaran Wabah Penyakit

Selanjutnya dipilih cara menampilkan datanya dengan melakukan penentuan metode yaitu berdasarkan *euclidean distance* dan dipilih kabupaten Lamongan seperti Gambar 5.65. Pada menu tersebut, dipilih pula normalisasi, sehingga pada Gambar 5.66 ditampilkan hasil normalisasinya.

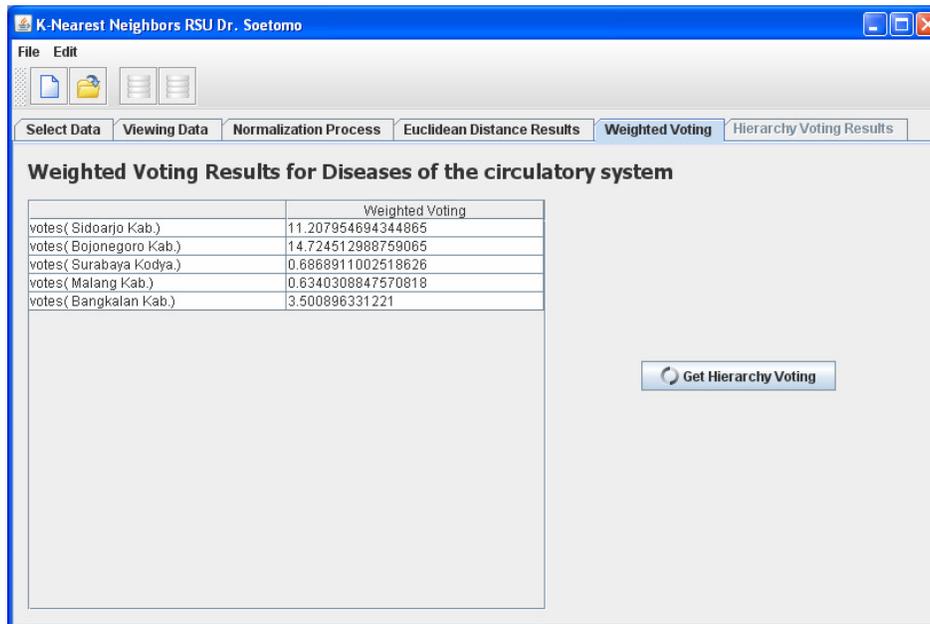


Gambar 5.66. Hasil Normalisasi pada Penyebaran Wabah Penyakit

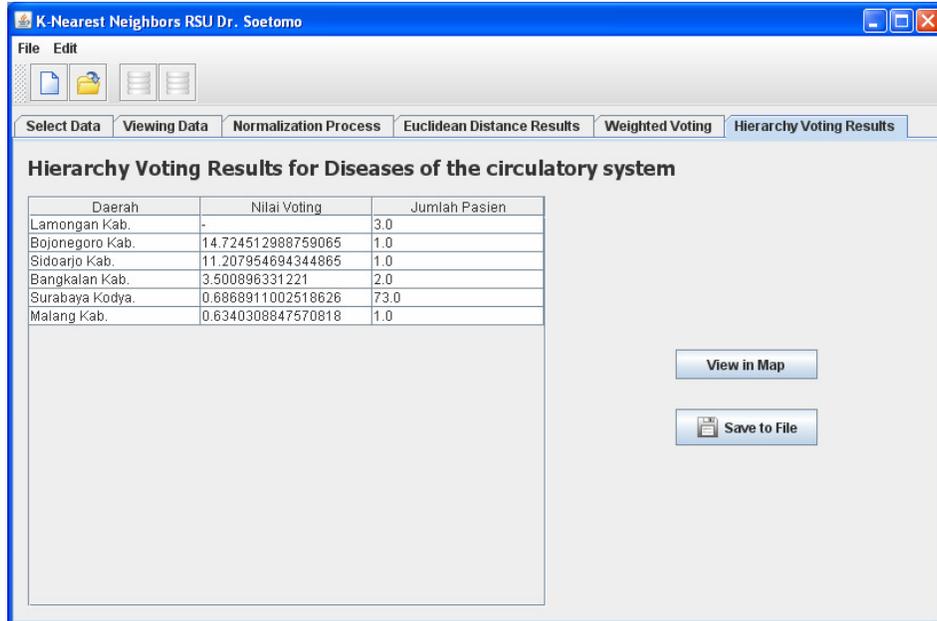
Hasil *euclidean distance* ditampilkan hanya untuk kabupaten Lamongan saja seperti Gambar 5.67. Selain itu, dihasilkan pula weighed voting yaitu kabupaten yang berada di sekitar kabupaten Lamongan. Kabupaten-kabupaten tersebut adalah Sidoarjo, Bojonegoro, Surabaya, Malang, dan Bangkalan (Gambar 5.68).



Gambar 5.67. Hasil *Euclidean Distance* pada Penyebaran Wabah Penyakit

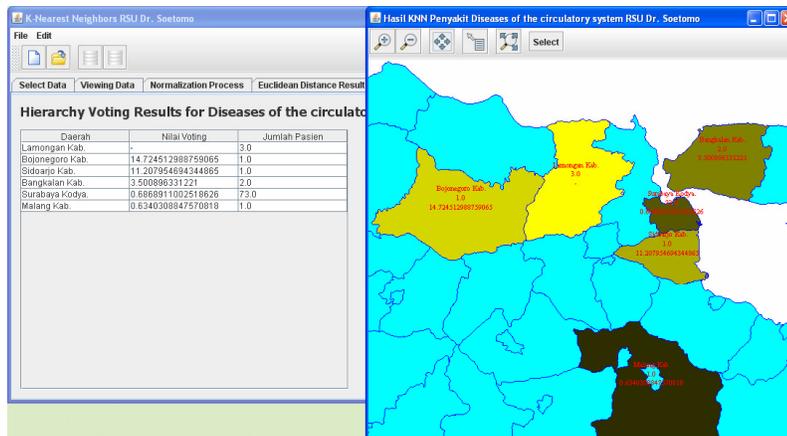


Gambar 5.68. Hasil *Weighted Voting* pada Penyebaran Wabah Penyakit



Gambar 5.69. Hirarki Hasil Voting pada Penyebaran Wabah Penyakit

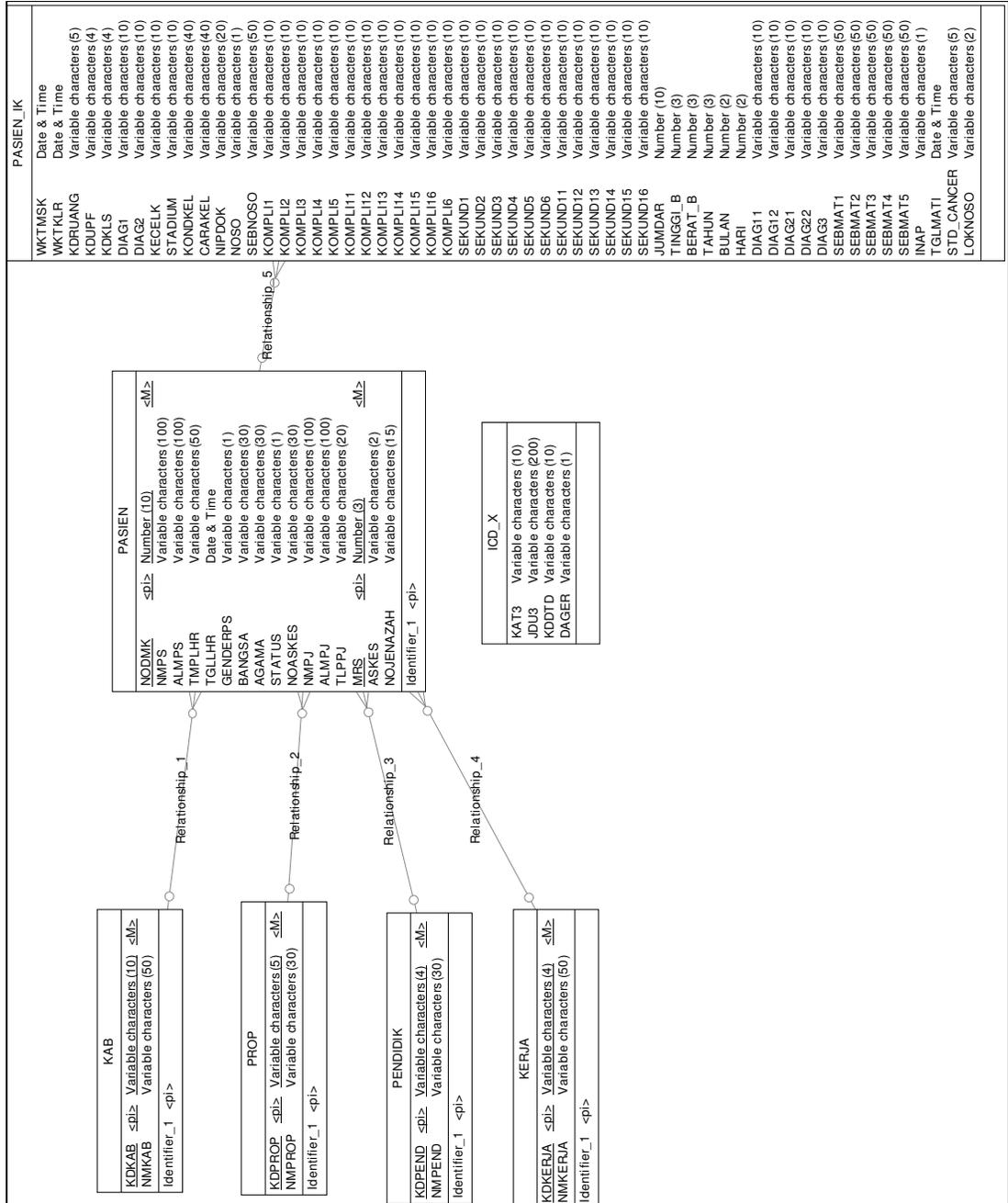
Setelah diberi bobot, maka perlu ditentukan urutan bobot dari kabupaten sekitar seperti Gambar 5.69. Hasil pemetaan dapat dilihat ditunjukkan dengan memperlihatkan hasil hirarki *voting* (Gambar 5.70).



Gambar 5.70. Hirarki Hasil Voting pada Penyebaran Wabah Penyakit

## 5.4. Sub Sistem *Sequential Mining* Untuk Mengenali Pola Sebab-Akibat Terjangkitnya Penyakit Pada Pasien

### 5.4.1. Conceptual Model

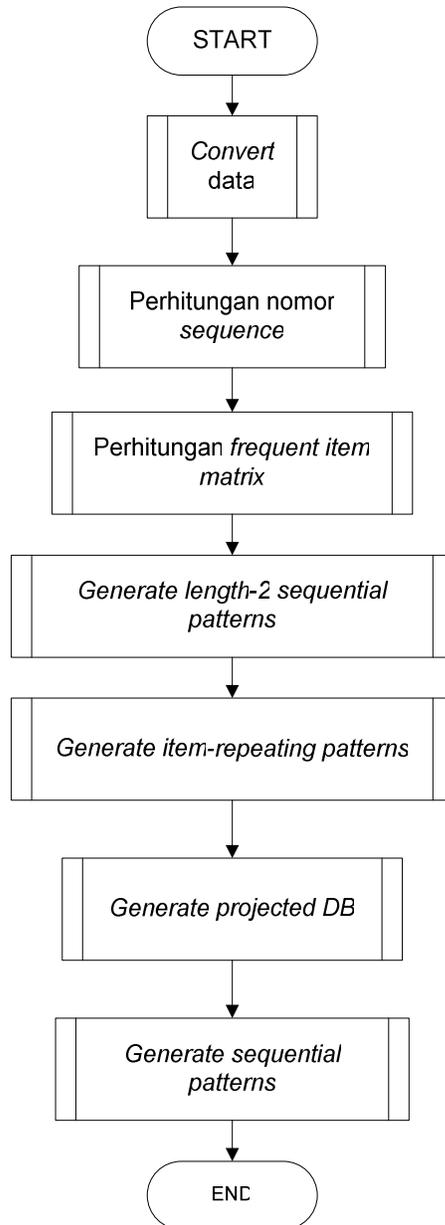


Gambar 5.71. Conceptual Model Pola Penyakit

#### 5.4.2. Algoritma FreeSpan dalam Menentukan *Sequential Patterns*

a. *Flowchart* untuk prosedur utama

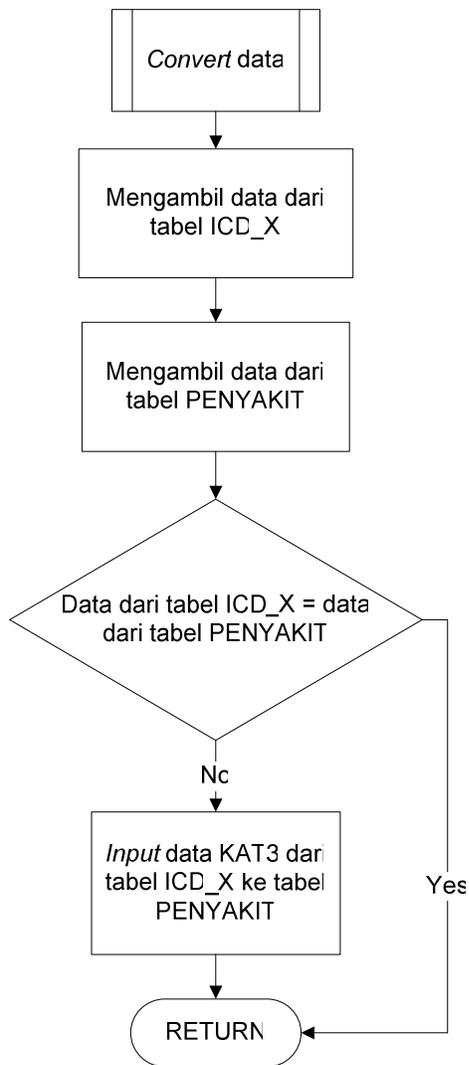
Prosedur utama pada bagian utama ini yaitu berisi proses-proses analisis yang menghasilkan *sequential patterns*, dimana awali dengan *convert data* yang dapat diartikan sebagai pemrosesan data awal (Gambar 5.72).



Gambar 5.72. Flowchart prosedur utama

b. Pemrosesan data awal

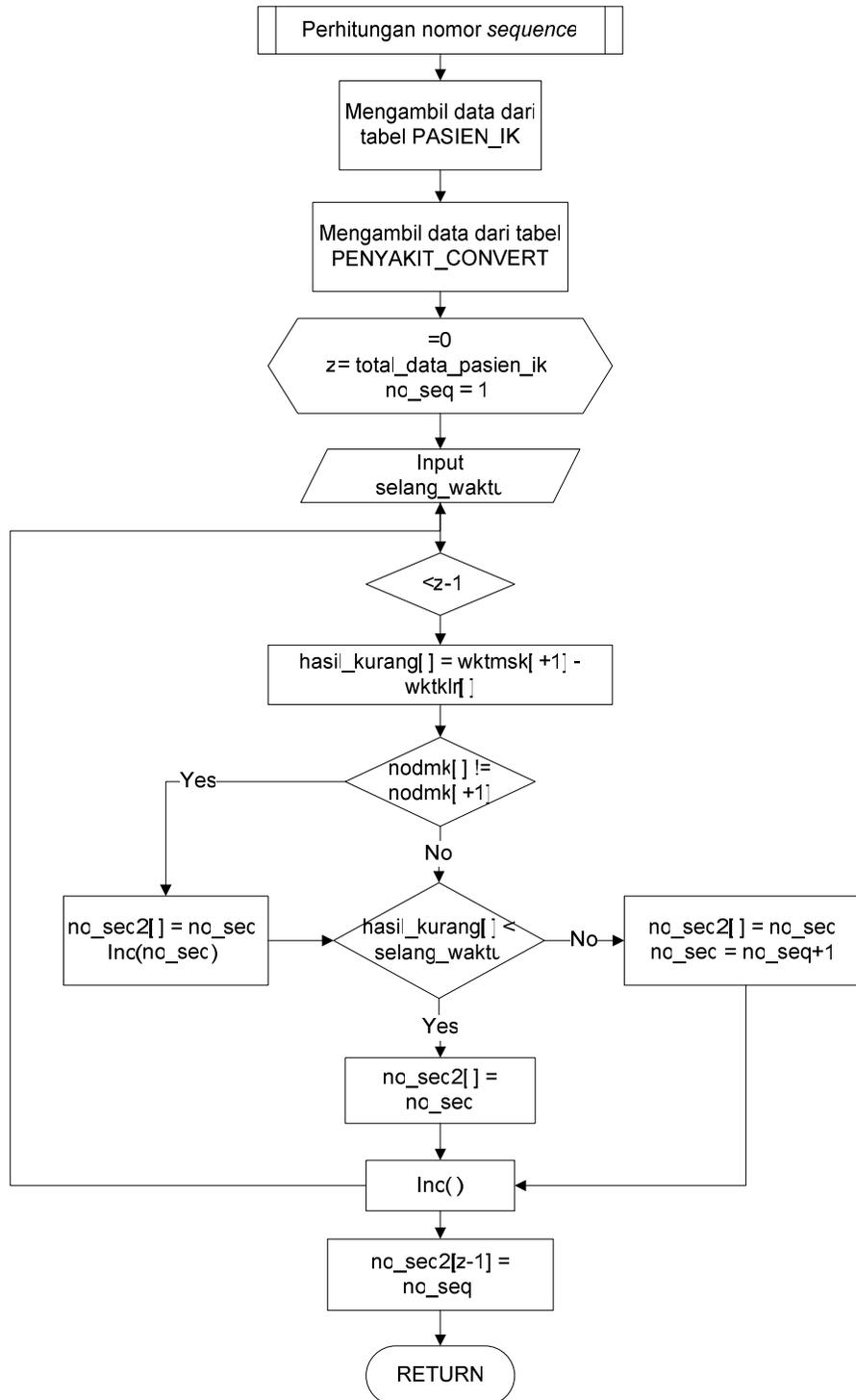
Proses pada Gambar 5.73 dapat dikatakan sebagai *convert* data, dimana pertama kali mengambil isi *field* KAT3 dari tabel ICD\_X dan mengambil isi *field* KAT3 dari tabel PENYAKIT\_CONVERT. Kemudian data tersebut dibandingkan, apabila isi *field* KAT3 dari tabel ICD\_X sama dengan isi *field* KAT3 dari tabel PENYAKIT\_CONVERT, maka data yang sama tersebut tidak dimasukkan dan berarti bahwa kode penyakit dari tabel ICD\_X telah mempunyai kode pengganti di tabel PENYAKIT\_CONVERT. Apabila isi *field* KAT3 dari tabel ICD\_X tidak sama dengan isi *field* KAT3 dari tabel PENYAKIT\_CONVERT, maka isi *field* KAT3 dari tabel ICD\_X dimasukkan ke dalam tabel PENYAKIT\_CONVERT.



Gambar 5.73. Flowchart pemrosesan data awal

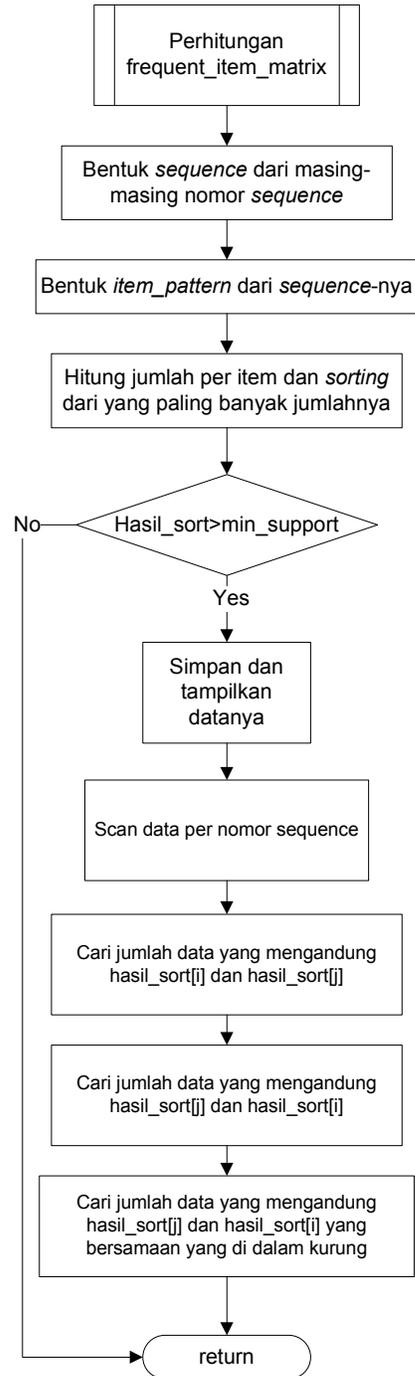
c. Pemberian nomor *sequence* pada data

Proses Gambar 5.74 merupakan perhitungan untuk memberikan nomor *sequence* pada data berdasarkan *input* selang waktu dari *user*.



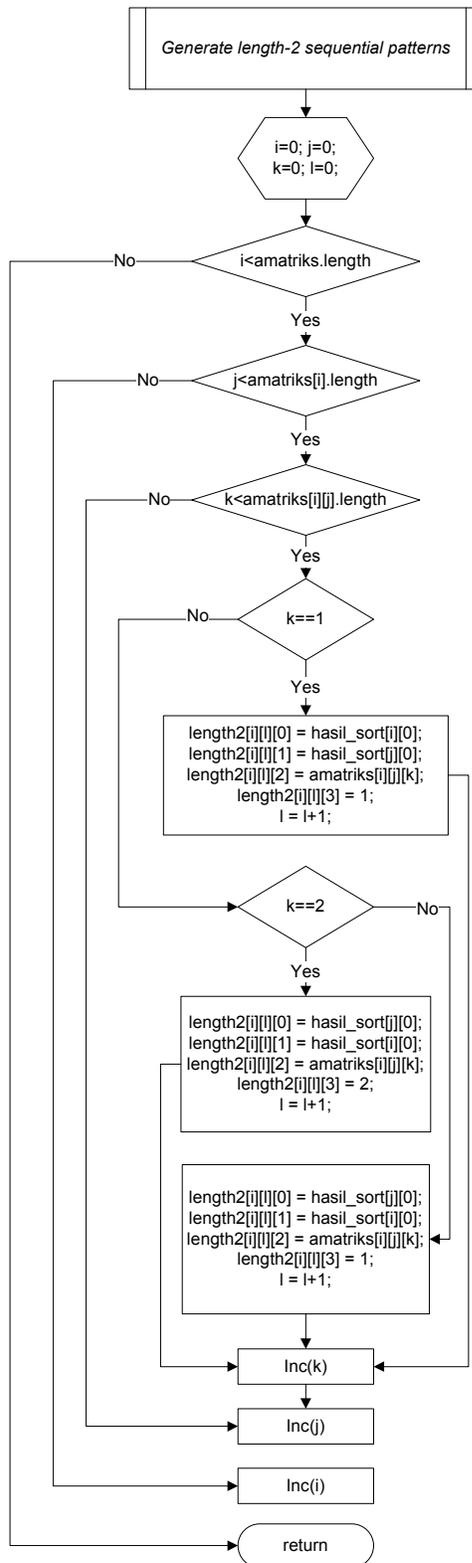
Gambar 5.74. Flowchart pemberian nomor *sequence*

d. Perhitungan *frequent item matrix*



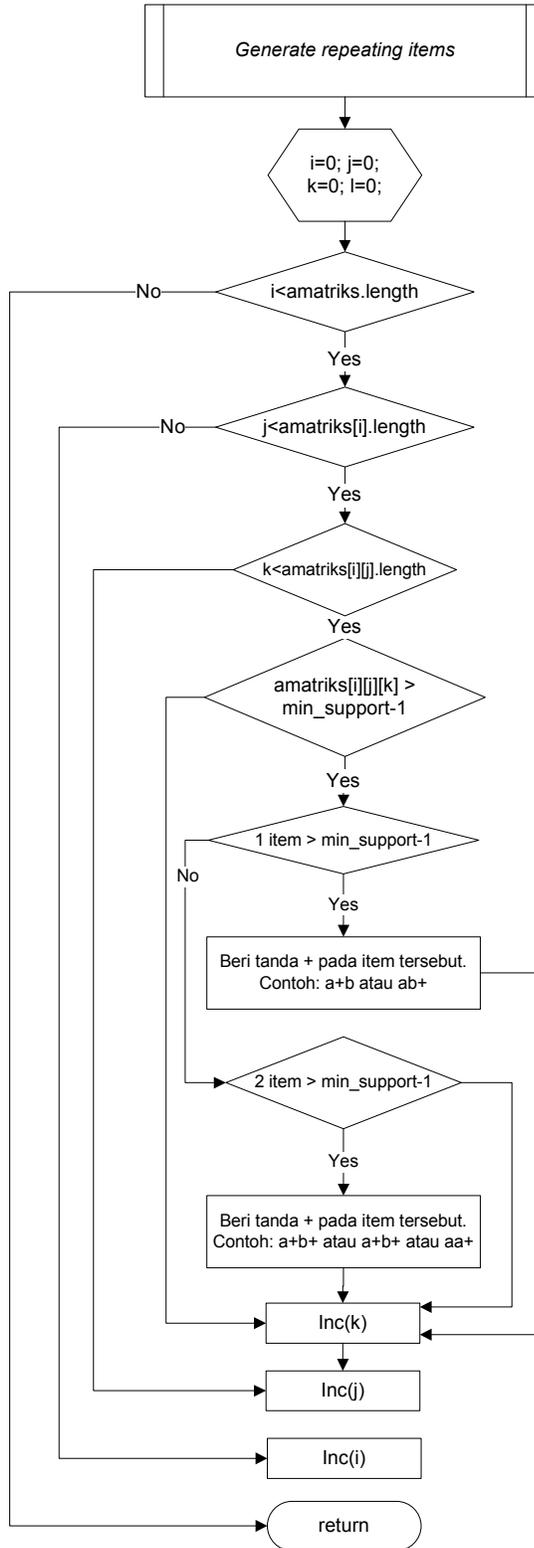
Gambar 5.75. Flowchart Frequent Item Matrix

e. *Generate length-2 sequential patterns*



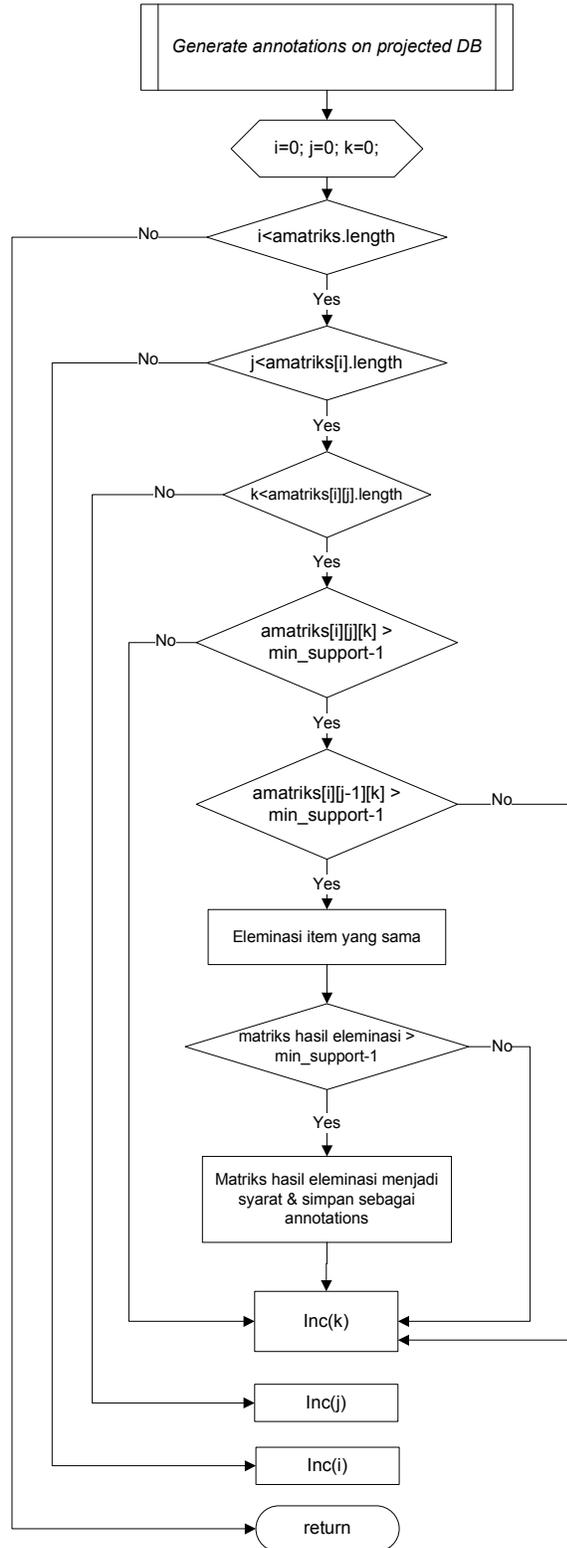
Gambar 5.76. Flowchart Generate Length-2 Sequential Patterns

f. *Generate repeating item*



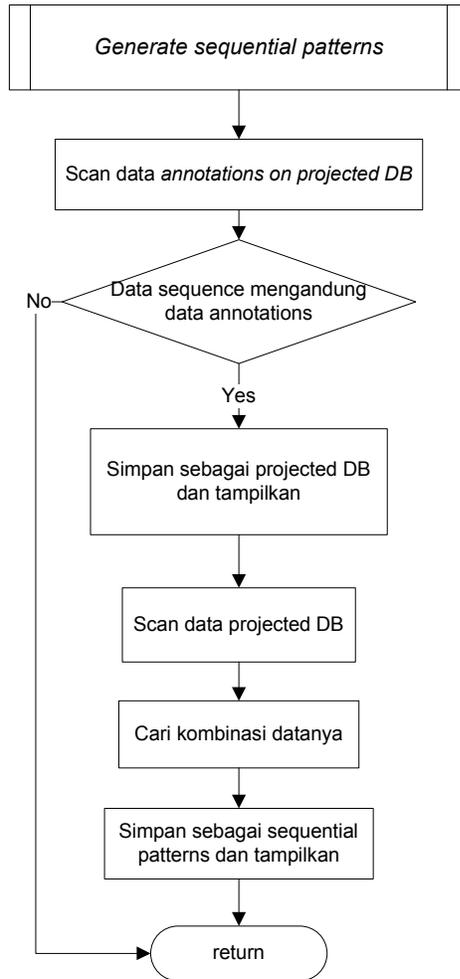
Gambar 5.77. Repeating Items

g. *Generate projected DB*



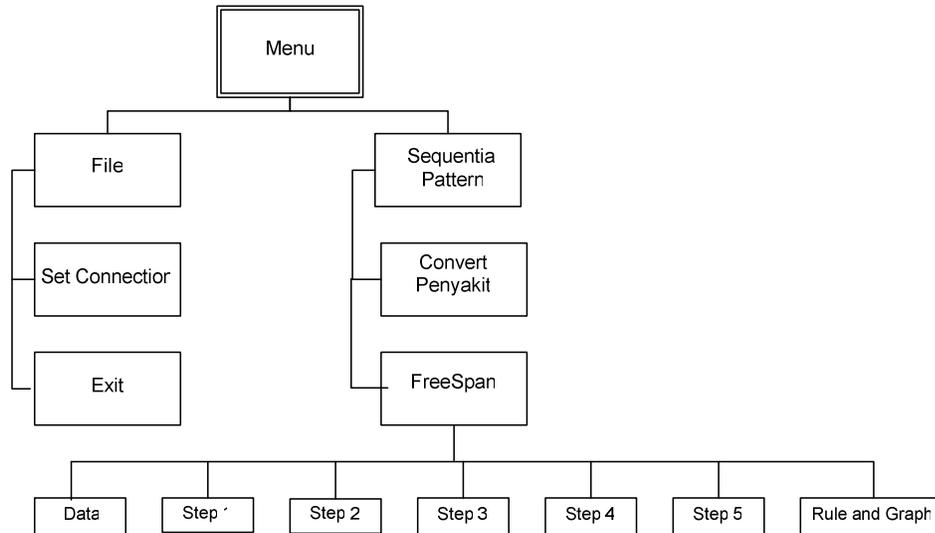
Gambar 5.78. Generate annotations on projected DB

h. *Generate sequential patterns*



Gambar 5.79. Generate sequential patterns

5.4.3. Struktur Menu



Gambar 5.80. Struktur Menu Penentuan Pola Penyakit

#### 5.4.4. Struktur Menu Pengenalan Pola Sebab-Akibat Terjangkitnya Penyakit Pada Pasien

Untuk memulai perhitungan dalam aplikasi ini dibutuhkan beberapa *inputan* dari *user* seperti Start Date, End Date, selang waktu yang dapat berupa hari, bulan, atau tahun, *minimum support*, dan apakah mau mengetahui *sequence* per daerah atau pasien secara keseluruhan. Jika ingin melihat *sequence* per daerah, maka *combobox* use propinsi perlu dicentang, jika tidak maka dilakukan perhitungan terhadap seluruh pasien tanpa memperhitungkan daerahnya (Gambar 5.81).

Gambar 5.81. Inputan awal user

Setelah *user* menentukan pilihannya maka data dari tabel PASIEN\_IK akan *ditransform* yang bertujuan memudahkan dalam pengolahan data yang dapat dilihat pada Gambar 5.82.

NODMK	NO SEQUENCE	MRS	KET DIAG	PENYAKIT	WAKTU MASUK	WAKTU KELUAR
111	1	1	1	7	2009-01-01	2009-01-03
111	1	1	2	9	2009-01-01	2009-01-03
111	1	2	1	8	2009-01-04	2009-01-06
111	1	3	1	7	2009-01-08	2009-01-10
111	1	4	1	6	2009-01-15	2009-01-17
111	1	4	2	8	2009-01-15	2009-01-17
111	2	5	1	7	2009-12-01	2009-12-05
111	2	5	2	11	2009-12-01	2009-12-05
111	2	6	1	8	2009-12-07	2009-12-10
111	2	6	2	10	2009-12-07	2009-12-10
111	2	7	1	7	2009-12-15	2009-12-17
111	2	8	1	11	2009-12-20	2009-12-25
111	2	8	2	12	2009-12-20	2009-12-25

Gambar 5.82. Data awal dan data yang diolah

Dari hasil data yang ditransformasi akan dihasilkan *frequent item matrix* seperti pada Gambar 5.83.

Sequence ID	Sequence	Item Pattern
1	(7 9) (8) (7) (6 8)	6 7 8 9
2	(7 11) (8 10) (7) (11 12)	7 8 10 11 12
3	(6 13) (7 11) (6) (7) (11)	6 7 11 13
4	(7 10) (8 10) (9)	7 8 9 10
5	(6) (7 9) (7) (8) (7) (6 9 10)	6 7 8 9 10

Frequent item matrix:

	7	8	6	9	10	11
4						
4 3 0		1				
3 2 0		2 1 1	2			
2 2 2		2 2 0	1 2 1	1		
3 1 1		1 1 2	1 0 1	1 1 1	1	
2 2 2		1 1 0	1 1 0	0 0 0	1 1 0	2

Gambar 5.83. Item pattern dan frequent item matrix

Dari *frequent item matrix* dengan *minimum support* = 2, maka hasil *length-2 sequential patterns* dapat dilihat pada Gambar 5.84. Hasil dari *length-2 sequential patterns* ini menampilkan kombinasi item yang hanya lebih besar dari *minimum support*.

Item	Length-2 sequential patterns
7	<7 7>: 4
8	<7 8>: 4 <8 7>: 3
6	<7 6>: 3 <6 7>: 2 <8 6>: 2 <6 6>: 2
9	<7 9>: 2 <9 7>: 2 <(7 9)>: 2 <8 9>: 2 <9 8>: 2 <9 6>: 2
10	<7 10>: 3 <(8 10)>: 2
11	<7 11>: 2 <11 7>: 2 <(7 11)>: 2 <11 11>: 2

Gambar 5.84. Length-2 sequential patterns

Kemudian dibentuk *repeating items* yang dapat dilihat pada Gambar 5.85.

Generate Item-Repeating Items	
Item	Repeating Items
7	<7 7+>
8	{7+ 8}
6	{6+ 7+} <8 6+> <6 6+>
9	{7+ 9} <9 6+>
10	<7+ 10>
11	{7+ 11+} <11 11+>

Gambar 5.85. Repeating items

Dalam membentuk *annotation projected DB* dibutuhkan *length-2 sequential patterns* sehingga dapat menghasilkan seperti pada Gambar 5.86.

Generate Projected DBs	
Item	Projected DBs
6	<8 6> : {7}
9	{8 9} : {7}
9	<9 6> : {7}
9	<9 6> : {8}
10	<(8 10)> : {7}

Gambar 5.86. Annotation projected DB

*Annotation DB* menghasilkan 2 jawaban yaitu *projected DB* yang dapat dilihat pada Gambar 5.86 dan *sequential patterns* yang dapat dilihat pada Gambar 5.87.

Projected DB	
Item	Projected DB
<8 6> : {7}	7 8 7 6
<8 6> : {7}	7 7 8 7 6
{8 9} : {7}	(7 9) 8 7 8
{8 9} : {7}	7 8 9
{8 9} : {7}	(7 9) 7 8 7 9
<9 6> : {7, 8}	(7 9) 8 7 (6 8)
<9 6> : {7, 8}	(7 9) 7 8 7 6
<(8 10)> : {7}	7 (8 10) 7
<(8 10)> : {7}	7 (8 10)

Gambar 5.87. Projected DB

Sequential Patterns		
Item	Sequential Pattern	Jumlah
<8 6>:{7}	< 7 8 6 >	: 2
<8 6>:{7}	< 8 7 6 >	: 2
<8 6>:{7}	< 7 8 7 6 >	: 2
{8 9}:{7}	< (7 9) 8 >	: 2
{8 9}:{7}	< (7 9) 8 7 >	: 2
{8 9}:{7}	< (7 9) 7 8 >	: 2
{8 9}:{7}	< 9 8 7 >	: 2
{8 9}:{7}	< 9 7 8 >	: 2
{8 9}:{7}	< 7 8 9 >	: 2
<9 6>:{7, 8}	< (7 9) 6 >	: 2
<9 6>:{7, 8}	< (7 9) 8 6 >	: 2
<9 6>:{7, 8}	< (7 9) 7 6 >	: 2
<9 6>:{7, 8}	< 9 8 6 >	: 2
<9 6>:{7, 8}	< 9 7 6 >	: 2
<9 6>:{7, 8}	< (7 9) 8 7 6 >	: 2
<9 6>:{7, 8}	< 9 8 7 6 >	: 2
<(8 10)>:{7}	< 7 (8 10) >	: 2

Gambar 5.88. Sequential Patterns dalam bentuk angka

Pada Gambar 5.88 hasil dari *sequential patterns* masih dalam bentuk angka. Oleh karena itu data yang semula dalam bentuk angka perlu diubah dalam bentuk kode penyakit yang sesungguhnya sehingga memudahkan *user* dalam memahami pola suatu penyakit (Gambar 5.89).

Sequential Patterns		
Item	Sequential Pattern	Jumlah
<c a >:{b}	b c a	: 2
<c a >:{b}	c b a	: 2
<c a >:{b}	b c b a	: 2
{c d}:{b}	(b d) c	: 2
{c d}:{b}	(b d) c b	: 2
{c d}:{b}	(b d) b c	: 2
{c d}:{b}	d c b	: 2
{c d}:{b}	d b c	: 2
{c d}:{b}	b c d	: 2
<d a >:{b, c}	(b d) a	: 2
<d a >:{b, c}	(b d) c a	: 2
<d a >:{b, c}	(b d) b a	: 2
<d a >:{b, c}	d c a	: 2
<d a >:{b, c}	d b a	: 2
<d a >:{b, c}	(b d) c b a	: 2
<d a >:{b, c}	d c b a	: 2
<(c e)>:{b}	b (c e)	: 2

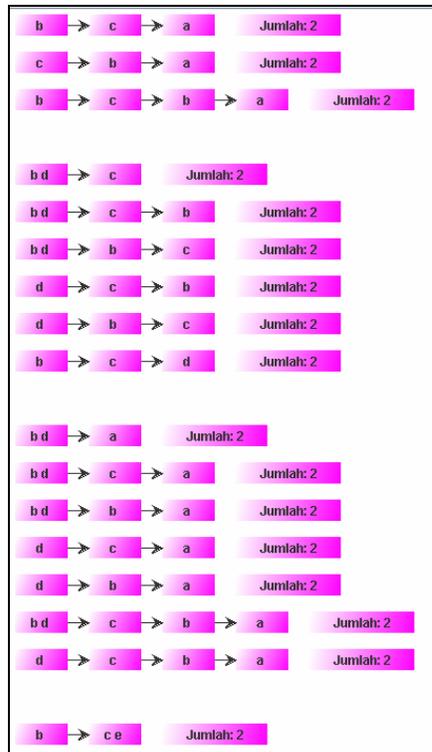
Gambar 5.89. Sequential Patterns dalam bentuk Kode Penyakit

Dari *sequential patterns* yang diperoleh maka dapat membentuk *rule* seperti pada Gambar 5.90.

Rule	View Graph
Rule	
Diawali penyakit b, kemudian penyakit c, a	
Diawali penyakit c, kemudian penyakit b, a	
Diawali penyakit b, kemudian penyakit c, b, a	
Diawali penyakit b dan d, kemudian penyakit c	
Diawali penyakit b dan d, kemudian penyakit c, b	
Diawali penyakit b dan d, kemudian penyakit b, c	
Diawali penyakit d, kemudian penyakit c, b	
Diawali penyakit d, kemudian penyakit b,c	
Diawali penyakit b, kemudian c, d	
Diawali penyakit b dan d, kemudian penyakit a	
Diawali penyakit b dan d, kemudian penyakit c,a	
Diawali penyakit b dan d, kemudian penyakit b,a	
Diawali penyakit d, kemudian penyakit c, a	
Diawali penyakit d, kemudian penyakit b, a	
Diawali penyakit b dan d, kemudian penyakit c, b, a	
Diawali penyakit d, kemudian penyakit c, b, a	
Diawali penyakit b, kemudian penyakit c dan e	

Gambar 5.90. Rule

Selain bentuk *rule*, dapat digambarkan dalam bentuk *graph* seperti pada Gambar 5.91.



Gambar 5.91. Graph Sequential Patterns

Gambar 5.92 merupakan data awal RSUD Dr. Soetomo yang digunakan sebagai *input* dalam pengolahan data, dimana *minimum support*=2, selang waktunya antara 1 Januari 2002 sampai dengan 31 Desember 2002, dengan jumlah minimal pasien masuk ke rumah sakit (*mrs*) minimal 3.

Tabel PASIEN_IK setelah di TRANSFORM						
NODMK	NO SEQUENCE	MRS	PENYAKIT	WAKTU MASUK	WAKTU KELUAR	
145291	1	1	4081	2002-03-19	2002-04-15	
145291	1	1	3209	2002-03-19	2002-04-15	
145291	2	2	72	2002-06-12	2002-06-20	
145291	2	2	4888	2002-06-12	2002-06-20	
145291	2	2	4972	2002-06-12	2002-06-20	
145291	3	3	4888	2002-07-11	2002-07-13	
145291	3	3	4194	2002-07-11	2002-07-13	
145291	4	4	4888	2002-08-15	2002-08-17	
145291	4	4	2130	2002-08-15	2002-08-17	
10003555	5	5	11330	2002-01-18	2002-01-23	
10003555	6	6	11330	2002-04-18	2002-04-25	
10003555	7	7	11330	2002-05-23	2002-05-27	
10003555	8	8	11330	2002-05-23	2002-05-27	

Gambar 5.92. Input data awal

Setelah data awal diproses maka dapat ditentukan *sequence*, *item pattern* dan *frequent item matrix* (Gambar 5.93).



Kemudian dapat dibentuk *repeating items* sesuai dengan *frequent item matrix*-nya (Gambar 5.95).

Generate Item-Repeating Items	
Item	Repeating Items
11328	<11328 11328+>
1913	<1913+ 11328+> <1913 1913+>
11331	<11331+ 11328+> <11331 11331+>
11330	{11330+ 1913+} <11330 11330+>
11335	<11335 1913+>
1244	{1244 1913+} <1244 11330+>
1830	{1830 11328+} <1830 11331+> <1830 11330+>
11329	<11329+ 11331+> <11329 11329+>
1434	<1434 1434+>
1257	<1257 1913+>
1134	<1134+ 11328+> {1134+ 11329+} <1134 1134+>
1246	{1246 1913+} <1246 11330+>
11327	<11327 1913+>
2130	<2130 11328+> <2130 1134+>
1447	<1447+ 11330+> <1447 1447+>
1596	
1900	<1900 1900+>
234	<234 11328+> <234 1434+> <234 1900+>
1130	<1130+ 1134+> <1130 1130+>
1405	<1405 1405+>
72	<72+ 234+> <72 72+>
1243	<1243 1913+>
1268	<1268 1913+>
2426	{2426 72+}
4888	<4888 72+>
6224	
6091	
6139	<6139 1913+>
11334	
1020	<1020 1913+>
1233	
2090	<2090+ 234+> <2090+ 6224+> {2090+ 6091+} <2090 2090+>
4486	<4486 1134+>
6141	<6141 1913+> <6141 11330+>
6144	<6144 1913+>

Gambar 5.95. *Repeating Items*

Dalam membentuk *annotation projected DB* dibutuhkan *length-2 sequential pattern* sehingga dapat menghasilkan seperti pada Gambar 5.96.

Generate Projected DBs	
Item	Projected DBs
1244	<1244 11330> : {1913}
1830	<{11331 1830}> : {11328}
1830	<{1244 1830}> : {11330}
1246	<1246 11330> : {1913}
1246	<{1830 1246}> : {11330}
2130	<{1134 2130}> : {11328}
2426	<2426 72> : {234}
6224	<{2130 6224}> : {11335}
6224	<{234 6224}> : {11335}
6091	<{6224 6091}> : {234}
2090	<{6224 2090}> : {234}
2090	<2090 6091> : {234}
2090	<2090 6091> : {6224}
4486	<{2130 4486}> : {1134}
6141	<{11330 6141}> : {1913}
3818	<3818 6091> : {234}
3818	<3818 2090> : {234}
3818	<3818 2090> : {6091}
1935	<{1913 1935}> : {11328}
1935	<{11330 1935}> : {1913}
6162	<{4486 6162}> : {6139}
2433	<{6162 2433}> : {11335}
4972	<{2130 4972}> : {11335}
4972	<{4888 4972}> : {2130}
4194	<{4971 4194}> : {4888}

Gambar 5.96. *Annotation Projected DB*

*Annotation projected DB* menghasilkan 2 jawaban yaitu *projected DB* pada Gambar 5.97 dan *sequential pattern* yang dapat dilihat pada Gambar 5.98.

Projected DB	
Item	Projected DB
<1244 11330>:{1913}	1244 (11330 1913)   (1244 1913) ...
<1246 11330>:{1913}	1246 (11330 1913)   1246 (1246 ...
<2426 72>:{234}	(2426 234) 72
<(2130 6224)>:{11335}	(11335 2130 6224)
<3818 2090>:{234, 6091}	(3818 2090) (2090 6091)   (6091 ...
<(1913 1935)>:{11328}	(11328 1913 1935)   (11328 1913...
<(6162 2433)>:{11335}	(11335 6162 2433)
<(4888 4972)>:{2130}	(4888 4972 2130)
<(4971 4194)>:{4888}	(4888 4971 4194)

Gambar 5.97. *Annotation Projected DB*

Sequential Patterns		
Item	Sequential Pattern	Jumlah
<1244 11330>:{1913}	< 1244 (11330 1913) >	: 2
<1246 11330>:{1913}	< 1246 (11330 1913) >	: 2
<3818 2090>:{234, 6091}	< 3818 (2090 6091) >	: 2

Gambar 5.98. *Sequential Patterns*

Pada Gambar 5.99 hasil dari *sequential patterns* masih dalam bentuk angka. Oleh karena itu data yang semula dalam bentuk angka perlu diubah dalam bentuk kode penyakit yang sesungguhnya sehingga memudahkan *user* dalam memahami pola suatu penyakit pada Gambar 5.100.

Sequential Patterns		
Item	Sequential Pattern	Jumlah
<C53.1 Z51.3 >:{D64.9}	C53.1 (Z51.3 D64.9)	: 2
<C53.9 Z51.3 >:{D64.9}	C53.9 (Z51.3 D64.9)	: 2
<I10.X E11.2 >:{A41.9, ...}	I10.X (E11.2 N08.3)	: 2

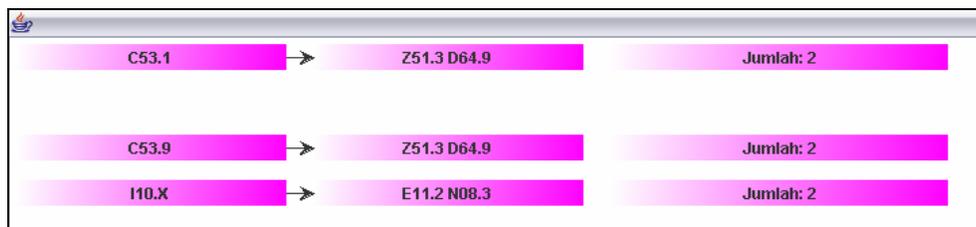
Gambar 5.99. *Sequential Patterns* dalam bentuk Kode Penyakit

Dari *sequential patterns* yang diperoleh maka dapat membentuk *rule* seperti pada Gambar 5.101.

Rule	View Graph
Rule	
Diawali penyakit C53.1, kemudian Z51.3 dan D64.9	
Diawali penyakit C53.9, kemudian Z51.3 dan D64.9	
Diawali penyakit I10.X, kemudian E11.2 dan N08.3	

Gambar 5.100. *Rule*

Selain itu juga dapat digambarkan dalam bentuk *graph* seperti pada Gambar 5.101.



Gambar 5.101. *Graph Sequential Patterns*

## BAB 6. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian terhadap penelitian, maka disimpulkan bahwa aplikasi tersebut dapat menghasilkan *data warehouse* yang dibutuhkan untuk menganalisa data rumah sakit, dimana *data warehouse* ini membantu ekstraksi data lebih cepat dan mudah daripada apabila harus dilakukan dari database utama. Dengan demikian RSUD Dr. Soetomo dapat menggunakan *knowledge* tersebut untuk mengambil tindakan pencegahan atau penyembuhan terhadap suatu penyakit.

Algoritma *k-nearest neighbors* digunakan untuk melakukan pengelompokan penyebaran wabah penyakit agar dapat melihat keterkaitan antar kabupaten. Algoritma FreeSpan yang digunakan untuk mengenali pola terjangkitnya penyakit pada suatu daerah yang ditampilkan dalam bentuk rule dan graph untuk membantu visualisasi sehingga mempermudah pihak RSUD Dr. Soetomo dalam memahami.

Saran pengembangan aplikasi adalah dengan mengimplementasikan metode data mining lain agar dapat membantu rumah sakit untuk mendapatkan metode yang terbaik dalam melihat pola penyakit.

## DAFTAR PUSTAKA

- Han, Jiawei and Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.
- Han, Jiawei. 2000. FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining. Retrieved June 7, 2009, from <http://www.cs.sfu.ca/~jpei/publications/freespan.pdf>
- Harmon, John E. & Steven J. Anderson. The Design and Implementation of Geographic Information Systems. John Wiley & Sons, Inc., 2003.
- Hendra Pesiwarissa. Perancangan dan Pembuatan Aplikasi Data Mining dalam Menganalisa Track Record Penyakit Pasien di RSUD dr. Haulussy Ambon dengan Menggunakan Fuzzy Association Rule Mining. 20 Desember 2005, Universitas Kristen Petra.
- Indikator Indonesia Sehat 2010 dan Pedoman Penetapan Indikator Provinsi Sehat dan Kabupaten/Kota Sehat, Keputusan Menteri Kesehatan No. 1202/Menkes/SK/VIII/2003, Departemen Kesehatan RI, 2003.
- Intan, Rolly. A Proposal of an Algorithm for Generating Fuzzy Association Rule Mining in Market Basket Analysis. The 3<sup>rd</sup> International Conference on Computational Intelligence, Robotics and Autonomous Systems, Desember 2005, Singapore.
- Kendall, Kenneth E. and Julie E. Kendall. System Analysis and Design. New Jersey: Prentice Hall, 1992.
- Larose, Daniel T. Discovering Knowledge in Data: An Introduction to Data Mining. John Wiley & Sons, Inc., 2005.
- Lin T.Y., Yao Yiyu, Zadeh, L.A., Data Mining, Rough Sets, and Granular Computing, Physica-Verlag Heidelberg, Germany, 2002.
- Prayitno, Osman. Perancangan dan Pembuatan Aplikasi Data Mining dengan konsep Fuzzy C-Covering Untuk Membantu Analisis Market Basket Pada Perusahaan X. 15 Juni 2004, Universitas Kristen Petra.

Rusly, Hendra. Pembuatan Software Data Mining untuk Membantu Pengambilan Keputusan dengan Menggunakan Metode Market Basket Analysis pada Perusahaan "X". 19 Desember 2003, Universitas Kristen Petra.

Sistem Informasi Rumah Sakit di Indonesia (Sistem Pelaporan Rumah Sakit Revisi V), Keputusan Menteri Kesehatan Republik Indonesia No. 1410/Menkes/SK/X/2003, Departemen Kesehatan Republik Indonesia, Direktorat Jenderal Pelayanan Medik, 2003.

Sucahyo, Yudho Giri. Data Mining: Menggali Informasi yang Terpendam.

Yudho. Data Mining. <http://ilmukomputer.com/populer/yudho-datamining.php>

## LAMPIRAN

### SARANA

Komputer yang dimiliki dan digunakan untuk mengembangkan sistem aplikasi data mining adalah Pentium IV dengan RAM 2GB (hasil up grade dari 256 MB menggunakan anggaran penelitian tahun pertama), hard disk 500GB (hasil up grade dari 30GB menggunakan anggaran penelitian tahun pertama). Printer yang dimiliki adalah Samsung ML-2010 Laser Jet.

### PERSONALIA PENELITIAN

#### 1. Ketua Peneliti:

- a. Nama Lengkap : Oviliani Yenty Yuliana
- b. Golongan pangkat dan NIP : IV/C (Pembina Utama Muda) dan 94014
- c. Jabatan Fungsional : Lektor Kepala
- d. Jabatan Struktural : Koordinator Bidang Studi Sistem Informasi
- e. Fakultas/Program Studi : Teknologi Industri/Teknik Informatika
- f. Perguruan Tinggi : Universitas Kristen Petra
- g. Bidang Keahlian : Database, Programming
- h. Waktu untuk penelitian ini : 10 jam/ minggu

#### 2. Anggota Peneliti I:

- a. Nama Lengkap : Silvia Rostianingsih
- b. Golongan pangkat dan NIP : IV/A (Pembina Utama Muda) dan 01043
- c. Jabatan Fungsional : Lektor
- d. Jabatan Struktural : Kepala Laboratorium Pemrograman
- e. Fakultas/Program Studi : Teknologi Industri/Teknik Informatika
- f. Perguruan Tinggi : Universitas Kristen Petra
- g. Bidang Keahlian : Database, Sistem Informasi Geografis
- h. Waktu untuk penelitian ini : 10 jam/ minggu

#### 3. Anggota Peneliti II:

- a. Nama Lengkap : Karjono
- b. Golongan pangkat dan NIP : IV/A (Pembina) dan 140 120 909
- c. Jabatan Fungsional : -

- d. Jabatan Struktural : Kepala Bidang Perencanaan Program dan Rekam Medik RSUD Dr. Soetomo Plt Kepala Instalasi Teknologi Informasi
- e. Fakultas/Program Studi : -
- f. Perguruan Tinggi : Akademi Perkam Medik dan Informatika Kesehatan Surabaya
- g. Bidang Keahlian : Rekam Medik dan Sistem Informasi Manajemen Rumah Sakit
- h. Waktu untuk penelitian ini : 10 jam/ minggu

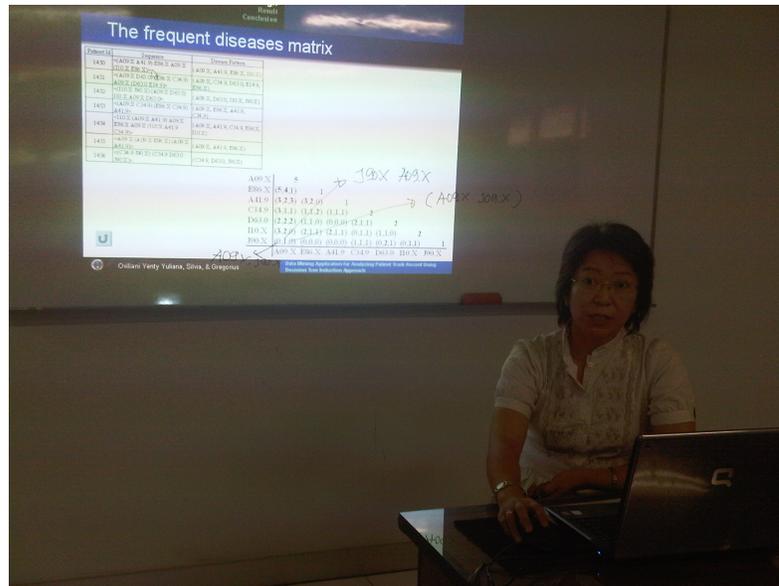
Foto Kegiatan



**Presentasi hasil penelitian di RSUD Dr. Soetomo**



***Knowledge workers RSUD Dr. Soetomo***



**Presentasi Hasil Penelitian *Discovering Sequential Disease Patterns in Medical Databases Using FreeSpan Mining Approach***



**Presentasi Hasil Penelitian Analisis Track Record Penyakit dengan Menggunakan *Metode K-Nearest Neighbors***