

3 ANALISIS DAN DESAIN SISTEM

3.1 Analisis Permasalahan

RSU Dr. Soetomo telah memiliki *database* Oracle 9i untuk melakukan pencatatan data pasien. Seluruh transaksi/*event* yang terjadi di rumah sakit disimpan dalam *database* (data perawatan pasien, tagihan pasien, dan lain-lain). Sistem informasi yang dimiliki oleh RSU Dr. Soetomo saat ini digunakan untuk menghasilkan laporan sesuai dengan standar Departement Kesehatan (laporan *bed occupancy rate*/tingkat penggunaan ruangan, laporan jumlah pasien yang masuk, dan lain-lain). RSU Dr. Soetomo belum memiliki sistem informasi yang dapat digunakan untuk menghasilkan analisa yang lebih lanjut dan dapat digunakan untuk perencanaan dan pengambilan keputusan.

Diperlukan sebuah *software* sistem informasi yang mampu menghasilkan informasi yang bersifat analitik. Salah satu informasi yang bisa dihasilkan dari data yang ada adalah pemetaan penyebaran penyakit. *Software* yang dihasilkan juga dapat menganalisa tingkat ketahanan hidup pasien/*survival analysis* dan menghasilkan *output* berupa grafik.

3.2 Analisis Kebutuhan

Subbab ini menjelaskan kebutuhan apa saja yang harus dipenuhi oleh *software*. Kebutuhan didapat dari analisis permasalahan pada subbab 3.1.

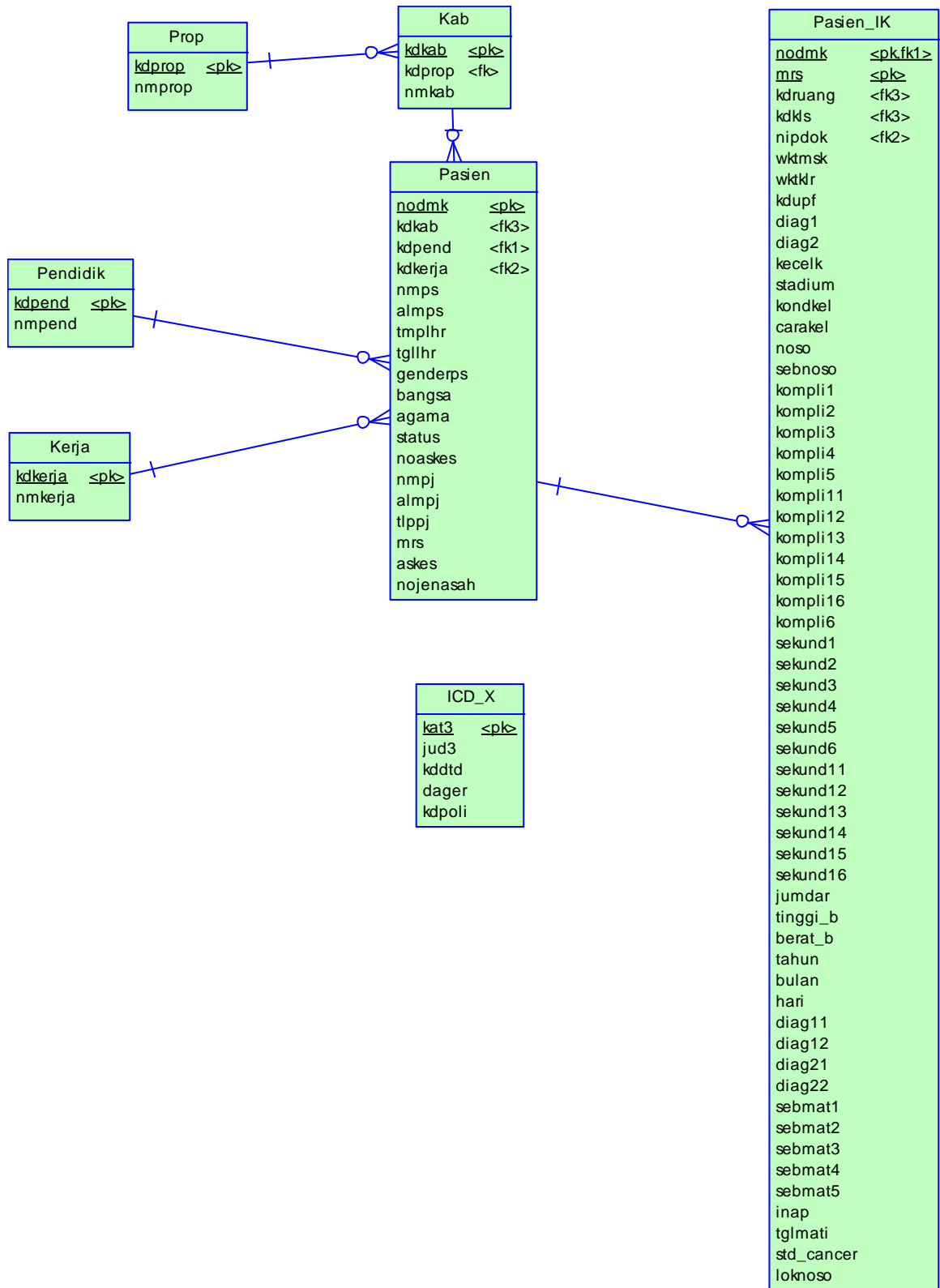
1. Aplikasi bisa membaca *file* peta digital dengan format *shapefile* dan menampilkannya (dalam bentuk peta dan tabel).
2. Aplikasi dapat melakukan pemetaan penyebaran penyakit dengan metode Kriging dan menampilkan hasilnya pada peta digital.
 - a. Data yang dibutuhkan metode Kriging adalah sebagai berikut:
 - i. Nama kelurahan
 - ii. Jumlah pasien suatu penyakit di kelurahan tersebut
 - iii. Jumlah penduduk di kelurahan
 - iv. Proporsi (jumlah pasien / jumlah penduduk)
 - v. Koordinat bujur dan lintang dari suatu kelurahan

- b. *Database* pasien hanya menyimpan alamat pasien, jadi dari alamat tersebut harus didapatkan kelurahannya.
 - c. Perlu dilakukan pembuatan tabel baru dan penambahan *field* pada tabel di *database* RSUD Dr. Soetomo untuk menyimpan data kelurahan pasien.
 - d. Aplikasi mampu mengoreksi data alamat yang memiliki kesalahan atau variasi penulisan. Hal ini diperlukan untuk mendapatkan kelurahan pasien secara tepat.
 - e. Aplikasi memiliki menu untuk melakukan *geocoding*/mendapatkan koordinat bujur dan lintang dari suatu kelurahan.
3. Aplikasi dapat melakukan *survival analysis* dan menampilkan grafik *survival* pasien hasil estimasi Kaplan-Meier.
- a. Data yang dibutuhkan untuk melakukan *survival analysis* adalah sebagai berikut:
 - i. Lama rawat pasien
 - ii. Kolom/*field* yang menandakan hidup/mati pasien (jika bernilai 1 berarti pasien mati, 0 berarti pasien hidup)
 - iii. Kolom-kolom lain yang digunakan sebagai kategori analisa (jenis kelamin, usia pasien, dan lain-lain).
 - b. Semua data yang dibutuhkan sudah ada dalam *database* RSUD Dr. Soetomo, jadi untuk *survival analysis* tidak diperlukan penambahan tabel apapun.
4. Dapat membuat grafik yang bisa dikustomisasi (jenis tampilan maupun isi grafiknya). Ada 2 kategori grafik:
- a. *Predefined* – grafik yang termasuk kategori kategori ini adalah grafik yang sudah disediakan oleh aplikasi. *User* tinggal memilih dari menu grafik:
 - i. Grafik jumlah pasien (bisa berdasarkan kategori tertentu, seperti jenis kelamin, pekerjaan, bulan, tahun, dan lain-lain). Pengguna bisa memilih tampilan grafik yang diinginkan (*bar chart*, *pie chart*, atau *line chart*)
 - ii. *Boxplot* lama rawat pasien

- iii. *Scatter plot*
 - iv. Histogram usia dan lama rawat pasien
 - b. *Custom* – user bisa membuat membuat grafik sendiri jika grafik yang diinginkan tidak ada pada pilihan grafik *predefined*.
5. Aplikasi juga memiliki fitur-fitur dasar yang dimiliki oleh *software* GIS lain, seperti:
- a. User bisa memilih obyek pada peta:
 - i. Secara *graphical* (dengan melakukan klik pada obyek di peta)
 - ii. Berdasarkan atribut (berdasarkan nilai atribut suatu *field*, misalnya memilih kelurahan pada peta yang jumlah pasiennya di atas 200 orang)
 - b. *Zoom in, zoom out, pan* peta
 - c. *Map layer* (peta bisa mempunyai banyak *layer*, misalnya layer kota yang di atasnya terdapat layer jalan, layer bangunan, dll)
 - d. Bisa mengatur *style/tampilan* peta
 - e. Bisa menampilkan *legend*

3.3 ERD Database RSUD Dr. Soetomo

Database yang digunakan adalah *database* rawat inap dari tahun 2000-2009. Namun tidak semua tabel dari *database* digunakan. Gambar 3.1 menunjukkan ERD berisi tabel-tabel yang digunakan:



Gambar 3.1 ERD Database Rawat Inap

Keterangan:

1. Tabel PROP menyimpan daftar provinsi seluruh Indonesia
2. Tabel KAB menyimpan daftar kabupaten seluruh Indonesia
3. Tabel KERJA menyimpan daftar pekerjaan
4. Tabel PENDIDIK menyimpan daftar tingkat pendidikan
5. Tabel PASIEN menyimpan data individu pasien (nama, alamat, jenis kelamin, dll.). Tabel ini berelasi dengan tabel KAB, KERJA, dan PENDIDIK, yang berarti data kabupaten, pekerjaan, dan pendidikan pasien juga disimpan dalam *database*.
6. Tabel ICD_X menyimpan daftar penyakit (kode dan nama penyakit) sesuai dengan daftar *International Classification of Diseases* revisi 10 (ICD-10) dari WHO
7. Tabel PASIEN_IK menyimpan data diagnosa pasien rawat inap. Tabel ini menyimpan nomor pasien yang dirawat, nomor kunjungan (yang menandakan ini kunjungan ke berapa), waktu masuk pasien, waktu keluar pasien, dan kode penyakit yang diderita pasien.

Untuk penjelasan kegunaan masing-masing *field* pada tabel dapat dilihat pada Tabel 3.1 sampai dengan Tabel 3.7:

Tabel 3.1 Tabel PROP

Nama Field	Tipe Data	Keterangan
KDPROP	VARCHAR2(5)	Kode provinsi
NMPROP	VARCHAR2(30)	Nama provinsi

Tabel 3.2 Tabel KAB

Nama Field	Tipe Data	Keterangan
KDKAB	VARCHAR2(10)	Kode kabupaten
KDPROP	VARCHAR2(5)	Kode provinsi
NMKAB	VARCHAR2(50)	Nama kabupaten

Tabel 3.3 Tabel KERJA

Nama Field	Tipe Data	Keterangan
KDKERJA	VARCHAR2(4)	Kode pekerjaan
NMKERJA	VARCHAR2(50)	Nama pekerjaan

Tabel 3.4 Tabel PENDIDIK

Nama Field	Tipe Data	Keterangan
KDPEND	VARCHAR2(4)	Kode pendidikan
NMPEND	VARCHAR2(30)	Tingkat pendidikan

Tabel 3.5 Tabel PASIEN

Nama Field	Tipe Data	Keterangan
NODMK	NUMBER(10)	Nomor ID pasien
KDKAB	VARCHAR2(10)	Kode kabupaten asal pasien
KDKERJA	VARCHAR2(4)	Kode pekerjaan pasien
KDPEND	VARCHAR2(4)	Kode pendidikan pasien
NMPS	VARCHAR2(100)	Nama pasien
ALMPS	VARCHAR2(100)	Alamat pasien
TMLHR	VARCHAR2(50)	Tempat lahir
TGLLHR	DATE	Tanggal lahir
GENDERPS	VARCHAR2(1)	Jenis kelamin
BANGSA	VARCHAR2(30)	Kebangsaan
AGAMA	VARCHAR2(30)	Agama pasien
STATUS	VARCHAR2(1)	Menikah/ belum
NOASKES	VARCHAR2(30)	Nomor Askes pasien
NMPJ	VARCHAR2(100)	Tidak digunakan
ALMPJ	VARCHAR2(100)	Tidak digunakan
TLPPJ	VARCHAR2(20)	Tidak digunakan
MRS	NUMBER(3)	Jumlah masuk rumah sakit
ASKES	VARCHAR2(2)	Tidak digunakan
NOJENAZAH	VARCHAR2(15)	Nomor jenazah pasien (jika pasien mati)

Tabel 3.6 Tabel ICD_X

Nama Field	Tipe Data	Keterangan
KAT3	VARCHAR2(10)	Kode penyakit
JUD3	VARCHAR2(200)	Nama penyakit
KDDTD	VARCHAR2(10)	Tidak digunakan
DAGER	VARCHAR2(1)	Tidak digunakan
KDDPOLI	VARCHAR2(10)	Tidak digunakan

Tabel 3.7 Tabel PASIEN_IK

Nama Field	Tipe Data	Keterangan
NODMK	NUMBER(10)	Nomor ID pasien
MRS	NUMBER(3)	Jumlah kunjungan pasien (1 berarti kunjungan pertama)
KDRUANG	VARCHAR2(5)	Kode ruangan
KDKLS	VARCHAR2(4)	Kode kelas ruangan
WKTMSK	DATE	Waktu masuk
WTKLR	DATE	Waktu keluar
KDUPF	VARCHAR2(4)	Kode Unit Pelaksana Fungsional (UPF)
DIAG1	VARCHAR2(10)	Kode diagnosa utama
DIAG2	VARCHAR2(10)	Tidak digunakan
KECELK	VARCHAR2(10)	Tidak digunakan
STADIUM	VARCHAR2(10)	Stadium penyakit
KONDKEL	VARCHAR2(40)	Kondisi pasien saat keluar dari rumah sakit (sembuh, belum sembuh, mati)
CARAKEL	VARCHAR2(40)	Cara pasien keluar dari rumah sakit (dipulangkan, pulang paksa, lari, dirujuk)
NIDPOK	VARCHAR2(20)	NIP dokter yang menangani pasien
NOSO	VARCHAR2(1)	Tidak digunakan
SEBNOSO	VARCHAR2(50)	Tidak digunakan
KOMPLI1	VARCHAR2(10)	Kode penyakit komplikasi ke-1 (tidak digunakan)
KOMPLI2	VARCHAR2(10)	Kode penyakit komplikasi ke-2 (tidak digunakan)
KOMPLI3	VARCHAR2(10)	Kode penyakit komplikasi ke-3 (tidak digunakan)
KOMPLI4	VARCHAR2(10)	Kode penyakit komplikasi ke-4 (tidak digunakan)
KOMPLI5	VARCHAR2(10)	Kode penyakit komplikasi ke-5 (tidak digunakan)
KOMPLI11	VARCHAR2(10)	Kode penyakit komplikasi ke-11 (tidak digunakan)
KOMPLI12	VARCHAR2(10)	Kode penyakit komplikasi ke-12 (tidak digunakan)
KOMPLI13	VARCHAR2(10)	Kode penyakit komplikasi ke-13 (tidak digunakan)
KOMPLI14	VARCHAR2(10)	Kode penyakit komplikasi ke-14 (tidak digunakan)
KOMPLI15	VARCHAR2(10)	Kode penyakit komplikasi ke-15 (tidak digunakan)
KOMPLI16	VARCHAR2(10)	Kode penyakit komplikasi ke-16 (tidak digunakan)
KOMPLI6	VARCHAR2(10)	Kode penyakit komplikasi ke-6 (tidak digunakan)
SEKUND1	VARCHAR2(10)	Kode penyakit sekunder ke-1 (tidak digunakan)
SEKUND2	VARCHAR2(10)	Kode penyakit sekunder ke-2 (tidak digunakan)
SEKUND3	VARCHAR2(10)	Kode penyakit sekunder ke-3 (tidak digunakan)
SEKUND4	VARCHAR2(10)	Kode penyakit sekunder ke-4 (tidak digunakan)
SEKUND5	VARCHAR2(10)	Kode penyakit sekunder ke-5 (tidak digunakan)
SEKUND6	VARCHAR2(10)	Kode penyakit sekunder ke-6 (tidak digunakan)
SEKUND11	VARCHAR2(10)	Kode penyakit sekunder ke-11 (tidak digunakan)
SEKUND12	VARCHAR2(10)	Kode penyakit sekunder ke-12 (tidak digunakan)
SEKUND13	VARCHAR2(10)	Kode penyakit sekunder ke-13 (tidak digunakan)

Tabel 3.7 Tabel PASIEN_IK (sambungan)

Nama Field	Type Data	Keterangan
SEKUND14	VARCHAR2(10)	Kode penyakit sekunder ke-14 (tidak digunakan)
SEKUND15	VARCHAR2(10)	Kode penyakit sekunder ke-1 5 (tidak digunakan)
SEKUND16	VARCHAR2(10)	Kode penyakit sekunder ke-16 (tidak digunakan)
JUMDAR	NUMBER(10)	Tidak digunakan
TINGGI_B	NUMBER(3)	Tinggi badan
BERAT_B	NUMBER(3)	Berat badan
TAHUN	NUMBER(3)	Usia pasien pada saat dirawat
BULAN	NUMBER(2)	Usia pasien pada saat dirawat
HARI	NUMBER(2)	Usia pasien pada saat dirawat
DIAG11	VARCHAR2(10)	Tidak digunakan
DIAG12	VARCHAR2(10)	Tidak digunakan
DIAG21	VARCHAR2(10)	Tidak digunakan
DIAG22	VARCHAR2(10)	Tidak digunakan
SEBMAT1	VARCHAR2(50)	Kode penyakit penyebab kematian ke-1
SEBMAT2	VARCHAR2(50)	Kode penyakit penyebab kematian ke-2
SEBMAT3	VARCHAR2(50)	Kode penyakit penyebab kematian ke-3
SEBMAT4	VARCHAR2(50)	Kode penyakit penyebab kematian ke-4
SEBMAT5	VARCHAR2(50)	Kode penyakit penyebab kematian ke-5
INAP	VARCHAR2(1)	Pasien rawat inap/ tidak (nilainya selalu 'Y')
TGLMATI	DATE	Tanggal mati pasien (jika mati)
STD_CANCER	VARCHAR2(5)	Tidak digunakan
LOKNOSO	VARCHAR2(2)	Tidak digunakan

3.4 Database Usulan Untuk Aplikasi

Diperlukan pembuatan tabel baru dan penambahan *field* pada tabel yang sudah ada di *database* RSUD Dr. Soetomo. Hal ini karena metode Kriging membutuhkan data lokasi berupa kelurahan tempat tinggal pasien sedangkan *database* hanya menyimpan data alamat saja. Subbab 3.4.1 sampai dengan 3.4.5 menjelaskan mengenai tabel apa saja yang perlu dibuat. Subbab 3.4.6 menjelaskan ERD *database* setelah dilakukan penambahan tabel.

3.4.1 Tabel WILAYAH

Tabel WILAYAH membagi sebuah kabupaten menjadi beberapa wilayah (misal kabupaten Surabaya dibagi menjadi 5 wilayah, yaitu Surabaya Utara, Surabaya Selatan, Surabaya Barat, Surabaya Timur, dan Surabaya Pusat). Tabel ini merupakan *child* dari tabel KAB (Tabel 3.2).

Tabel 3.8 Tabel WILAYAH

Nama Field	Tipe Data	Keterangan
Kode	int	Kode wilayah
Kode_kabupaten	int	Kode kabupaten suatu wilayah
Nama_wilayah	varchar2(100)	Nama wilayah

3.4.2 Tabel KECAMATAN

Tabel KECAMATAN membagi sebuah wilayah menjadi beberapa kecamatan. Tabel ini merupakan *child* dari tabel WILAYAH.

Tabel 3.9 Tabel KECAMATAN

Nama Field	Tipe Data	Keterangan
Kode	int	Kode kecamatan
Kode_wilayah	int	Kode wilayah dari suatu kecamatan
Nama_kecamatan	varchar2(100)	Nama kecamatan

3.4.3 Tabel KELURAHAN

Tabel KELURAHAN membagi sebuah kecamatan menjadi beberapa kelurahan. Tabel ini merupakan *child* dari tabel KECAMATAN.

Tabel 3.10 Tabel KELURAHAN

Nama Field	Tipe Data	Keterangan
Kode	int	Kode kelurahan
Kode_kecamatan	int	Kode kecamatan dari suatu kelurahan
Nama_kelurahan	varchar2(100)	Nama kelurahan

3.4.4 Tabel DAFTAR_JALAN

Tabel DAFTAR_JALAN menyimpan daftar nama jalan yang ada di kota Surabaya dan kelurahan dari jalan tersebut. Tabel DAFTAR_JALAN diperlukan untuk mendapatkan kelurahan tempat tinggal pasien dari data alamat pasien.

Tabel 3.11 Tabel DAFTAR_JALAN

Nama field	Tipe Data	Keterangan
Nama_jalan	varchar2(100)	Nama jalan

Kode_kelurahan	int	Kode kelurahan dari suatu jalan
----------------	-----	---------------------------------

3.4.5 Tabel JUMLAH_PENDUDUK_KELURAHAN

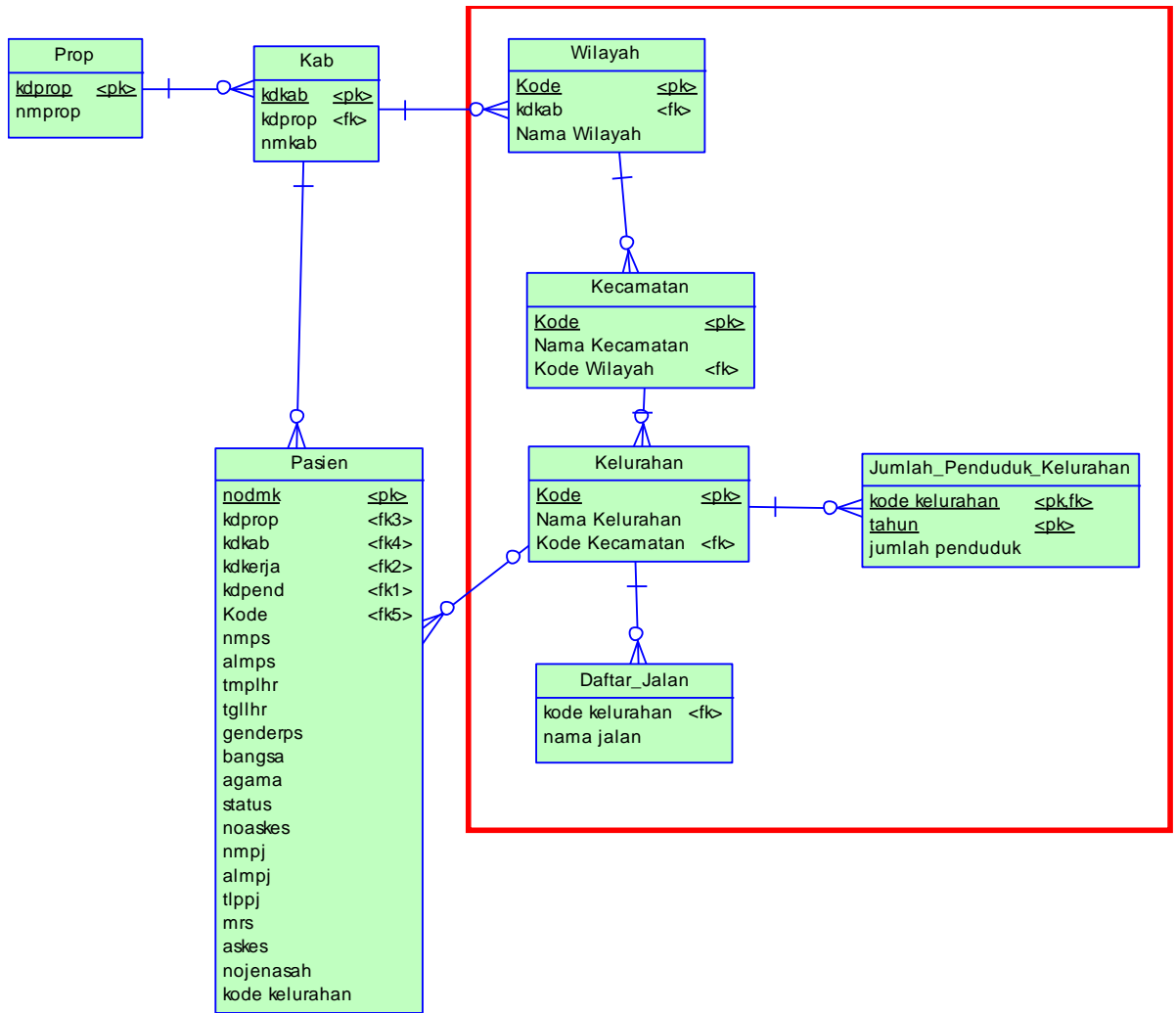
Tabel JUMLAH_PENDUDUK_KELURAHAN menyimpan jumlah penduduk suatu kelurahan pada tahun tertentu. Tabel JUMLAH_PENDUDUK_KELURAHAN diperlukan untuk mendapatkan proporsi penderita suatu penyakit pada kelurahan (proporsi adalah jumlah penderita di suatu kelurahan dibagi dengan jumlah penduduk pada kelurahan tersebut).

Tabel 3.12 Tabel JUMLAH_PENDUDUK_KELURAHAN

Nama <i>field</i>	Tipe Data	Keterangan
Kode_kelurahan	int	Kode kelurahan
Jumlah_Penduduk	int	Jumlah penduduk pada suatu kelurahan
Tahun	int	Menandakan tahun data jumlah penduduk

3.4.6 ERD *Database Usulan*

Gambar 3.2 menunjukkan ERD *database* yang sudah ditambahi dengan tabel-tabel baru. Tabel baru digambarkan di dalam kotak merah. Tabel-tabel yang tidak berubah (seperti tabel PASIEN_IK, KERJA) tidak digambarkan.



Gambar 3.2 ERD dengan Tabel Baru

3.5 Flowchart Sistem

Subbab ini menjelaskan *flowchart* sistem. *Flowchart* dibagi menjadi empat bagian utama, yaitu *flowchart* untuk proses Kriging, *flowchart* untuk proses *survival analysis*, *flowchart* untuk pembuatan grafik, dan *flowchart* untuk proses lain yang tidak termasuk tiga kategori di atas (misal untuk proses koreksi data alamat pasien dengan algoritma Levenshtein, dan lain-lain).

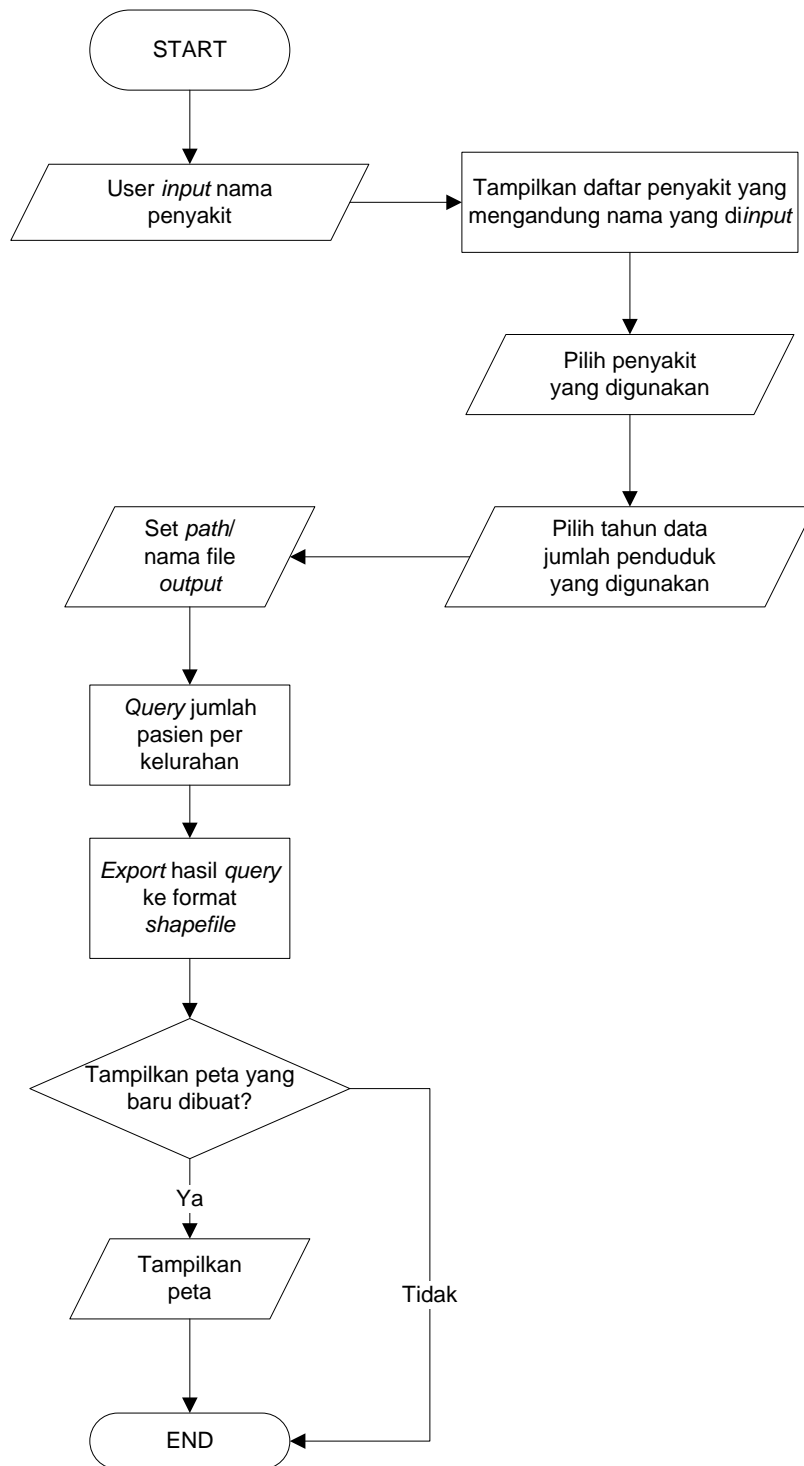
3.5.1 Flowchart untuk Proses Kriging

Proses Kriging dibagi menjadi 2 tahap, yaitu *data preparation* dan proses utama.

3.5.1.1 *Flowchart Data Preparation Kriging*

Tahap *data preparation* bertujuan untuk membuat peta vektor/*shapefile* bertipe titik. Masing-masing obyek titik mewakili satu kelurahan di Surabaya dan menyimpan jumlah pasien, jumlah penduduk, dan proporsi pasien (jumlah pasien / jumlah penduduk) untuk kelurahan tersebut. Pada tahap ini, pengguna menginputkan penyakit yang mau dianalisa, memilih tahun untuk data jumlah penduduk yang digunakan, dan mengatur letak *file output*-nya.

Flowchart untuk tahap *data preparation* tampak pada Gambar 3.3.



Gambar 3.3 Flowchart Data Preparation Kriging

3.5.1.2 Flowchart Proses Utama Kriging

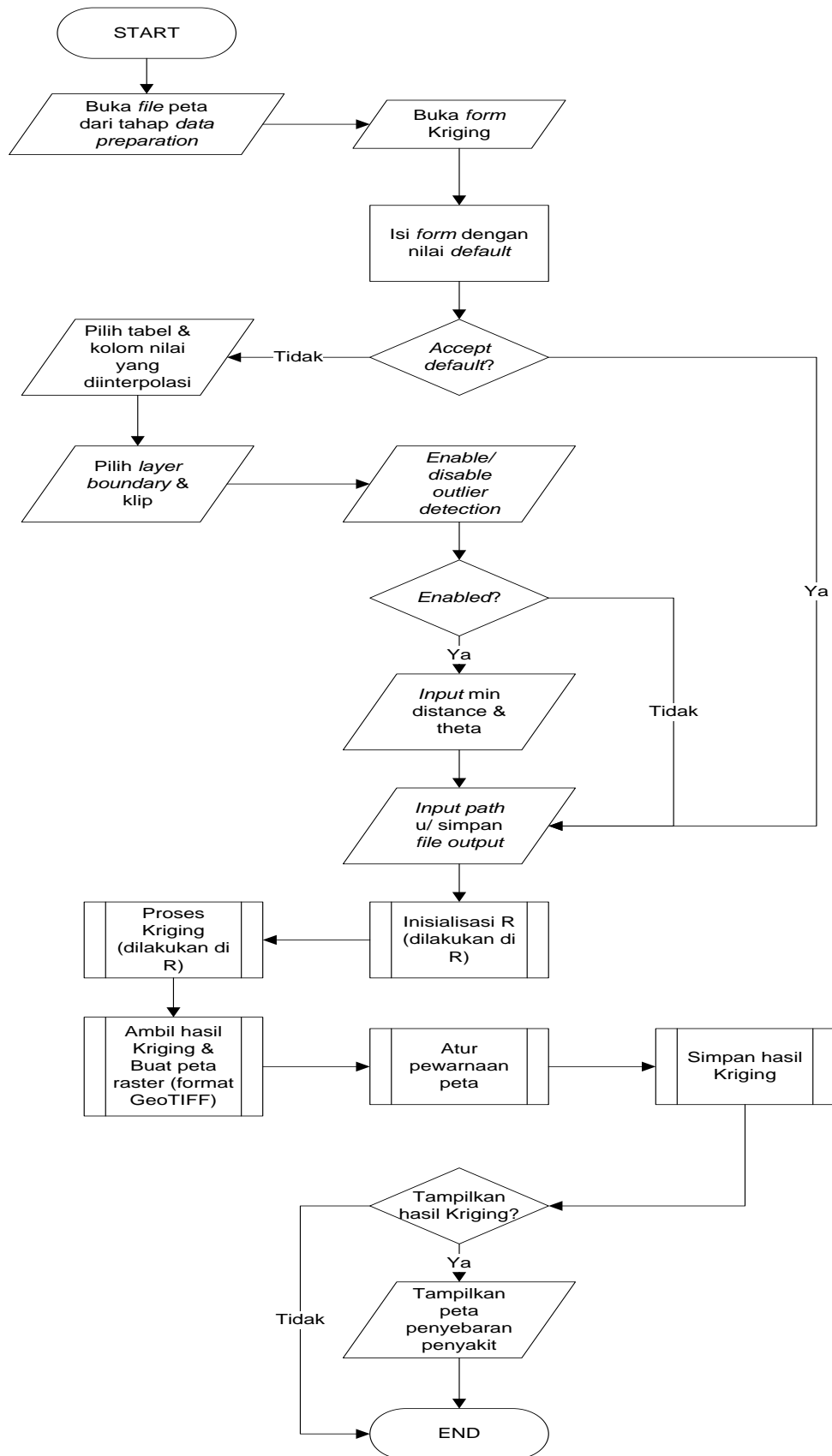
Tahap ini menghasilkan peta penyebaran penyakit berupa peta *raster* (dengan format GeoTIFF). Pengguna harus membuka *file* peta yang dihasilkan

dari tahap *data preparation*. Setelah itu, pengguna memilih *form* Kriging. *Form* Kriging akan muncul dengan nilai *default* yang bisa diubah.

Pengguna bisa memilih tabel dan *field* yang menyimpan nilai yang ingin diinterpolasi (*default*-nya proporsi pasien). Pengguna juga bisa memilih *layer* yang dijadikan sebagai *boundary* dari proses Kriging. *Layer boundary* akan digunakan untuk batasan area yang diproses. Pengguna juga bisa memilih *layer* klip. *Layer* ini akan digunakan untuk memotong hasil Kriging yang berada di luar *layer*. Pengguna juga bisa mengatur ukuran *file output* yang digunakan, tempat menyimpan *file output*, dan parameter untuk *outlier detection*.

Setelah memasukkan parameter untuk Kriging, proses akan berjalan. Pertama, aplikasi melakukan inisialisasi R. Kemudian, perhitungan Kriging dilakukan di dalam R. Hasilnya akan dikirim kembali ke Java dan digunakan untuk membuat peta raster (format GeoTIFF). Pengguna kemudian bisa mengatur pewarnaan peta. Lalu peta raster dan aturan pewarnaan akan disimpan ke file. Terakhir, pengguna diberi pilihan untuk langsung menampilkan hasil Kriging.

Flowchart proses utama Kriging tampak pada Gambar 3.4.

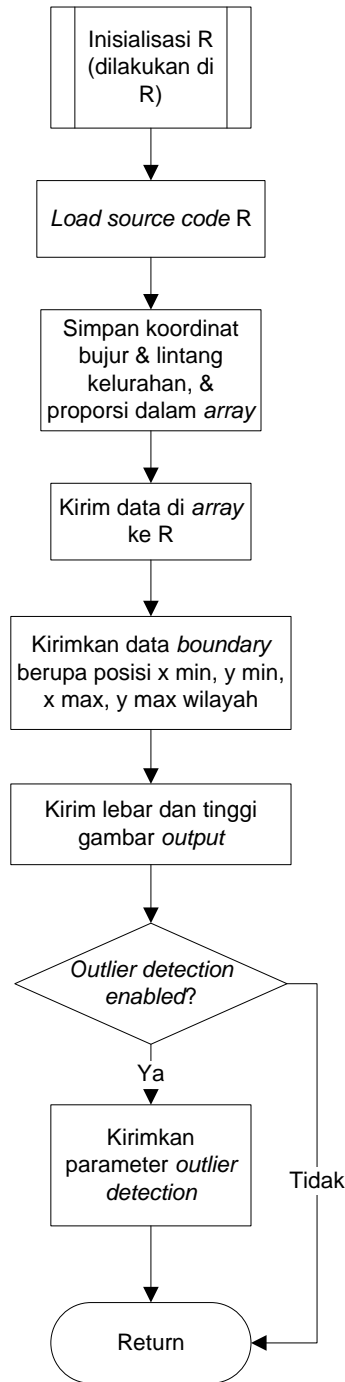


Gambar 3.4 Flowchart Proses Utama Kriging

3.5.1.2.1 *Flowchart* Inisialisasi R

Inisialisasi R dilakukan setelah pengguna selesai memasukkan semua parameter untuk Kriging. Aplikasi Java akan membaca *source code* R yang menyimpan fungsi untuk Kriging, kemudian data koordinat bujur-lintang dan proporsi pasien untuk setiap kelurahan akan di simpan dalam *array*. Data di *array* dikirim ke R. Koordinat *boundary* (batas wilayah yang di buat peta penyebaran penyakitnya) diambil dari *layer boundary* dan dikirim ke R dalam bentuk x minimal, y minimal, x maksimal (x minimal + panjang wilayah), dan y maksimal (y minimal + lebar wilayah). Lebar dan tinggi gambar output juga dikirim. Jika *user* memilih untuk melakukan *outlier detection*, maka parameter *outlier detection* juga akan dikirim.

Flowchart inisialisasi R tampak pada Gambar 3.5



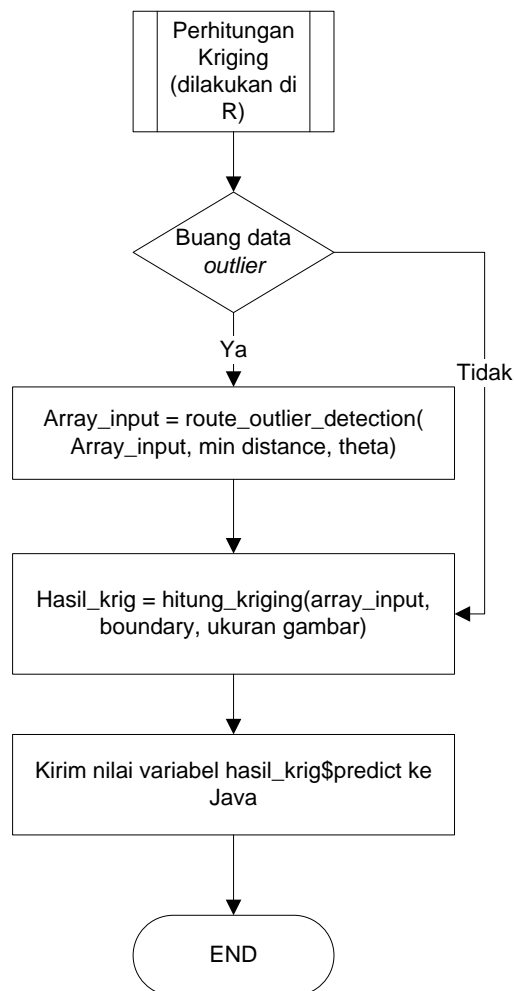
Gambar 3.5 *Flowchart* Inisialisasi R

3.5.1.2.2 *Flowchart* Perhitungan Kriging

Perhitungan Kriging dilakukan oleh R. Proses ini menerima input dari proses inisialisasi R. Sistem akan mengecek apakah *user* memilih untuk membuang data *outlier*. Jika ya, sistem akan memanggil fungsi

route_outlier_detection. Fungsi ini membuang data *outlier* dari *array* input menggunakan algoritma *route outlier detection* (lihat subbab *Outlier Detection* pada halaman 15). Lalu sistem akan memanggil fungsi *hitung_kriging* dengan parameter *array* berisi data, koordinat *boundary* (x min, y min, x max, y max), serta lebar dan tinggi gambar *output*. Fungsi *hitung_kriging* akan mengembalikan obyek dari *class* *kriging*. Obyek dari *class* ini memiliki variabel *predict* yang merupakan *array* berisi nilai prediksi/ hasil interpolasi. Isi variabel *predict* ini dikirimkan kembali ke Java.

Flowchart proses ini tampak pada Gambar 3.6.

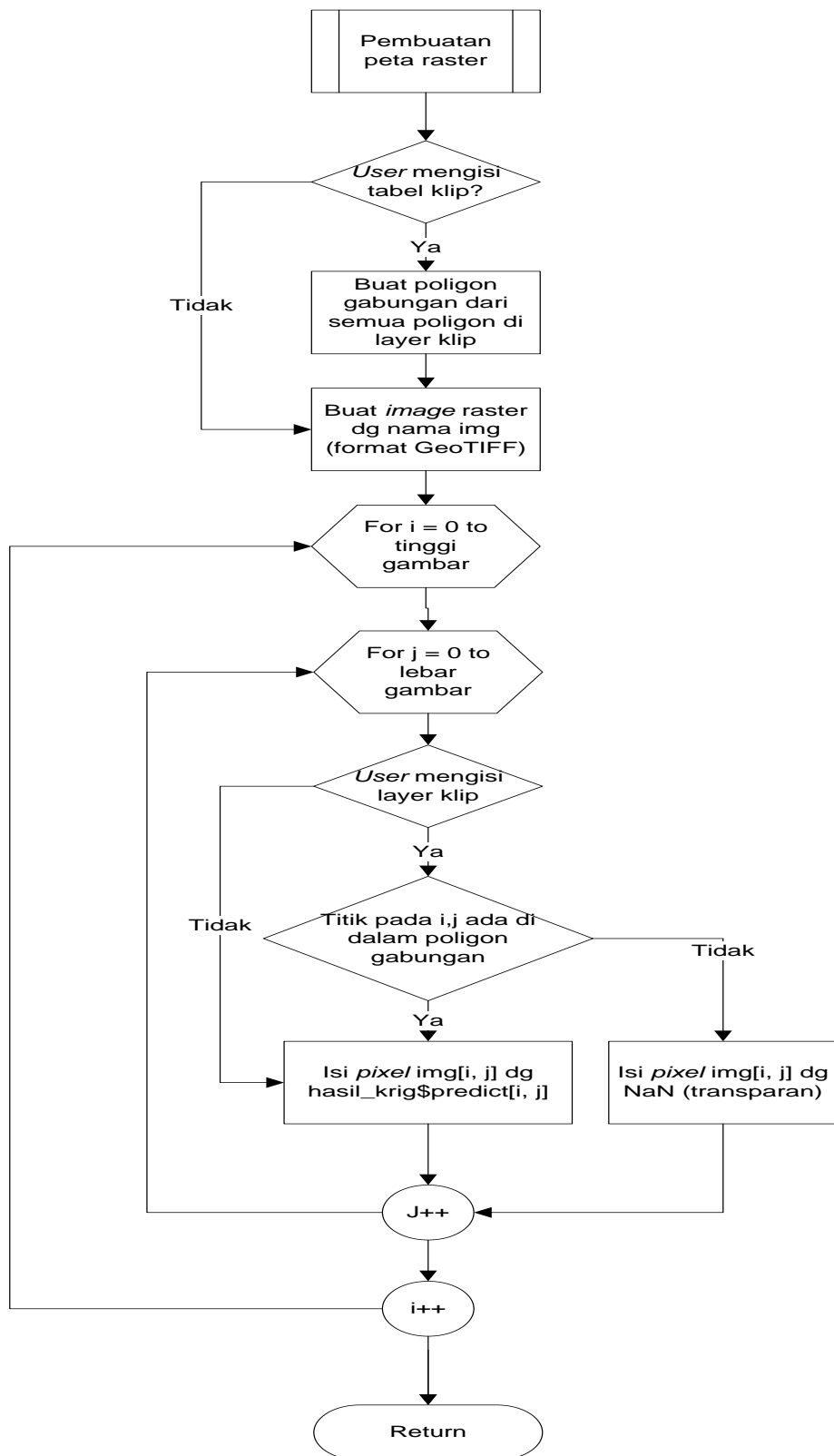


Gambar 3.6 Flowchart Perhitungan Kriging

3.5.1.2.3 *Flowchart* Pembuatan Peta Raster

Proses pembuatan peta raster dilakukan di Java. Peta raster dibuat dengan input hasil perhitungan Kriging. Hasil perhitungan berupa *array* 2D. Baris pertama berisi koordinat bujur, baris kedua berisi koordinat lintang, dan baris ketiga berisi nilai prediksi/ hasil interpolasi untuk suatu koordinat bujur dan lintang. Sistem akan membuat sebuah peta raster yang masing-masing sel-nya berisi nilai prediksi pada koordinat tertentu. Jika user mengisi *layer* klip, maka akan dilakukan pengecekan tambahan: titik yang berada di luar *layer* klip akan diberi nilai NaN (diberi warna transparan).

Flowchart pembuatan peta raster tampak pada Gambar 3.7.

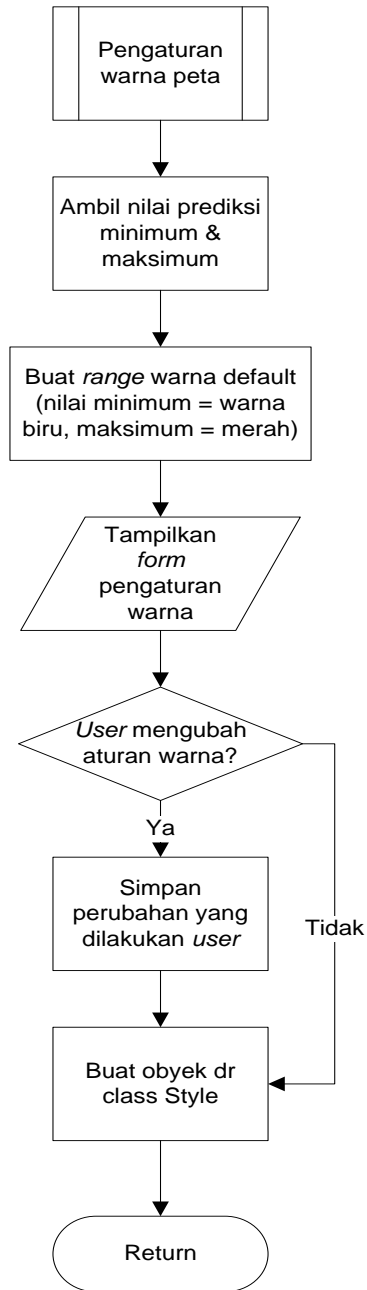


Gambar 3.7 Flowchart Pembuatan Peta Raster

3.5.1.2.4 *Flowchart* Pengaturan Pewarnaan Peta

Tahap keempat dari proses utama Kriging adalah pengaturan pewarnaan peta. *Form* pengaturan warna peta akan muncul secara otomatis dengan warna *default* (warna *default*-nya adalah warna biru untuk daerah pada peta dengan nilai terendah, warna merah untuk nilai tertinggi, dan warna hijau untuk nilai tengah). User bisa mengganti warna yang digunakan. *Form* ini akan menghasilkan obyek *class Style* yang akan disimpan dengan *file* peta.

Flowchart pengaturan warna peta tampak pada Gambar 3.8.



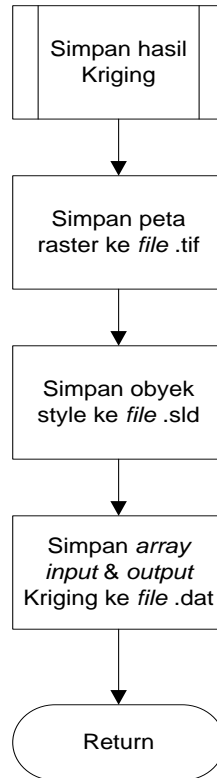
Gambar 3.8 *Flowchart* Proses Pengaturan Warna

3.5.1.2.5 **Flowchart Penyimpanan Hasil Kriging**

Ada tiga file yang disimpan pada tahap ini: *file* peta raster (format GeoTIFF), *file* aturan pewarnaan peta (format .sld), dan *file* metadata Kriging (format .dat, berisi data jumlah penduduk untuk masing-masing kelurahan dan array 2D hasil output proses Kriging). *File* peta raster menyimpan peta penyebaran penyakit, sedangkan aturan pewarnaannya disimpan pada *file* .sld.

File metadata Kriging digunakan untuk *tool* prediksi jumlah pasien pada *window* peta (lihat subbab Desain *Window* Peta pada halaman 15).

Flowchart penyimpanan hasil Kriging tampak pada Gambar 3.9



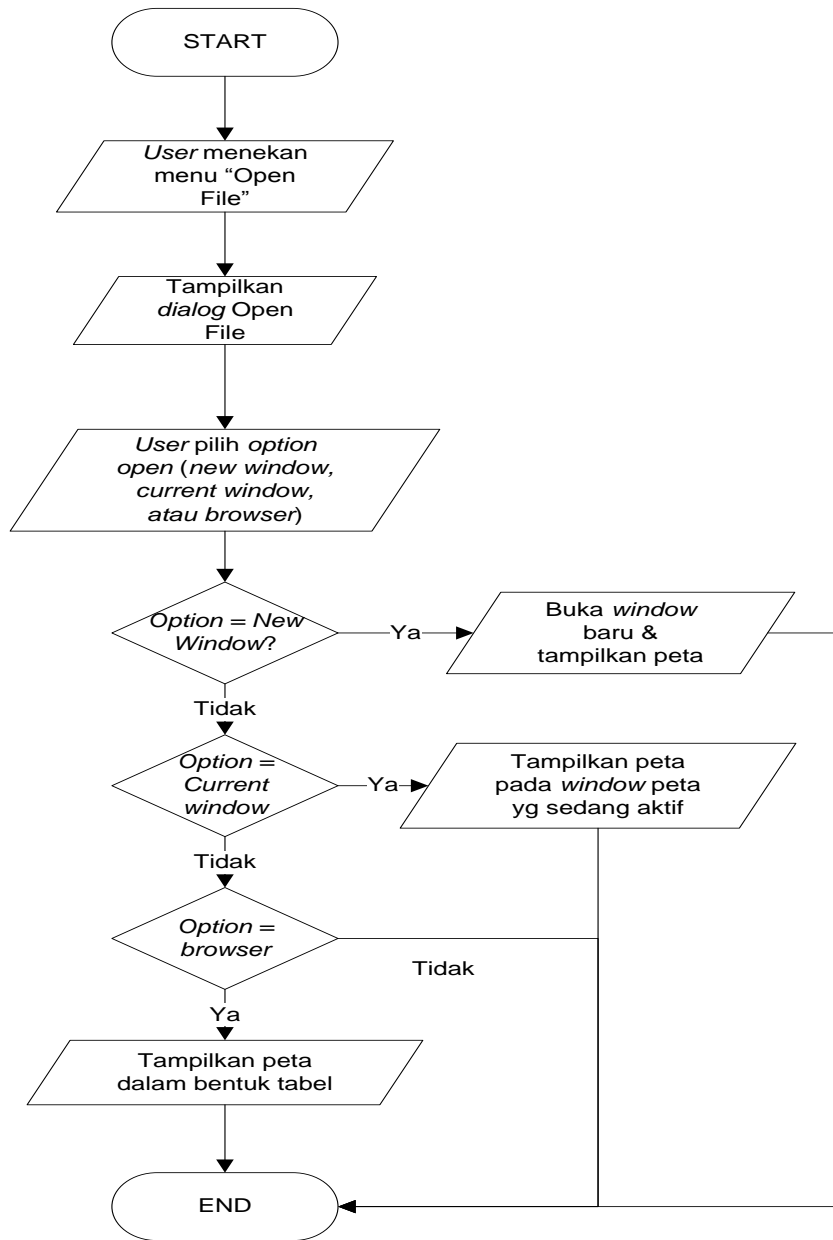
Gambar 3.9 *Flowchart* Penyimpanan Hasil Kriging

3.5.2 *Flowchart* Proses Membuka Peta

Proses kriging menghasilkan dua buah peta: peta bertipe vektor/*shapefile* yang dihasilkan tahap *data preparation* (berisi obyek titik yang berisi jumlah pasien untuk masing-masing kelurahan, digunakan untuk input bagi proses utama Kriging), dan peta bertipe raster/GeoTIFF yang dihasilkan proses utama (berisi peta penyebaran penyakit). Kedua peta ini disimpan sebagai sebuah *file*.

User bisa membuka *file* peta dengan beberapa option, yaitu membuka peta pada *window* peta baru, atau membuka peta pada *window* peta yang sedang aktif (peta akan ditambahkan sebagai *layer* baru pada *window* peta yang sedang aktif), atau menampilkan isi tabel dari peta.

Gambar 3.10 menunjukkan *flowchart* untuk proses membuka peta:



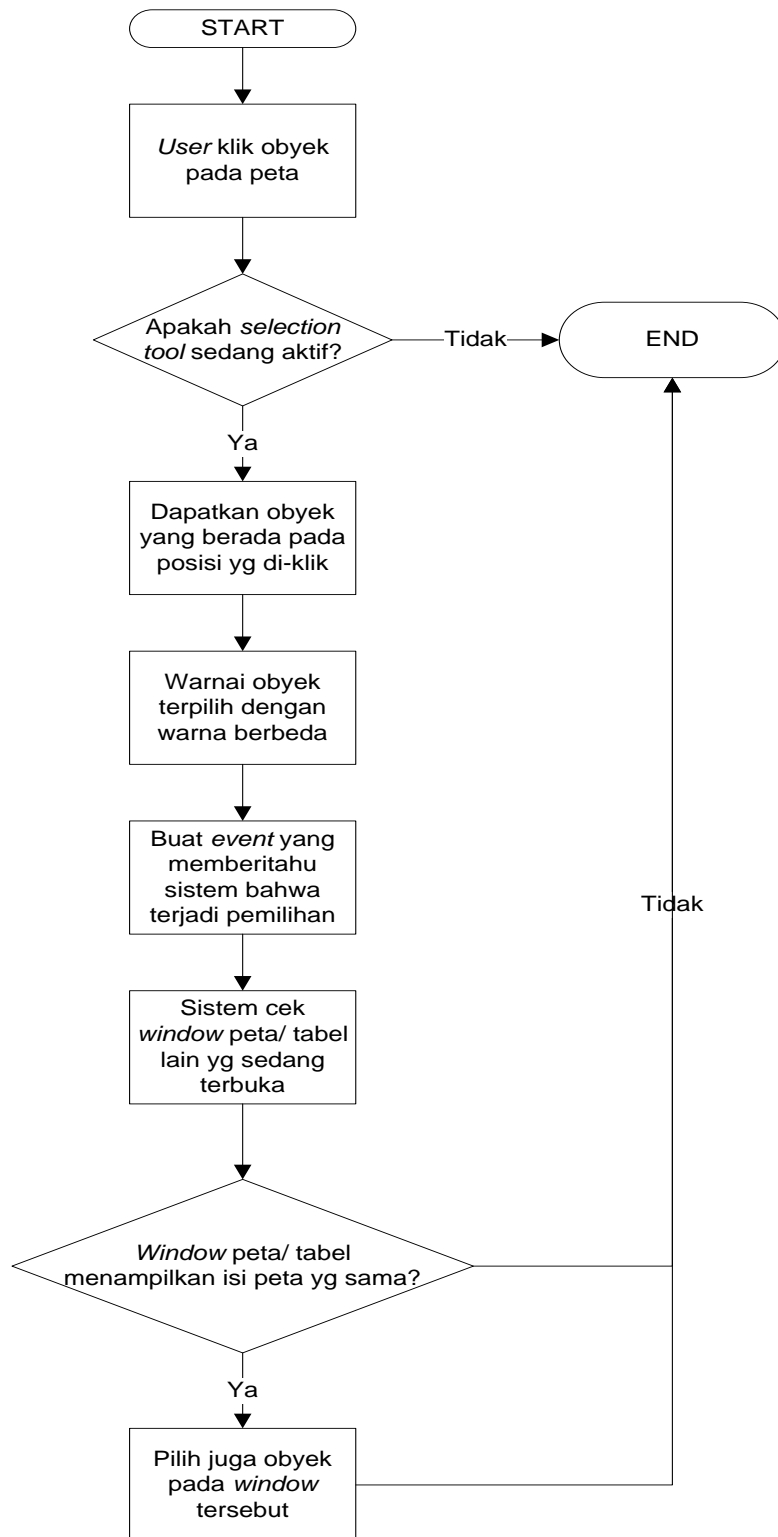
Gambar 3.10 *Flowchart* untuk Membuka Peta

3.5.3 *Flowchart* Pemilihan Obyek Peta

User bisa memilih obyek pada peta. Caranya adalah dengan memilih *tool selection* pada *window* peta (lihat subbab Desain *Window* Peta pada halaman 15). Obyek yang terpilih akan diberi warna yang berbeda. Selain memilih obyek pada *window* yang diklik *user*, jika ada *window* peta/tabel lain yang sedang

menampilkan isi peta yang sama, maka obyek di *window* lain tersebut otomatis ikut terpilih.

Gambar 3.11 menunjukkan *flowchart* proses pemilihan obyek pada peta:



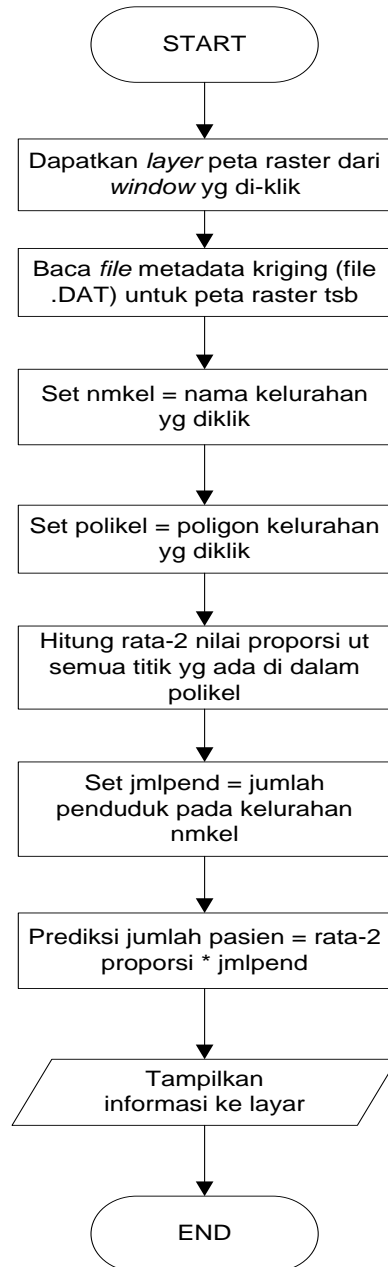
Gambar 3.11 *Flowchart* Proses Pemilihan Obyek pada Peta

3.5.4 *Flowchart* Prediksi Jumlah Pasien

User bisa menampilkan informasi prediksi jumlah pasien dari peta penyebaran penyakit yang dihasilkan proses Kriging. Caranya adalah dengan memilih *tool* prediksi jumlah pasien pada *window* peta (lihat subbab Desain *Window* Peta pada halaman 15). Informasi yang ditampilkan berupa:

- Nama kelurahan yang di-klik
- Jumlah penduduk di kelurahan tersebut
- Hasil interpolasi Kriging di titik yang di-klik
- Prediksi jumlah pasien pada titik yang di-klik
- Prediksi rata-rata jumlah pasien pada kelurahan yang di-klik

Gambar 3.12 menunjukkan *flowchart* untuk proses prediksi jumlah pasien:



Gambar 3.12 *Flowchart* Proses Prediksi Jumlah Pasien

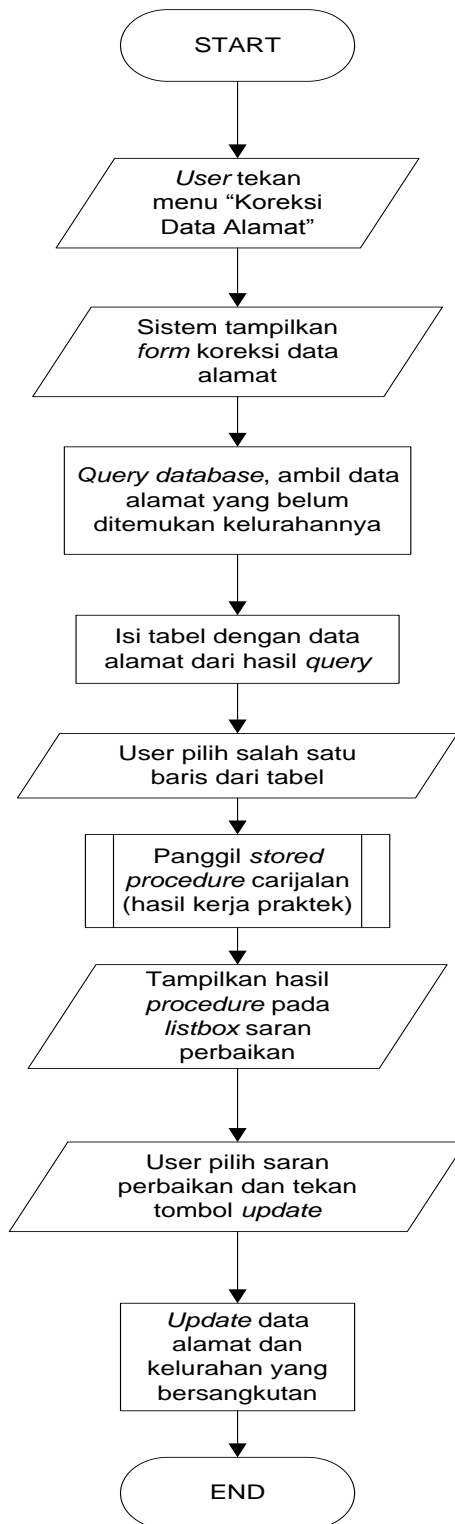
3.5.5 *Flowchart* Proses Koreksi Data Alamat

Proses koreksi data alamat diawali oleh *user* dengan membuka menu koreksi data alamat. Sistem akan menampilkan *form* koreksi data alamat. Sistem melakukan *query* ke *database* untuk mengambil daftar alamat yang belum ditemukan kelurahannya. Daftar alamat tersebut ditampilkan dalam tabel di *form*. Jika *user* memilih salah satu baris dari daftar alamat, maka sistem akan

memanggil *stored procedure* cari_jalan untuk mencari nama jalan dari *database* yang paling mirip dengan alamat yang dipilih. *Stored procedure* cari_jalan menggunakan *script SQL* yang dihasilkan oleh kerja praktek yang tergabung dengan penelitian (dengan perubahan yang diperlukan).

Sepuluh nama jalan yang termirip akan ditampilkan sebagai saran perbaikan. *User* bisa memilih salah satu saran tersebut dan menekan tombol *update* untuk meng-*update* data alamat dan kelurahan pasien.

Gambar 3.13 menunjukkan *flowchart* proses koreksi data alamat:



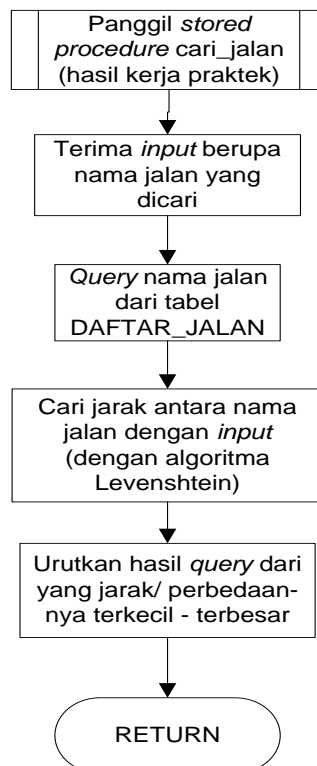
Gambar 3.13 *Flowchart* Proses Koreksi Data Alamat

3.5.5.1 Flowchart Stored Procedure Cari_Jalan

Inti proses koreksi data alamat ada di *stored procedure* cari_jalan. *Stored procedure* ini menerima input berupa nama jalan yang belum ditemukan kelurahannya, dan mengembalikan 10 nama jalan yang paling mirip sebagai saran perbaikan. *Stored procedure* ini dibuat menggunakan *script* SQL hasil kerja praktek yang tergabung dalam penelitian (dengan perubahan yang diperlukan).

Stored procedure cari_jalan menerima parameter input berupa nama jalan yang belum ditemukan kelurahannya. *Stored procedure* akan mengambil daftar jalan yang ada dari tabel DAFTAR_JALAN (Tabel 3.11). Lalu jarak antara input dengan masing-masing jalan akan dihitung menggunakan algoritma Levenshtein (lihat subbab Algoritma Levenshtein pada halaman 15). Setelah itu daftar jalan dari tabel DAFTAR_JALAN akan diurutkan dari yang jaraknya terkecil (berarti kemiripan nama jalannya tinggi) ke terbesar. Lalu *stored procedure* mengembalikan 10 nama jalan dengan jarak terkecil sebagai saran terbaik.

Gambar 3.14 menunjukkan *flowchart* dari *stored procedure* cari_jalan:



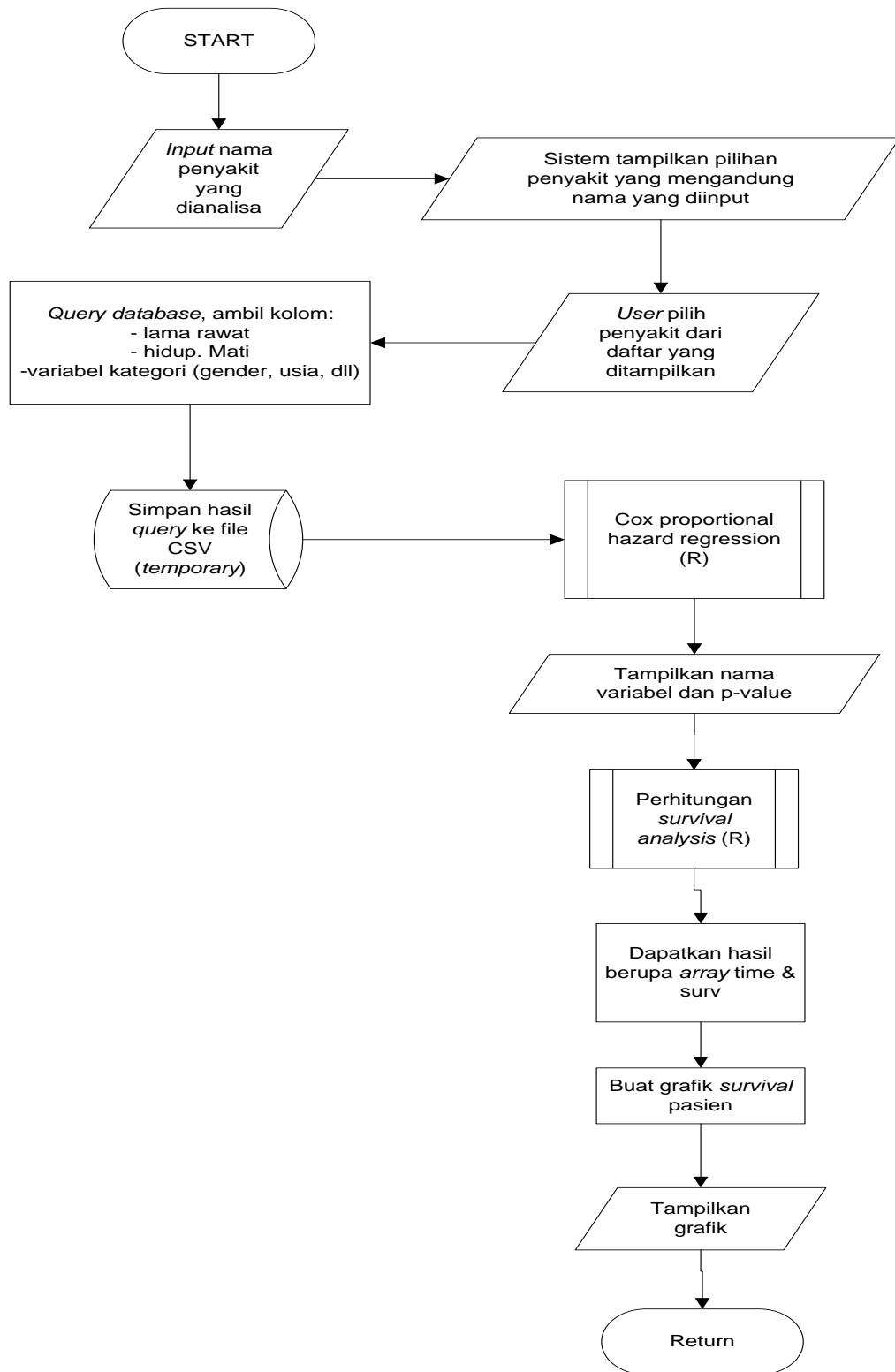
Gambar 3.14 Flowchart Stored Procedure cari_jalan

3.5.6 *Flowchart Proses Survival Analysis*

Proses *survival analysis* menerima input dari pengguna berupa kode dan nama penyakit yang dianalisa. Aplikasi akan melakukan *query* ke *database*. Hasil *query* berupa kolom lama rawat, status hidup/mati pasien (1 jika mati, 0 jika hidup), dan kolom-kolom untuk variabel kategori analisa (gender pasien, pekerjaan, usia, dan lain-lain). Hasil *query* kemudian disimpan ke sebuah *file temporary* bertipe CSV. Aplikasi Java menginisialisasi R, dan kemudian R membaca *file* CSV. Kemudian R melakukan perhitungan *cox proportional hazard regression* untuk menghitung pengaruh setiap variabel kepada *survival* pasien. Hasil perhitungan R akan ditampilkan dalam bentuk tabel berisi nama variabel dan *p-value* (semakin kecil *p-value*, berarti variabel tersebut diperkirakan semakin berpengaruh terhadap *survival* pasien).

Pengguna bisa memilih kategori analisa. Aplikasi Java kemudian memanggil fungsi-fungsi R untuk *survival analysis*. Hasilnya berupa dua buah *array* bertipe *double*, *time* dan *surv*. *Array time* menyimpan waktu lama rawat, dan *array surv* menyimpan prediksi peluang hidup pasien pada waktu lama rawat tertentu. Kedua *array* ini digunakan untuk membuat grafik *survival* pasien.

Flowchart proses *survival analysis* tampak pada Gambar 3.15.

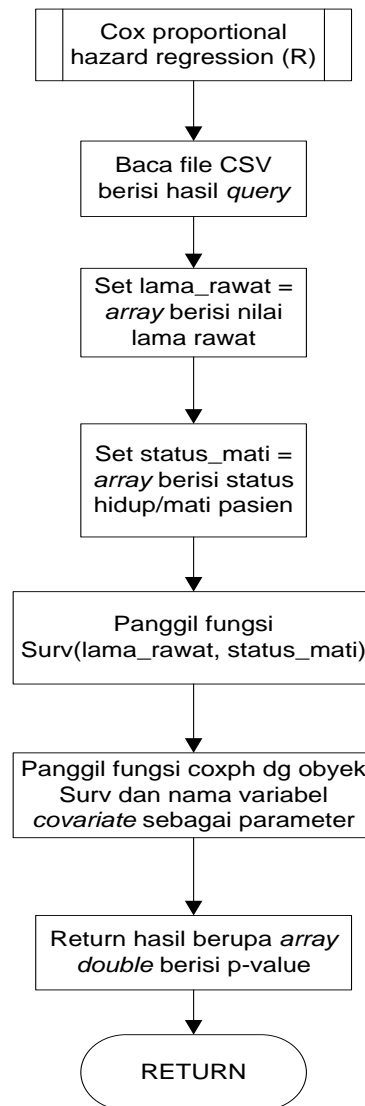


Gambar 3.15 *Flowchart Proses Survival Analysis*

3.5.6.1 Flowchart Cox Proportional Hazard Regression

Perhitungan *cox proportional hazard regression* dilakukan untuk mengetahui variabel apa saja yang mempengaruhi *survival* pasien. Perhitungan dilakukan di dalam R. Hasil perhitungan ditampilkan dalam bentuk tabel.

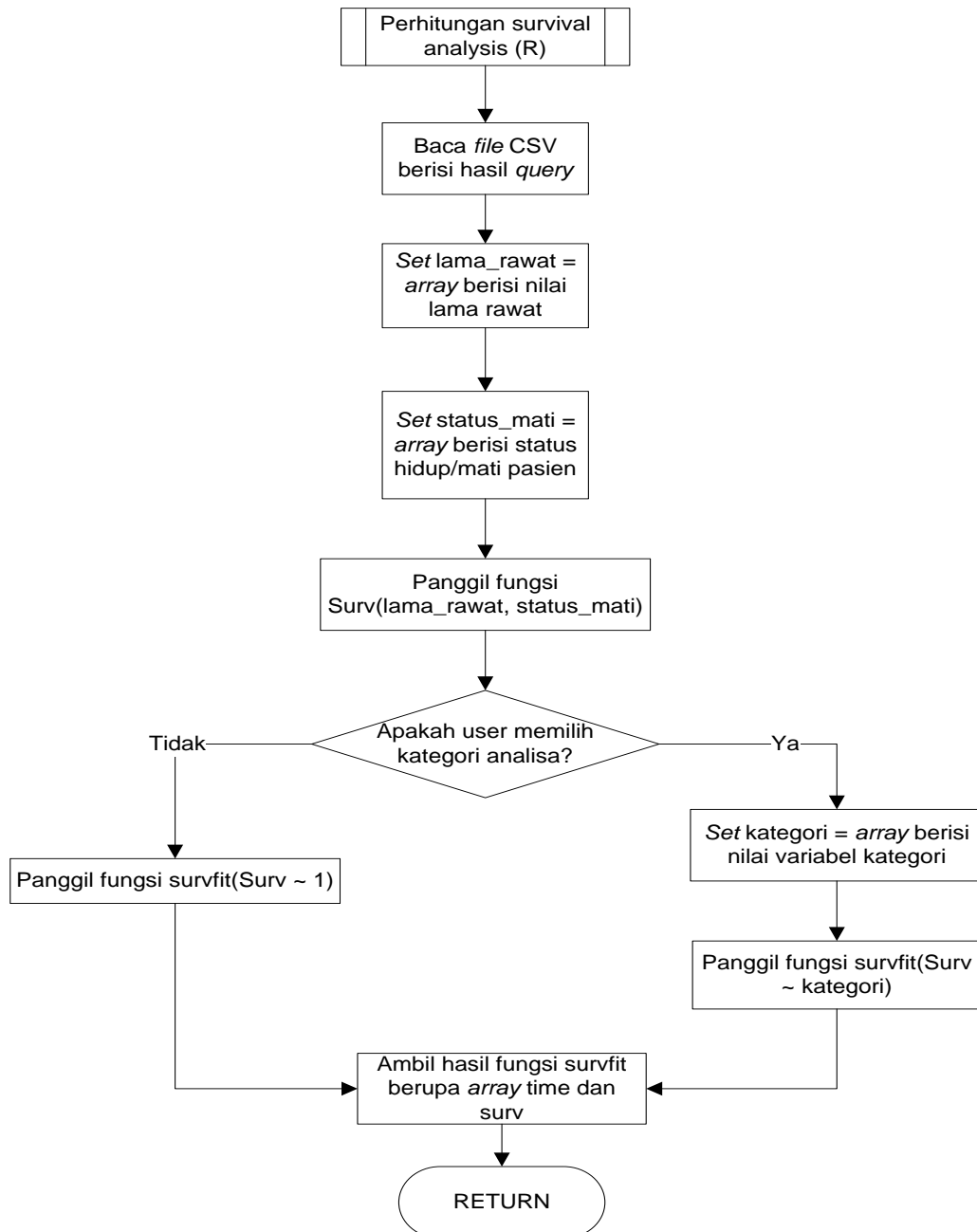
Flowchart proses perhitungan *cox proportional hazard regression* tampak pada Gambar 3.16.



Gambar 3.16 *Flowchart* Perhitungan *Cox Proportional Hazard Regression*

3.5.6.2 Flowchart Perhitungan Survival Analysis

Proses perhitungan *survival analysis* dijalankan setelah user menekan tombol OK pada form *survival analysis*. Proses perhitungan dilakukan di dalam R.



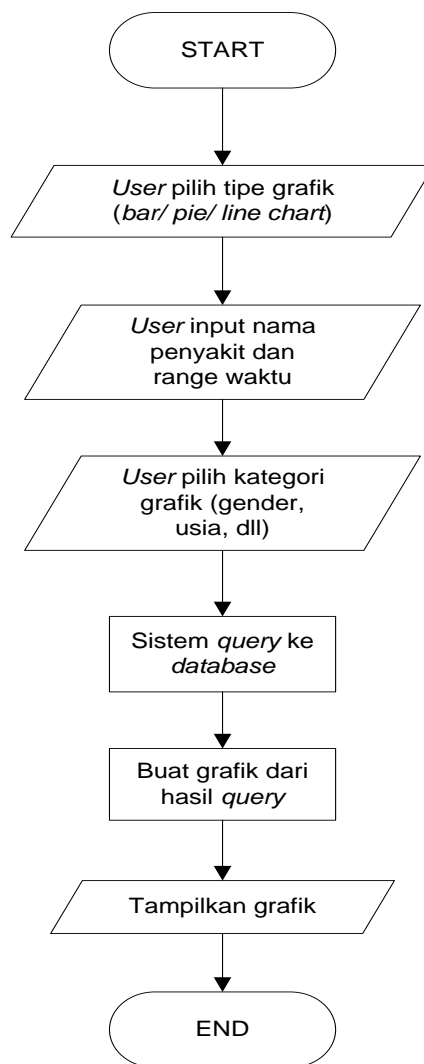
Gambar 3.17 Flowchart Proses Perhitungan Survival Analysis di R

3.5.7 Flowchart Pembuatan Grafik

Proses pembuatan grafik dibagi menjadi dua macam, yaitu pembuatan grafik *predefined* dan *custom*.

3.5.7.1 Flowchart Pembuatan Grafik *Predefined*

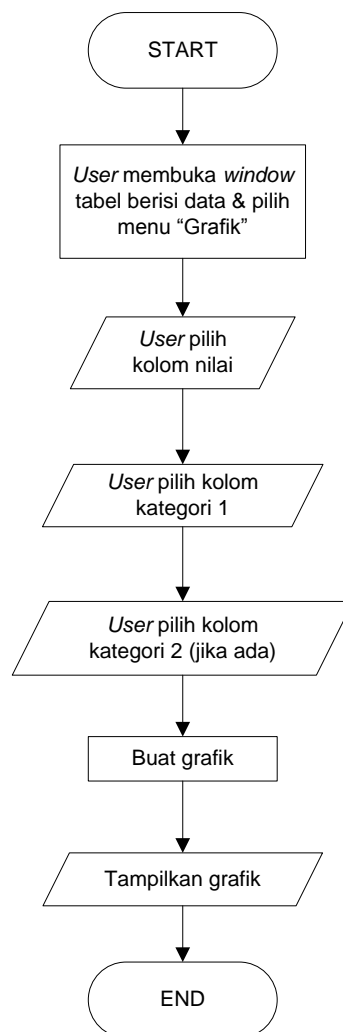
Untuk proses pembuatan grafik *predefined* hanya diberikan *flowchart* untuk pembuatan grafik jumlah pasien. Grafik lain memiliki alur proses yang mirip jadi tidak digambarkan di sini.



Gambar 3.18 *Flowchart* Pembuatan Grafik *Predefined*

3.5.7.2 Flowchart Pembuatan Grafik Custom

Untuk membuat grafik *custom*, *user* harus membuka *window* tabel berisi data yang ingin dibuat grafiknya. *Window* tabel bisa berasal dari hasil *query* menggunakan SQL Wizard (lihat subbab Desain *Form* SQL Wizard halaman 15). Kemudian *user* memilih menu “Grafik”. Setelah itu, *user* harus memilih kolom yang menyimpan nilai (misal kolom JUMLAH_PASIEN), kolom kategori pertama grafik (misal kolom GENDER), dan kolom kategori kedua (jika ada). Setelah itu, grafik akan ditampilkan pada *window* grafik.



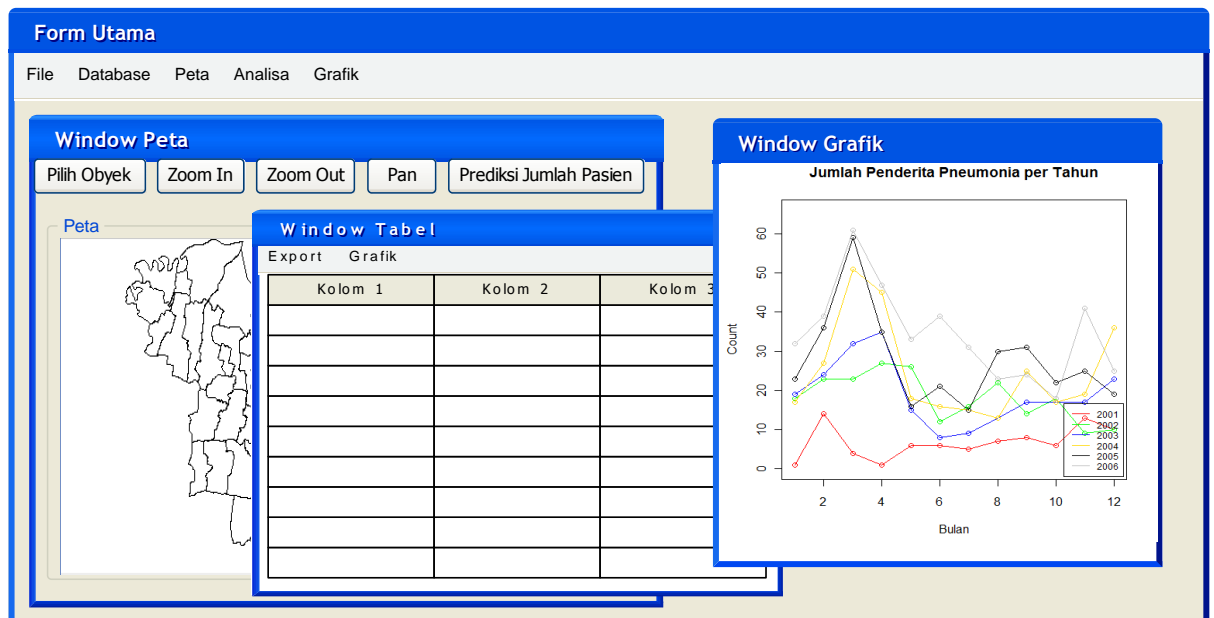
Gambar 3.19 Flowchart Pembuatan Grafik Custom

3.6 Desain User Interface Aplikasi

Subbab ini menjelaskan desain *form-form* yang ada pada *software*.

3.6.1 Desain Form Utama

Gambar 3.20 menunjukkan desain *form* utama. *Form* utama bertipe *form* MDI (*multiple document interface*), yang berarti *form* utama bisa memiliki beberapa *child window*. *Child window* ini bisa berupa *window* peta (yang menampilkan peta digital), *window* tabel (yang menampilkan tabel dari hasil *query* SQL ataupun tabel dari sebuah peta digital), dan *window* grafik (menampilkan grafik).



Gambar 3.20 Desain *Form* Utama

Form utama memiliki 5 menu. Berikut ini akan dijelaskan menu apa saja yang ada serta kegunaannya:

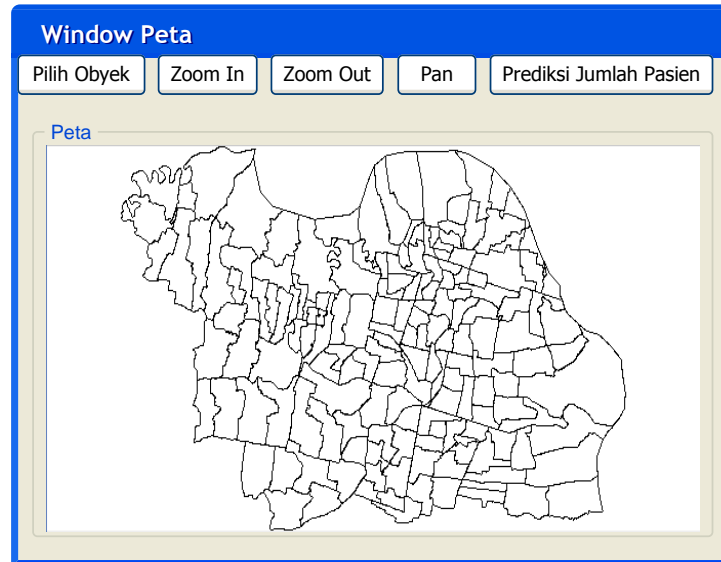
1. Menu File

- Menu *Open File* - digunakan untuk membuka *file* peta digital. Saat membuka peta, *user* bisa memilih apakah akan menampilkan peta atau isi tabel-nya.
- Menu *Close File* – digunakan untuk menutup *file* yang sedang dibuka oleh user

2. Menu Database
 - a. Menu Konek ke *Database*
 - b. Menu *Disconnect*
 - c. Menu *Data Preparation* Kriging – digunakan untuk tahap *data preparation* Kriging
 - d. Menu *SQL Wizard* – digunakan jika user ingin melakukan *query* SQL dengan menggunakan *wizard* seperti di Microsoft Access
 - e. Menu Koreksi Data Alamat
3. Menu Peta
 - a. Menu *Layer and Style Control* – digunakan untuk *manage layer* pada peta, dan juga untuk mengatur tampilan peta
 - b. Menu *Show Legend* – digunakan untuk menampilkan legenda pada peta
 - c. Menu *Show All Layer* – digunakan untuk *re-set zoom* peta kembali ke keadaan awal (semua obyek ditampilkan)
 - d. Menu *Select By Attribute* – digunakan untuk memilih obyek pada peta berdasarkan kriteria tertentu (misal memilih area yang jumlah penduduknya > 10.000 orang)
4. Menu Analisa
 - a. Menu Kriging
 - b. Menu Survival Analysis
5. Menu Grafik – digunakan untuk menampilkan grafik *predefined*. Jika user ingin membuat grafik *custom*, maka user bisa menggunakan menu grafik pada *window* tabel.
 - a. Grafik Jumlah Pasien per Bulan
 - b. Grafik Jumlah Pasien (dengan kategori)
 - c. *Boxplot* Lama Rawat
 - d. *Histogram* Usia Pasien

3.6.2 Desain *Window* Peta

Gambar 3.21 menunjukkan desain *window* peta. Pada *window* peta terdapat sejumlah *tool* yang bisa digunakan *user*, yaitu *tool selection* (untuk memilih obyek pada peta), *zoom in*, *zoom out*, *pan*, dan prediksi jumlah pasien.



Gambar 3.21 Desain *Window* Peta

3.6.3 Desain *Window* Tabel

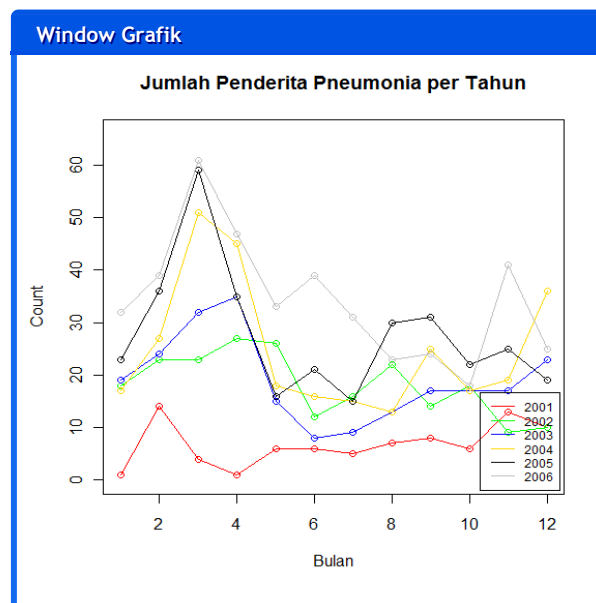
Window tabel digunakan untuk 2 hal: menampilkan data peta dalam bentuk tabel dan menampilkan data hasil query SQL. Menu ini memiliki menu Export dan Grafik. Menu Export digunakan untuk mengekspor data di tabel ke format Excel/ CSV. Menu Grafik digunakan untuk membuat grafik *custom*.

Window Tabel		
Export Grafik		
Kolom 1	Kolom 2	Kolom 3

Gambar 3.22 Desain Window Tabel

3.6.4 Desain Window Grafik

Window grafik digunakan untuk menampilkan semua jenis grafik, baik itu grafik *predefined*, ataupun grafik *custom*.

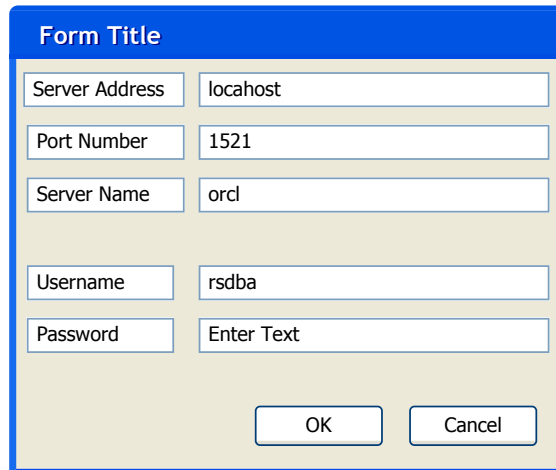


Gambar 3.23 Desain Window Grafik

3.6.5 Desain *Form* untuk Menu *Database*

3.6.5.1 Desain *Form* Konek ke *Database*

Gambar 3.24 menunjukkan desain *form* untuk koneksi ke *database*.



Form Title	
Server Address	localhost
Port Number	1521
Server Name	orcl
Username	rsdba
Password	Enter Text
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Gambar 3.24 Form Koneksi ke *Database*

3.6.5.2 Desain *Form* *Data Preparation* Kriging

Gambar 3.25 menunjukkan desain *form* untuk Kriging. *User* bisa memilih penyakit yang dianalisa dengan memasukkan nama penyakit dan menekan tombol tambah. Setelah itu, pilihan penyakit yang mengandung *string* yang diinputkan *user* akan muncul. Kemudian penyakit yang dipilih *user* akan ditambahkan pada tabel.

The image shows a software window titled "Form Title" with a light beige background and a blue border. At the top left, there is a text input field labeled "Nama Penyakit" and a "Tambah" button to its right. Below this is a table with two columns: "Kode Penyakit" and "Nama Penyakit". The table has 10 rows, with the first row containing headers and the rest being empty. To the right of the table is a "Hapus" button. Below the table is a dropdown menu labeled "Daftar tahun (u/ data jumlah penduduk)". At the bottom left, there is a text input field labeled "Path file output" and a "Set File" button to its right. At the very bottom, there are "OK" and "Cancel" buttons.

Gambar 3.25 Desain *Form Data Preparation* Kriging

3.6.5.3 Desain *Form* Koreksi Data Alamat

Gambar 3.26 menunjukkan desain *form* perbaikan data alamat. Daftar alamat yang harus diperbaiki ditampilkan pada tabel di sebelah kiri. *User* bisa menekan tombol View Saran untuk menampilkan daftar saran perbaikan. Setelah *user* memilih salah satu saran, *user* bisa menekan tombol *update* untuk meng-*update* data di database.

Form Title

Tampilkan data yang kelurahannya kosong saja?

Jumlah Item Saran

Daftar Alamat yang Perlu Diperbaiki

Saran perbaikan (ditampilkan dari yang paling mirip)

View Saran

Update Otomatis

Alamat Baru

Kelurahan Baru

Update

Update data lain yang nama jalannya sama?

Gambar 3.26 Desain *Form* Perbaikan Data Alamat

3.6.6 Desain *Form* untuk Menu Analisa

3.6.6.1 Desain *Form* Kriging

Gambar 3.27 menunjukkan desain *form* Kriging. Alur kerja proses ditunjukkan pada Gambar 3.4. Setelah user menekan tombol OK, proses Kriging akan berjalan. Kemudian *form* pengaturan warna peta akan muncul. Desain *form* pengaturan warna peta tampak pada Gambar 3.28.

Gambar 3.27 Desain *Form Kriging*

Nilai	Warna

Gambar 3.28 Desain *Form Pengaturan Warna*

3.6.6.2 Desain *Form Survival Analysis*

Gambar 3.29 menunjukkan desain *form* survival analysis. *User* harus menginputkan kode penyakit yang ingin dianalisa dengan menekan tombol “Input Nama Penyakit”. Kemudian *user* bisa memilih kategori analisa pada tabel. Jika *user* ingin mengubah *query* yang digunakan untuk mendapatkan data lama rawat,

dan status hidup/mati pasien, user bisa menekan tombol *custom query* (ini akan memunculkan SQL Wizard dari subbab 3.6.9).

Nama Variabel	P-Value

Gambar 3.29 Desain *Form Survival Analysis*

3.6.7 Desain *Form* untuk Menu Peta

3.6.7.1 Desain *Form Layer and Style Control*

Gambar 3.30 menunjukkan desain form *layer and style control*. Pada form ini, *user* bisa menambahkan/ menghapus *layer* peta. *User* juga bisa mengatur tampilan peta dengan menekan tombol Set Style, Set Selected Style (mengatur tampilan untuk obyek peta yang sedang terpilih), dan Set Label.

Form Layer & Style Control

Daftar Layer

Nama Layer	Visible?

Buttons: Add Layer, Hapus Layer, Set Style, Set Selected Style, Set Label

Gambar 3.30 Desain *Form Layer and Style Control*

3.6.7.2 Desain *Form Select by Attribute*

Gambar 3.31 menunjukkan desain *form Select by Attribute*. Pada *form* ini, *user* bisa mengisi kriteria untuk memilih obyek pada peta. *User* harus memilih tabel dan kolom yang digunakan. Setelah *user* menekan tombol OK, obyek pada peta yang memenuhi kriteria akan dipilih.

Form Select By Attribute

Daftar Tabel

Daftar Kolom

Operator

Buttons: =, !=, LIKE, AND, >, >=, IN (), OR, <, <=, NOT

Kondisi Filter

JUMLAH_PENDUDUK > 10.000

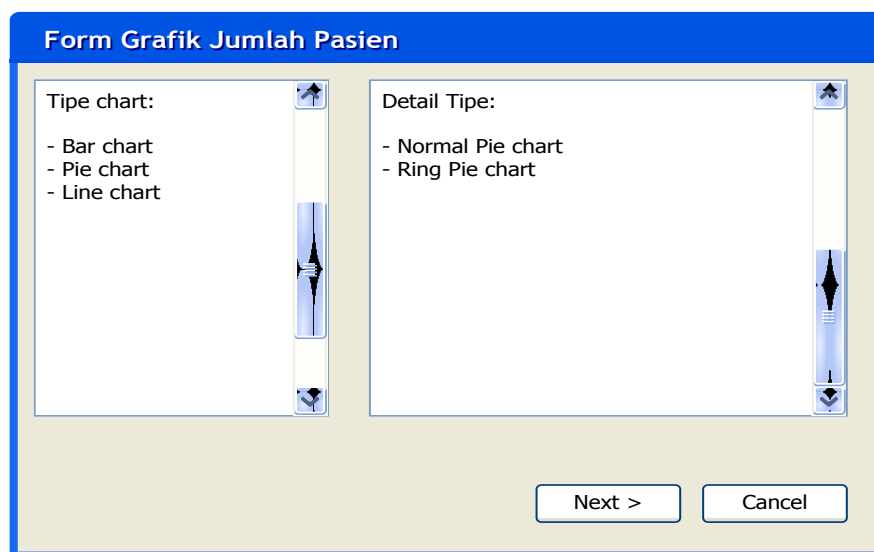
Buttons: OK, Cancel

Gambar 3.31 Desain *Form Select by Attribute*

3.6.8 Desain *Form* untuk Menu Grafik

3.6.8.1 Desain *Form* Grafik Jumlah Pasien (dengan Kategori)

Form untuk pembuatan grafik jumlah pasien dibagi menjadi 3 halaman. Pada halaman pertama (Gambar 3.32), *user* memilih jenis grafik yang diinginkan. Pada halaman kedua (Gambar 3.33), *user* harus memasukkan kode penyakit yang diinginkan dan *range* waktu (jika dikosongi berarti mengambil semua data). Pada halaman ketiga (Gambar 3.34), *user* memilih kategori grafik.



The image shows a software dialog box titled "Form Grafik Jumlah Pasien". The dialog is divided into two main sections. The left section, labeled "Tipe chart:", contains a list of three options: "- Bar chart", "- Pie chart", and "- Line chart". The right section, labeled "Detail Tipe:", contains a list of two options: "- Normal Pie chart" and "- Ring Pie chart". Both sections have vertical scroll bars. At the bottom right of the dialog, there are two buttons: "Next >" and "Cancel".

Gambar 3.32 Desain *Form* Grafik Jumlah Pasien Halaman 1

Form Grafik Jumlah Pasien

Nama Penyakit

Kode Penyakit	Nama Penyakit

Tanggal Awal

Tanggal Akhir

Gambar 3.33 Desain *Form* Grafik Jumlah Pasien Halaman 2

Form Grafik Jumlah Pasien

Kategori 1:

- Gender
- Usia
- Pekerjaan
- dll

Kategori 2 (series):

- Gender
- Usia
- Pekerjaan
- dll

Gambar 3.34 Desain *Form* Grafik Jumlah Pasien Halaman 3

3.6.8.2 Desain *Form* Grafik Histogram Usia Pasien

Gambar 3.35 menunjukkan desain *form* histogram usia pasien. Desain *form* mirip dengan desain *form* grafik jumlah pasien halaman 2 (Gambar 3.33),

tetapi perbedaannya adalah dalam *form* histogram *user* bisa mengatur *range* histogram dengan menekan tombol Atur Range. Jika tombol Atur Range ditekan, maka form Atur Range Histogram akan muncul (Gambar 3.36).

The screenshot shows a window titled "Form Histogram Usia Pasien". At the top left, there is a text input field labeled "Nama Penyakit" and a "Tambah" button. Below this is a table with two columns: "Kode Penyakit" and "Nama Penyakit". The table has several empty rows. To the right of the table is a "Hapus" button. At the bottom left, there are two date input fields: "Tanggal Awal" and "Tanggal Akhir". At the bottom right, there are three buttons: "Atur Range", "OK", and "Cancel".

Gambar 3.35 Desain *Form* Histogram Usia Pasien

The screenshot shows a window titled "Form Title". It contains several input fields: "Jumlah Bin", "Nilai minimum", and "Nilai maksimum". There is also a dropdown menu labeled "Cara pembagian range". Below these is a table with two columns: "Nilai awal bin" and "Nilai akhir bin". The table has several empty rows. At the bottom, there are two buttons: "OK" and "Cancel".

Gambar 3.36 Desain *Form* Atur Range Histogram

3.6.8.3 Desain *Form* untuk Grafik Lain

Untuk grafik lainnya (grafik jumlah pasien per bulan, *boxplot* lama rawat, grafik *scatter plot*) desain *form*-nya mirip dengan desain *form* grafik jumlah pasien halaman 2 (Gambar 3.33). User harus meng-input penyakit yang mau dibuat grafiknya dan mengisi *range* waktu. Desain *form* untuk grafik yang lainnya tidak ditampilkan di sini.

3.6.9 Desain *Form* SQL Wizard

Form SQL Wizard digunakan jika *user* ingin melakukan *query* ke *database*. *Form* ini diperlukan jika *user* ingin membuat grafik sendiri (yang tidak ada di menu grafik), atau jika *user* ingin melakukan tahap *data preparation* untuk Kriging atau *survival analysis* dengan *query* yang berbeda dengan *query default* (misalkan suatu saat ada perubahan pada tabel yang lama sehingga *query* untuk Kriging dan *survival analysis* harus diubah).

User interface form ini mirip dengan *user interface query wizard* pada Microsoft Access. *User* tidak perlu mengerti *syntax* SQL untuk bisa melakukan *query*.

Form ini dibagi menjadi 6 halaman, yaitu:

1. Halaman Pemilihan Tabel – di halaman ini *user* memilih tabel yang digunakan. *User* tidak perlu mengatur kondisi *join* antar tabel (otomatis dideteksi berdasarkan *primary key/foreign key* tabel).

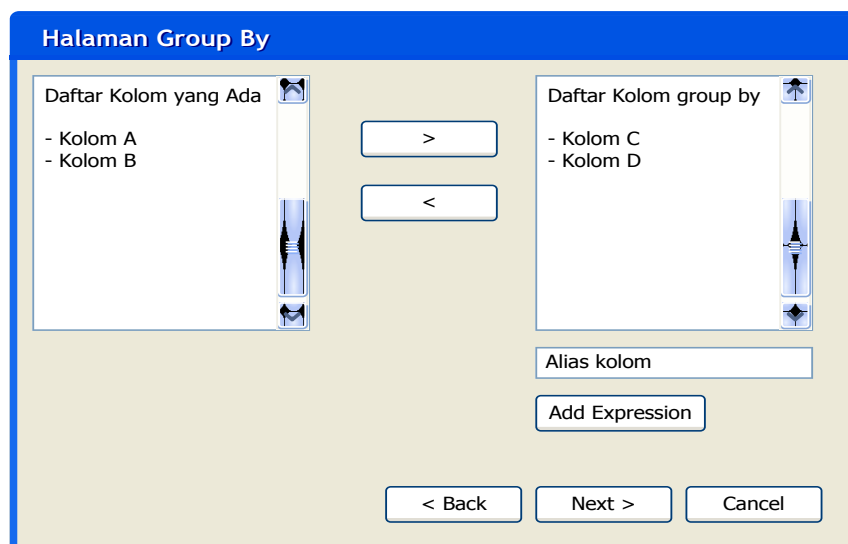
Gambar 3.37 Desain *Form* SQL Wizard Halaman Pemilihan Tabel

2. Halaman Group By – di halaman ini user memilih kolom untuk bagian GROUP BY pada SQL. User boleh tidak memilih kolom apapun. Tersedia tombol untuk maju ke halaman berikutnya dan tombol untuk kembali ke halaman sebelumnya.



Gambar 3.38 Desain *Form* SQL Wizard Halaman Group By

3. Halaman Select – di halaman ini *user* memilih kolom untuk ditampilkan (sama seperti bagian SELECT pada SQL). Selain memilih dari daftar kolom yang ada, *user* juga bisa menambahkan *expression* yang terdiri dari gabungan beberapa kolom (misal SELECT gaji + bonus * 0.1)/ gabungan kolom dengan fungsi SQL (misal SELECT UPPER_CASE('haLO')) dengan menekan tombol Add Expression



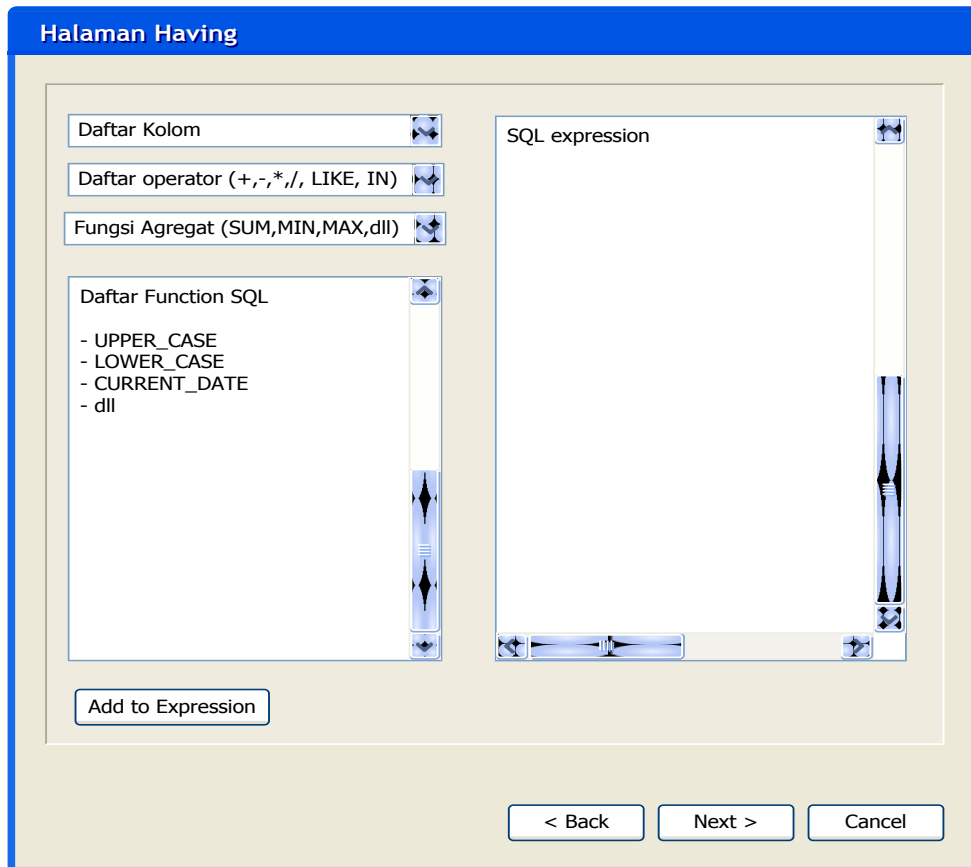
Gambar 3.39 Desain *Form* SQL Wizard Halaman SELECT

- Halaman Where – di sini *user* bisa memberi *filter* hasil *query* (sama seperti bagian WHERE pada SQL). Jika *user* tidak mengisi apa-apa, maka semua data akan ditampilkan

The image shows a software interface titled "Halaman Where". It features several interactive elements: a list of columns, a list of operators, a list of aggregate functions, and a list of SQL functions. A large text area on the right is for entering the SQL expression. Navigation buttons for "Back", "Next", and "Cancel" are located at the bottom.

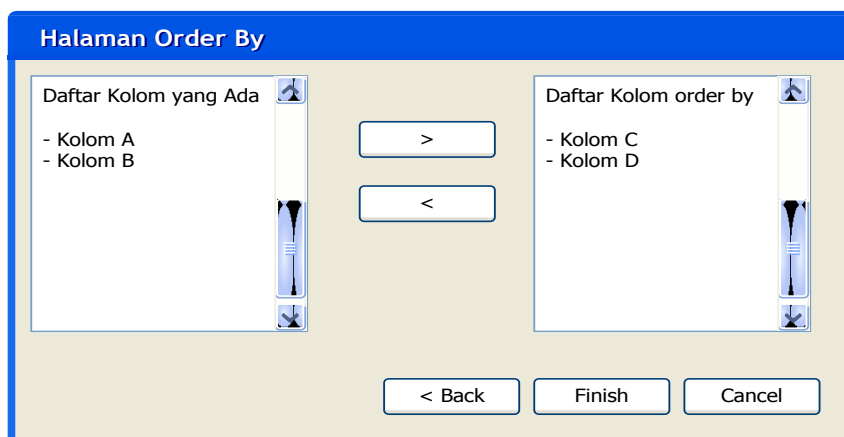
Gambar 3.40 Desain *Form* SQL Wizard Halaman Where

- Halaman Having – halaman ini akan muncul hanya jika *user* mengisi kolom *group by* pada halaman 2. Fungsinya mirip dengan halaman *where*, yaitu untuk melakukan *filter* data.



Gambar 3.41 Desain *Form* SQL Wizard Halaman Having

6. Halaman Order By – di halaman ini user bisa memilih hasil query diurutkan berdasarkan kolom apa.



Gambar 3.42 Desain *Form* SQL Wizard Halaman Order By