



UNIVERSITAS INDONESIA

**REPRESENTASI DATA SPASIAL DAN BAHASA QUERY
SPASIAL DENGAN POSTGRESQL**

SKRIPSI

**KARLINA
0305010319**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI MATEMATIKA
DEPOK
JULI 2010**



UNIVERSITAS INDONESIA

**REPRESENTASI DATA SPASIAL DAN BAHASA QUERY
SPASIAL DENGAN POSTGRESQL**

SKRIPSI


Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains

**KARLINA
0305010319**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI MATEMATIKA
DEPOK
JULI 2010**

DEPOK
JULI 2010

HALAMAN PERNYATAAN ORISINALITAS



Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Karlina

NPM : 0305010319

Tanda Tangan :

Tanggal : 5 Juli 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Karlina
NPM : 0305010319
Program Studi : Matematika
Judul Skripsi : Representasi data spasial dan bahasa query spasial dengan PostgreSQL

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dra. Yahma Wisnani, M.Kom (.....)
Pembimbing : Dr. Rokhmatuloh (.....)
Penguji : Dra. Siti Aminah, M.Kom (.....)
Penguji : Drs. Suryadi, M.T (.....)
Penguji : Dr. Yudi Satria, M.T (.....)

Ditetapkan di : Depok
Tanggal : 5 Juli 2010

KATA PENGANTAR

Alhamdulillahirabil'aalamiin, segala puji dan syukur saya panjatkan hanya kepada Allah penguasa semesta alam yang telah memberikan rahmat, dan nikmat-Nya. Dengan izin-Nya saya dapat mengerjakan skripsi ini mulai dari memilih pembimbing, topik, program, proses penulisan sampai ujian, semua itu terlaksana tidak lain hanya karena petunjuk dan pertolongan-Nya. Adapun penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Sains Jurusan Matematika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

Tulisan yang sederhana sekalipun, pasti ada kontribusi orang lain didalamnya. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

Almarhum Ayah saya yang semasa hidupnya senantiasa berjuang tanpa lelah untuk keluarga, semoga senantiasa dalam cahaya-Nya dan dikumpulkan bersama orang-orang sholeh di halaman syurga-Nya. Ibu dan Kakak saya, terima kasih atas curahan kasih sayang yang begitu besar, serta dukungan finansial dan iringan do'a yang selalu mengiringi langkah saya.

Ibu Yahma Wisnani dan Bapak Rokhmatuloh selaku dosen pembimbing yang telah banyak membantu dan menyediakan waktu, tenaga, pikiran untuk mengarahkan saya dalam penyusunan skripsi ini, juga pengarahan hidup dan motivasi.

Ibu Siti Nurrohmah selaku pembimbing akademis yang telah mendukung dan membimbing saya selama menjalani perkuliahan di Fakultas Matematika dan Ilmu Pengetahuan Alam.

Mohammad Layin suami kebanggaan atas segala perhatian, kasih sayang, dan limpahan pengertian yang membuat saya tak berhenti berbahagia dan bersyukur, dan mohon maaf jika selama ini saya begitu merepotkan dan belum bisa banyak berbuat, semoga keberkahan melimpahi keluarga kita di dunia dan di

akhirat. Ibu mertua dan ayah mertua serta adik-adik ipar yang baik hati terima kasih atas do'a dan dukungannya.

Segegap dosen, staf tata usaha, dan penguji, Bapak Suryadi, Bapak Yudi, Bapak Ari, dan Ibu Siti Aminah yang telah banyak membantu selama saya kuliah di Departemen Matematika sampai ujian berlangsung.

Orang-orang yang belum pernah saya temui namun selalu siap membantu, Bapak Firman Hadi dosen ITB selaku pembimbing skripsi secara implisit yang tak pernah bosan menjawab pertanyaan saya walaupun hanya lewat dunia maya. Juga Hendra Waskita mahasiswa fasilkom atas inspirasi skripsinya.

Teman-teman dan adik-adik di Departemen Matematika. Teman-teman seangkatan 2005, Nurma, Aya, Yuni, Mia, Desti, Shally, Yanu, Trian, Inul. Teman-teman seperjuangan yang menyelesaikan tugas akhir pada semester ini, Kiki, Fia, Bian, Dian, Hamdan, Hairu, Gyo, Ote, Nisma, Melati, Fika. Adik-adik angkatan 2006, Nisa, Dian, Yuri, Yuko, Lena, Helmet, Tami, Ita, Puspa. Adik-adik angkatan 2007, Arif, Hikmah, Widya, Nedia, Amanda yang selalu memberi semangat.

Teman-teman jurusan Geografi, Yuli, Anin, Weni, Toni, Amri yang telah banyak memberikan data-data dan pengarahan yang bermanfaat. Rekan-rekan FMA. Saudari satu lingkaran dan murobbiyah yang membuat hati ini selalu berwarna. Sahabatku Dea terima kasih atas do'a dan sms semangatnya. Segegap staf dan pengajar Nurul Fikri atas dispensasi jadwal mengajar. Adik-adik penghibur hati Cempaka delapan belas, Shinta, Tami, Moki, Ria, Laili, Lina, Sekar, dan barisan Rakarapati. Pihak-pihak lain yang belum disebutkan yang telah membantu dalam penulisan skripsi ini.

Akhir kata saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu matematika.

Depok, 5 Juli 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Karlina
NPM : 0305010319
Program Studi : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

“ Representasi data spasial dan bahasa query spasial dengan PostgreSQL ”

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 5 Juli 2010

Yang menyatakan,

(Karlina)

ABSTRAK

Nama : Karlina
Program Studi : Matematika
Judul : Representasi Data Spasial dan Bahasa Query Spasial dengan PostgreSQL

Data spasial atau data geografi memiliki peranan penting dalam kehidupan manusia. Pada awalnya, data spasial disimpan dengan menggunakan peta, namun seiring berkembangnya teknologi dan kebutuhan data yang cepat, data spasial diambil, disimpan, dan diolah dengan teknologi komputer yang dinamakan Sistem Informasi Geografi (SIG). Dengan semakin meningkatnya volume data geografi, maka dibutuhkan suatu sistem yang dapat menyimpan sekaligus mengelola dengan efisien untuk kebutuhan suatu badan swasta atau pemerintah. *Database Management System* (DBMS) adalah pilihan yang tepat untuk melakukan tugas tersebut. DBMS relasional yang biasa digunakan dalam dunia bisnis tidak sesuai untuk diaplikasikan pada tipe data spasial, sehingga diperlukan perluasan teknologi DBMS untuk informasi spasial. Untuk mengimplementasikan dalam program komputer, PostgreSQL merupakan *open source* basis data yang dapat menyimpan tipe data kompleks seperti data spasial beserta bahasa query spasial untuk tahap manipulasi data.

Pada tugas akhir ini dideskripsikan representasi data spasial dalam basis data dan bahasa query spasial yang digunakan dalam pengambilan informasi spasial dari basis data dan melakukan implementasi dengan sistem basis data *object-relational* PostgreSQL.

Kata kunci : Data Spasial, basis data spasial, bahasa query spasial, *Structured Query Language*, Sistem Informasi Geografi, PostgreSQL
viii+52 halaman : 29 gambar; 5 tabel
Daftar Pustaka : 13 (1991-2010)

ABSTRACT

Name : Karlina
Program Study : Mathematic
Title : Spatial Data Representation and Spatial Query Language with PostgreSQL

Spatial data or geographic data has an important role in human life. In the beginning, spatial data stored within the map, but with a growing technology and the need for fast data, spatial data captured, stored and analyzed with computer technology called Geographic Information Systems (GIS). Due to the ever-increasing volume of geographic data, so needed a system that can store both manage it efficiently to the needs of a private or government entity. Database Management System (DBMS) is the perfect choice to perform that task. Relational DBMS which is used in the business world is not suitable to be applied on spatial data types, so that the necessary to extend of DBMS technology for spatial information. To implement in computer programs, PostgreSQL is an open source database that can store complex data types such as spatial data along with the query language for spatial data manipulation stages. In this thesis described the representation of spatial data in a database and spatial query language used in the retrieval of spatial information from the database and implement the computer system PostgreSQL database.

Keywords : Spatial data, spatial database, query language, Structured Query Language, Geographic Information System, PostgreSQL
ix+52 pages : 29 pictures; 5 tables
Bibliography : 13 (1991-2010)

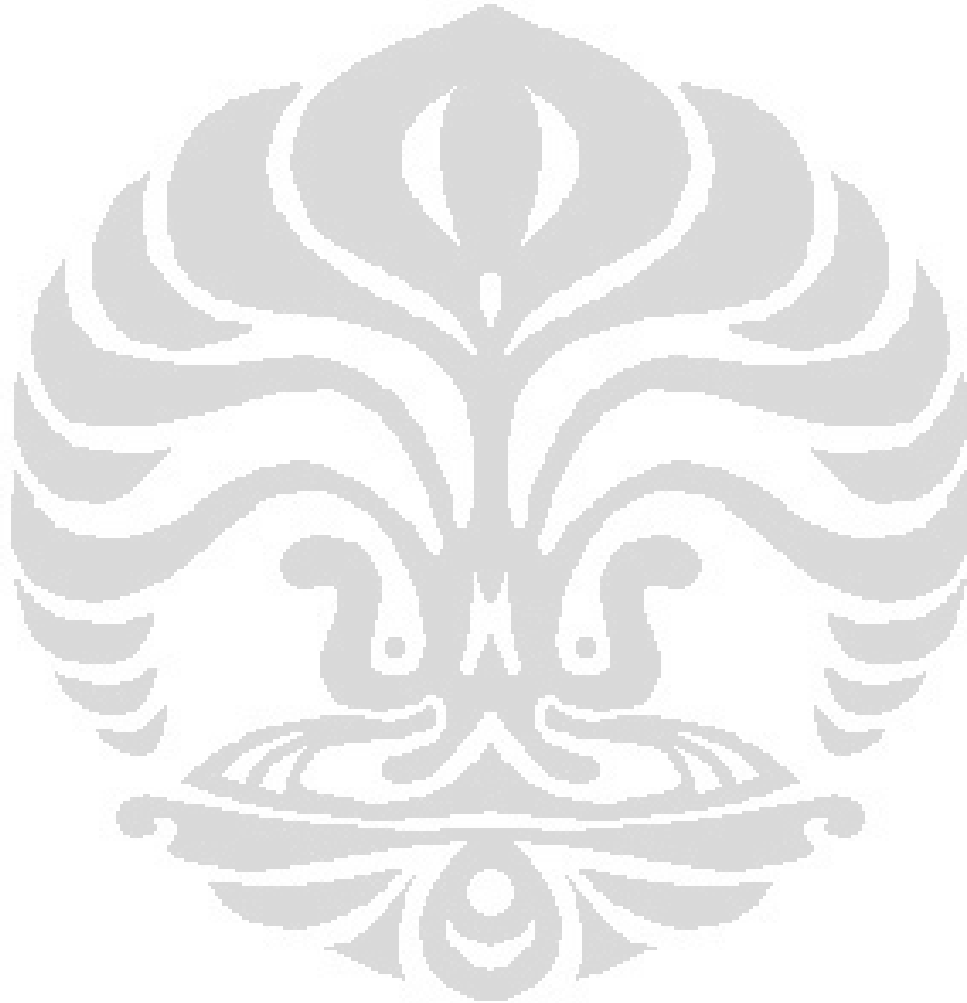
DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Ruang Lingkup Penelitian	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penelitian	3
BAB 2 LANDASAN TEORI	5
2.1 Data Spasial	5
2.1.1 Model Data Spasial	5
2.1.1.1 Model Data Vektor	5
2.1.1.2 Model Data Raster	6
2.2 Basis Data	7
2.2.1 Model <i>Entity-Relationship</i>	8
2.2.2 Model <i>Object-Oriented</i>	9
2.2.3 Model Basis Data <i>Object-Relational</i>	9
2.2.4 Bahasa Query	10
2.2.4.1 Aljabar Relasional	10
2.2.4.2 <i>TupleRelational Calculus</i>	11
2.2.4.3 SQL (<i>Structured Query Language</i>)	12
2.3 Basis Data Spasial	13
2.3.1 Indeksing Data Spasial	13
2.3.2 Indeks Spasial <i>R-Tree</i>	14
2.3.3 Hubungan Topologi Spasial	15
2.4 Sistem Informasi Geografi	18
2.5 PostgreSQL	20

BAB 3 REPRESENTASI DATA SPASIAL DAN BAHASA QUERY	
3.1 Representasi Dimensi Objek Spasial	21
3.2 Representasi Objek Spasial dalam Model Vektor	22
3.2.1 Struktur Notasi	23
3.3 Representasi Objek Spasial dalam Basis Data	25
3.3.1 Model <i>Entity Relationship</i> DKI Jakarta	25
3.4 Bahasa Query Spasial	27
3.4.1 Perbandingan Query dalam SQL,Aljabar Relasional dan Tuple Relational Calculus	27
3.4.2 SQL untuk Data Spasial	29
3.4.3 Data Manipulation Language	21
3.4.4 SQL untuk Data Spasial	23
BAB 4 POSTGRESQL	
4.1 Skema dalam Basis Data	32
4.2 Penyimpanan Data dalam PostgreSQL	35
4.3 <i>Data Definition Language</i> pada PostgreSQL	37
4.4 <i>Data Manipulation Language</i> pada PostgreSQL	37
BAB 5 KESIMPULAN DAN SARAN	
5.1 Kesimpulan	43
5.2 Saran	43
DAFTAR PUSTAKA	44

DAFTAR TABEL

Tabel 2.1 Enam belas spesifikasi hubungan topologi spasial berdasar pada kriteria <i>empty</i> dan <i>non-empty</i> perpotongan <i>boundary</i> dan interior	7
Tabel 2.2 Pembagian kerja antara SDBMS dan SIG (Yeung, 2007)	19
Tabel 3.1 Representasi data spasial dalam basis data <i>object-relational</i>	20
Tabel 3.2 Operasi Spasial dalam standar OGC untuk SQL	25
Tabel 4.1 Geometri envelope dari kelurahan Ciracas	47

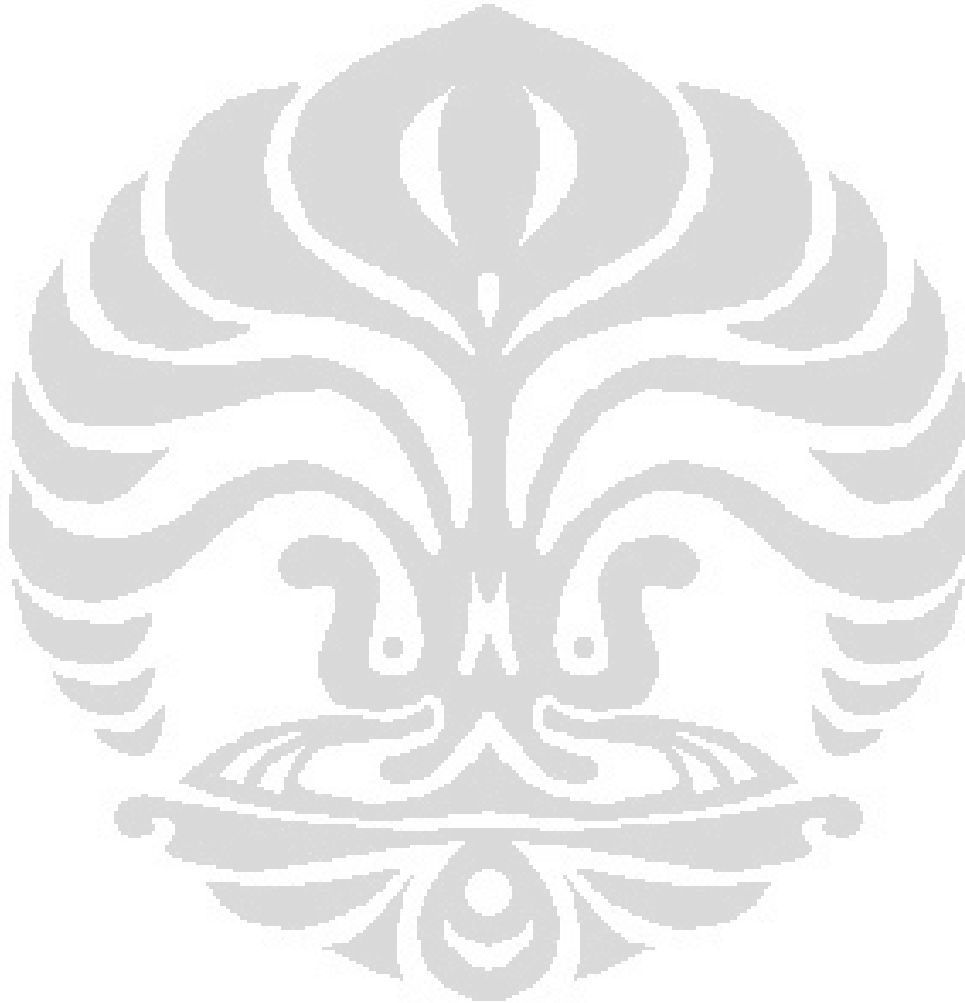


DAFTAR GAMBAR

Gambar 2.1 Data vektor	6
Gambar 2.2 Data Raster	6
Gambar 2.3 Diagram E-R untuk sungai-sungai yang terdapat di suatu kota.	9
Gambar 2.4 Contoh <i>minimum bounding box</i> dari objek spasial	14
Gambar 2.5 R-Tree untuk gambar 2.4	15
Gambar 2.6 Delapan hasil dari hubungan topologi spasial	17
Gambar 3.1 Contoh Objek Dimensi 1	21
Gambar 3.2 Contoh Objek Dimensi 2	22
Gambar 3.3 Representasi Vektor untuk Poligon	23
Gambar 3.4 Contoh Polyline	24
Gambar 3.5 Contoh Polygon	24
Gambar 3.6 Diagram ER Skema DKI Jakarta.....	26
Gambar 4.1 Peta Administrasi DKI Jakarta	33
Gambar 4.2 Peta Banjir tahun 2002 DKI Jakarta	34
Gambar 4.3 Peta Jalan Tol dan Jalan Utama di DKI Jakarta	34
Gambar 4.4 Peta Lokasi Fasilitas Umum di DKI Jakarta	35
Gambar 4.5 Tabel Entitas Administrasi.....	36
Gambar 4.6 Tabel Entitas Jalan.....	36
Gambar 4.7 Tabel Entitas Fasilitas Umum	36
Gambar 4.8 Output visual query envelope	47
Gambar 4.9 Output visual query boundary	47
Gambar 4.10 Hasil query touch	48
Gambar 4.11 Hasil query cross	48
Gambar 4.12 Hasil query contains	49
Gambar 4.13 Hasil query overlap	50
Gambar 4.14 Output visual query overlap	51
Gambar 4.15 Hasil query buffer	51
Gambar 4.16 Output visual query convexhull	52
Gambar 4.17 Hasil query intersection dan length	52

DAFTAR LAMPIRAN

Lampiran 1 Algoritma pencarian titik dengan <i>R-Tree</i>	46
Lampiran 2 Hasil query spasial dengan PostgreSQL	47



BAB 1 PENDAHULUAN

1.1.Latar Belakang

Sebuah organisasi atau perusahaan besar maupun menengah telah mempelajari dan mengetahui bahwa informasi spasial dengan pemetaan dapat menjadi sumber yang akurat untuk menganalisis data. Dengan teknologi komputer yang disebut Sistem Informasi Geografi (SIG), perusahaan dapat dengan mudah memperoleh informasi spasial atau informasi sebuah lokasi. Misalnya sebuah perusahaan yang bergerak dibidang kuliner masakan Italia seperti pizza ingin mengembangkan usahanya dengan menambah cabang baru di suatu kota, hal ini dapat dilakukan dengan mem-plot alamat para penggemar pizza kedalam sebuah peta, pemilik restoran pizza dapat melihat dimana lokasi yang tepat untuk menambah cabang restoran mereka. Namun informasi tersebut belumlah cukup, harus ditambahkan informasi pendapatan, karena akan lebih baik jika penempatan cabang berada di daerah para penggemar pizza yang mempunyai pendapatan tinggi. Mereka dapat menambahkan tanda tertentu dalam peta lokasi, namun menjadi lebih kompleks, belum lagi kebutuhan informasi jumlah keluarga, tingkat pemesanan pizza, dan sebagainya. Selain harus fokus pada peta lokasi, mereka tetap harus memperhatikan variabel-variabel lain. Hal ini mengakibatkan pengambilan data dan analisis data menjadi kurang efisien.

Saat ini data disimpan dan diolah dengan basis data. Basis data adalah kumpulan data yang saling berhubungan. Pada kasus restoran pizza, basis data dapat menjadi solusi untuk menyimpan dan mengolah data spasial dan non spasial secara bersamaan yang selanjutnya disebut basis data spasial. Basis data spasial memberikan perangkat untuk penyimpanan dan manipulasi. Basis data spasial tidak mengharuskan untuk menampilkan data secara visual. Pengelola dapat menyimpan berbagai atribut masyarakat pecinta pizza seperti level pendapatan, jumlah anak, jumlah pemesanan, taburan pizza yang disukai di daerah tersebut, dan sebagainya.

Basis data spasial memiliki bahasa query spasial untuk berinteraksi dengan penggunaanya. Query spasial akan memberi jawaban seperti pertanyaan “berikan nama-nama pecinta pizza yang letak rumahnya 5 kilometer dari setiap cabang restoran dan nama cabangnya” atau “berikan nama penggemar pizza yang paling banyak melakukan pemesanan pizza pada bulan Mei”. Selain itu basis data spasial juga memiliki indeksing spasial yang membuat proses query spasial menjadi lebih cepat.

Model basis data yang dapat menyimpan objek geometri adalah basis data *object-relational*. Saat ini telah banyak program komputer basis data yang menyediakan sistem basis data *object-relational*. Contohnya PostgreSQL, Oracle dan ArcGIS. Namun dari ketiganya, PostgreSQL merupakan *open source* atau program komputer yang bisa diunduh oleh siapa saja tanpa mengeluarkan biaya pembelian. PostgreSQL termasuk *open source* yang paling banyak digunakan sekarang ini karena kelebihan dari fitur-fitur yang dimiliki termasuk fitur data spasial.

Tugas akhir ini mendeskripsikan representasi data spasial dan bahasa query spasial yang digunakan dengan menggunakan sistem basis data PostgreSQL.

1.2 Perumusan Masalah

Dalam melakukan penyimpanan basis data spasial dan manipulasi data dilakukan pemecahan masalah menjadi beberapa masalah yang lebih kecil dan sederhana.

Beberapa masalah tersebut:

- Bagaimana representasi data spasial dalam basis data
- Bagaimana formula bahasa query spasial
- Bagaimana PostgreSQL menyimpan dan memanipulasi data

1.3 Tujuan penelitian

Tujuan akhir penelitian ini adalah mendeskripsikan representasi data spasial dan bahasa query spasial dengan PostgreSQL.

1.4. Ruang Lingkup Penelitian

Batasan-batasan dalam tugas akhir ini adalah:

- Input data spasial menggunakan format data Arcview shp
- Tipe data yang digunakan adalah data vektor
- Data spasial yang digunakan adalah peta wilayah DKI Jakarta tahun 2002

1.5 Metodologi Penelitian

Metodologi dalam mengerjakan tugas akhir ini adalah:

- Studi literature
 Pada tahap ini penulis melakukan studi literature mengenai representasi objek spasial, bahasa *query* basis data, model basis data, hubungan antara basis data spasial dengan Sistem Informasi Geografi dan manual PostgreSQL.
- Implementasi sistem basis data spasial
 Pada tahap ini dilakukan implementasi kedalam PostgreSQL, proses implementasi terdiri dari:
 - Tahap penyimpanan basis data spasial.
 - Tahap manipulasi data dengan formula query spasial yang terdapat di PostgreSQL

1.6 Sistematika Penulisan

Sistematika dari penulisan laporan ini adalah

- Bab I Pendahuluan :
 Bab ini menjelaskan mengenai latar belakang diadakannya penelitian, perumusan masalah yang akan diteliti berdasar latar belakang, tujuan dari diadannya penelitian, ruang lingkup yang membatasi penelitian, metodologi penelitian, dan sistematika penulisan.
- Bab II Landasan Teori :
 Bab ini menjelaskan teori-teori yang digunakan dalam penelitian yaitu basis data spasial dan sistem informasi geografi.

- Bab III Representasi Data Spasial dan Bahasa Query :
Menjelaskan mengenai representasi data spasial dalam basis data dan bahasa query untuk mengolah data spasial yang telah disimpan.
- Bab IV Basis Data Spasial dengan PostgreSQL:
Menjelaskan mengenai cara penyimpanan dan manipulasi basis data spasial dengan menggunakan PostgreSQL
- Bab V Penutup
Bab ini berisi tentang kesimpulan dari proses penelitian. Selain itu terdapat saran untuk pengembangan aplikasi selanjutnya.



BAB 2

LANDASAN TEORI

2.1 Data Spasial

Data spasial adalah data yang dapat ditampilkan, dimanipulasi, dan dianalisis melalui atribut spasial yang menunjukkan lokasi permukaan bumi (K.W. Yeung, 2007).

Data spasial mempunyai dua property (K.W. Yeung, 2007) :

1. Mengacu pada *geographic space*, artinya data terdaftar pada sistem koordinat geografi yang telah ada.
2. Data spasial di representasikan dalam skala geografi yang bervariasi.

2.1.1 Model Data Spasial

Pada pemanfaatannya data spasial yang diolah dengan menggunakan komputer (data spasial digital) menggunakan model sebagai pendekatannya (Gumelar, 2004). *Economic and Social Commission for Asia and the Pasific* (1996), mendefinisikan model data sebagai suatu set logika atau aturan dan karakteristik dari suatu data spasial. Model data merupakan representasi hubungan antara dunia nyata dengan data spasial.

Terdapat dua model dalam data spasial, yaitu model data raster dan model data vektor. Keduanya memiliki karakteristik yang berbeda, selain itu dalam pemanfaatannya tergantung dari masukan data dan hasil akhir yang akan dihasilkan. Model data tersebut merupakan representasi dari obyek-obyek geografi yang terekam sehingga dapat dikenali dan diproses oleh komputer (Gumelar, 2004).

2.1.1.1 Model Data Vektor

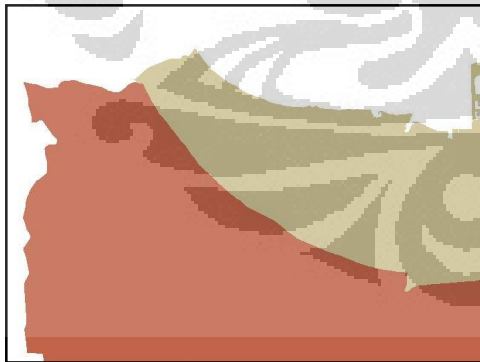
Dalam menyimpan informasi spasial seperti informasi jalan, bangunan, dan informasi geografis lainnya yang terdapat di dalam peta agar dapat diolah dan disimpan dalam suatu komputer dibutuhkan suatu tipe data tertentu. Tipe data vektor adalah salah satu tipe data yang digunakan dalam menyimpan informasi spasial tersebut.

Di dalam model data vektor, informasi spasial tersimpan di dalam komputer sebagai data koordinat kartesius. Data-data koordinat tersebut membentuk simbol simbol sebagai berikut (Riyanto, 2009) :

1. Data Titik (*Point*): Merupakan sepasang koordinat (X, Y) .
2. Data Garis (*Line*): Terdiri dari 2 buah koordinat yang menggambarkan titik awal dan titik akhir $(X_1, Y_1; X_2, Y_2)$.
3. Data Area (*Polygon*): Terdiri dari kumpulan pasangan titik koordinat dimana titik koordinat awal sama dengan titik koordinat akhir $(X_1, Y_1; X_2, Y_2; \dots; X_n, Y_n)$ dimana $(X_1, Y_1 = X_n, Y_n)$.
4. Data permukaan (*Surface*): Merupakan suatu area yang terdiri dari 3 titik koordinat kartesius (X, Y, Z) .

2.1.1.2 Model Data Raster

Model data raster mempunyai struktur data yang tersusun dalam bentuk piksel dan membentuk grid. Setiap piksel memiliki nilai tertentu dan memiliki atribut sendiri, termasuk nilai koordinat yang unik. Tingkat keakurasian model ini sangat tergantung pada ukuran piksel atau biasa disebut dengan resolusi (Gumelar, 2004).



Gambar 2.1 Data vektor



Gambar 2.2 Data raster

Dari gambar 2.1 dan gambar 2.2 merupakan gambar potongan wilayah Jakarta dalam bentuk vektor dan raster. Terlihat pada garis batas wilayah perbedaan antara data vektor dan raster. Pada data raster terlihat susunan piksel yang berbentuk kotak-kotak, semakin kecil ukuran piksel maka semakin mendekati bentuk objek sesungguhnya. Garis batas pada data vektor terlihat lebih halus, karena terdiri dari susunan titik-titik, keakurasian data vektor tergantung dari jarak antara titik-titiknya.

2.2 Basis Data

Basis data adalah kumpulan dari data yang saling berhubungan. Sedangkan data adalah segala fakta yang dapat direkam dan memiliki arti yang implicit. Contohnya adalah nama, nomor telepon, dan alamat. Semua data yang saling berhubungan tersebut disebut dengan basis data (Elmasri, 2007).

Basis data memiliki sifat-sifat sebagai berikut (Elmasri, 2007):

- Basis data merepresentasikan beberapa aspek dari dunia nyata, yang biasa disebut dengan istilah *MiniWorld*.
- Basis data secara logis merupakan suatu kumpulan data yang saling berkaitan dan memiliki arti tertentu.
- Basis data dirancang, dibangun, dan berkembang dengan data tertentu.
- Basis data diperuntukkan untuk sekelompok orang atau aplikasi tertentu.

Database Management System (DBMS) adalah kumpulan data-data yang saling berhubungan dan kumpulan program untuk mengolah data tersebut. DBMS merupakan perangkat lunak yang memiliki tujuan umum dan fasilitas untuk :

- Mendefinisikan (memberikan spesifikasi tipe data, struktur data, dan batasan/*constraint* data).
- Membentuk (proses menyimpan data ke dalam media penyimpanan).
- Memanipulasi (memberikan query/perintah untuk mengambil data, mengubah data, dan membuat laporan).
- Mendistribusikan data (membagikan data kepada banyak orang atau aplikasi).

Keuntungan menggunakan basis data (Elmasri, 2007) :

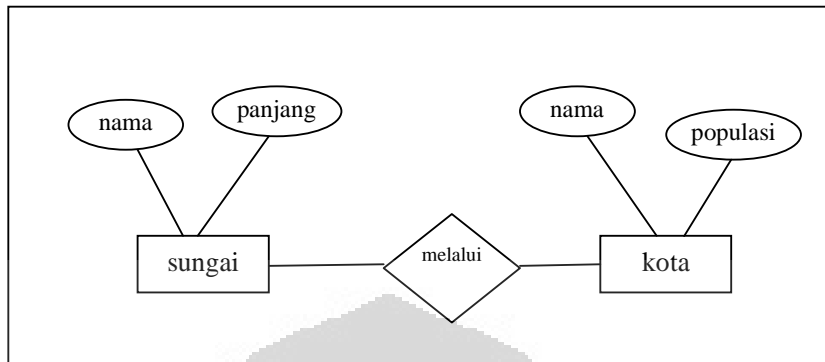
1. Mengontrol *redundancy* (perulangan data).
2. Membatasi akses bagi yang tidak berkepentingan.
3. Memberikan tempat penyimpanan yang *persistent* (data tidak hilang ketika program berhenti).
4. Memberikan struktur penyimpanan yang efisien untuk pemrosesan query
5. Menyediakan *backup* (cadangan data) dan *recovery* (pengembalian data ke kondisi sebelum terjadi kegagalan).
6. Menyediakan *interface* untuk banyak pengguna.
7. Mampu merepresentasikan hubungan yang kompleks diantara data-data
8. Memaksakan integrasi data sesuai dengan *constrains* (batasan-batasan) tertentu.
9. Mampu melakukan pengambilan keputusan dengan menggunakan aturan-aturan tertentu.

2.2.1 Model *Entity-Relationship*

Model *Entity-Relationship* (E-R) berdasar pada persepsi dunia nyata yang mengandung objek-objek dasar yang disebut *entities* dan *relationship* antar objek. *Entity* (Entitas) adalah objek yang dapat dibedakan dari objek lain dengan himpunan atribut (Korth, 1991). Misalnya atribut kecamatan dan *atribut* populasi menjelaskan *entity* wilayah tertentu di Jakarta.

Struktur model E-R secara keseluruhan dapat digambarkan dengan *E-R diagram* yang terdiri dari (Korth, 1991):

- Persegi panjang, yang merepresentasikan *entity*
- Elips, yang merepresentasikan atribut
- Berlian, yang merepresentasikan *relationships* antar *entity*.
- Garis, yang menghubungkan atribut ke *entity* dan *entity* ke *relationships*.



Gambar 2.3 Diagram E-R untuk sungai-sungai yang terdapat di suatu kota

Contoh diagram E-R pada gambar 2.3 berarti terdapat dua *entity* yaitu sungai dan kota. *Entity* sungai mempunyai dua atribut yaitu nama sungai dan panjang sungai. *Entity* kota mempunyai atribut nama kota dan jumlah populasinya. Kedua *entity* tersebut dihubungkan dengan *relationship* “melalui”, artinya sungai-sungai tersebut melalui atau mengalir di kota.

2.2.2 Model *Object-Oriented*

Model *Object-Oriented* (OO-model) berdasar pada kumpulan objek. Objek mengandung nilai yang disimpan dalam *instance variables* di dalam objek. Objek juga mengandung kode untuk menjalankan objek yang disebut *methods*. Objek yang mengandung tipe yang sama dan *methods* yang sama di kelompokkan bersama dalam *klas*. Tidak seperti dalam model E-R, tiap objek mempunyai identitas masing-masing yang unik disebut objek ID (Korth, 1991).

2.2.3 Model Basis Data *Object-Relational*

Object-relational database (ORDBMS) merupakan sistem yang mencoba untuk memperluas sistem basis data dengan fungsi yang diperlukan untuk mendukung kelas aplikasi yang lebih luas, dalam berbagai cara, menjembatani antara relasional dan *object-oriented* (Chatarji, 2003).

ORDBMS dibuat untuk menangani tipe data baru seperti audio, video, dan gambar yang tidak bisa ditangani oleh basis data relasional. Selain itu, pengembangannya berhasil meningkatkan penggunaan bahasa pemrograman *object-oriented* (Chatarji, 2003).

2.2.4 Bahasa Query

Bahasa query adalah bahasa yang digunakan pengguna untuk memperoleh informasi dari basis data. Bahasa query dapat dibagi menjadi dua kategori yaitu prosedural dan non-prosedural. Pada bahasa prosedural, pengguna menginstruksikan sistem untuk melakukan urutan operasi pada basis data untuk menghitung hasilnya. Pada non-prosedural, pengguna menggambarkan informasi tanpa memberikan prosedur khusus untuk mendapatkan informasi tersebut (Korth, 1991).

Contoh bahasa prosedural yaitu Aljabar Relasional dan bahasa non-prosedural yaitu Tuple relational calculus, Domain Relational Calculus. Selain itu terdapat bahasa query komersial yang digunakan dalam program komputer yaitu SQL.

2.2.4.1 Aljabar Relasional

Aljabar relasional adalah bahasa query prosedural yang mengandung himpunan operasi yang mengambil satu atau dua hubungan sebagai input dan menghasilkan suatu hubungan baru sebagai hasilnya (Korth, 1991).

Operasi dasar dalam aljabar relasional yaitu (Korth, 1991):

- Operasi *select*, memilih tupel yang memenuhi predikat yang diberikan, dinotasikan dengan sigma (σ).
- Operasi *project*, mengembalikan argumen relasi dengan kolom tertentu, dinotasikan dengan pi (Π).
- Operasi produk kartesian, mengkombinasikan informasi dari beberapa relasi, dinotasikan dengan cross (\times).

- Operasi *rename*, menamakan kembali atribut untuk mencegah kemungkinan ambigu, dinotasikan dengan (ρ)
- Operasi *set difference*, memungkinkan untuk menemukan tupel yang berada di satu relasi tetapi tidak ada di relasi lain, dinotasikan dengan (-).

Tambahan untuk operasi lainnya adalah *set intersection*, *union*, *natural join*, *division*, dan *assignment*.

2.2.4.2 Tuple Relational Calculus

Bahasa query ini merupakan bahasa query non-prosedural yang menggambarkan informasi yang diinginkan tanpa memberikan prosedur tertentu untuk mendapatkan informasi. Query dalam *tuple relational calculus* diekspresikan sebagai $\{t|P(t)\}$ dimana P adalah *formula*. Beberapa variabel tupel mungkin muncul dalam formula. Variable tupel dikatakan variabel bebas, kecuali jika dihitung oleh \exists atau \forall .

Formula *tuple relational calculus* di bangun dari *atoms*. Atom mempunyai satu dari bentuk berikut:

- $s \in r$, dimana s adalah variable tupel dan r adalah relasi.
- $s[x] \Theta u[y]$, dimana s dan u adalah variable tupel, x adalah atribut dimana s didefinisikan, y adalah atribut dimana u didefinisikan, dan Θ operator perbandingan ($<$, \leq , $=$, \neq , $>$, \geq). Atribut x dan y harus mempunyai domain yang dapat dibandingkan dengan Θ .
- $s[x] \Theta c$, dimana s adalah variable tupel, x adalah atribut dimana s didefinisikan, Θ adalah operator, dan c adalah konstanta dalam domain atribut x .

Formula dibangun dari atom menggunakan aturan berikut:

- Atom merupakan formula
- Jika P_1 formula, maka $\neg P_1$ dan (P_1) juga formula
- Jika P_1 dan P_2 formula, maka $P_1 \wedge P_2$, $P_1 \vee P_2$, $P_1 \Rightarrow P_2$ formula

- Jika $P_1(s)$ adalah formula mengandung variable bebas s , maka $\exists s \in r(P_1(s))$ dan $\forall s \in r(P_1(s))$ juga formula.

2.2.4.3 SQL (*Structured Query Language*)

Structured query language (SQL) adalah bahasa *query* pertama yang dikembangkan oleh IBM. SQL berkembang dari SEQUEL (*Structured English Query Language*) yang merupakan bagian dari IBM's sistem - part of IBM's SYSTEM R, 1974 (Nizar, 2007).

SQL kemudian berkembang menjadi SQL2. SQL2 didukung oleh banyak DBMS. DBMS pendukung SQL2 adalah ORACLE, SYBASE, INFORMIX, IBM DB2, SQL SERVER, MS Access, MySQL dan lain-lain. SQL2 dikenal dengan sebutan SQL/92 merupakan perkembangan dari SQL/86, SQL/89, yang semuanya sesuai dengan standar ANSI & ISO (Nizar, 2007). Sekarang sedang dikembangkan SQL3/SQL-99 dengan tambahan *feature object oriented*. Dengan adanya tambahan *feature* ini diharapkan dapat memodelkan data dengan lebih baik (Nizar, 2007).

SQL menggunakan kombinasi aljabar relasional dan kalkulus relasional (Korth, 1991).

SQL mempunyai dua bagian komponen (Shekhar, 2003):

1. *Data definition language* (DDL)

DDL digunakan untuk *create* (membuat), *delete* (menghapus), dan memodifikasi definisi tabel dalam basis data.

2. *Data manipulation language* (DML)

Dalam DML, queri dimainkan, baris di *insert* dan di *hapus* dari table yang telah di spesifikasikan pada DDL.

Struktur dasar SQL terdiri dari tiga klausa yaitu (Korth, 1991):

- **select** berkoresponden dengan operasi *project* pada aljabar relasional. Klausa ini digunakan untuk mendaftar atribut yang diinginkan dalam hasil query.
- **form** berkoresponden dengan produk kartesian pada aljabar relasional. Klausa ini mendaftar relasi yang akan di amati.
- **where** berkoresponden dengan predikat *selection* pada aljabar relasional. Klausa ini mengandung predikat yang melibatkan atribut relasi yang dipilih pada **from**.

Bentuk query SQL:

```
select A1, A2, ..., An
from r1, r2, ..., r3
where P
```

dimana A_i merepresentasikan atribut dan r_i merepresentasikan relasi. P adalah predikat.

Query ini ekuivalen dengan ekspresi aljabar relasional:

$$\Pi_{A_1, A_2, \dots, A_n}(\sigma_P(r_1 \times r_2 \times \dots \times r_m))$$

Variabel Tupel

SQL meminjam gagasan variabel tuple dari tuple relasional kalkulus. Variabel tupel dalam SQL harus di hubungkan dengan relasi tertentu. Variabel tupel didefinisikan dalam klausa **from**, digunakan untuk membandingkan dua tupel dalam relasi yang sama.

2.3 Basis Data Spasial

Basis data spasial adalah basis data yang mendefinisikan tipe data untuk objek geometri dan memungkinkan untuk menyimpan data geometri dalam tabel basis data. Basis data spasial menyediakan fungsi dan indeks untuk *querying* dan manipulasi data spasial menggunakan SQL. (Regina, 2009)

Guting (1994) mendefinisikan sistem basis data spasial sebagai kelas dari basis data yang mempunyai tiga karakteristik:

1. Sistem basis data spasial adalah sistem basis data
2. Menawarkan tipe data spasial dalam model data dan bahasa *query*-nya
3. Mendukung tipe data spasial dalam implementasinya, menyediakan indeksing spasial dan algoritma yang efisien untuk *spatial joins*.

2.3.1 Indeksing Data Spasial

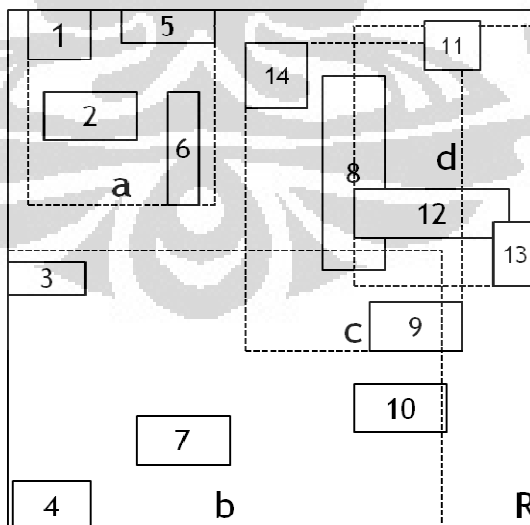
Indeksing spasial bekerja dengan tujuan yang sama seperti indkxing tabel yaitu mempercepat akses ke data dan mengembalikan data ke user dari basis data. Konsep dasar indkxing spasial yaitu menggunakan pendekatan dimana proses akses spasial secara berangsur-angsur mempersempit wilayah pencarian samapai objek basis data ditemukan. Metode spasial yang umum dipakai adalah *R-tree* (dimana R adalah *region*) (Yeung, 2007).

2.3.2 Indeks Spasial *R-Tree*

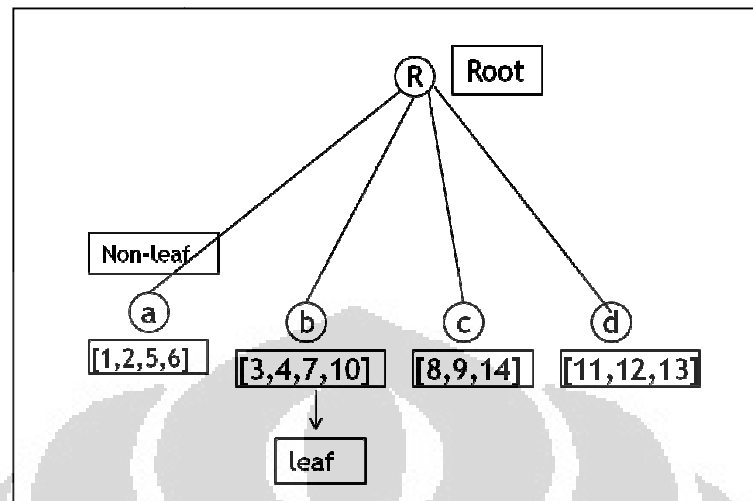
Properti struktur *R-tree* (Rigaux, 2002):

- setiap *node* (kecuali *root*), jumlah *entry* antara m dan M , dimana $m \in [0, M/2]$.
- dalam *non-leaf node* N berisi $(dr, nodeid)$, dr yaitu *directory rectangle* dari *child node* N , yang menunjuk alamat *nodeid*.
- setiap *leaf* berisi (mbb, oid) , mbb adalah *minimal bounding box* dari objek spasial yang disimpan pada *oid*.

Gambar 2.4 merupakan contoh kumpulan objek berjumlah 14 dengan *minimum bounding box* pada masing-masing objek id.



Gambar 2.4 Contoh *minimum bounding box* dari objek spasial



Gambar 2.5 R-Tree untuk gambar 2.4

2.3.3 Hubungan Topologi Spasial

Dalam menampilkan, memproses, dan menganalisis informasi spasial, pengguna mengambil data spasial dengan query. Query data spasial berdasarkan konsep spasial. Pertanyaan query spasial secara eksplisit memasukkan hubungan spasial untuk menggambarkan batasan objek spasial yang akan di analisis atau di tampilkan.

Berdasar model data yang mendasarinya, dipilih himpunan bagian dari ruang topologi. Pendekatan *point-set* merupakan model paling umum untuk merepresentasikan wilayah topologi spasial. Model hubungan topologi spasial yang digunakan didasarkan pada topologi *point-set* dari *interior* dan *boundary*.

Berikut ini merupakan definisi ruang topologi. Diberikan himpunan tak-kosong X , suatu koleksi \mathcal{T} yang berisikan himpunan-himpunan bagian dari X dikatakan topologi pada X , jika memenuhi:

- X dan himpunan kosong termuat di dalam \mathcal{T} .
- Gabungan (berhingga ataupun tak h.ingga) dari himpunan-himpunan di \mathcal{T} termuat di \mathcal{T} pula.

- Irisan dari dua himpunan di \mathcal{A} berada di \mathcal{A} pula.
Pasangan (X, \mathcal{A}) dikatakan ruang topologi.

Interior

Diberikan $Y \subset X$, interior Y dinotasikan dengan Y° , didefinisikan sebagai gabungan dari himpunan terbuka yang terkandung dalam Y .

Closure

Closure Y dinotasikan dengan \bar{Y} , didefinisikan sebagai perpotongan dari semua himpunan tertutup yang mengandung Y .

Boundary

Boundary Y , dinotasikan dengan ∂Y , merupakan perpotongan dari *closure* Y dan *closure* dari komplemen Y .

Hubungan antara *interior*, *closure*, dan *boundary* adalah:

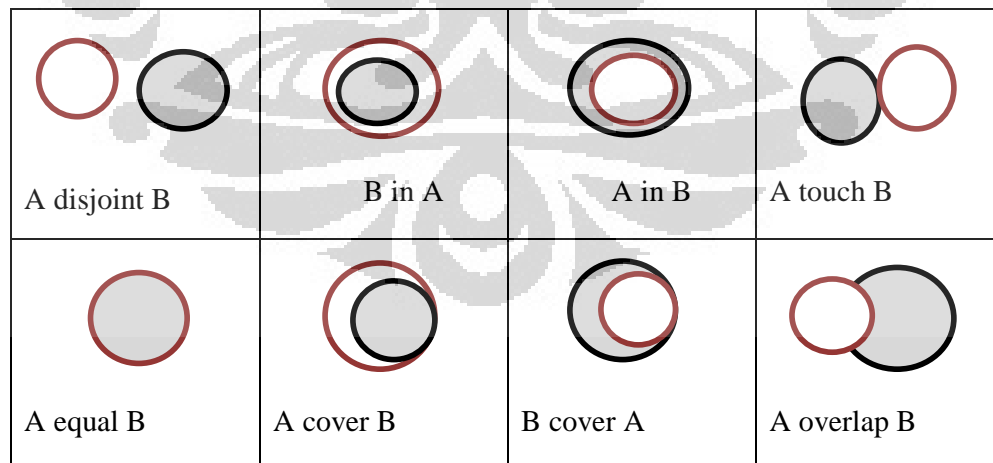
- $Y^\circ \cap \partial Y = \emptyset$
- $Y^\circ \cup \partial Y = \bar{Y}$

Ambil dua himpunan bagian, A dan B , dari ruang topologi X . Hubungan topologi spasial antara A dan B digambarkan dengan empat perpotongan antara *boundary* dan *interior* dari A dan B , yaitu $\partial A \cap \partial B$, $A^\circ \cap B^\circ$, $\partial A \cap B^\circ$, dan $A^\circ \cap \partial B$. Bentuk pertama disebut perpotongan *boundary-boundary*, kedua disebut perpotongan *interior-interior*, ketiga disebut perpotongan *boundary-interior*, dan keempat adalah perpotongan *interior-boundary*. Empat perpotongan tersebut di nilai dengan property himpunan *empty* dan *non-empty*. *Empty* didefinisikan sebagai himpunan kosong yang berarti tidak berpotongan dan dinotasikan dengan \emptyset . *Non-empty* merupakan himpunan tidak kosong yang berarti berpotongan dan dinotasikan dengan $\neq \emptyset$.

Enam belas kombinasi dari hubungan topologi spasial di rangkum dalam tabel 2.1.

Tabel 2.1 Enam belas spesifikasi hubungan topologi spasial berdasar pada kriteria *empty* dan *non-empty* antara perpotongan *boundary* dan *interior*.

$\partial A \cap \partial B$	$\partial A \cap B^\circ$	$A^\circ \cap \partial B$	$A^\circ \cap B^\circ$	Relationships
\emptyset	\emptyset	\emptyset	\emptyset	A disjoint B
\emptyset	\emptyset	\emptyset	$\neq \emptyset$	Tidak terjadi
\emptyset	\emptyset	$\neq \emptyset$	\emptyset	Tidak terjadi
\emptyset	\emptyset	$\neq \emptyset$	$\neq \emptyset$	B in A
\emptyset	$\neq \emptyset$	\emptyset	\emptyset	Tidak terjadi
\emptyset	$\neq \emptyset$	\emptyset	$\neq \emptyset$	A in B
\emptyset	$\neq \emptyset$	$\neq \emptyset$	\emptyset	Tidak terjadi
\emptyset	$\neq \emptyset$	$\neq \emptyset$	$\neq \emptyset$	Tidak terjadi
$\neq \emptyset$	\emptyset	\emptyset	\emptyset	A touch B
$\neq \emptyset$	\emptyset	\emptyset	$\neq \emptyset$	A equal B
$\neq \emptyset$	\emptyset	$\neq \emptyset$	\emptyset	Tidak terjadi
$\neq \emptyset$	\emptyset	$\neq \emptyset$	$\neq \emptyset$	A cover B
$\neq \emptyset$	$\neq \emptyset$	\emptyset	\emptyset	Tidak terjadi
$\neq \emptyset$	$\neq \emptyset$	\emptyset	$\neq \emptyset$	B cover A
$\neq \emptyset$	$\neq \emptyset$	$\neq \emptyset$	\emptyset	Tidak terjadi
$\neq \emptyset$	$\neq \emptyset$	$\neq \emptyset$	$\neq \emptyset$	A overlap B



Gambar 2.6 Delapan hasil dari hubungan topologi spasial berdasar pada kriteria *empty* dan *non-empty* antara perpotongan *boundary* dan *interior*.

2.4 Sistem Informasi Geografi

Sistem informasi geografis adalah suatu alat yang berguna untuk mengolah data spasial menjadi suatu informasi, pada umumnya informasi tersebut berkaitan erat dan digunakan untuk mengambil suatu keputusan mengenai suatu hal (Demers, 2005).

Sebelum sistem informasi geografis dibuat, manusia menggunakan peta untuk menyimpan data spasial yang dimilikinya. Peta pertama yang ditemukan adalah peta yang tergambar di dinding gua Lascaux, Perancis yang dibuat 3500 tahun yang lalu oleh para pemburu Cromagnon. Peta tersebut menggambarkan hewan buruan dan garis yang dipercaya sebagai rute migrasi hewan buruan mereka. Ilmu perpetaan berkembang hingga pada tahun 1700-an, penggunaan teknik survei modern untuk memetakan topografis diterapkan dan pada tahun ini pula berkembang versi awal dari peta tematis (peta yang menunjukkan hubungan ruang dalam bentuk atribut tunggal atau hubungan atribut) yang digunakan untuk keilmuan dan data sensus (Riyanto, 2009).

Sistem informasi geografis diawali pada tahun 1960 dimana pemerintah Kanada melalui Departemen Kehutanan dan Pengembangan Desa (*Department of forestry and Rural Development*) mengadakan suatu proyek besar untuk mengatur sumber daya alam. Salah satu tugas awal adalah melakukan inventarisasi dan memetakan hutan, sumber daya mineral yang tersedia, habitat hewan liar, ketersediaan dan kualitas air. Data-data tersebut lalu dibandingkan untuk memprediksi keberadaan sumber daya alam dan membangun strategi manajemen untuk memastikan keberlangsungan sumber daya alam baik yang dapat diperbaharui maupun yang tidak dapat diperbaharui. Tugas tersebut membutuhkan tenaga, biaya dan waktu yang sangat besar. Waktu yang besar mengakibatkan informasi yang didapat akan lebih lambat daripada kejadian nyata dilapangan, sumber daya alam tersebut bisa saja sudah habis ketika proses pemetaan selesai. Pertemuan eksekutif IBM dan direktur Departemen Kehutanan dan Pembangunan Pedesaan menghasilkan suatu kesepakatan penggunaan teknologi komputer untuk mengatasi masalah tersebut. Maka lahirlah CGIS(*Canada Geographic Information System*) sistem informasi

geografis pertama yang bertugas untuk melakukan klasifikasi dan memetakan sumber daya lahan di Kanada (Demers, 2005)

Suatu sistem informasi geografis terdiri dari subsistem sebagai berikut (Demers, 2005):

1. Subsistem data *input* : Subsistem ini memasukkan dan melakukan pemrosesan awal dari data spasial dari berbagai sumber. Subsistem ini juga bertanggung jawab dalam melakukan transformasi dari data spasial.
2. Subsistem penyimpanan data dan pengambilan data : Subsistem ini bertanggung jawab dalam pengaturan data spasial sehingga mudah untuk melakukan pengambilan data, memperbaharui data dan mengubah data.
3. Subsistem manipulasi dan analisis data : Subsistem ini bertugas untuk melakukan pengumpulan, pemisahan, memperkirakan parameter dan batasan dan melakukan fungsi modeling.
4. Subsistem pelaporan: Subsistem ini bertugas menampilkan semua atau sebagian data dalam bentuk tabel , grafik, atau peta.

Tabel 2.2 Pembagian kerja antara SDBMS dan SIG (Yeung, 2007)

Sistem	Pekerjaan
Sistem Informasi Geografi	<ul style="list-style-type: none"> • Mengambil dan edit data • Analisis data • Membangun peta dan produk kartografi
Sistem Basis Data Spasial	<ul style="list-style-type: none"> • Menyimpan dan mengelola data • Indeksing spasial • Keamanan dan integritas data • Query data spasial

2.5 PostgreSQL

PostgreSQL adalah sistem basis data *object-relational*. PostgreSQL menyatakan dirinya basis data *open source* yang paling terdepan (Regina, 2009).

Fitur-fitur PostgreSQL (Regina, 2009):

1. Variasi pilihan untuk memilih dalam menulis fungsi basis data yang dapat mengembalikan nilai skalar sederhana
2. Dapat mendefinisikan array
3. Memungkinkan untuk mendefinisikan fungsi *aggregate* yang diambil dari lebih dari satu kolom.

PostgreSQL dipilih sebagai pasangan yang cocok untuk PostGIS karena mendukung untuk menambah tipe data baru, mendefinisikan fungsi, operator, dan indeks yang bekerja pada tipe baru dan memenuhi untuk pengguna query.

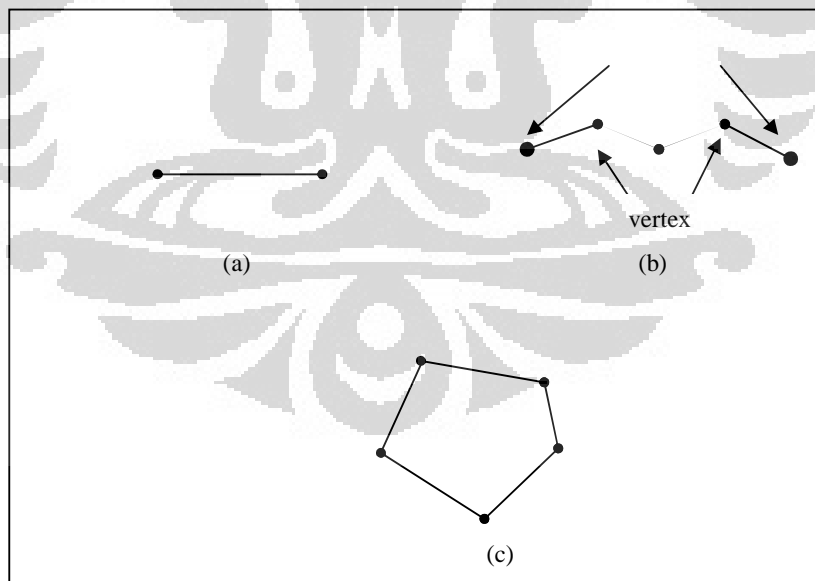
PostGIS menyediakan lebih dari 300 operator spasial, fungsi spasial, tipe data spasial dan indeks spasial. PostGIS dan PostgreSQL juga menyesuaikan standar industri seperti ANSI SQL dan OGC (*Open Geospatial Consortium*). OGC merupakan sebuah badan yang keberadaannya untuk menstandarisasi bagaimana mengakses dan mendistribusi geografi dan data spasial (Regina, 2009)

BAB 3 REPRESENTASI DATA SPASIAL DAN BAHASA QUERY

3.1 Representasi Dimensi Objek Spasial

Keseluruhan himpunan (identitas, objek spasial, dan deskripsi) membentuk objek geografi disebut juga *entity* atau *feature*. Secara umum definisi objek spasial adalah himpunan titik, namun dalam penerapannya menggunakan salah satu dari tipe berikut:

- Objek dimensi-nol atau titik.
Titik digunakan untuk merepresentasikan entitas lokasi yang dianggap sangat kecil atau dianggap tidak begitu berguna dalam sisi bentuknya. Kota, bangunan, adalah contoh entitas yang direpresentasikan dengan titik.
- Objek dimensi-satu atau objek linear
Objek ini digunakan untuk merepresentasi *network* (jalan, saluran air, sungai, dll). *Polyline* didefinisikan sebagai himpunan segmen garis hingga atau edge, sedemikian sehingga tiap titik ujung atau *vertex* terletak di dua garis kecuali *extreme point* terletak di satu garis.

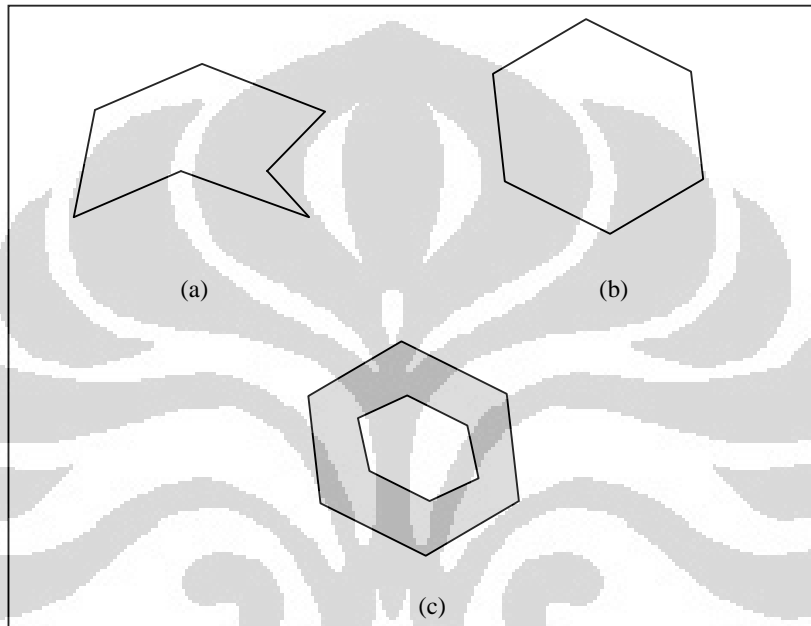


Gambar 3.1 Contoh objek dimensi-satu : (a) segmen garis,
(b) *polyline*, (c) *polyline* sederhana tertutup

- Objek dimensi-dua atau objek permukaan

Digunakan untuk merepresentasikan wilayah yang besar, seperti negara, propinsi.

Polygon merupakan tipe geometri utama dari objek ini. *Polygon* adalah wilayah bidang yang dibatasi oleh *polyline* tertutup.



Gambar 3.2 Contoh Objek Dimensi-dua: (a) *polygon* sederhana, (b) *polygon* konveks, (c) *polygon* dengan lubang

3.2 Representasi Objek Spasial dalam Model Vektor

Titik direpresentasikan oleh koordinat, sedangkan objek linear dan bidang di representasikan oleh struktur titik (list, himpunan, array).

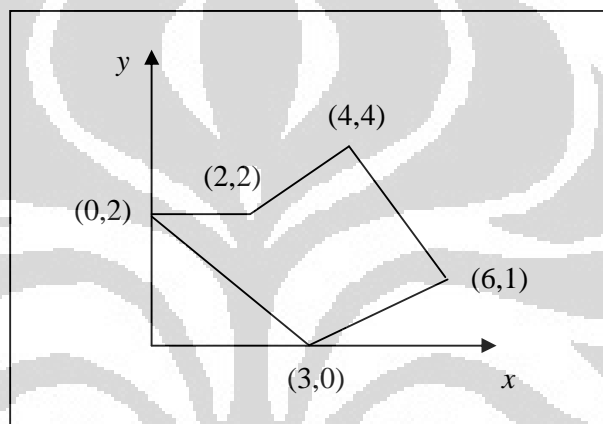
Berikut ini bentuk representasi objek:

- *Polyline* direpresentasikan oleh *list points* $\langle p_1, \dots, p_n \rangle$, tiap p_i merupakan vertex. Masing-masing pasangan (p_i, p_{i+1}) , dengan $i < n$ merepresentasikan edge dari *polyline*.
- *Polygon* direpresentasikan oleh *list points*. Yang berbeda adalah list merepresentasikan *polyline* tertutup, dan pasangan (p_n, p_1) adalah edge.
- *Region* adalah himpunan *polygon*.

3.2.1 Struktur Notasi

Titik dinotasikan oleh $[]$, list oleh $\langle \rangle$, dan himpunan oleh $\{ \}$. Dengan menggunakan notasi ini struktur objek dapat dijabarkan seperti berikut:

- Titik : $[x:real, y:real]$
- *Polyline* : $\langle point \rangle$
- *Polygon*: $\langle point \rangle$
- *Region* : $\{polygon\}$



Gambar 3.3 Representasi Vektor untuk *Polygon P*

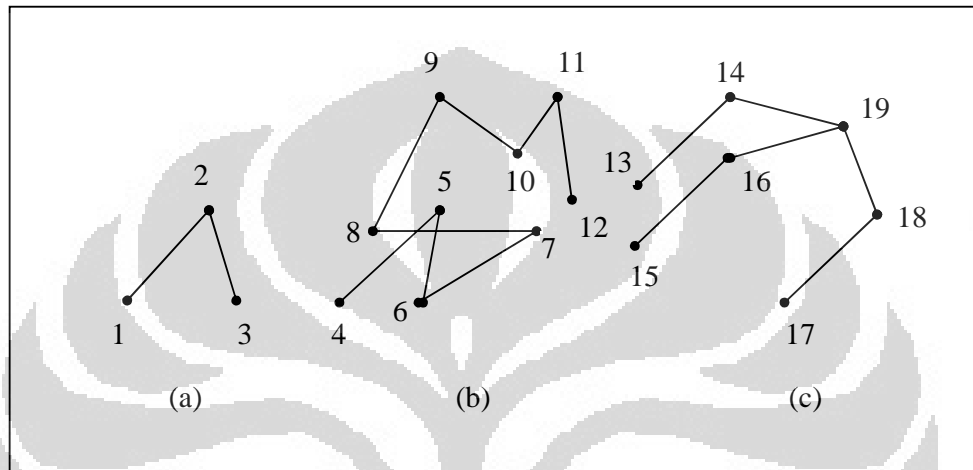
Gambar 3.3 menunjukkan representasi *polygon P* dalam vektor yang dijabarkan dalam pasangan koordinat :

$$\langle [4,4], [6,1], [3,0], [0,2], [2,2] \rangle$$

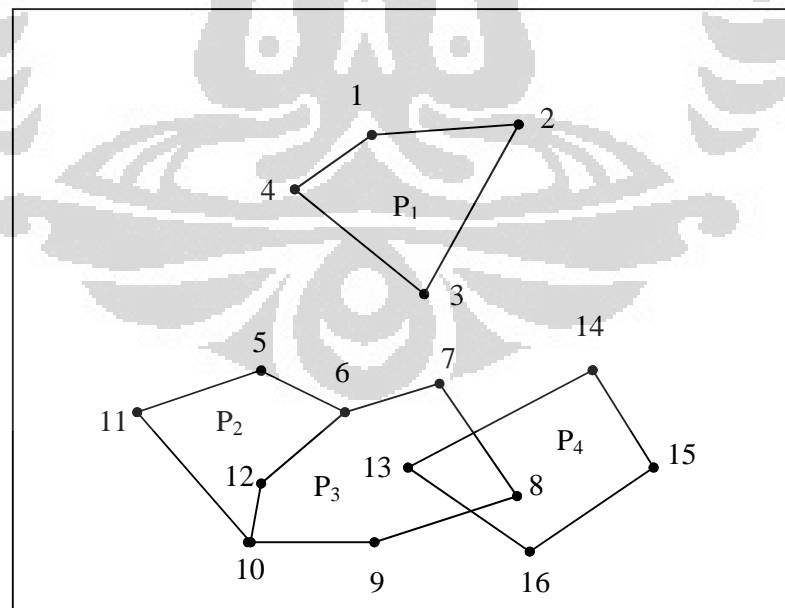
Gambar 3.4 merupakan contoh *polyline* dalam vektor:

- $L_1 = \langle 1,2,3 \rangle$ merupakan *polyline* dengan dua segmen garis.
- $L_2 = \langle 4, 5, 6, 7, 8, 9, 10, 11, 12 \rangle$ merupakan *polyline* tidak sederhana.
- L_3 bukan *polyline*, vertex 19 merupakan *endpoint* dari tiga *edge*. Maka L_3 adalah himpunan *polyline*.

Gambar 3.5 adalah contoh objek bidang (*polygon* dan *region*) dalam representasi vektor, menggunakan notasi vertex. *Region* mengandung *polygon* P_2 dan P_3 yang bersisian, dapat di jabarkan menjadi $\{ \langle 5, 6, 12, 10, 11, 5 \rangle, \langle 6, 7, 8, 9, 10, 12, 6 \rangle \}$. *Polygon* tunggal $P_4 \{ \langle 13, 14, 15, 16, 13 \rangle \}$ dan $P_1 \{ \langle 1, 2, 3, 4, 1 \rangle \}$.



Gambar 3.4 (a) L_1 , (b) L_2 , (c) L_3



Gambar 3.5 Contoh *Polygon*

3.3 Representasi Objek Spasial dalam Basis Data

Misalkan terdapat entitas administrasi Jakarta yang mempunyai empat atribut yaitu kecamatan, kelurahan, wilayah dan bentuk geometri. Entitas ini merupakan *multipolygon*. Kecamatan berisi nama-nama kecamatan yang ada di Jakarta, begitu juga untuk atribut kelurahan, sedangkan atribut wilayah berisi pembagian wilayah Jakarta misalnya Jakarta Timur, Jakarta Pusat. Setiap satu *polygon* merepresentasikan satu kelurahan, penyimpanan dalam basis data adalah sebagai berikut:

Tabel 3.1 Representasi Data Spasial dalam Basis Data Objek-Relasional

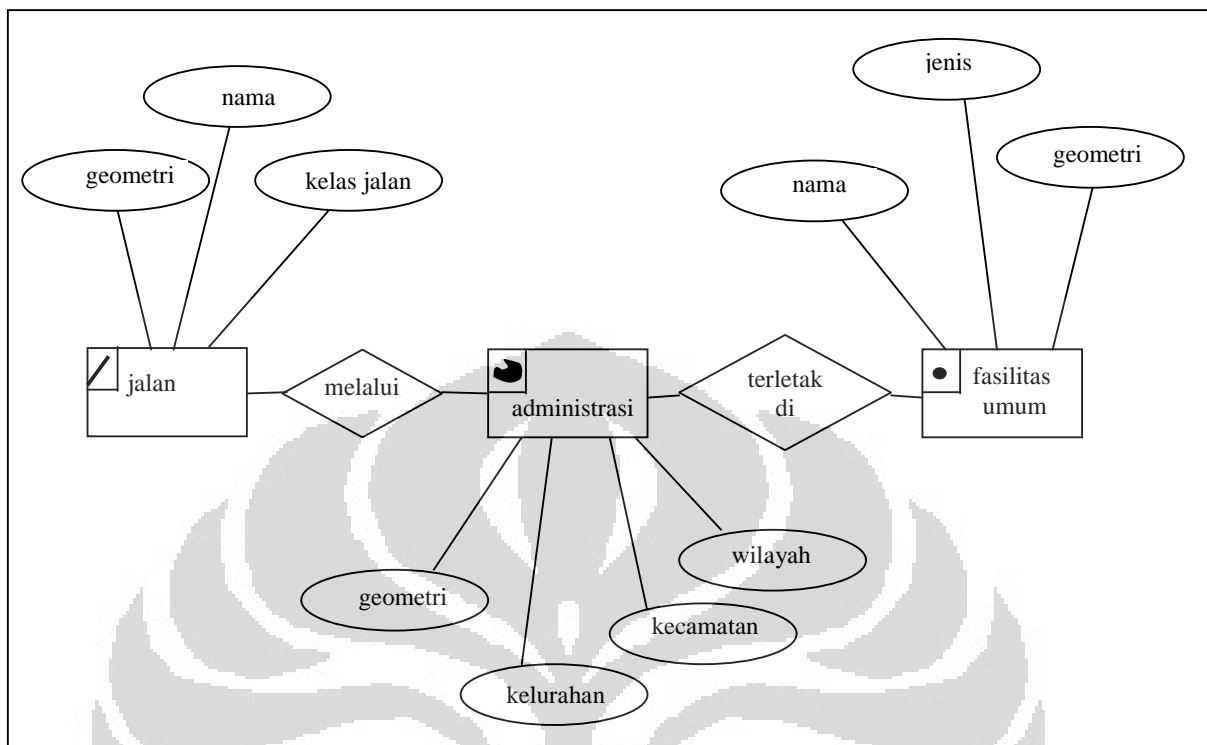
Objek ID	Wilayah	Kecamatan	Kelurahan	Geometri
1	Jakarta Timur	Pasar Rebo	Kalisari	((0,2), (2,2), (4,4), (6,1), (3,0),(0,2))

Dalam tabel tersebut disimpan data non-spasial yaitu atribut objek ID, wilayah, kecamatan, kelurahan, dan data spasial yaitu atribut geometri yang berupa *polygon*. Informasi spasial yang disimpan dalam basis data adalah titik-titik koordinat dari objek spasial. Titik koordinat yang ditampilkan pada tabel merupakan titik ujung dari list *edge* pada *polygon*. *Edge* pada *polygon* merupakan garis batas *polygon* (seperti pada gambar 3.5).

Model basis data yang digunakan untuk basis data spasial adalah model basis data objek-relasional. Berbasis objek, karena memakai objek spasial yang setiap objeknya diberi identitas tersendiri, berbasis relasional karena basis data spasial dapat dibuat hubungan relasionalnya dan dapat menggunakan bahasa query relasional.

3.3.1 Model Entity Relationship DKI Jakarta

Sebelum menjelaskan bahasa query spasial, akan dibuat terlebih dahulu entity relationship dari basis data yang telah dipilih. Dalam penelitian ini, data terdiri dari tiga entitas, yaitu entitas administrasi, jalan, dan fasilitas umum, seperti terlihat dalam Gambar 3.6.



Gambar 3.6 Diagram E-R Skema DKI Jakarta

Gambar 3.6 merupakan diagram E-R skema data dalam penelitian ini yaitu DKI Jakarta. Dalam skema basis data DKI Jakarta terdapat tiga entitas yaitu jalan, administrasi dan fasilitas umum. Entitas jalan merepresentasikan lintasan jalan yang melalui daerah DKI Jakarta yang mempunyai atribut nama jalan, kelas jalan, dan geometri yang berupa *polyline*. Entitas administrasi merepresentasikan pembagian wilayah DKI Jakarta yang terdiri atas nama kelurahan, nama kecamatan, nama wilayah bagian dan geometri yang berupa *region* (himpunan *polygon*). Entitas fasilitas umum merepresentasikan bangunan-bangunan yang merupakan fasilitas umum yang ada di Jakarta. Entitas fasilitas umum mempunyai atribut nama, jenis fasilitas umum, dan geometri yang berupa titik.

Basis data spasial mempunyai hubungan relasional antar datanya, karena setiap skema basis data spasial mempunyai sistem koordinat yang sama dan saling berhubungan walaupun geometri tiap entitas berbeda. yang menjadi penghubung dalam basis data spasial adalah atribut geometri. Hal ini berbeda dengan basis data

relasional non-spasial, yang menjadi penghubungnya adalah *foreign key* yaitu atribut yang sama antara dua tabel, sehingga terjadi duplikasi atribut dalam basis data, hal ini yang menjadi kelemahan basis data relasional dibandingkan basis data objek-relasional yang digunakan pada basis data spasial.

3.4 Bahasa Query Spasial

Bahasa query yang digunakan untuk mengolah data spasial merupakan bahasa query yang telah diperluas. Bahasa query yang diperluas dengan query spasial adalah bahasa SQL. Bahasa query ‘murni’ seperti aljabar relasional, tuple relasional kalkulus dan domain relasional kalkulus, belum ada sumber yang membahas bahasa query tersebut untuk data spasial.

3.4.1 Perbandingan Query dalam SQL, Aljabar Relasional, dan *Tuple Relational Calculus*

- **Data Definition Language**

Contoh pendefinisian entitas administrasi dengan SQL:

```
create table administrasi {
    wilayah varchar (20)
    kecamatan varchar (20)
    kelurahan varchar (20)
    geometri Polygon}
```

create table adalah klausa untuk mendefinisikan tabel entitas, nama tabelnya adalah administrasi. Tabel terdiri dari empat kolom, setiap kolom diberi nama dan spesifikasi tipe datanya. Kolom geometri berisi tipe geometri dari administrasi dalam hal ini *polygon*. Pada aljabar relasional dan *tuple relational calculus* tidak terdapat pendefinisian data.

- **Data Manipulation Language**

1. Menyisipkan data ke dalam tabel administrasi

SQL : **insert into** administrasi (wilayah, kecamatan, kelurahan,
geometri)

values ('Jakarta Timur', 'Pasar Rebo', 'Kalisari', polygon)

Setelah data didefinisikan dalam DDL, selanjutnya data di tambahkan ke tabel dengan perintah **insert into**.

Aljabar relasional : $administrasi \leftarrow administrasi \cup \{("Jakarta Timur", "Pasar Rebo", "Kalisari")\}$

Bahasa aljabar relasional tidak dapat menyimpan data geometri.

Pada *tuple relational calculus* tidak terdapat formula untuk menyisipkan data.

2. Perintah menghapus

SQL : **delete** administrasi
where kecamatan='Pasar Rebo'

Seluruh baris yang memiliki kecamatan Pasar Rebo akan dihapus dari tabel.

Aljabar relasional : $administrasi \leftarrow administrasi - \sigma_{kecamatan="Pasar Rebo"}$

Pada *tuple relational calculus* tidak terdapat formula untuk menghapus data.

3. Perintah memodifikasi selanjutnya berdasar dari tiga klausa yaitu:

select, from, where. SQL juga mempunyai fungsi mengurutkan, mencari rata-rata, *up date*, dan mencari jumlah maksimum, minimum.

```
select kecamatan
from administrasi
where kecamatan="Pasar Rebo"
```

Keterangan : Ekspresi SQL ini mengembalikan nama kecamatan yang mempunyai kelurahan Ciracas pada tabel administrasi.

Aljabar Relasional : $\Pi_{kecamatan}(\sigma_{kelurahan="Ciracas"}(administrasi))$

Tuple relational calculus : $\{t | \exists t \in administrasi(t[kelurahan] = s[kelurahan] \wedge s[kelurahan] = "Ciracas")\}$

Berikut ini bahasa query yang terdapat pada SQL namun tidak ada pada aljabar relasional dan *tuple relational calculus* :

1. **select avg (length)**

from jalan

where kelas_jalan='Jalan Tol'

Keterangan : Query ini menghitung rata-rata dari panjang jalan yang diasumsikan kolomnya ada pada tabel jalan. Selain **avg** operasi aggregate lainnya yaitu **min, max, count, sum**.

2. **select** kelurahan, kecamatan

from administrasi

order by kecamatan asc

Keterangan : Query ini mengurutkan nama kelurahan beserta kecamatan, dan diurutkan berdasar alphabet kecamatan.

3. **select** wilayah, **count**(Kelurahan)

from administrasi

group by wilayah

Keterangan : Query ini mendaftarkan jumlah kelurahan yang dimiliki ditiap wilayah.

3.4.2 SQL untuk Data Spasial

Bahasa SQL yang telah diperluas untuk data spasial berbeda dengan data non-spasial, karena berkaitan dengan data ruang dua dimensi dalam koordinat. Dalam data non-spasial, untuk menghubungkan dua tabel dilakukan *join* tabel. Pada data spasial ketika menghubungkan dua tabel berarti menggabungkan dua *layer* atau peta dan melakukan *spatial relationship*. Oleh karena itu membutuhkan operator spasial untuk mendapatkan fungsi *spatial relationship* seperti *intersection*, *overlap*, dan sebagainya.

Open Geospatial Consortium (OGC) yang merupakan lembaga yang mengelola *open source* geografi telah menstandarisasi operator spasial untuk SQL yang ditunjukkan dalam tabel 3.2. SQL spasial dan non-spasial dapat digunakan secara bersamaan. Berikut ini contoh query basis data spasial :

1. **select** a.kecamatan

```
from administrasi a, administrasi b
where touch(a.geometri,b.geometri)
group by a.kecamatan
having count(b.kecamatan)=1
```

Keterangan: Query ini mengembalikan nama kecamatan yang hanya mempunyai tetangga kecamatan berjumlah 1

2. **create view** kecamatan_neighbor **as**

```
select a.kecamatan, count(b.kecamatan) as num_neighbor
from administrasi a, administrasi b
where touch(a.geometri,b.geometri)
group by a.kecamatan
```

```
select a.kecamatan, num_neighbor
from kecamatan_neighbor
where num_neighbor= (select max(num_neighbor)
from kecamatan_neighbor)
```

Keterangan: Query ini mengembalikan nama kecamatan yang mempunyai jumlah tetangga yang paling besar. **create view** merupakan perintah untuk melihat tabel sebagai hasil bentukan baru yang memenuhi kriteria tertentu.

Tabel 3.2 Operasi Spasial dalam Standar OGC untuk SQL

Basic Function	SpatialReference	Referensi sistem koordinat geometri
	Envelope	Minimum bounding box dari geometri
	IsEmpty	True jika geometri himpunan kosong
	IsSimple	True jika geometri sederhana (tidak berpotongan)
	Boundary	Batas geometri
Topological/ Set Operators	Equal	True jika dua batas dan interior dua geometri sama
	Disjoint	True jika dua geometri tidak berpotongan
	Intersect	True jika dua geometri berpotongan
	Touch	True jika batas geometri berpotongan tetapi interior tidak berpotongan
	Cross	True jika polygon berpotongan dengan line
	Whitin	True jika interior geometri yang diberikan tidak berpotongan dengan eksterior geometri lain
	Contains	True geometri yang diberikan mengandung geometri lain
	Overlap	True jika dua geometri berpotongan dengan dimensi yang sama
Spatial Analysis	Distance	Jarak terdekat dari dua geometri
	Buffer	Geometri yang mengandung semua point yang jaraknya lebih kecil atau sama dengan jarak yang ditentukan
	Convexhull	Geometri konveks terkecil dari himpunan geometri
	Intersection	Geometri dari dua geometri yang saling berpotongan
	Union	Gabungan dua geometri
	Difference	Bagian geometri yang tidak berpotongan dengan geometri yang diberikan
	Symdiff	Bagian dari dua geometri yang tidak saling berpotongan

BAB 4 POSTGRESQL

4.1 Skema dalam Basis Data

PostgreSQL dapat menyimpan data spasial jika telah terintegrasi dengan PostGIS, maka sebelum memulai membuat tabel data spasial, terlebih dahulu pastikan PostgreSQL telah terintegrasi bersama PostGIS. PostGIS merupakan program *open source* pelengkap PostgreSQL untuk perluasan bahasa SQL.

Data vektor yang akan di simpan, didigitasi menggunakan Arcview dengan format *shapefile*. Arcview adalah program SIG yang salah satu fungsinya untuk melakuka digitasi peta dan merubahnya ke dalam bentuk file digital. Proses digitasi adalah proses membuat peta dari bentuk *image* ke bentuk vektor. *Shapefile* merupakan salah satu jenis file untuk menyimpan data vektor. Dalam penelitian ini data spasial yang diperoleh merupakan data sekunder. Proses pendigitasian data vektor dilakukan di laboratorium SIG Departemen Geografi Universitas Indonesia.

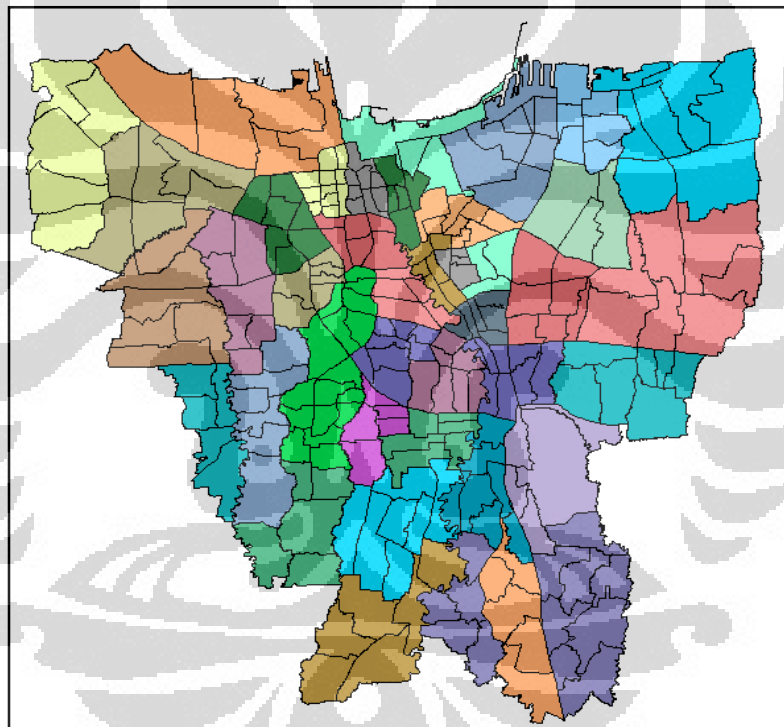
Data vektor yang diambil adalah data peta DKI Jakarta tahun 2002,yang terdiri dari tiga peta tematik yaitu:

1. Peta fasilitas umum yang ada di Jakarta yang merepresentasikan objek titik.
2. Peta jalan-jalan yang melewati Jakarta sebagai representasi objek garis.
3. Peta administrasi DKI Jakarta seperti wilayah kecamatan dan kelurahan yang ada di Jakarta sebagai representasi poligon.
4. Peta banjir wilayah Jakarta yang merepresentasikan objek poligon.

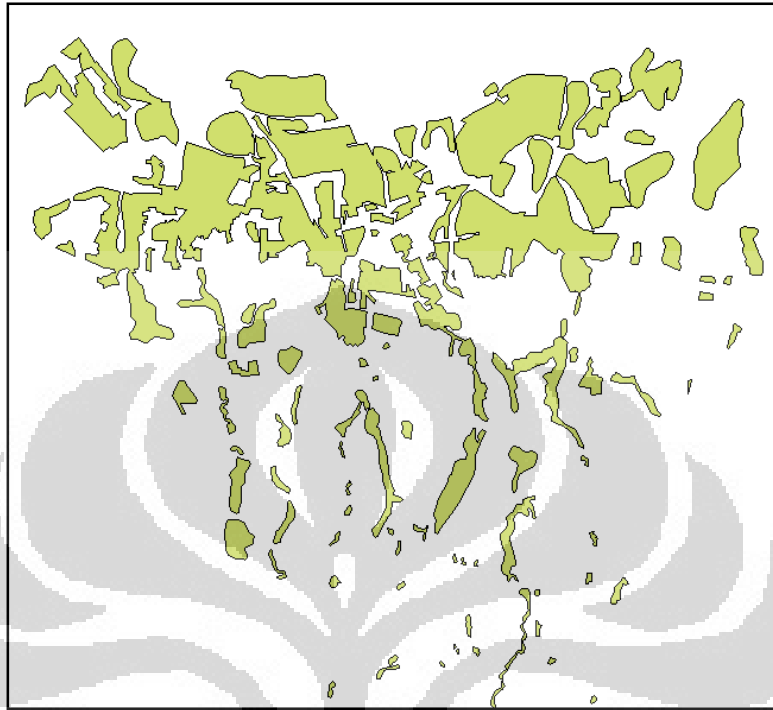
Peta yang terdapat pada Arcview juga terdapat atribut tabel dari masing-masing peta, namun belum terdapat kolom atribut spasial. Titik koordinat atau informasi spasial pada Arcview telah tersimpan dalam sistem.

Atribut tabel pada Arcview pada masing-masing peta yaitu:

1. Peta administrasi mempunyai enam atribut yaitu wilayah, kecamatan, kelurahan, kode wilayah, kode kecamatan, kode kelurahan.
2. Peta jalan mempunyai empat atribut yaitu panjang jalan, kode jalan, nama jalan, dan kelas jalan.
3. Peta fasilitas umum mempunyai empat atribut yaitu jenis fasilitas umum, kode, nama, alamat.



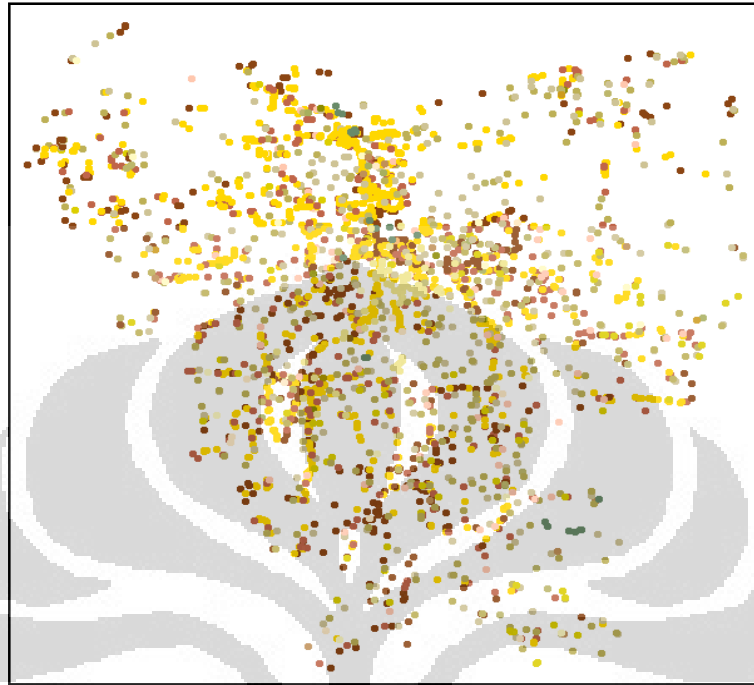
Gambar 4.1 Peta Administrasi DKI Jakarta



Gambar 4.2 Peta Banjir tahun 2002 DKI Jakarta



Gambar 4.3 Peta Jalan Tol dan Jalan Utama di DKI Jakarta



Gambar 4.4 Peta Lokasi Fasilitas Umum di DKI Jakarta

4.2 Penyimpanan Data dalam PostgreSQL

Data yang telah didigitasi dalam Arcview, yang disimpan dalam format shp (*shapefile*) kemudian di import dengan menggunakan *PostGIS import database*. Setelah di import, secara otomatis PostgreSQL telah menyimpan tabel basis data dari masing-masing peta, yang kemudian disebut entitas. Entitas dalam basis data merupakan peta dalam Arcview.

Berikut ini adalah entitas dan atribut pada basis data, yaitu:

1. Entitas **administrasi**

Atribut: **gid, wilayah, kecamatan, kelurahan, kd_wil, kd_kec, kd_kel, the_geom.**

2. Entitas **jalan**

Atribut: **gid, length, jn_code, nama_jalan, kelas_jala, the_geom.**

3. Entitas **fasilitas umum**

Atribut: **gid, jenis, kode, nama, alamat, the_geom.**

4. Entitas **banjir tahun 2002**

Masing-masing entitas mempunyai atribut yang sama seperti pada Arview, namun setiap entitas akan ditambahkan atribut kolom untuk informasi spasial, yaitu atribut *the_geom* dan *gid*. *The_geom* mendefinisikan titik-titik koordinat yang dimiliki oleh masing-masing objek geometri. *Gid* atau *geometry id* mendefinisikan kode pada masing-masing geometri, yang juga sebagai *primary key* pada tiap entitas.

gid integer	wilayah character varying(15)	kecamatan character varying(20)	kelurahan character varying(25)	kd_wil character varying(2)	kd_kec character varying(3)	kd_kel character varying(3)	the_geom geometry
1	JAKARTA UTARA	KELAPA GADING	PEGANGSAAN DUA	75	050	003	010600000001000000010300000001
2	JAKARTA UTARA	KELAPA GADING	KELAPA GADING BARAT	75	050	001	010600000001000000010300000001
3	JAKARTA UTARA	CILINCING	SUKAPURA	75	060	001	010600000001000000010300000001
4	JAKARTA TIMUR	PULO GADUNG	KAYU PUTIH	72	090	006	010600000001000000010300000001
5	JAKARTA TIMUR	PASAR REBO	KALISARI	72	010	002	010600000001000000010300000001
6	JAKARTA TIMUR	PASAR REBO	GEDONG	72	010	005	010600000001000000010300000001
7	JAKARTA TIMUR	PASAR REBO	CIJANTUNG	72	010	004	010600000001000000010300000001
8	JAKARTA TIMUR	MATRAMAN	UTAN KAYU SELATAN	72	100	005	010600000001000000010300000001
9	JAKARTA TIMUR	MATRAMAN	PISANGAN BARU	72	100	003	010600000001000000010300000001

Gambar 4.5 Tabel Entitas Administrasi

gid integer	length double precis	jn_code integer	nama_jalan character varying(32)	kelas_jala character varying(3)	the_geom geometry
1	0.00034	20	Bandengan Utara	Jalan Utama	010500000001000000010200
2	0.000222	20	Jembatan Dua	Jalan Utama	010500000001000000010200
3	0.000293	20	Pluit Selatan Raya	Jalan Utama	010500000001000000010200
4	0.001023	20	Pluit Selatan Raya	Jalan Utama	010500000001000000010200
5	0.007043	20	Jalan Baru	Jalan Utama	010500000001000000010200
6	0.004166	20	Kamal Muara	Jalan Utama	010500000001000000010200
7	0.000994	20	Kamal Muara	Jalan Utama	010500000001000000010200

Gambar 4.6 Tabel Entitas Jalan

gid integer	jenis character varying(15)	kode integer	nama character varying(20)	alamat character varying(20)	the_geom geometry
1	Ktr. Pemerintah	55	Puspenmas Jak-utara	Sungai Landak	010100000023918E80F4CE254
2	Ktr. Pemerintah	55	Dinas Pemakaman	Pemakaman	01010000000811F86F490C9254
3	Ktr. Pemerintah	55	Sudin Kebakaran DKI	Baru Kosambi	01010000005E073D4CF5C3254
4	Ktr. Pemerintah	55	Dinas Kehutanan	Kali Baru Barat 2	010100000072C790D6ACBC254
5	Ktr. Pemerintah	55	Ktr.,Depdikbud	Cilincing	0101000000CFF3457BECDE254
6	Ktr. Pemerintah	55	PT Telkom	Sungai Tiram Raya	0101000000B47C55CDBEDD254
7	Ktr. Pemerintah	55	Ktr Telekomunikasi	Sarang Bango No.10	010100000074FBAAB86BDE254
8	Ktr. Pemerintah	55	PT PAM Jawa	Cakung Cilincing	01010000001E3738DC2FCF254

Gambar 4.7 Tabel Entitas Fasilitas Umum

Gambar 4.5, 4.6, dan 4.7 merupakan tabel basis data yang tersimpan dalam PostgreSQL. Atribut *gid* merupakan kode geometri, yang dibaca geometri ke 1, 2, 3 dan seterusnya. Atribut *the_geom* merupakan titik koordinat pada masing-masing *gid*. Titik koordinat yang terlihat pada tabel merupakan kode ASCII, kode

ini dapat di ubah kedalam bentuk titik koordinat yang dimengerti pengguna dengan query.

4.3 Data Definition Language pada PostgreSQL

Data spasial yang digunakan merupakan data yang telah di-*import* dari Arcview dengan PostGIS, maka tahap DDL telah dilakukan di Arcview sehingga pendefinisian data pada PostgreSQL tidak perlu lagi. Hal ini lebih efisien jika dalam mendefinisikan tabel dilakukan dengan query di dalam PostgreSQL.

Akan ditunjukkan contoh *query* untuk DDL data spasial dalam PostgreSQL jika mendefinisikan tabel secara manual:

```
create table jalan(
    jalan_id integer,
    nama_jalan varchar)

select addgeometrycolumn('jalan','jalan_geom',-1,'linestring',2)
insert into jalan (jalan_id, jalan_geom, nama_jalan)
values (1, st_geomfromtext('linestring(191232 243118, 191108 243242)',-
1,'Mampang Prapatan 1'))
```

Keterangan: untuk mendefinisikan data spasial, setelah mendefinisikan tabel, ditambahkan kolom untuk menyimpan data geometri, kemudian disisipkan. Pada contoh di atas, didefinisikan tabel jalan dengan atribut jalan_id dan nama_jalan kemudian didefinisikan bentuk geometri jalan yaitu *linestring* dan disisipkan koordinat kedua titik ujung dari garis.

4.4 Data Manipulation Language pada PostgreSQL

Contoh Query SQL dalam PostgreSQL:

Cari kecamatan yang ada di Jakarta Timur

```
select kecamatan
from administrasi
where wilayah='JAKARTA TIMUR'
```

PostgreSQL mempunyai *sensitive case* maksudnya dalam pengambilan tupel dalam kolom yang berisi string harus sesuai dengan huruf besar dan kecilnya, karena akan mempengaruhi hasil *query*. Bentuk sintaks untuk query pada PostgreSQL sama seperti pada SQL, dan pada umumnya bentuk sintaks SQL sama untuk semua sistem basis data.

Untuk query spasial diperlukan operator spasial, operator spasial telah di standarisasi oleh lembaga yang bernama OGC (*Open Geospatial Consortium*).

Operasi spasial dalam OGC terbagi menjadi tiga kategori:

1. Operasi Dasar

- Spatial Reference()

Menunjukkan sistem koordinat pada data.

```
select srid(the_geom)
from jalan
```

SRID adalah *Spatial Reference System Identifier*. Setiap objek spasial mempunyai pengenal referensi spasial (SRID). SRID sesuai dengan sistem referensi spasial berdasarkan elipsoid tertentu digunakan baik untuk pemetaan bumi datar atau bulat pemetaan bumi. Sebuah kolom dapat berisi objek spasial dengan SRID berbeda. Namun, hanya contoh spasial dengan SRID yang sama dapat digunakan ketika melakukan operasi query data spasial SQL. Yang paling umum unit pengukuran dari SRID adalah meter atau meter persegi. Jika dua contoh spasial tidak memiliki SRID yang sama, hasil dari query spasial yang digunakan akan bernilai NULL.

- Envelope()

Mengembalikan koordinat batas minimum dari *orthogonal bounding rectangel* geometri.

```
select envelope(the_geom)
from administrasi
where kelurahan='CIRACAS'
```

- Boundary()

Mengembalikan koordinat batas dari geometri.

```
select asewkt(boundary(k.the_geom))
```

```
from administrasi k
```

```
where k.kelurahan='CIRACAS'
```

Perintah `asewkt` digunakan untuk mengembalikan titik koordinat meter dari bentuk koordinat kode ASCII.

- IsSimple()

Mengembalikan nilai *true* jika geometri sederhana (tidak berpotongan).

Misalkan *polygon* yang digunakan dalam penelitian ini merupakan geometri simple karena tidak mengandung *hole*.

- IsEmpty()

Mengembalikan nilai *true* jika geometri merupakan himpunan kosong. Data dalam penelitian ini tidak ada geometri yang berisi himpunan kosong.

2. Himpunan Operator

- Equal

True jika interior dan batas geometri A sama dengan geometri B.

- Disjoint

True jika interior dan batas geometri A dan geometri B tidak berpotongan.

Apakah jalan Bekasi Raya melewati kecamatan Cilandak?

```
select disjoint(w.the_geom,j.the_geom)
```

```
from administrasi w,jalan j
```

```
where w.kecamatan='CILANDAK' and j.nama_jalan='Bekasi Raya';
```

Hasil query: *True* (jalan Bekasi Raya tidak melalui Cilandak)

- Intersect

True jika geometri A berpotongan dengan geometri B.

- Touch

True jika batas geometri A berpotongan dengan batas geometri B.

Cari kecamatan yang bersebelahan dengan kecamatan Pasar Rebo?

```
select distinct c1.kecamatan as PASARREBO_Neighbors
```

```
from administrasi c1, administrasi c2
```

```
where touches(c1.the_geom,c2.the_geom)='T' and c2.kecamatan='PASAR  
REBO';
```

Output: Gambar 4.8 Hasil query touch (terlampir).

- Cross

True jika poligon berpotongan dengan garis.

Cari nama jalan dan kecamatan yang melewati Jakarta Timur:

```
select j.nama_jalan, w.kecamatan
```

```
from jalan j, administrasi w
```

```
where crosses(j.the_geom,w.the_geom)='T'  
and wilayah='JAKARTA TIMUR'
```

Output: Gambar 4.9 Hasil query cross (terlampir).

- Within (A,B)

true jika geometri A seluruhnya ada didalam geometri B.

- Contains (B,A)

true jika geometri A seluruhnya ada di dalam geometri B.

Cari nama fasilitas umum dan jenisnya yang lokasinya ada di kecamatan Pasar Rebo?

```
select c.jenis,c.nama
```

```
from fasum c, administrasi k
```

```
where contains(k.the_geom,c.the_geom)='T'  
and k.kecamatan='PASAR REBO';
```

Hasil query : Gambar 4.10 Hasil query contains (terlampir).

- Overlap

True jika geometri A berpotongan dengan geometri B dengan A dan B berdimensi sama.

Cari kelurahan yang terkena banjir di wilayah Jakarta Timur?

```
select distinct c.kelurahan
```

```
from administrasi c, banjir2002 d
```

```
where overlaps(c.the_geom, d.the_geom)='T' and c.wilayah='JAKARTA  
TIMUR';
```

Hasil query: Gambar 4.11 Hasil query overlap (terlampir)

3. Analisis Spasial

- Distance

Mengembalikan nilai jarak terdekat dari dua geometri.

Tentukan jarak dari apotik Cipayung dan apotik Cibubur?

```
select distance(w.the_geom,j.the_geom)
```

```
from fasum w,fasum j
```

```
where w.nama='Apotik Cipayung' and j.nama='Apotik Cibubur';
```

Hasil query : 2410.43103227511 meter

Jarak yang diberikan merupakan jarak terdekat yang dihitung dengan rumus matematika jarak antara dua titik. Untuk jarak terdekat dengan syarat melewati jalan tidak dapat diaplikasikan, karena geometri fasilitas umum yang berupa titik tidak terdapat pada garis jalan. Namun kasus seperti itu dapat dilakukan dengan spasial *network* yang mungkin dibahas di penelitian berikutnya.

- Buffer

Mengembalikan geometri yang terdiri dari titik titik yang jaraknya sama dengan atau kurang dari geometri yang diberikan .

Cari nama fasilitas umum dan jenisnya yang lokasinya berjarak 100 m dari jalan Lentengagung Barat?

```
select c.jenis,c.nama
```

```
from fasum c, jalan k
```

```
where intersects(c.the_geom,buffer(k.the_geom,100))='T'
```

and k.nama_jalan='Lentengagung Barat';

Hasil query: Gambar 4.13 Hasil query buffer (terlampir)

- Convexhull

Mengembalikan geometri konveks terkecil.

Tentukan convexhull dari kelurahan Balekambang.

```
select asewkt(convexhull(the_geom))
```

```
from administrasi
```

```
where kelurahan='BALE KAMBANG';
```

Hasil query terlampir.

- Intersection

Mengembalikan hasil geometri dari dua geometri yang berpotongan.

Berikan nama jalan dan panjangnya yang melalui di setiap kecamatan?

```
select r.nama_jalan, c.kecamatan,
```

```
length(intersection(r.the_geom,c.the_geom))
```

```
from jalan r, administrasi c
```

```
where crosses(r.the_geom,c.the_geom)='T';
```

Hasil query terlampir.

- Union

Mengembalikan geometri dari dua geometri yang bergabung.

- Difference

Mengembalikan bagian geometri A yang tidak berpotongan dengan geometri B.

- SymDiff

Mengembalikan bagian geometri A dan B yang tidak berpotongan.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini dapat disimpulkan bahwa data vektor dapat disimpan dan diolah dengan basis data objek relasional. Bahasa query untuk mengolah data spasial dalam basis data menggunakan bahasa SQL yang diperluas. Topologi hubungan spasial merupakan dasar dalam klasifikasi operator query spasial.

PostgreSQL adalah salah satu sistem basis data objek relasional yang handal dan murah yang dapat menyimpan serta mengolah data spasial dan data non-spasial secara bersamaan. PostgreSQL dapat berintegrasi dengan *open source* berbasis Sistem Informasi Geografi yang merupakan dua sistem yang berbeda namun saling melengkapi sehingga dapat menciptakan efisiensi pengolahan data spasial yang semakin meningkat.

5.2 Saran

Proses digitasi peta ke dalam data vektor sangat menentukan pengolahan data spasial dalam basis data, sehingga proses digitasi harus lebih akurat. Penelitian ini masih dapat dikembangkan dari berbagai aspek, baik aspek data maupun aspek sistem basis data. Semoga penelitian ini dapat dimanfaatkan untuk pengembangan penelitian berikutnya.

DAFTAR PUSTAKA

- Chatarji, Jagadish. (2003). *Introduction to RDBMS, OODBMS, and ORDBMS*. Mei 28, 2010. <http://www.aspfree.com/c/a/Database/introduction-to-RDBMS-OODBMS-and-ORDBMS/3/>
- Egenhofer, J Max dan Franzosa D Robert. (1991). Point-set Topological Spatial Relations. *International Journal for Geographical Information Systems*, 5(2):161-174.
- Elmasri, Ramez dan Navathe, Shamkant B. (2007). *Fundamental of Database System 5th Edition*. USA: Addison Wesley.
- Guting, Ralf Harmut. 1994. *An Introduction to Spatial Database System*. VLDB Journal (Vol. 3, No.4, October 1994). Germany.
- Hendra, Antonius.(2009). *Pengelolaan Bencana: Identifikasi Wilayah Bencana dengan menggunakan Sistem Informasi Geografis dan Inferensi*. Laporan Tugas Akhir Universitas Indonesia.
- Obe, Regina dan Hsu, Leo. 2009. *PostGIS in Action*. Boston: Manning Publications Co.
- PostGIS 1.5.1 Manual PDF Documentation. 2010. Open Source
- Riyanto et al .(2009). *Pengembangan Aplikasi Sistem Informasi Geografis Berbasis Desktop dan Web*. Yogyakarta: Gava Media.
- Rigaux, Philippe.(2002). *Spatial Databases With Application to GIS*. USA: Morgan Kaufmann Publisher.
- Rokhmatuloh .(2010). *Kumpulan Informasi Jakarta CD*. Universitas Indonesia: Laboratorium SIG Departemen Geografi

Shekhar, Shashi dan Cawla, Sanjay. (2003). *Spatial Databases: A Tour*. Prentice Hall

Sherman, Gary.(2008). *Desktop GIS Mapping the Planet with Open Source Tools*.
Texas: The Pragmatic Bookself.

Wisnani, Yahma. (2007). *Diktat Mata Kuliah Basis Data*. Depok: Departemen
Matematika Universitas Indonesia

Yeung, Albert, & Hall, G Brent.(2007). *Spatial Database Systems Design,
Implementation and Project Management*. Netherlands: Springer.



Lampiran 1

Algoritma pencarian titik dengan *R-tree*:

- RT-POINTQUERY (*P*: point): set (oid)

```

begin
  result = ∅
  // Step 1: Traverse the tree from the root, and compute SL
  // the set of leaves whose dr contains P
  SL = RTREETRAVERSAL (root, P)
  // Step 2: scan the leaves, and keep the entries that contain P
  for each L in SL do // Scan the entries in the leaf L
    for each e in L do
      if (e.mbb contains P) then result += {e.oid}
    end for
  end for
  return result
end

```
- RTREETRAVERSAL (nodeid: PageID, P: point): set of leaves

```

begin
  result = ∅
  // Get the node page
  N = READPAGE (nodeid)
  if (N is a leaf) return {N}
  else
    // Scan the entries of N, and visit those that contain P
    for each e in N do
      if (e.dr contains P) then
        result += RTREETRAVERSAL (e.nodeid, P)
      end if
    end for
  end if
  return
  result end

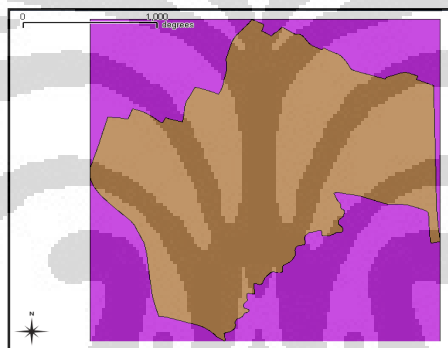
```

Lampiran 2

Hasil query spasial dengan PostgreSQL:

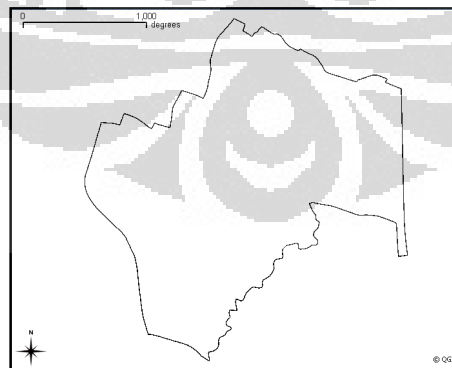
Tabel 4.1 Geometri envelope dari kelurahan Ciracas

Koordinat dalam ASCII	Koordinat Kartesian
"01030000000100000005000000104CEC34CC8C2541B83D15FC7CBC6141104CEC34CC8C2541DBA574F9DCBD614144393F323DA12541DBA574F9DCBD614144393F323DA12541B83D15FC7CBC6141104CEC34CC8C2541B83D15FC7CBC6141"	"POLYGON((706150.1033653049298919.87759291,706150.1033653049301735.79548924,708766.5981386099301735.79548924,708766.5981386099298919.87759291,706150.1033653049298919.87759291))"



Keterangan: Ungu: envelope, coklat: kelurahan Ciracas

Gambar 4.8 Output visual query envelope dengan perluasan PostgreSQL dengan QGIS



Keterangan: garis hitam: garis batas kelurahan CIRACAS

Gambar 4.9 Output visual query boundary

	pasarrebo_neighbors character varying(20)
1	JAGAKARSA
2	KRAMAT JATI
3	PASAR REBO
4	CIRACAS
5	PASAR MINGGU

Keterangan: Nama kecamatan yang berdampingan dengan kecamatan Pasar Rebo

Gambar 4.10 Hasil query touch

	nama_jalan character varying(32)	kecamatan character varyin
1	Perintis Kemerdekaan	PULO GADUNG
2	Let. Jend. Suprpto	PULO GADUNG
3	Jend. Ahmad Yani	PULO GADUNG
4	Ir. Wiyoto Wiyono (Toll)	PULO GADUNG
5	Ir. Wiyoto Wiyono (Toll)	PULO GADUNG
6	Jend. Ahmad Yani	PULO GADUNG
7	Laks. M. Yos Sudarso	PULO GADUNG
8	Laks. M. Yos Sudarso	PULO GADUNG
9	Perintis Kemerdekaan	PULO GADUNG
10	NA	PASAR REBO
11	T.B. Simatupang	PASAR REBO
12	Tol Lingkar Luar	PASAR REBO
13	Tol Lingkar Luar	PASAR REBO
14	Bogor Raya	PASAR REBO
15	Bogor Raya	PASAR REBO
16	Tambak	MATRAMAN
17	Tambak	MATRAMAN

Keterangan: Sebagian nama jalan yang melalui setiap kecamatan di Jakarta Timur

Gambar 4.11 Hasil query cross

	jenis character varying(15)	nama character varying(20)
1	Apotik	Apotik Jaga Waras
2	Apotik	Apotik Elyana
3	Apotik	Apotik Gongseng Farm
4	Bank	Bank Exim
5	Bank	Bank BNI
6	Bank	Bank BRI
7	Bank	Bank BRI
8	Ktr. Pemerintah	Kt.Depsos
9	Ktr. Pemerintah	Kt.P & K
10	Ktr. Pemerintah	Kt.LAPAN
11	Ktr. Pemerintah	Dinas Kebakaran
12	Kantor Lurah	Kel. Gedong
13	Kantor Lurah	Kel. Cijantung
14	Kantor Lurah	Kel. Kali Sari
15	Kantor Lurah	Kel. Baru
16	Kantor Camat	Kec. Pasar Rebo
17	Kantor Lurah	Kel. Pekayon
18	Puskesmas	Kel. Gedong
19	Puskesmas	Kel. Cijantung 1
20	Puskesmas	Kel. Cijantung 2
21	Puskesmas	Kel. Kalisari
22	Puskesmas	Kec. Pasar Rebo
23	Puskesmas	Kel. Baru
24	RS Umum	RSU Pasar Rebo
25	RS Umum	RS St. Carolus

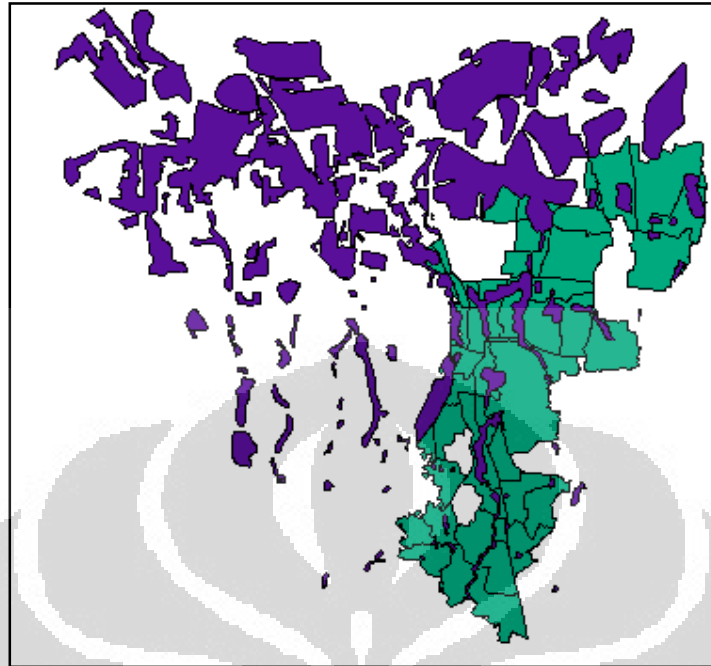
Keterangan: Nama fasilitas umum yang ada di Pasar Rebo

Gambar 4.12 Hasil query contains

	kelurahan character varying(25)
1	KAMPUNG MELAYU
2	UJUNG MENTENG
3	CILILITAN
4	PULO GADUNG
5	KAYU PUTIH
6	MAKASAR
7	KRAMAT JATI
8	KEBON PALA
9	KELAPA DUA WETAN
10	CEGER
11	BAMBU APUS
12	PALMERIAM
13	BATU AMPAR
14	CIPINANG BESAR SELATAN
15	CAKUNG TIMUR
16	CIRACAS
17	BARU
18	BIDARA CINA
19	RAWA BUNGA
20	DUKUH
21	CIPAYUNG
22	BALE KAMBANG
23	RAWA TERATE
24	PINANG RANTI
25	PONDOK KELAPA
26	BALI MESTER
27	CIPINANG BESAR UTARA
28	HALIM PERDANA KUSUMAH
29	JATINEGARA
30	KEMASER

Keterangan: Sebagian nama kelurahan di Jakarta timur yang terkena banjir

Gambar 4.13 Hasil query overlap



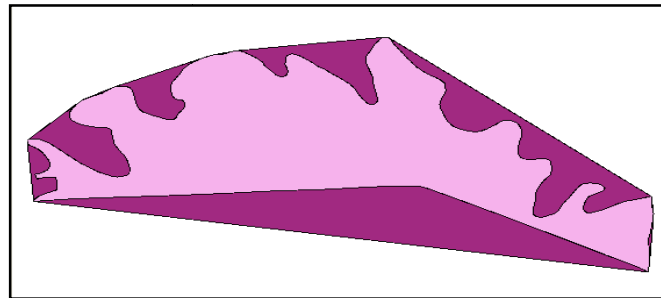
Keterangan: Biru: daerah banjir.
Hijau: kelurahan di Jakarta Timur yang terkena banjir.

Gambar 4.14 Output visual query overlap

	jenis character varying(15)	nama character varying(20)
1	Apotik	Apotik Rasuna
2	Ktr. Pemerintah	Kantor PLN
3	Apotik	Apotik
4	Apotik	Apotik Rasuna
5	Apotik	Apotik Lenteng Agung

Keterangan: Fasilitas umum yang jaraknya 100 meter dari jalan Lentengagung Barat

Gambar 4.15 Hasil query buffer



Keterangan: Ungu: convexhull, pink: kelurahan Bale Kambang

Gambar 4.16 Hasil query convexhull

	nama_jalan character varying(32)	kecamatan character varying(20)	length double precision
1	Boulevard Timur	KELAPA GADING	111.39769794567
2	Pegangsaan Dua	KELAPA GADING	608.080625587269
3	Danau Sunter	KELAPA GADING	90.5400464122397
4	Danau Sunter	KELAPA GADING	92.223327958053
5	Laks. M. Yos Sudarso	KELAPA GADING	152.160818671804
6	Boulevard Barat	KELAPA GADING	326.965405226474
7	Perintis Kemerdekaan	KELAPA GADING	66.3897024423089
8	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	185.146056694221
9	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	184.452308160232
10	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	181.775106548015
11	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	182.913310697131
12	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	1891.04418649587
13	Ir. Wiyoto Wiyono (Toll)	KELAPA GADING	1888.22810876644
14	Laks. M. Yos Sudarso	KELAPA GADING	856.838128168133
15	Laks. M. Yos Sudarso	KELAPA GADING	572.173801700016
16	Laks. M. Yos Sudarso	KELAPA GADING	6.58618447958909
17	Perintis Kemerdekaan	KELAPA GADING	162.384524354073
18	Perintis Kemerdekaan	KELAPA GADING	125.389696798552
19	Pahlawan	CILINCING	24.2715483875972
20	Tipar Cakung	CILINCING	260.915666804198
21	Perintis Kemerdekaan	PULO GADUNG	70.2425121376776
22	Let. Jend. Suprpto	PULO GADUNG	103.633259465541
23	Jend. Ahmad Yani	PULO GADUNG	2441.88316041452
24	Ir. Wiyoto Wiyono (Toll)	PULO GADUNG	2484.70208404208
25	Ir. Wiyoto Wiyono (Toll)	PULO GADUNG	2484.16530162395
26	Jend. Ahmad Yani	PULO GADUNG	709.922530039068
27	Laks. M. Yos Sudarso	PULO GADUNG	36.5694802431237

Keterangan: Sebagian nama jalan dan panjang jalan yang melalui tiap kecamatan

Gambar 4.17 Hasil query *intersection* dan *length*