

BASIS DATA SPASIAL

Outline

- Data Spasial : Data Geografis
- Representasi Data Geografis
- Basis Data Spasial
- Sistem Manajemen Basis Data Spasial
- Tipe Queri Spasial
- Bahasa Queri Spasial
- Contoh Queri Spasial
- Pengindeksan Data Spasial

Data Spasial

- Dua tipe data spasial :
 - Data geografis (2D): peta jalan, peta penggunaan tanah, peta kepemilikan tanah, peta politis yg menunjukkan perbatasan, citra medis, cuaca dll.
 - Data *Computer-aided Design* (CAD) (2D/3D): informasi spasial mengenai konstruksi dari sebuah objek, seperti bangunan, mobil, pesawat terbang.

Data Geografis

2 tipe data geografis :

1. Data raster, terdiri atas data peta piksel (2/3D). Cth, citra satelit mengenai awan (2D), citra mengenai suhu pada wilayah dgn ketinggian berbeda.
2. Data vektor, data ini terbentuk berdasarkan objek geometris dasar seperti garis, titik, segitiga atau bentuk poligon lainnya dalam 2D. Atau bentuk silinder, bola, kubus & bentuk lainnya dalam 3D.

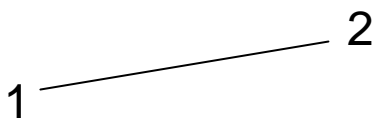
Representasi Data Geometris

- Sebuah segmen garis dapat direpresentasikan dgn koordinat dari titik2 akhirnya.

Objek

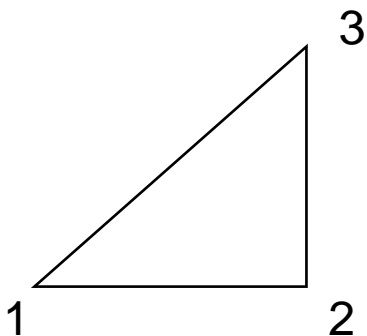
Representasi

- Segmen garis



$\{(x_1,y_1),(x_2,y_2)\}$

- Segitiga



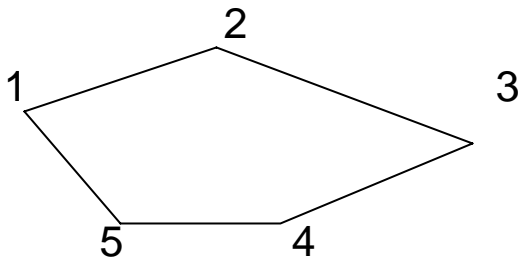
$\{(x_1,y_1),(x_2,y_2),(x_3,y_3)\}$

Representasi Data Geometris

Objek

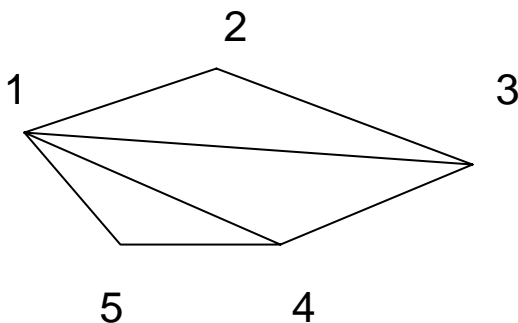
Representasi

- Poligon



$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)\}$

- Poligon



$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ID1\}$
 $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ID2\}$
 $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ID3\}$

Basis Data Spasial

- Basis data spasial merupakan kumpulan dari tipe data spasial, operator, indice, strategi pemrosesan dll. Dapat bekerja pada bahasa pemrograman Java, Visual Basic dll.
- Basis data spasial yg khusus menyimpan data geografis terdapat pada *Geographic Information System (GIS)*.
- Objek spasial
 - Mempunyai bentuk geometris, ukuran & lokasi (koordinat)
 - Berubah menurut waktu, cth penggunaan tanah, wilayah penduduk

Basis Data Spasial (lanj.)

- Isu
 - **Representasi**; bagaimana merepresentasikan gunung, danau?
 - **Bahasa query**; bagaimana menspesifikasikan query spasial?
 - Temukan semua danau di dekat gunung ?
 - **Indeksing**; bagaimana meningkatkan metode pengaksesan spasial?
 - **Pengumpulan data**; bagaimana mendigitasi citra satelit, peta pada kertas, dll

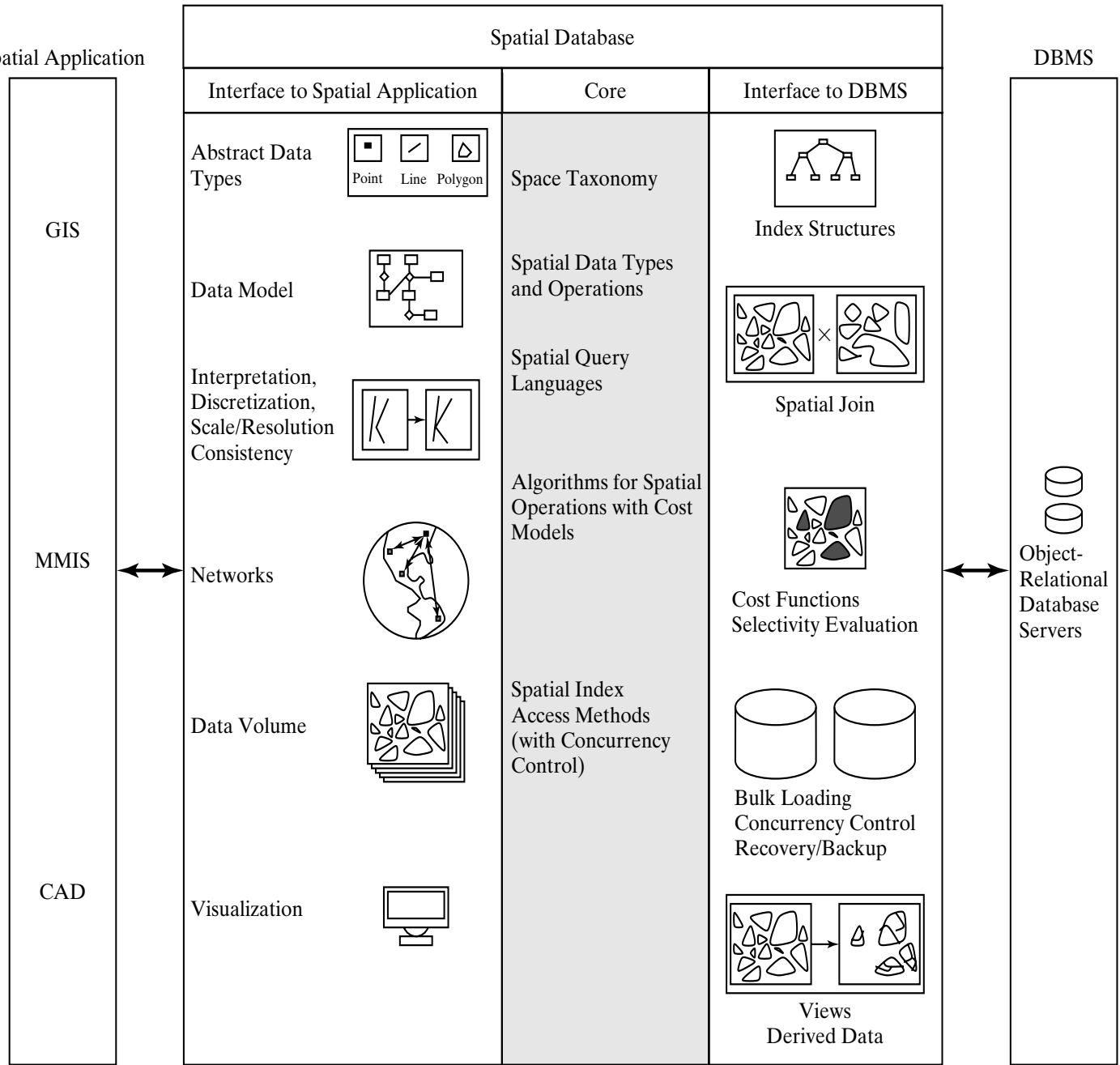
Sistem Manajemen Basis Data Spasial (SMBDS)

- SMBDS merupakan suatu perangkat lunak :
 - Dapat bekerja seperti DBMS yg ada
 - Mendukung model data spasial, tipe data abstrak spasial (ADT/*Abstract Data Type*) & bahasa query yg dapat memanggil ADT
 - Mendukung indeksing spasial, algoritma pemrosesan operasi spasial yg efisien, & aturan domain yg khusus utk optimasi query.
 - Digunakan pada GIS ataupun aplikasi lainnya.

Sistem Manajemen Basis Data Spasial (SMBDS)

- Contoh: *Oracle Spatial data cartridge*, ESRI SDE
 - Dapat bekerja dgn Oracle 8i DBMS
 - Mempunyai tipe data spasial (mis. poligon), operasi (cth. overlap) yg dpt dikerjakan dgn SQL3
 - Mempunyai indeksing spasial, mis. R-trees

Arsitektur SMBDS



Tipe Queri Spasial

- *Neareness queries*

Meminta objek yg berada dekat lokasi tertentu. Cth, queri utk mencari semua restoran yg berada dlm jarak ttt dari suatu titik ttt.

- *Region queries*

Meminta objek yg berada sebagian atau keseluruhan pada area ttt. Cth, queri utk mencari semua toko eceran di dalam suatu kota.

Tipe Queri Spasial

- *Queries that request intersections and unions of regions*

Meminta objek yg berada pada area yg beririsan atau gabungan dari beberapa area. Cth, dari informasi tahunan mengenai curah hujan & kepadatan penduduk, diajukan queri utk mendapatkan semua area dengan curah hujan rendah & kepadatan penduduk tinggi.

Perhitungan queri ini dgn melakukan join dari 2 relasi spasial.

Bahasa Query Spasial

- Bahasa Query Spasial
 - Tipe data spasial, mis poin, linestring, poligon
 - Operasi spasial, cth. *overlap*, *distance*, *nearest neighbor*.
 - Dapat dipanggil dari bahasa query, cth. SQL3

```
SELECT Sname
FROM Senator S
WHERE S.district.Area() > 300
```
- Standar
 - Bahasa query standar : SQL3
 - OGIS (*Open Geodata Interchange Standar*) : standar utk tipe data spasial & operator

Contoh Querri Spasial

- Contoh Join Spasial

```
SELECT S.name FROM Senator S,  
Business B  
WHERE S.district.Area() > 300  
AND Within(B.location, S.district)
```

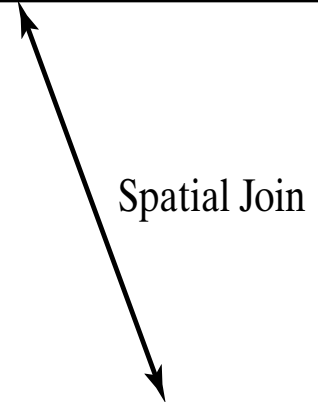
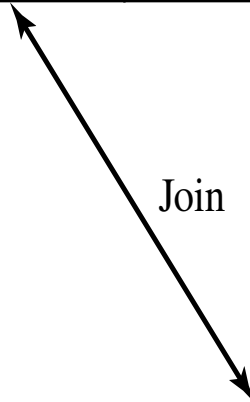
- Contoh Join Non-Spasial

```
SELECT S.name FROM Senator S,  
Business B  
WHERE S.soc-sec = B.soc-sec  
AND S.gender = 'Female'
```

Contoh Querri Spasial (lanj.)

SENATOR

NAME	SOC.-SEC	GENDER	DISTRICT (POLYGON)
------	----------	--------	--------------------

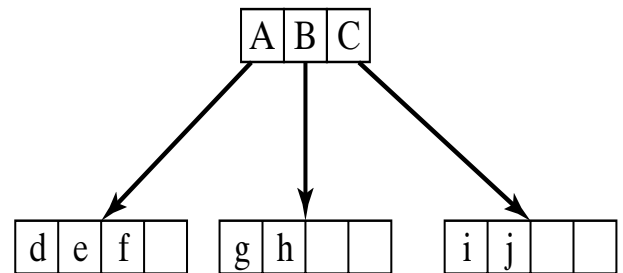
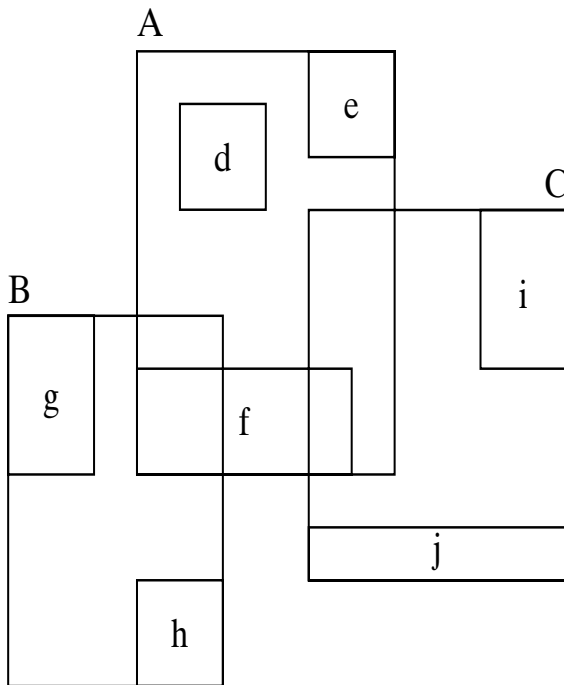


BUSINESS

B-NAME	OWNER	SOC-SEC	LOCATION (POINT)
--------	-------	---------	------------------

Pengindeksan Data Spasial

- Untuk mengindeks data spasial (poligon, persegiipjg) digunakan *R-tree* (perluasan *B-tree*)
- R-tree merupakan hirarki dari kumpulan persegiipjg.



Data Spasial OGIS

- Terdiri atas klas geometri & 4 subklas :
 - Point, Curve, Surface & Geometry Collection*
- 3 kategori operasi :
 - Diaplikasikan utk semua tipe geometri
 - SpatialReference, Envelope, Export, IsSimple, Boundary*
 - Predikat utk relationship topologis
 - Equal, Disjoint, Intersect, Touch, Cross, Within, Contains*
 - Analisis data spasial
 - Distance, Buffer, Union, Intersection, ConvexHull, SymDiff*

Contoh SQL Sederhana SELECT_FROM_WHERE

- Operasi analisis spasial
 - Operator Unary : *Area*
 - Operator Binary : *Distance*
- Operasi spasial topologis boolean –
clause WHERE
 - Touch*
 - Cross*
- Menggunakan analisis spasial & operasi topologis
 - Buffer, overlap*
- Contoh SQL kompleks
 - Aggreagate SQL queries*
 - Nested queries*

Operasi Spasial pada clause SELECT

- Queri: tampilkan nama, populasi & area untuk setiap tabel Country.

```
SELECT C.Name,C.Pop,  
Area(C.Shape)AS "Area"  
FROM Country C
```

– Cat: *Area()* : operasi spasial

Operator Spasial *Distance*

- Queri: tampilkan GDP & jarak ibukota negara ke ekuator dari semua negara.

```
SELECT Co.GDP,
Distance(Point(0,Ci.Shape.y),Ci.Shape)
AS "Distance"
FROM Country Co, City Ci
WHERE Co.Name = Ci.Country
AND Ci.Capital = 'Y'
```

Co. Name	Co. GDP	Dist-to-Eq (in Km).
Havana	16.9	2562
Washington, D.C.	8003	4324
Brasilia	1004	1756
Ottawa	658	5005
Mexico City	694.3	2161
Buenos Aires	348.2	3854

Operasi Spasial pada clause WHERE

- Queri: Cari nama semua ibukota negara yg bertetangga dgn United States (USA) pada tabel Country.

```
SELECT C1.Name AS "Neighbors of  
USA" FROM Country C1, Country C2  
WHERE Touch(C1.Shape, C2.Shape)=1  
AND C2.Name ='USA '
```

- Cat: Operator spasial *Touch()* digunakan utk men-join tabel Country table dengan dirinya sendiri (rekursif).

Querri Spasial dengan Tabel Multipel

- Querri: untuk semua sungai yang terdapat pada tabel River, cari daerah yg dilalui oleh sungai tsb.

```
SELECT R.Name, C.Name  
FROM River R, Country C  
WHERE Cross(R.Shape,C.Shape)=1
```

- Cat: Operasi spasial *Cross* utk men-join tabel River & Country.

Contoh Query Spasial

Buffer & Overlap

- Query: sungai St. Lawrence dapat memasok air ke kota2 dalam jarak 300 km. Tampilkan kota2 yg dapat menggunakan air dr sungai tsb.

```
SELECT Ci.Name  
FROM City Ci, River R WHERE  
Overlap(Ci.Shape, Buffer(R.Shape,300))  
= 1 AND R.Name = 'St.Lawrence '
```


Operasi Spasial pada *aggregate query*

- Queri: Tampilkan semua daerah, susun berdasarkan jumlah daerah tetangganya.

```
SELECT Co.Name, Count(Co1.Name)
FROM Country Co, Country Co1
WHERE Touch(Co.Shape,Co1.Shape)
GROUP BY Co.Name
ORDER BY Count(Co1.Name)
```

Using Spatial Operation in *Nested Queries*

- Queri: Untuk setiap sungai, identifikasikan kota terdekat.

```
SELECT C1.Name, R1.Name  
FROM City C1, River R1  
WHERE Distance (C1.Shape,R1.Shape)  
<= ALL ( SELECT  
Distance(C2.Shape)FROM City  
C2WHERE C1.Name <> C2.Name)
```

- Cat: Operasi spasial *Distance* digunakan dalam konteks *nested query*.

Nested Spatial Query

- **Queri:** Tampilkan daerah yg hanya mempunyai satu tetangga saja. Suatu daerah merupakan tetangga daerah lainnya jika tanah mereka saling berbatasan. Berarti, daerah kepulauan tidak mempunyai tetangga, seperti Iceland.

```
SELECT Co.Name FROM Country Co
WHERE Co.Name IN (SELECT Co.Name
FROM Country Co,Country Co1
WHERE Touch(Co.Shape,Co1.Shape)
GROUP BY Co.NameHAVING Count(*)=1)
```

Referensi

1. Korth, H.; **Database System Concept**,
Mc Graw Hill, 4th edition, New York,
2002