

# Konsep Basis Data dalam GIS

**Arif Basofi**  
**PENS 2015**





# Contents

1

Proyeksi Peta

2

Jenis Proyeksi Peta

3

Pemilihan Proyeksi Peta

4

Sistem Proyeksi Peta Indonesia



# Trend Basis Data Spasial

- ❖ Hampir semua perangkat lunak SIG telah memiliki **format sendiri** untuk **menyimpan datanya**.
- ❖ Namun, untuk **data atribut**, mereka menggunakan sistem pengelolaan basis data yang sudah ada. Contoh: **ArcGIS** menggunakan format **.dbf** yang merupakan format dari DBMS “**DBase**”.
- ❖ **Keuntungan menggunakan sistem basis data lain adalah:**
  - ✧ Pembuat perangkat lunak SIG dapat berkonsentrasi untuk mengembangkan fungsi-fungsi inti dari SIGnya → tidak perlu meneliti dan mengembangkan sistem basis data sendiri.
  - ✧ Sebagian besar arsitektur table relasional sudah terbuka, sehingga tidak ada kesulitan dalam menggunakan sistem basis data tertentu ataupun berpindah antara satu DBMS ke DBMS yang lain.
  - ✧ Dapat memilih berbagai macam DBMS mulai dari yang gratis dengan fitur sederhana sampai dengan yang sangat mahal dengan fitur yang sangat lengkap.



# Spatial Extension

- ❖ Sebagian besar DBMS tidak dapat secara langsung mendukung penyimpanan data spasial → harus menggunakan **Spatial Extension**
- ❖ Berikut ini beberapa DBMS yang dapat digunakan untuk menyimpan data spasial:
  - ✧ MySQL dengan MySQL Spatial Support
  - ✧ Oracle dengan Oracle Spatial
  - ✧ PostGreSQL dengan PostGIS
  - ✧ Microsoft Access
  - ✧ Microsoft SQL Server dengan MSSQLSpatial
  - ✧ IBM DB2 dengan Spatial Extender & Geodetic Extender
- ❖ Beberapa tipe DBMS hanya dapat menyimpan data spasial saja, tanpa kemampuan untuk melakukan analisa spasial.



# Spasial DBMS

- ❖ **Spasial DBMS (SDBMS):** merupakan DBMS yang selain menyediakan dalam pengelolaan basis data biasa, juga memiliki kemampuan dalam penyimpanan dan pengelolaan data spasial.
- ❖ Dapat juga berupa *middleware* (contoh: ArcSDE).
- ❖ Dapat diimplementasikan baik sebagai *thick* maupun *thin* client (Contoh: CGI vs Java).
- ❖ **SDBMS:**
  - ❖ Bekerja diatas DBMS biasa
  - ❖ Mengijinkan model dan tipe data spasial
  - ❖ Mendukung bahasa untuk meng-query tipe data spasial
  - ❖ Mendukung pembuatan indeks spasial
  - ❖ Mempunyai algoritma yang efisien untuk operasi-operasi spasial
  - ❖ Memiliki aturan-aturan khusus untuk optimasi query



# Keuntungan SDBMS

- ❖ SDBMS menyediakan struktur untuk **penyimpanan** dan **analisa data spasial**.
- ❖ **Data spasial** terdiri dari objek-objek dalam ruang **multi-dimensi**. Tidak seperti table konvensional yang hanya memiliki dua dimensi (baris dan kolom).
- ❖ Menyimpan data spasial dalam DBMS standar **membutuhkan tempat penyimpanan** dalam jumlah yang **sangat besar**.
- ❖ Mengambil dan menganalisa data spasial dari DBMS standar membutuhkan **waktu yang lama** serta **sulit** dilakukan → dapat mengakibatkan **banyak kesalahan**.
- ❖ SDBMS menyediakan tempat penyimpanan, proses pengambilan, dan proses analisa data spasial yang **jauh lebih efisien**.



# Tipe Data yang Disimpan dalam SDBMS

## ❖ Data spasial dua dimensi:

- ❖ Koordinat geography
- ❖ Koordinat cartesian (2D)
- ❖ Jaringan
- ❖ Direction (arah)

## ❖ Data spasial tiga dimensi:

- ❖ Cuaca
- ❖ Koordinat Cartesian (3D)
- ❖ Topologi
- ❖ Foto satelit



# Penggunaan dan Pengguna SDBMS

## ❖ Tiga tipe penggunaan SDBMS:

- ❖ Mengelola data spasial
- ❖ Menganalisa data spasial
- ❖ Penggunaan data spasial tingkat tinggi

## ❖ Beberapa contoh pengguna SDBMS:

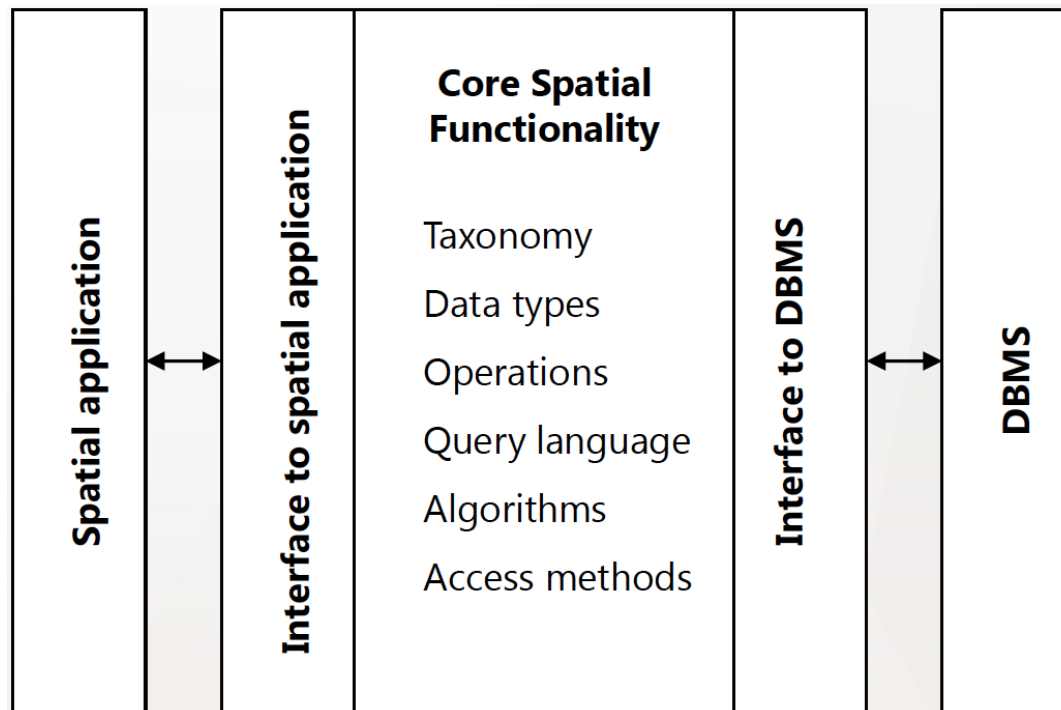
- ❖ Agen transportasi untuk memantau proyek-proyeknya
- ❖ Manajer asuransi untuk mempertimbangkan profil lokasi beresiko
- ❖ Dokter untuk membandingkan data Magnetic Resonance Images (MRI)
- ❖ Sistem tanggap darurat untuk mencari jalur tercepat ke korban
- ❖ Perusahaan selular untuk memantau penggunaan telepon





# SDBMS Three-layer Structure

- ❖ **SDBMS** bekerja dengan aplikasi spasial di **sisi depan** dan **DBMS** di **sisi belakang**.
- ❖ **SDBMS** memiliki tiga lapisan:
  - ❖ Antarmuka ke aplikasi spasial
  - ❖ Fungsi-fungsi inti terkait data spasial
  - ❖ Antarmuka ke DBMS





# Spatial Query Language

- ❖ Beberapa adaptasi dari SQL untuk data spasial:
  - ✧ Spatial Query Language
  - ✧ Temporal Query Language (TSQL2)
  - ✧ Object Query Language (OQL)
  - ✧ Object Oriented Structured Query Language (O<sub>2</sub>SQL)
- ❖ Spatial query language menyediakan peralatan dan struktur khusus untuk bekerja dengan data spasial.
- ❖ SQL3 menyediakan tipe-tipe data spasial 2D beserta fungsi-fungsinya



# Spatial Query Language...

## ❖ Tiga tipe query :

- ❖ Operasi-operasi dasar pada semua tipe data (misal: IsEmpty, Envelope, Boundary).
- ❖ Operasi topologi beserta serangkaian operatornya (misal: Disjoint, Touch, Contains)
- ❖ Analisa spasial (missal: Distance, Intersection, SymmDiff)



# Pembuatan Entitas Data Spasial

- ❖ Membuat entitas **KABUPATEN** untuk menyimpan nama kabupaten, nama propinsi, populasi, serta data geografinya:

```
CREATE TABLE Kabupaten (  
  Nama varchar(30) ,  
  Propinsi varchar(30) ,  
  Pop Integer ,  
  Shape Polygon) ;
```

- ❖ Membuat entitas **SUNGAI** untuk menyimpan nama sungai, panjang, serta data geografinya:

```
CREATE TABLE Sungai (  
  Nama varchar(30) ,  
  Panjang Integer ,  
  ShapeLine String) ;
```



# Contoh Query Spasial

- ❖ Cari semua kabupaten yang berbatasan dengan kabupaten Bojonegoro:

```
SELECT K1.Nama
FROM Kabupaten K1, Kabupaten K2
WHERE Touch(K1.Shape, K2.Shape) = 1 AND
      K2.Nama = 'Bojonegoro';
```

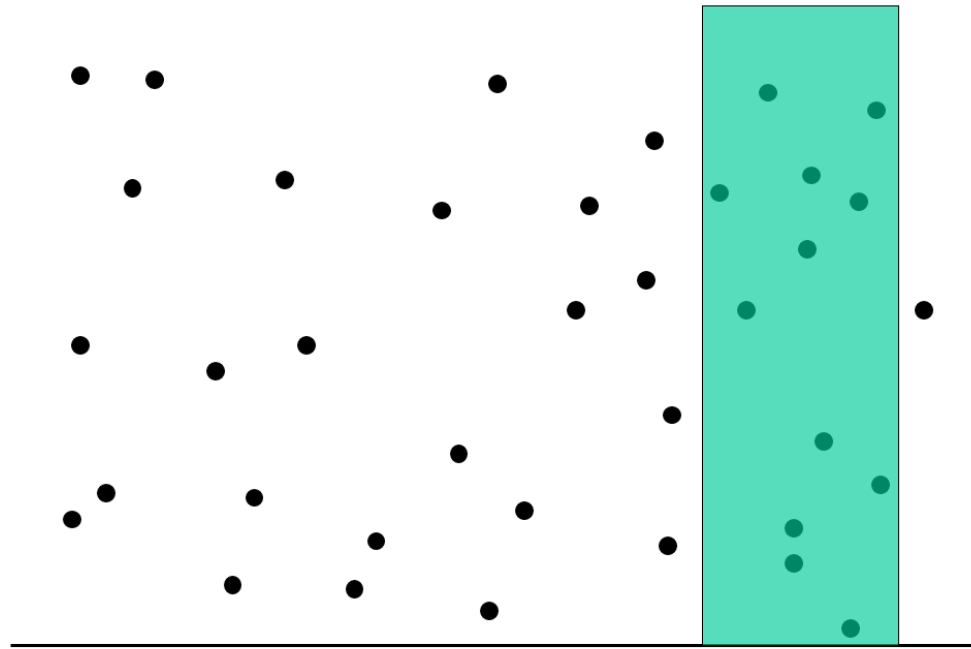
- ❖ Cari semua kabupaten yang dilewati sungai Brantas:

```
SELECT K.Nama, S.Nama
FROM Kabupaten K, Sungai S
WHERE Intersect(K.Shape, S.Shape) = 1 AND
      S.Nama = 'Brantas';
```



# Indeks pada data spasial

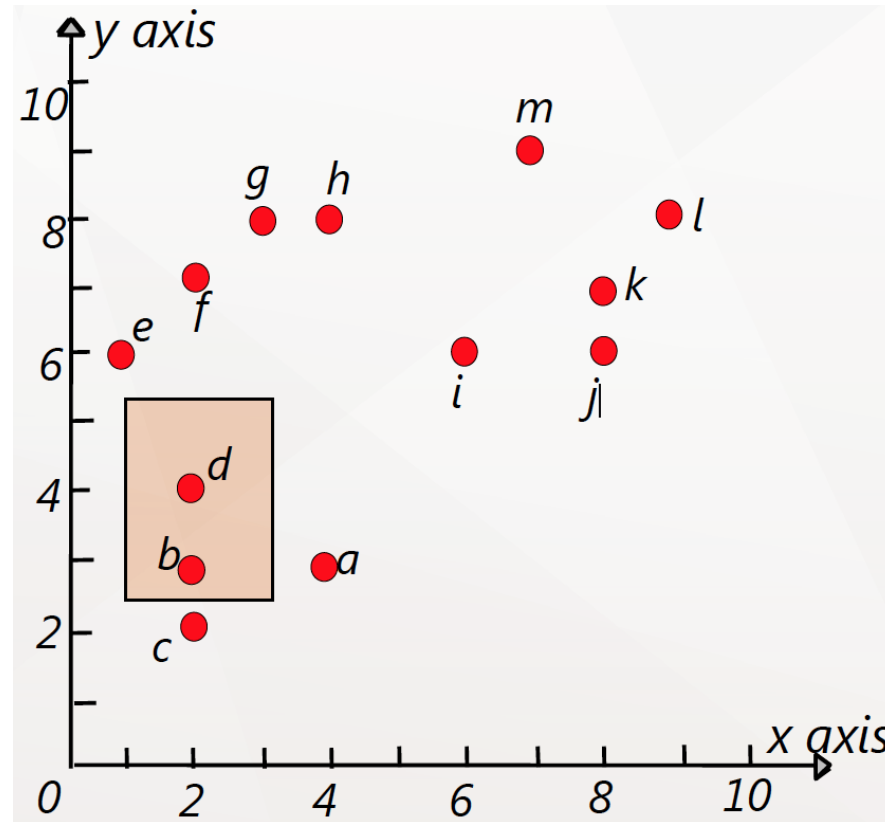
- ❖ Fungsi indeks pada basis data adalah untuk **mempercepat proses pencarian** data dalam query.
- ❖ **Indeks 1-dimensi** yang biasa digunakan pada basis data biasa (**B-Tree**) tidak dapat digunakan untuk data spasial.
- ❖ **Data spasial** menggunakan **R-Tree indeks**. Huruf **R** berarti **Rectangle**.



Suppose a B<sup>+</sup>-tree exists on X.



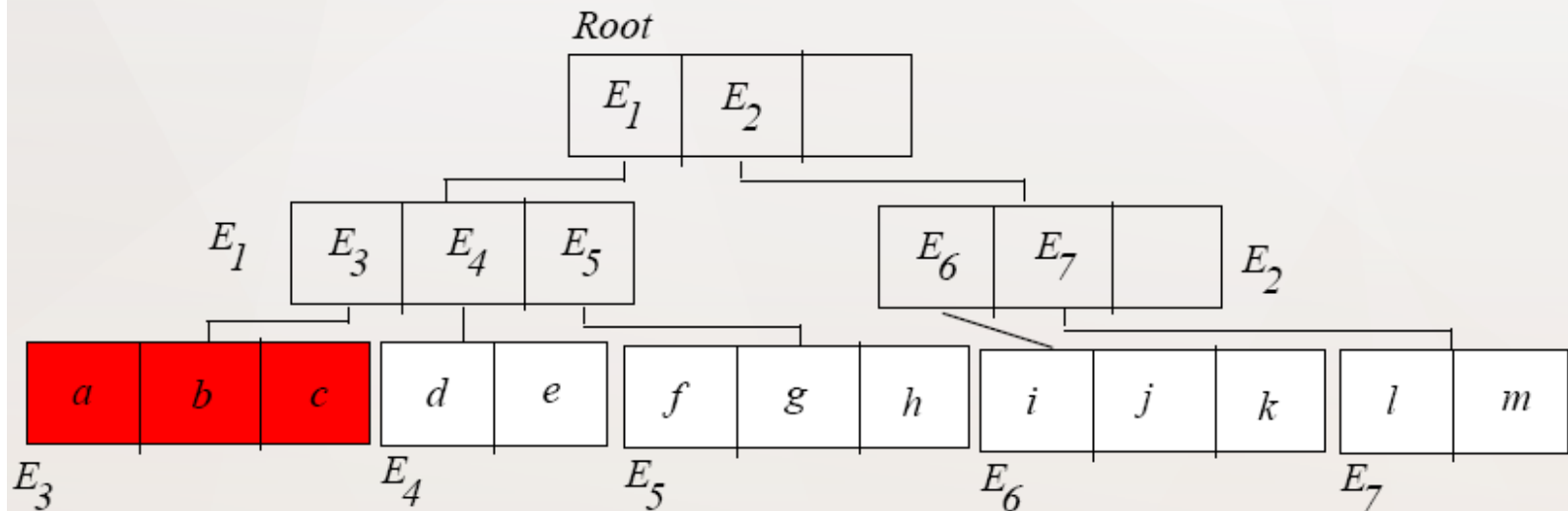
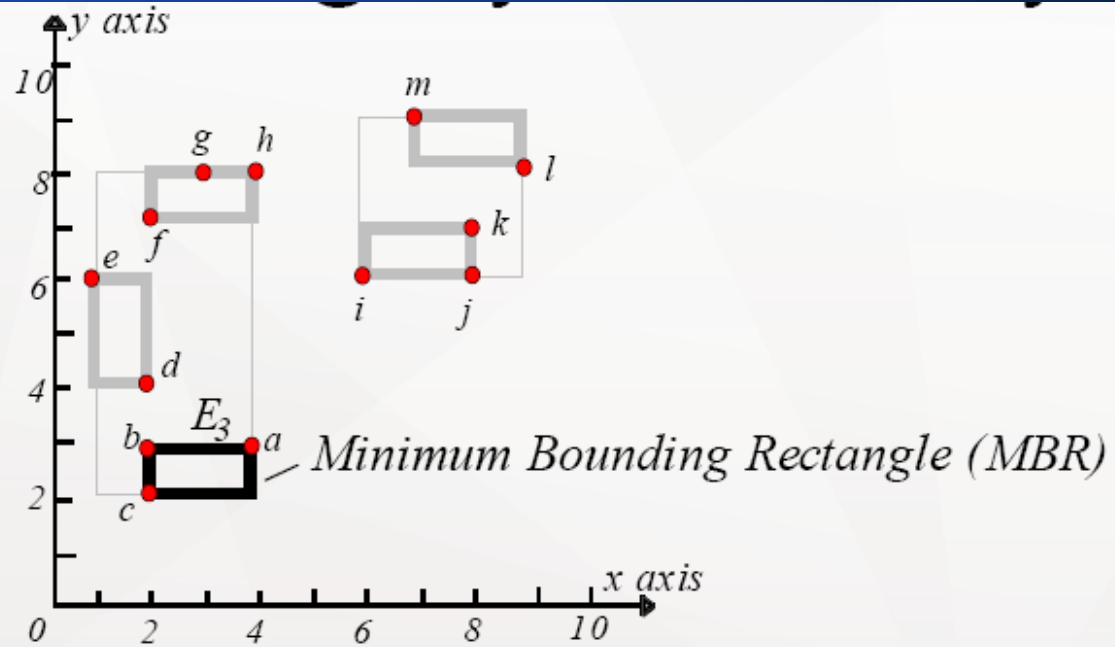
# R-Tree Motivation



- ❖ **Range query:** find the objects in a given range.
- ❖ E.g. find all hotels in Boston.
- ❖ No index: scan through all objects. NOT EFFICIENT!



# R-Tree: Clustering by Proximity

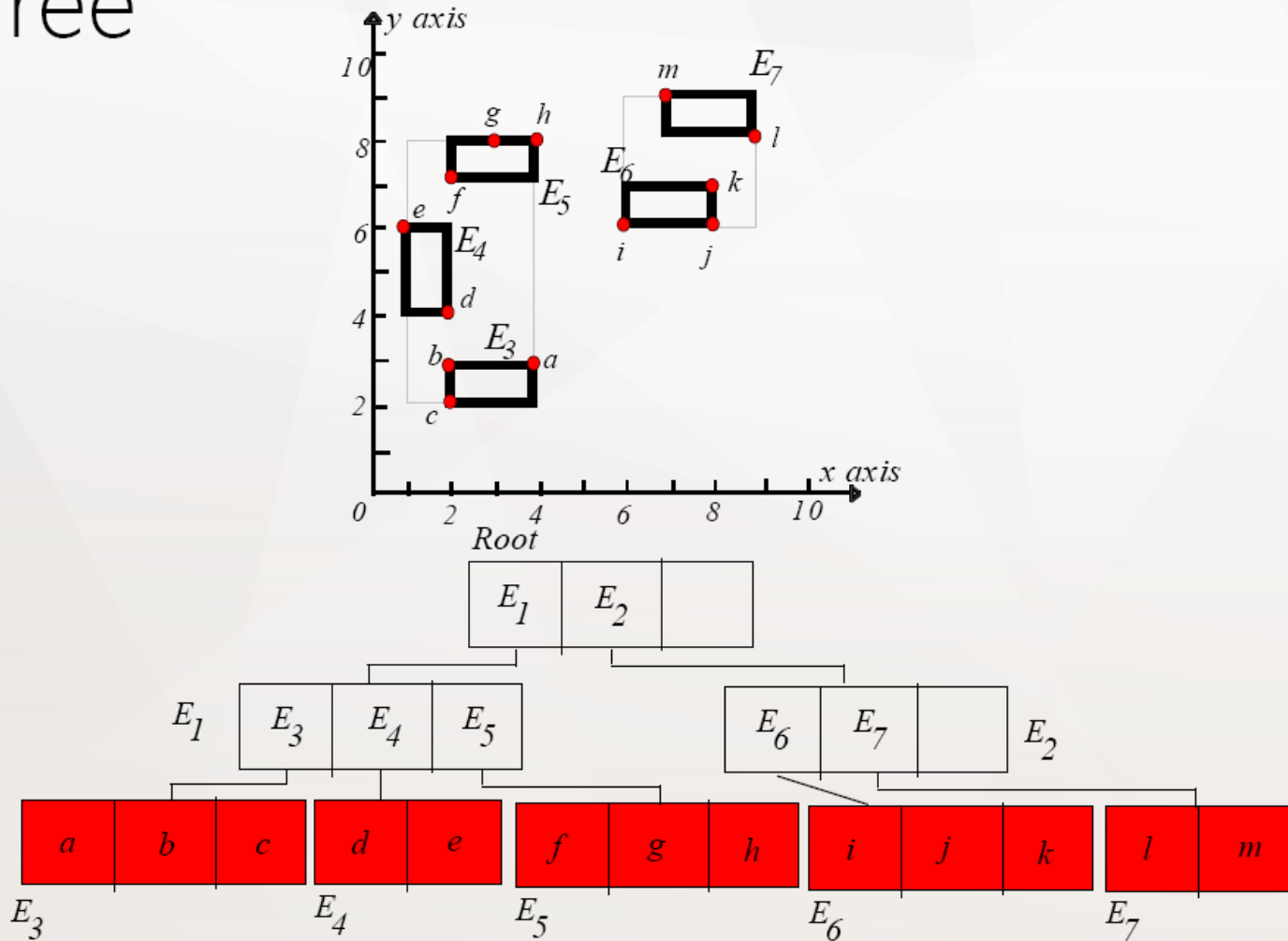






# R-Tree: Clustering by Proximity...

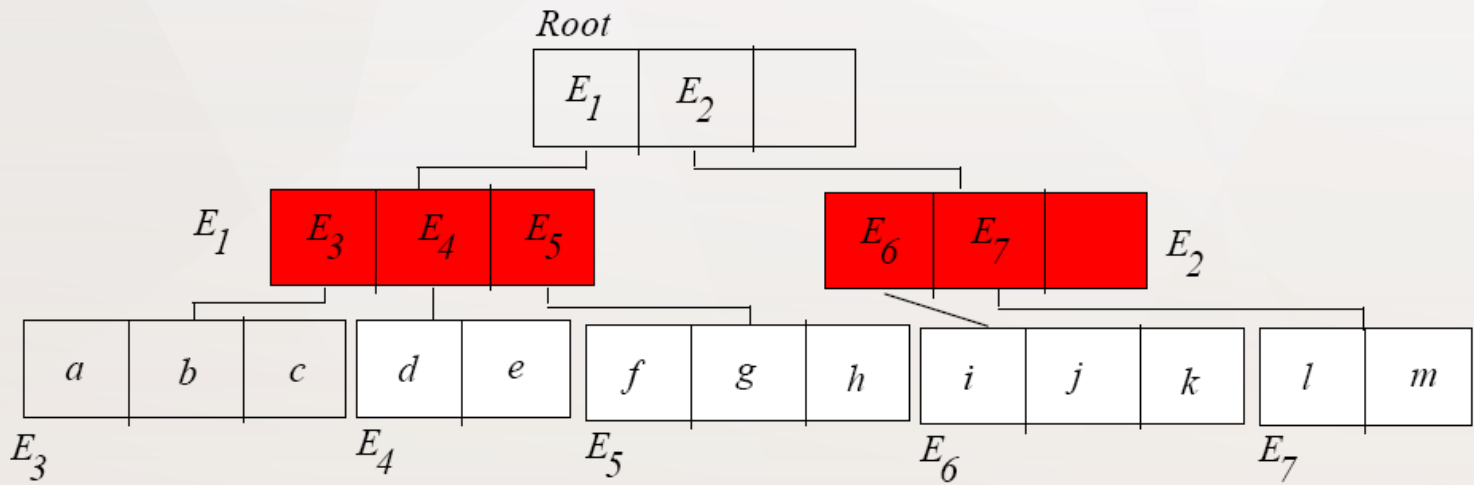
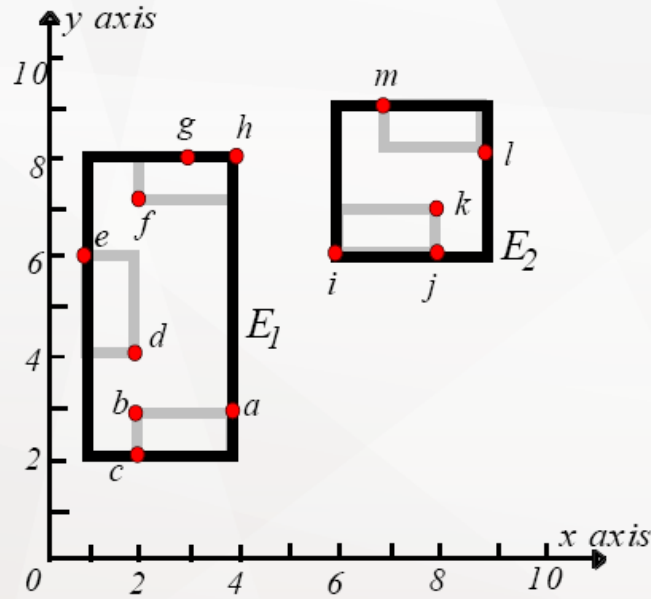
## R-Tree





# R-Tree: Clustering by Proximity...

## R-Tree



# Sampai Jumpa Lagi...

