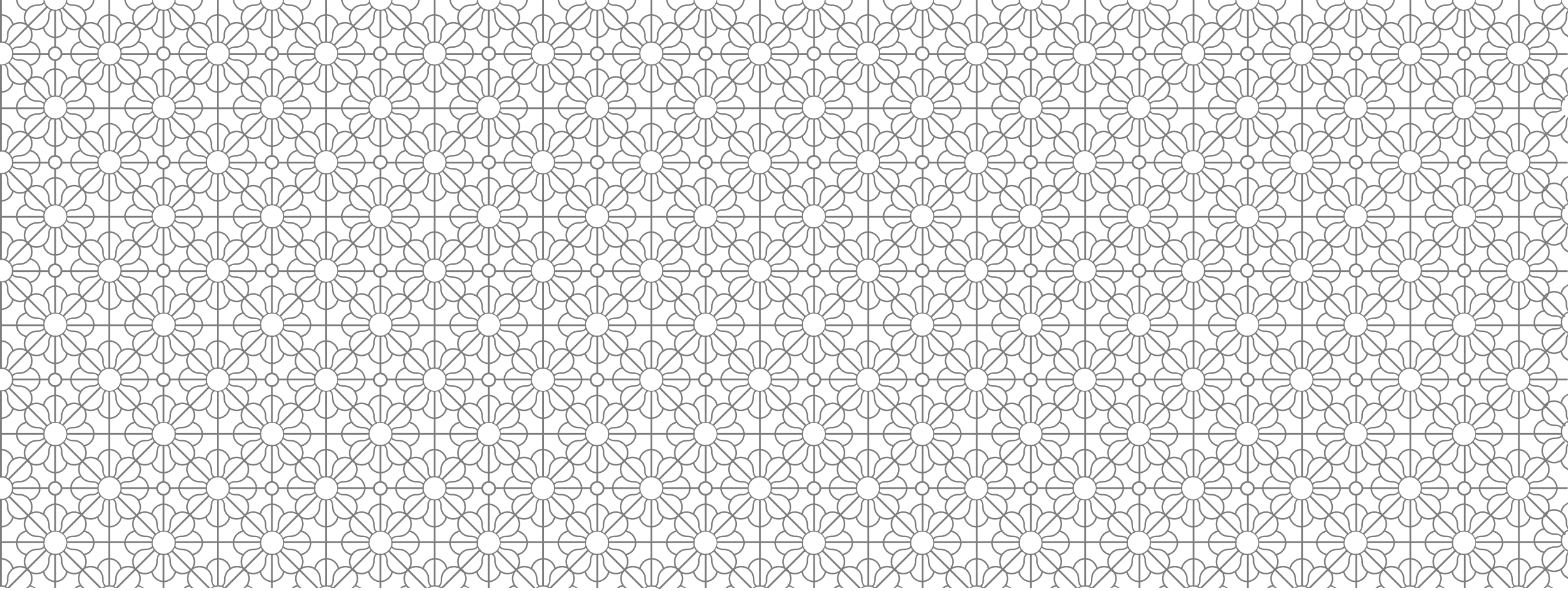


BASIS DATA

INTEGRITAS BASIS DATA

BEBERAPA KEKANGAN DALAM BASIS DATA

Terdapat beberapa kekangan yang harus dipatuhi pada file basis data agar dapat memenuhi kriteria sebagai suatu basis data. Beberapa kekangan itu berhubungan dengan masalah kerangkapan data, inkonsistensi data, data terisolasi, keamanan data, dan integritas data.

DATA REDUDANCY

Data Redudancy, yaitu munculnya data-data yang sama secara berulang-ulang pada file basis data yang semestinya tidak diperlukan.

DATA INCONSISTENCY

Data Inconsistency, yaitu munculnya data yang tidak konsisten pada medan yang sama untuk beberapa file dengan kunci yang sama. Ketidak-konsistenan data biasanya terjadi akibat kesalahan dalam pemasukan data (data entry) atau update anomaly, yaitu suatu proses untuk meng-update data, tetapi mengakibatkan munculnya data yang tidak konsisten atau kehilangan informasi tentang objek yang ditinjau

DATA INTEGRITY (INTEGRITAS DATA)

Informasi yang disimpan pada basis data bisa valid jika DBMS membantu mencegah pemasukan informasi yang tidak benar. Jika basis data memenuhi semua konstrain integritas yang dispesifikasikan pada skema basis data maka basis data adalah valid.

DBMS memaksakan konstrain integritas sehingga hanya mengijinkan basis data valid yang akan disimpan oleh DBMS.

Integritas Data adalah merupakan keutuhan dan kesatuan data dalam basis data sehingga data tersebut dapat menjadi sumber informasi yang dapat digunakan.

Batasan integritas merupakan aturan yang diberikan pada suatu tabel agar data yang dimasukkan terjamin validitasnya. Batasan integritas akan menjaga basis data dari kerusakan yang terjadi secara tidak sengaja dengan memastikan bahwa perubahan yang diperbolehkan tidak mengakibatkan terjadinya inkonsistensi data

JENIS BATASAN INTEGRITAS

Aturan integritas domain.

Domain adalah nilai – nilai yang dimungkinkan untuk setiap atribut. Setiap atribut harus mempunyai domain yang sama.

Aturan integritas entitas.

Misalkan, dalam penerapan model basis data relasional, tidak ada suatu atribut primary key yang memiliki nilai NULL, artinya bahwa setiap atribut primary key harus memiliki nilai tertentu.

Aturan integritas referensial.

Jika foreign key terdapat di relasi maka nilai foreign key harus cocok pada nilai primary key suatu tupel di relasi asal (home relation).

Aturan integritas yang didefiniskan pemakai.

adalah aturan – aturan tambahan yang dispesifikan pemakai atau administrator basis data

INTEGRITAS REFERENSIAL

Integritas referensial merupakan cara untuk menjaga agar Foreign Key suatu tabel dan Primary Key milik tabel yang direferensi, selalu konsisten.

Tujuan integritas referensial untuk menjamin dan memastikan agar entitas dalam suatu tabel yang menunjuk ke suatu pengenal unik pada suatu baris di tabel lain benar-benar menunjuk pada nilai yang memang ada.

Berdasarkan operasi yang dilakukan, jenis integritas referensial :

- penambahan (insert)
- penghapusan (delete)
- peremajaan (update)

Integritas referensial dapat dilaksanakan pada tabel yang memiliki relasi. Sehingga proses penghapusan / peremajaan suatu kolom juga akan terjadi pada kolom tabel lain yang mempunyai referensi dengannya

CONTOH

Membuat tabel mhs

CREATE TABLE mhs (nim varchar(8), namaMhs varchar(20), PRIMARY KEY (nim));

Membuat tabel mk

CREATE TABLE mk (kodeMK varchar(3), namaMK varchar(20), PRIMARY KEY (kodeMK));

Membuat tabel ambilMK

CREATE TABLE ambilMK (nim varchar(8), kodeMK varchar(3), nilai float(3,2), PRIMARY KEY (nim, kodeMK), FOREIGN KEY (nim) REFERENCES mhs (nim) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (kodeMK) REFERENCES mk (kodeMK) ON DELETE CASCADE ON UPDATE CASCADE);

Tindakan-tindakan yang dapat diatur oleh pemakai ini biasa disebut tindakan referensial.

[ON UPDATE {RESTRICT | CASCADE | SET NULL}]

[ON DELETE {RESTRICT | CASCADE | SET NULL}]

Keterangan :

UPDATE : menyatakan tindakan kalau pada tabel induk terjadi perubahan nilai.

DELETE : menyatakan tindakan kalau pada tabel induk terjadi penghapusan baris.

Adapun tindakan yang dapat didefinisikan pada tabel ON UPDATE maupun ON DELETE berserta penjelasannya sebagai berikut :

RESTRICT : menyatakan bahwa pengubahan atau penghapusan ditolak.

CASCADE : jika nilai kunci primer pada tabel induk berubah (UPDATE) maka kunci asing pada tabel yang mereferensi akan disesuaikan dengan nilai pada kunci primer tabel induk, sedangkan apabila terjadi proses DELETE semua kunci asing yang cocok dengan kunci primer pada tabel induk milik record yang dihapus akan ikut dihapus.

SET NULL : Menyatakan kunci asing akan diisi dengan NULL kalau kunci primer pada tabel induk yang nilainya sama dengan nilai pada kunci asing tersebut diubah atau dihapus.

DATA REDUNDANCY (REDUNDANSI DATA)

Adalah penyimpanan data yang sama secara berulang-ulang, baik dalam satu tabel / file maupun antar tabel / file yang berbeda. Redundansi menyebabkan terjadinya pemborosan tempat penyimpanan maupun inkonsistensi data.

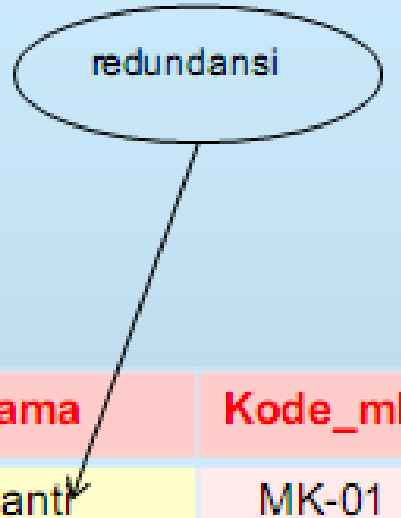
Contoh Bentuk Redundansi 1

IDSales	NamaSales	Telepon
ADN006	Yeni, SE	3517261
ADN006	Yeni, SE	3520165
ADN007	Memey	4744621
ADN007	Memey	08122861427
ADN008	Tina	08566241521
ADN009	Ir. Yanto	7265122
ADN009	Ir. Yanto	7123910
ADN010	Made	6723192

Contoh Bentuk Redundansi 2

mhs

NIM	Nama	Alamat	Tgl_lhr
A11.01	Santi	.....	.....
A11.02	Rudi	.....	.....
A11.03	Sandra	.....	.....
A11.04	Budi	.....	.....
A11.05	Santosa	.....	.....



krs

NIM	Nama	Kode_mk	Nilai
A11.01	Santi	MK-01	A
A11.01	Santi	MK-02	B
A11.02	Rudi	MK-01	A
A11.03	Sandra	MK-01	A
A11.03	Sandra	MK-02	B
A11.03	Sandra	MK-03	B

STRUCTURAL CONSTRAINTS

Dalam sebuah Relationship pada Basis Data terdapat batasan yang terstruktur (Structural Constraints). Tipe utama dari batasan disebut multiplicity yang mencerminkan aturan dari sistem yang akan dibuat oleh user. Multiplicity dibuat berdasarkan dua batasan yaitu Cardinality dan Participation.

Cardinality

Adalah nilai maximum occurrence dari sebuah Relationship antara dua entitas; contohnya: antara entitas Dosen dan Mata Kuliah terdapat Relationship "Mengajar" dengan multiplicity"0..5", artinya satu dosen boleh mengajar maximal 5 mata kuliah sedangkan sebuah mata kuliah bisa jadi belum memiliki dosen pengajarnya. Cardinality = 5 dan Participation = 0.

Participation

Adalah nilai minimum occurrence dari sebuah Relationship antara dua entitas; contohnya antara entitas Gedung dan Ruang Kelas terdapat Relationship "Terdiri Dari" dengan multiplicity "1..5", artinya satu Gedung bisa terdapat maximal 5 ruang kelas tapi satu ruang kelas hanya terdapat pada satu gedung. Cardinality = 5 dan Participation = 1

PERBEDAAN TOTAL CONSTRAINT DAN PARTIAL CONSTRAINT

Total Constraint : constraint dimana data dalam entitas yang memiliki constraint tersebut terhubung secara penuh ke dalam entitas dari relasinya

Partial Constraint : constraint dimana data dalam entitas yang memiliki constraint tersebut terhubung ke dalam entitas dari relasinya.

Contoh 1:

- Entity 1 to entity 2 : kardinalitas : *one to many* dengan detail minimal 0 dan maksimalnya banyak. Dependensi : entitas 1 dan entitas 2 tidak saling ketergantungan.
- Entity 2 to entity 1 : kardinalitas : *many to one* dengan detail minimal 1 dan maksimalnya 1. Dependensi : entitas 1 dan entitas 2 tidak saling ketergantungan.

Contoh 2:

- Entity 3 to entity 4 : kardinalitas : *one to many* dengan detail minimal 0 maksimalnya banyak. Dependensi : entitas 3 dan entitas 4 tidak saling ketergantungan.
- Entity 4 ke entity 3 : kardinalitas : *many to one* dengan detail minimal 1 maksimal 1. Dependensi : entitas 4 dan entitas 3 tidak saling ketergantungan.

Contoh 3:

Entity 5 to entity 6 dan Entity 6 to entity 5 : kardinalitas : *one to one* dengan detail minimal 0 maksimalnya 1. Dependensi : entitas 5 dan entitas 6 tidak saling ketergantungan.

Contoh 4:

Entity 7 to entity 8 : kardinalitas : *one to one* dengan detail minimal 0 maksimalnya 1. Dependensi : entitas 7 dan entitas 8 tidak saling ketergantungan.

Entity 8 ke entity 7 : kardinalitas : *one to one* dengan detail minimal 1 maksimal 1. Dependensi : entitas 8 dan entitas 7 tidak saling ketergantungan.

Contoh 5:

Entity 9 to entity 10 : kardinalitas : *many to many* dengan detail minimal 0 maksimalnya banyak. Dependensi : entitas 9 dan entitas 10 tidak saling ketergantungan.

Entity 10 ke entity 9 : kardinalitas : *many to many* dengan detail minimal 1 maksimal banyak. Dependensi : entitas 10 dan entitas 9 tidak saling ketergantungan.

CONTOH 6:

Entity 13 to entity 14 : kardinalitas : *one to many* dengan detail minimal 0 maksimalnya banyak. Dependensi : entitas 13 menjadi parent dari entitas 14.

Entity14 ke entity 13 : kardinalitas : *many to one* dengan detail minimal 1 maksimal 1. Dependensi : entitas 14 tergantung kepada entitas 13.

DEPENDENSI

Dependensi Fungsional

Dependensi Fungsional Penuh

Dependensi Fungsional Parsial

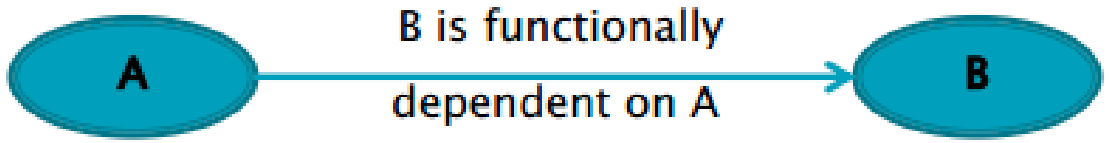
Dependensi Fungsional Transitif

DEPENDENSI FUNGSIONAL

Dependensi fungsional menggambarkan relasi / hubungan, batasan, dan keterkaitan antara atribut-atribut dalam suatu relasi. Suatu atribut dikatakan bergantung pada atribut lain secara fungsional jika kita menggunakan harga atribut yang lain.

Notasi dependensi fungsional yaitu A => B (A secara fungsional menentukan B)

Artinya bahwa A secara fungsional menentukan B **atau** B bergantung pada A. Jika ada dua baris data dengan nilai A yang sama, maka nilai B juga sama.



Tabel Nilai

MataKuliah	NIM	NamaMhs	NilaiHuruf
Struktur Data	980001	Ali Akbar	A
Struktur Data	980004	Indah Susanti	B
Basis Data	980001	Ali Akbar	
Basis Data	980002	Budi Haryanto	
Basis Data	980004	Indah Susanti	
Bahasa Indonesia	980001	Ali Akbar	B
Bahasa Indonesia	980003	Ali Akbar	B
Matematika 1	980002	Budi Haryanto	C
Matematika 1	980003	Ali Akbar	A

Dependensi yang ada:

NIM => NamaMhs

{MataKuliah,NIM} => NilaiHuruf

DEPENDENSI PENUH

B memiliki dependensi fungsional secara penuh pada A

B bukan memiliki dependensi terhadap subset A

Contoh:

NIM	Nama	IPK	idRuang	Dosen
A11.2078	Paijo	3.20	301	Agus C
A11.2906	Tukijo	2.02	602	Joko P
A11.3456	Parjo	3.09	302	Rima U
A11.3254	Ngatini	3.40	301	Santo M
A11.3098	Tumini	2.75	602	Setyo P

Dependensi Penuh: NIM => idRuang

DEPENDENSI PARSIAL

Dependensi Parsial merupakan ketergantungan fungsional dimana beberapa atribut dapat dihilangkan dari A dengan ketergantungan tetap dipertahankan.

B memiliki dependensi terhadap subset A

Contoh:

NIM	Nama	IPK	idRuang	Dosen
A11.2078	Paijo	3.20	301	Agus C
A11.2906	Tukijo	2.02	602	Joko P
A11.3456	Parjo	3.09	302	Rima U
A11.3254	Ngatini	3.40	301	Santo M
A11.3098	Tumini	2.75	602	Setyo P

Dependensi Parsial: {NIM, Nama} => idRuang

DEPENDENSI TRANSITIF

Transitif Dependensi adalah kondisi dimana A, B, C merupakan atribut sebuah relasi dimana A=>B dan B=>C

Maka C dikatakan dependensi transitif terhadap A melalui B

Contoh:

NIP	Nama	Jabatan	Gaji	kdCabang	almCabang
10.002.1	Paijo	Quality Control	1500	01	Semarang
11.089.2	Tukijo	Supervisor	1250	02	Kendal
11.098.1	Parjo	Quality Control	1500	01	Semarang
12.101.3	Ngatini	Cleaning Service	750	03	Tengaran
12.110.1	Tumini	Quality Control	1500	01	Semarang

NIP => {Nama, Jabatan, Gaji, kdCabang, almCabang}

kdCabang => almCabang