

BAB V MODEL DATA RELASIONAL

A. Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menjelaskan konsep basis data, dan merancang basis data mulai dari tahap pemodelan data dengan model E-R sampai model relasional, dimana hasil rancangan memenuhi kaedah normalisasi.

B. Sub-Capaian Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu menjelaskan konsep model basis data relasional

C. Indikator – Indikator Pembelajaran

Mahasiswa mampu:

1. Menjelaskan konsep model relasional
2. Menjelaskan Batasan Model Relasional

D. Materi

1. Pendahuluan

Pada bab ini dibahas mengenai basis data relasional. Model data relasional pertama kali diperkenalkan oleh Ted Codd dari IBM Research pada tahun 1970 dalam sebuah makalah klasik, dan itu menarik perhatian karena kesederhanaan dan dasar matematikanya. Model tersebut menggunakan konsep relasi matematis yang terlihat seperti tabel nilai, dan memiliki landasan teoretisnya seperti teori himpunan dan logika predikat orde pertama. Dalam bab ini juga dibahas karakteristik dasar model dan batasannya.

Implementasi pertama secara komersial dari model relasional ini dilakukan pada awal 1980-an, seperti sistem SQL/DS pada sistem operasi MVS oleh IBM dan Oracle DBMS. Sejak itu, model tersebut telah diimplementasikan pada sejumlah besar sistem komersial, serta sejumlah sistem *open source*. DBMS relasional komersial populer saat ini seperti Oracle (dari Oracle), Sybase DBMS (sekarang dari SAP), dan SQLServer dan Microsoft Access (dari Microsoft). Selain itu, beberapa sistem open source, seperti MySQL dan PostgreSQL, juga tersedia.

2. Konsep Model Relasional

Model relasional mewakili basis data sebagai kumpulan relasi. Secara informal, setiap relasi menyerupai tabel nilai atau, sampai batas tertentu, tabel dua dimensi yang terdiri atas record/baris data dan kolom data. Sebagai contoh seperti yang ditampilkan pada Gambar 5.1. Misalnya, tabel basis data yang ditunjukkan pada Gambar 5.1 merupakan representasi model relasional dasar. Namun, ada perbedaan penting antara relasi dan tabel, seperti yang akan dibahas selanjutnya.

MAHASISWA

NIM	Nama	Jenis Kelamin	Tgl_Lahir	IPK
2215101005	Ketut Yogi	Laki-laki	25-08-2004	3,58
2115101052	Putu Ayu Lestari	Perempuan	21-07-2003	2,90
2115011015	Santi Budaya	Laki-laki	09-11-2002	3,88

JURUSAN

NIM	NamaJur	Lokasi	KetuaJur
2215101005	Teknik Informatika	Singaraja	Luh Ayu Dewi
2115101052	Teknik Informatika	Singaraja	Luh Ayu Dewi
2115011015	Teknologi Industri	Singaraja	Kadek Raiswara

Gambar 5.1 Representasi Model Relasional

Ketika relasi dianggap sebagai sebuah tabel, setiap baris dalam tabel mewakili kumpulan nilai data terkait. Baris mewakili fakta yang biasanya sesuai dengan entitas atau hubungan dunia nyata. Nama tabel dan nama kolom digunakan untuk membantu menginterpretasikan arti dari nilai di setiap baris. Misalnya, tabel pertama pada Gambar 5.1 disebut MAHASISWA karena setiap baris mewakili fakta tentang entitas mahasiswa tertentu. Nama kolom, yaitu NIM, Nama, Jenis_kelamin, Tgl_Lahir, dan IPK menunjukkan cara menginterpretasikan nilai data di setiap baris, berdasarkan kolom setiap nilai berada. Semua nilai dalam kolom memiliki tipe data yang sama.

Dalam terminologi model relasional formal, baris disebut tupel, judul kolom disebut atribut, dan tabel disebut relasi. Tipe data yang menjelaskan tipe nilai yang dapat muncul di setiap kolom diwakili oleh domain nilai yang mungkin. Selanjutnya istilah-istilah (tupel, atribut, dan relasi) tersebut akan dibahas pada bagian berikut.

2.1 Domain, Atribut, Tupel dan Relasi

Sebuah domain D adalah sekumpulan nilai yang atomik/unik. Dengan sifat atomik tersebut, berarti bahwa setiap nilai dalam domain tidak dapat dibagi menjadi domain yang lebih sederhana. Metode umum untuk menentukan domain adalah dengan menentukan tipe data yang membentuk domain tersebut. Hal ini juga berguna untuk menentukan nama domain, untuk membantu dalam menafsirkan nilainya. Berikut ini adalah beberapa contoh dari domain, yaitu:

- Nomor_Induk_Mhs. Kumpulan nomor induk mahasiswa sepanjang 10 digit yang berlaku di sebuah universitas di Indonesia.
- Nama_Mahasiswa. Kumpulan string karakter yang mewakili nama seorang mahasiswa.
- Nomor_Hp. Kumpulan nomor sepanjang 12 digit yang mendeskripsikan sebagai nomor handphone yang berlaku di Indonesia.
- Nomor_Telp_Seluler. Himpunan 12 digit nomor telepon yang valid, dimana empat digit di awal mendeskripsikan sebagai kode area. Himpunan ini merupakan nomor telepon seluler yang berlaku di Indonesia.
- Alamat_Rumah. Sekumpulan string yang mendeskripsikan sebagai alamat rumah seseorang.
- Usia_Mahasiswa. Kemungkinan usia mahasiswa di sebuah universitas; masing-masing harus berupa nilai integer antara 18 dan 30.
- Tanggal_Lahir. Kumpulan angka yang mendeskripsikan sebagai tanggal lahir seseorang.
- Indeks_Prestasi_Kumulatif. Deretan angka dalam bentuk pecahan desimal antara 0 sampai 4,00 yang mendeskripsikan indeks prestasi kumulatif
- Nama_MataKuliah. Kumpulan nama mata kuliah di suatu jurusan di universitas tertentu, seperti contoh: “Matematika Dasar”, “Perancangan Antarmuka Web”, dan “Algoritma Pemrograman”.
- Nama_Jurusan. Himpunan nama jurusan akademik di suatu universitas, seperti contoh: Ilmu Komputer, Bahasa Indonesia, Ekonomi, dan Fisika.

Contoh di atas disebut sebagai definisi logis dari domain. **Tipe data** atau **format data** juga ditentukan untuk setiap domain. Misalnya, tipe data untuk domain NIM dapat dideklarasikan sebagai string karakter dalam bentuk *dddddddddd*, di mana setiap *d* adalah

digit numerik (desimal). Dua digit pertama menyatakan tahun angkatan, satu digit berikutnya menyatakan kode strata, satu digit berikutnya menyatakan kode fakultas, dua digit berikutnya menyatakan nomor urut kelahiran program studi pada sebuah fakultas, satu digit selanjutnya menyatakan semester ke- dimana mahasiswa memulai kuliah di universitas bersangkutan, dan tiga digit terakhir menyatakan nomor urut.

Tipe data untuk `Usia_Mahasiswa` adalah bilangan bulat antara 18 dan 30. Untuk `Tanggal_Lahir` menggunakan tipe data *Date*. `Indeks_Prestasi_Kumulatif` menggunakan tipe data pecahan desimal/riil antara 0 sampai dengan 4,00. Untuk `Nama_MataKuliah`, tipe datanya adalah kumpulan semua karakter string yang mewakili nama mata kuliah yang valid. Dan `Nama_Jurusan`, tipe datanya adalah kumpulan semua karakter string yang mewakili nama jurusan yang valid. Sebuah domain dengan demikian diberi nama, tipe data, dan format. Informasi tambahan untuk menafsirkan nilai-nilai domain juga dapat diberikan; misalnya, domain numerik seperti `Berat_Badan` harus memiliki unit pengukuran, seperti pound atau kilogram.

Skema relasi R , dilambangkan dengan $R(A_1, A_2, A_3, \dots, A_n)$, terdiri dari nama relasi R dan daftar atribut, $A_1, A_2, A_3, \dots, A_n$. Setiap **atribut** A_i adalah nama peran yang dimainkan oleh beberapa domain D dalam skema relasi R . D disebut **domain** dari A_i dan dilambangkan dengan $dom(A_i)$. Skema relasi digunakan untuk menggambarkan suatu relasi; R disebut nama relasi. Derajat (atau aritas) suatu relasi adalah jumlah n buah atribut dari skema relasinya.

Relasi derajat tujuh artinya ada tujuh atribut yang menggambarkan setiap mahasiswa, dimana ketujuh atribut tersebut menyimpan informasi tentang mahasiswa.

Contoh: MAHASISWA (Nama, NIM, No_Hp, Alamat, Telp_Kantor, Usia, IPK)

Untuk menggambarkan penggunaan tipe data dari setiap atribut, definisi terkadang ditulis sebagai:

MAHASISWA (Nama: string, NIM: string, No_Hp: string, Alamat: string,
Telp_Kantor: string, Usia: integer, IPK: float)

Untuk skema relasi ini, MAHASISWA adalah nama relasi yang memiliki tujuh atribut. Dalam definisi sebelumnya, telah ditunjukkan penggunaan tipe data generik seperti string atau integer pada atribut. Lebih tepatnya, domain dari setiap atribut untuk relasi MAHASISWA dapat ditentukan sebagai berikut:

$dom(Nama) = Nama_Mahasiswa;$
 $dom(NIM) = Nomor_Induk_Mhs;$
 $dom(No_Hp) = Nomor_Hp;$
 $dom(Alamat) = Alamat_Rumah;$
 $dom(Telp_Kantor) = Nomor_Telp_Seluler;$
 $dom(Usia) = Usia_Mahasiswa;$
 $dom(IPK) = Indeks_Prestasi_Kumulatif.$

Hal ini juga memungkinkan untuk merujuk pada atribut skema relasi berdasarkan posisinya dalam relasi. Jadi, atribut kedua dari relasi MAHASISWA adalah NIM, sedangkan atribut kelima adalah Telp_Kantor.

Gambar 5.2 menunjukkan contoh relasi MAHASISWA. Setiap tupel dalam relasi tersebut mewakili entitas mahasiswa tertentu. Dalam hal ini, relasi ditampilkan sebagai tabel, di mana setiap tupel ditampilkan sebagai baris data dan setiap atribut sesuai dengan judul kolom yang menunjukkan peran atau interpretasi nilai dalam kolom tersebut. Nilai NULL mewakili atribut yang nilainya tidak diketahui atau tidak ada untuk beberapa tupel MAHASISWA secara individual.

MAHASISWA						
Nama	NIM	No Hp	Alamat	Telp Kantor	Usia	IPK
Ketut Ayu	2115101007	087766669999	Jl. Cemara No. 2 Singaraja	NULL	18	3,57
Putu Pratama	2015101077	NULL	Jl. Kayumas No.8 Denpasar	082299666677	19	3,66
Kadek Diah	2015101081	085255552222	Jl. Kencana No. 5 Gianyar	081355660099	20	3,90
Gede Ageng	1915101001	082223234455	Jl. Pahlawan No. 2Y Klungkung	NULL	20	2,99

Gambar 5.2 Atribut dan Tupel dari Relasi MAHASISWA

Dalam hal ini, ada kemungkinan juga beberapa atribut memiliki domain yang sama. Misalnya, dalam relasi MAHASISWA, domain yang sama yaitu Nomor_Hp memainkan peran No_Hp mengacu pada telepon rumah seorang mahasiswa, dan peran Nomor_Telp_Seluler mengacu pada telepon kantor mahasiswa.

2.2 Karakteristik Relasi

Definisi relasi sebelumnya menyiratkan karakteristik tertentu yang membuat relasi berbeda daripada tabel. Selanjutnya, akan dibahas beberapa karakteristik yang dimaksud ini.

Pengurutan Tupel dalam Relasi. Sebuah relasi didefinisikan sebagai sekumpulan tupel. Tupel dalam suatu relasi tidak memiliki urutan tertentu. Dengan kata lain, suatu relasi tidak sensitif terhadap urutan tupel. Dapat dikatakan bahwa, pengurutan tupel bukan bagian dari definisi relasi karena relasi mencoba untuk merepresentasikan fakta pada tingkatan logis atau abstrak. Banyak urutan tupel dapat ditentukan pada relasi yang sama. Sebagai contoh, tupel dalam relasi MAHASISWA pada Gambar 5.2 dapat diurutkan berdasarkan nilai Nama, NIM, Usia, IPK, atau atribut lainnya.

Oleh karena itu, relasi yang ditampilkan pada Gambar 5.3 dianggap identik seperti yang ditunjukkan pada Gambar 5.2. Ketika suatu relasi diimplementasikan sebagai file atau tabel, urutan tertentu dapat ditentukan pada *record* file atau baris tabel.

Nama	NIM	No_Hp	Alamat	Telp_Kantor	Usia	IPK
Gede Ageng	1915101001	082223234455	Jl. Pahlawan No. 2Y Klungkung	NULL	20	2,99
Kadek Diah	2015101081	085255552222	Jl. Kencana No. 5 Gianyar	081355660099	20	3,9
Ketut Ayu	2115101007	087766669999	Jl. Cemara No. 2 Singaraja	NULL	18	3,57
Putu Pratama	2015101077	NULL	Jl. Kayumas No.8 Denpasar	082299666677	19	3,66

Gambar 5.3 Relasi MAHASISWA dari Gambar 5.2 dengan urutan tupel yang berbeda

Pengurutan Nilai dalam Tupel dan Definisi Alternatif dari Relasi. Menurut definisi relasi sebelumnya, n -tupel adalah daftar dengan n buah nilai yang terurut, sehingga pengurutan nilai dalam sebuah tupel adalah penting. Namun, pada tingkat yang lebih abstrak, urutan atribut dan nilainya tidak begitu penting sepanjang korespondensi antara atribut dan nilai tetap terjaga.

Nilai dan NULL di Tupel. Setiap nilai dalam tupel adalah nilai yang atomik/unik; yaitu, tidak dapat dibagi menjadi komponen-komponen yang lebih sederhana. Oleh karena itu, atribut komposit dan atribut bernilai banyak tidak diperbolehkan. Model ini kadang-kadang disebut model relasional datar (*flat relational model*). Banyak teori di balik model relasional dikembangkan dengan asumsi ini dalam pikiran, yang disebut asumsi **bentuk normal pertama**. Oleh karena itu, atribut bernilai banyak harus diwakili oleh hubungan

yang terpisah, dan atribut komposit diwakili hanya oleh atribut komponen sederhana mereka dalam model relasional dasar.

Konsep penting adalah nilai NULL, yang digunakan untuk mewakili nilai atribut yang mungkin tidak diketahui atau mungkin tidak berlaku untuk tupel. Nilai khusus, yang disebut NULL, digunakan dalam kasus ini. Misalnya, pada Gambar 5.2, beberapa tupel MAHASISWA memiliki NULL untuk telepon kantor mereka karena mereka tidak memiliki kantor (yaitu, telepon kantor tidak berlaku untuk mahasiswa ini). Mahasiswa lain memiliki NULL untuk nomor handphone/seluler, mungkin karena mahasiswa tersebut tidak memiliki handphone atau dia memilikinya tetapi tidak diketahui (nilainya tidak diketahui). Secara umum, nilai NULL memiliki beberapa arti, seperti nilai tidak diketahui, nilai ada tetapi tidak tersedia, atau atribut tidak berlaku untuk tupel ini (juga dikenal sebagai nilai tidak terdefinisi).

3. Batasan Model Relasional dan Skema Basis Data Relasional

Sejauh ini, telah dibahas karakteristik relasi tunggal. Dalam basis data relasional, biasanya akan ada banyak relasi, dan tupel dalam relasi tersebut biasanya terhubung dengan berbagai cara. Keadaan seluruh basis data akan sesuai dengan keadaan semua hubungannya pada kondisi tertentu. Umumnya ada banyak batasan atau kendala pada nilai aktual dalam basis data. Batasan ini diturunkan dari aturan pada *miniworld* yang diwakili oleh basis data.

Pada bagian ini, akan dibahas berbagai batasan pada data yang dapat dispesifikasikan pada basis data relasional dalam bentuk batasan. Batasan pada basis data secara umum dapat dibagi menjadi tiga kategori utama, yaitu:

- 1) Batasan yang melekat pada model data. Batasan ini disebut sebagai batasan berbasis model yang melekat (*inherent model-based constraint*) atau batasan implisit.
- 2) Batasan yang dapat diekspresikan secara langsung dalam skema model data, biasanya dengan menetakannya dalam *Data Definition Language* (DDL). Batasan ini disebut sebagai batasan berbasis skema (*schema-based constraints*) atau batasan eksplisit.
- 3) Batasan yang tidak dapat diekspresikan secara langsung dalam skema model data, dan karenanya harus diekspresikan oleh program aplikasi atau dengan cara lain. Batasan seperti ini disebut sebagai batasan aplikasi (*application-based*) atau semantik atau aturan bisnis.

Karakteristik relasi yang telah diskusikan di Bagian 2.2 adalah batasan yang melekat pada model relasional dan termasuk pada kategori pertama. Misalnya, suatu relasi tidak dapat memiliki tupel duplikat, ini masuk dalam batasan yang melekat. Batasan yang akan dibahas di bagian ini adalah batasan pada kategori kedua, yaitu batasan yang dapat diekspresikan dalam skema model relasional melalui DDL. Batasan dalam kategori ketiga lebih umum, berhubungan dengan makna serta perilaku atribut, dan sulit untuk diungkapkan dalam model data, sehingga biasanya diperiksa dalam program aplikasi yang melakukan pembaharuan/*update* basis data. Dalam beberapa kasus, batasan ini dapat ditentukan sebagai pernyataan dalam *Structure Query Language* (SQL).

Kategori batasan penting lainnya adalah ketergantungan data (*data dependency*), yang mencakup ketergantungan fungsional dan ketergantungan bernilai banyak. Kedua jenis ketergantungan ini digunakan terutama untuk menguji "kebaikan/*goodness*" dari desain basis data relasional dan digunakan dalam proses normalisasi.

Batasan berbasis skema termasuk pada batasan domain (*domain constraint*), batasan kunci (*key constraint*), batasan pada nilai NULL, batasan integritas entitas (*entity integrity constraint*), dan batasan integritas referensial (*referential integrity constraint*).

3.1 Batasan Domain (*domain constraint*)

Batasan domain menentukan bahwa dalam setiap tupel, nilai setiap atribut A harus berupa nilai atomik dari domain $dom(A)$. Tipe data yang terkait dengan domain biasanya mencakup tipe data numerik (*integer*) untuk bilangan bulat (seperti *short integer*, bilangan bulat/*integer*, dan *long integer*) dan bilangan real (*float*). Karakter (*Char*), Boolean, string dengan panjang tetap, dan string dengan panjang bervariasi juga tersedia, seperti halnya tanggal (*date*), waktu (*time*), dan tipe data khusus lainnya. Domain juga dapat dideskripsikan dengan nilai dalam selang (*subrange*) dari tipe data atau sebagai tipe data enumerasi di mana semua nilai yang mungkin dicantumkan secara eksplisit.

3.2 Batasan Kunci dan Batasan pada Nilai NULL

Dalam model relasional formal, sebuah relasi didefinisikan sebagai sekumpulan tupel. Semua tupel dalam suatu relasi harus berbeda. Ini berarti bahwa tidak ada dua tupel yang dapat memiliki kombinasi nilai yang sama untuk semua atributnya. Biasanya, ada **himpunan bagian dari atribut** (*subset of attribute*) lain dalam skema relasi R dengan properti bahwa tidak ada dua tupel dalam R yang memiliki kombinasi nilai yang sama untuk atribut-atributnya. Setiap relasi memiliki setidaknya satu *superkey* (kumpulan semua

atributnya). Namun, *superkey* dapat memiliki atribut yang berulang, jadi konsep yang lebih berguna adalah konsep kunci, yang tidak memiliki redundansi/perulangan.

Oleh karena itu, kunci memenuhi dua properti berikut:

- 1) Dua tupel berbeda dalam relasi mana pun tidak dapat memiliki nilai identik untuk (semua) atribut dalam kunci (*key*). Properti keunikan ini juga berlaku untuk *superkey*.
- 2) Ini adalah *superkey* minimal, yaitu *superkey* di mana kita tidak dapat menghapus atribut apa pun dan masih memiliki batasan keunikan. Properti minimalitas ini diperlukan untuk kunci tetapi opsional untuk *superkey*.

Oleh karena itu, kunci adalah *superkey* tetapi tidak sebaliknya. *Superkey* dapat berupa kunci (jika minimal) atau mungkin bukan kunci (jika tidak minimal). Perhatikan relasi MAHASISWA pada Gambar 5.2. Himpunan atribut {NIM} adalah kunci dari MAHASISWA karena tidak ada dua tupel mahasiswa yang dapat memiliki nilai NIM yang sama. Setiap himpunan atribut yang menyertakan NIM, misalnya {NIM, Nama, Usia} adalah *superkey*. Namun, *superkey* {NIM, Nama, Usia} bukanlah kunci dari MAHASISWA karena menghapus Nama atau Usia atau keduanya dari himpunan masih meninggalkan himpunan atribut tersebut dengan sebuah *superkey*, yaitu NIM. Secara umum, setiap *superkey* yang terbentuk dari atribut tunggal juga merupakan kunci. Kunci dengan banyak atribut harus mempertimbangkan, di mana semua atribut-atribut tersebut secara bersama-sama memiliki sifat yang unik.

Nilai atribut kunci dapat digunakan untuk mengidentifikasi secara unik setiap tupel dalam relasi. Misalnya, nilai NIM 1915101001 mengidentifikasi secara unik tupel yang berkorespondensi dengan Gede Ageng dalam relasi MAHASISWA. Perhatikan bahwa satu himpunan atribut yang merupakan kunci adalah properti dari skema relasi; itu adalah batasan yang harus dipegang pada setiap skema relasi yang valid. Sebuah kunci ditentukan dari arti atributnya, dan propertinya adalah waktu-invarian (*time-invariant*): Itu harus terus dikendalikan ketika tupel baru disisipkan dalam relasi. Sebagai contoh, atribut Nama pada relasi MAHASISWA pada Gambar 5.2 tidak boleh dipilih sebagai kunci karena ada kemungkinan bahwa ada dua mahasiswa memiliki nama yang sama.

Secara umum, skema relasi mungkin memiliki lebih dari satu kunci. Dalam hal ini, setiap kunci disebut sebagai kunci kandidat (*candidate key*). Sebagai contoh, relasi MOBIL pada Gambar 5.4 memiliki dua *candidate key*, yaitu No_Plat dan No_Mesin. Adalah umum untuk menunjuk salah satu *candidate key* sebagai *primary key* pada relasi. Ini adalah

candidate key yang nilainya digunakan untuk mengidentifikasi tupel dalam relasi. *Primary key* dari skema relasi diberi garisbawah, seperti yang ditunjukkan pada Gambar 5.4. Perhatikan bahwa ketika skema relasi memiliki beberapa *candidate key*, *primary key* harus dipilih *candidate key* yang ada. *Primary key* harus terdiri dari satu atribut atau sejumlah kecil atribut. *Candidate key* lainnya ditetapkan sebagai **kunci unik** dan tidak digarisbawahi. Pada Gambar 5.4, atribut No_Plat dipilih sebagai *primary key* untuk relasi MOBIL.

MOBIL

<u>No_Plat</u>	No_Mesin	Pembuat	Model	Tahun
D 4419 KF	JPM2E-2046858	Toyota	Kijang Super	1998
F 9090 IK	KPJ6E-2055581	Daihatsu	Zenia	2020
DK 8080 VF	LMP9F-9090782	Toyota	Agya	2009
DK 3311 MB	JKP8T-8076763	Suzuki	R Tiga	2019
B 7776 IY	TRW6G-4568231	Honda	HRV	2006

Gambar 5.4 Relasi MOBIL dengan *candidate key*: No_Plat dan No_Mesin

3.3 Basis Data Relasional dan Skema Basis data Relasional

Batasan yang telah dibahas sebelumnya berlaku untuk relasi tunggal dan atributnya. Basis data relasional biasanya berisi banyak relasi, dengan tupel dalam relasi yang terhubung dengan berbagai cara. Pada bagian ini, akan dibahas basis data relasional dan skema basis data relasional.

Pada Gambar 5.5 menunjukkan sebuah skema basis data relasional yang disebut PERUSAHAAN = {KARYAWAN, DEPARTEMEN, LOKASI_DEP, PROYEK, KERJA_DI, TANGGUNGAN}. Dalam setiap skema relasi, atribut yang digarisbawahi mewakili *primary key*. Gambar 5.6 menunjukkan status basis data relasional yang sesuai dengan skema PERUSAHAAN.

Ketika basis data relasional dirujuk, skema dan pernyataannya saat ini akan disertakan secara implisit. Pernyataan basis data yang tidak mematuhi semua batasan integritas disebut **tidak valid**, dan pernyataan yang memenuhi semua batasan dalam rangkaian batasan integritas yang ditentukan disebut **pernyataan valid**.

Pada Gambar 5.5, atribut NomorDep pada DEPARTEMEN dan LOKASI_DEP mewakili konsep dunia nyata yang sama, yaitu nomor yang diberikan ke departemen. Konsep yang sama ada pada NoDep di KARYAWAN dan No_Dep di PROYEK. Atribut

yang mewakili konsep dunia nyata yang sama mungkin memiliki nama yang sama atau berbeda dalam relasi yang berbeda. Atau, atribut yang mewakili konsep yang berbeda mungkin memiliki nama yang sama dalam relasi yang berbeda.

KARYAWAN

<u>ID</u>	NamaD	NamaT	NamaB	Tgl_Lhr	Gelar	Alamat	J_Kel	Hobi	ID_Aatasan	NoDep
-----------	-------	-------	-------	---------	-------	--------	-------	------	------------	-------

DEPARTEMEN

NamaDep	<u>NomorDep</u>	ID_Mgr	Tgl_Mulai_Mgr
---------	-----------------	--------	---------------

LOKASI_DEP

<u>NomorDep</u>	<u>LokasiDep</u>
-----------------	------------------

PROYEK

NamaPro	<u>NomorPro</u>	LokasiPro	No_Dep
---------	-----------------	-----------	--------

KERJA_DI

<u>ID_Kar</u>	<u>NoPro</u>	Jam
---------------	--------------	-----

TANGGUNGAN

<u>ID_Kar</u>	<u>Nama Tanggungan</u>	J_Kel	Tgl_Lhr	Jenis_Hubungan
---------------	------------------------	-------	---------	----------------

Gambar 5.5 Diagram Skema untuk Skema Basis Data Relasional PERUSAHAAN

Contoh:

Atribut yang menunjukkan sebagai nama dapat digunakan, misalnya NamaPro di PROYEK, dan NamaDep pada DEPARTEMEN. Dalam hal ini, ada dua atribut yang memiliki nama yang sama tetapi mewakili konsep dunia nyata yang berbeda, yaitu nama proyek dan nama departemen.

Dalam beberapa versi model relasional yang telah dibahas sebelumnya, asumsi dibuat bahwa konsep dunia nyata yang sama, ketika diwakili oleh atribut, akan memiliki nama atribut yang identik di semua relasi. Ini menciptakan masalah ketika konsep dunia nyata yang sama digunakan dalam peran (makna) yang berbeda dalam relasi yang sama. Misalnya, konsep Identitas (ID) muncul dua kali dalam relasi KARYAWAN Gambar 5.5, yaitu sekali dalam peran ID karyawan, dan sekali dalam peran ID atasan. Dalam hal ini, dicoba untuk memberi nama atribut yang berbeda, yaitu masing-masing ID dan ID_Aatasan.

Hal ini dilakukan karena atribut ID muncul dalam relasi yang sama dengan arti yang berbeda.

3.4 Integritas Entitas, Integritas Referensial, dan Kunci Asing (*Foreign Key*)

Batasan integritas entitas menyatakan bahwa tidak ada nilai *primary key* yang berupa NULL. Ini karena nilai *primary key* digunakan untuk mengidentifikasi tupel secara individu dalam suatu relasi. Memiliki nilai NULL untuk *primary key* menyiratkan bahwa beberapa tupel tidak dapat diidentifikasi. Misalnya, jika dua atau lebih tupel memiliki NULL untuk *primary key* mereka, maka tupel-tupel tersebut tidak dapat dibedakan ketika dihubungkan ke relasi lain.

Key constraint dan *entity integrity constraint* ditentukan pada relasi tunggal. *Referential integrity constraint* ditentukan antara dua relasi dan digunakan untuk menjaga konsistensi antara tupel dalam dua relasi tersebut. Secara informal, *referential integrity constraint* menyatakan bahwa tupel dalam satu relasi yang merujuk ke relasi lain harus merujuk ke tupel yang ada dalam relasi tersebut.

Contoh:

Misalnya, pada Gambar 5.5, atribut NoDep dari KARYAWAN adalah nomor departemen yang merupakan tempat dari setiap karyawan bekerja. Oleh karena itu, nilainya di setiap tupel KARYAWAN harus sesuai dengan nilai NomorDep dari beberapa tupel dalam relasi DEPARTEMEN.

Untuk menentukan batasan-batasan ini, pertama-tama pemahaman yang jelas tentang arti atau peran yang dimainkan setiap atribut atau kumpulan atribut dalam berbagai skema relasi basis data harus dimiliki. *Referential integrity constraint* biasanya muncul dari hubungan antara entitas yang diwakili oleh skema relasi.

Contoh:

Perhatikan basis data yang ditunjukkan pada Gambar 5.5. Dalam relasi KARYAWAN, atribut NoDep mengacu pada departemen tempat seorang karyawan bekerja; oleh karena itu, ditetapkan bahwa NoDep sebagai *foreign key* pada KARYAWAN yang merujuk pada relasi DEPARTEMEN. Ini berarti bahwa nilai NoDep dalam sembarang tupel pada relasi KARYAWAN harus sesuai dengan nilai *primary key* dari DEPARTEMEN, yaitu atribut NomorDep dalam beberapa tupel yang ada pada relasi DEPARTEMEN, atau nilai NoDep dapat berupa nilai NULL jika karyawan bukan merupakan anggota departemen bersangkutan.

Misalnya, pada Gambar 5.5 di mana tupel untuk karyawan 'Wayan Langgeng Sugih' merujuk tupel untuk departemen 'Penelitian', yang menunjukkan bahwa 'Wayan Langgeng Sugih' bekerja untuk departemen ini.

Perhatikan bahwa *foreign key* dapat merujuk ke relasinya sendiri. Misalnya, atribut ID_Aatasan pada KARYAWAN mengacu pada atasan dari seorang karyawan; ini adalah karyawan lain, diwakili oleh tupel dalam relasi KARYAWAN. Oleh karena itu, ID_Aatasan adalah *foreign key* yang merujuk ke relasi KARYAWAN itu sendiri. Pada Gambar 5.6, tupel untuk karyawan 'Wayan Langgeng Sugih' merujuk tupel untuk karyawan 'Kadek Bagus Genjing,' yang menunjukkan bahwa 'Kadek Bagus Genjing' adalah atasan dari 'Wayan Langgeng Sugih'.

Referential integrity constraint dapat ditampilkan dalam bentuk diagram yaitu dengan menggambar busur terarah dari setiap *foreign key* ke relasi yang dirujuknya. Untuk lebih jelasnya, panah dapat menunjuk ke *primary key* dari relasi yang dihubungkan. Gambar 5.7 menunjukkan skema dari Gambar 5.5 dimana *referential integrity constraint* ditampilkan dengan cara tersebut.

KARYAWAN

ID	NamaD	NamaT	NamaB	Tgl_Lhr	Gelar	Alamat	J_Kel	Hobi	ID_Aatasan	NoDep
123456789	Ketut	Ayu	Laksmi	1979-01-09	S. T	Jl. Pahlawan X Denpasar	Perempuan	Memasak	122557721	6
122557721	Wayan	Langgeng	Sugih	1975-12-08	S. T	Jl. Nuri 3 Klungkung	Laki-laki	Olah Raga	878787909	6
987654321	Gede	Ageng	Lemuh	1978-01-19	S. Kom	Jl. Sutomo 5X Singaraja	Laki-laki	Olah Raga	909090776	5
909090776	Royal	A	Simon	1981-06-20	S. Ikom	Jl. Nyuri XX Gianyar	Laki-laki	Mendaki	878787909	5
707080801	Luh Ayu	Sinar	Wati	1982-09-15	S. T	Jl. Cempaka 4Z Jakarta	Perempuan	Bernyanyi	122557721	6
777677768	Diah	Candra	Dewi	1978-07-31	S. ST	Jl. Sunari Denpasar	Perempuan	Chatting	NULL	6
555566664	Mustika	G	Ageng	1989-03-29	S. Par	Jl. Bima 7X Singaraja	Laki-laki	Volly	122557721	5
878787909	Kadek	Bagus	Genjing	1991-11-10	S. Kom	Jl. Arjuna No.9 Amplapura	Laki-laki	Basket	NULL	1

DEPARTEMEN

NamaDep	NomorDep	ID_Mgr	Tgl_Mulai_Mgr
Penelitian	2	122557721	2002-05-22
Administrasi	5	878787909	2012-01-01
Teknologi Informasi	6	909090776	2010-06-19

TANGGUNGAN

ID_Kar	Nama_Tanggungan	J_Kel	Tgl_Lhr	Jenis_Hubungan
122557721	Putu Jegeg	Perempuan	2004-04-05	Anak
122557721	Kadek Bagus	Laki-laki	2008-10-25	Anak
122557721	Komang Ayu	Laki-laki	1980-05-03	Istri
909090776	Jacklin	Perempuan	1990-08-19	Istri
123456789	Diah Rahayu	Perempuan	2005-04-25	Anak
123456789	Ayu Suksma	Perempuan	2007-12-21	Anak
123456789	Sinar Bintang	Perempuan	1991-11-29	Istri

LOKASI_DEP

NomorDep	LokasiDep
2	Jakarta Pusat
5	Jakarta Pusat
5	Denpasar
6	Surabaya

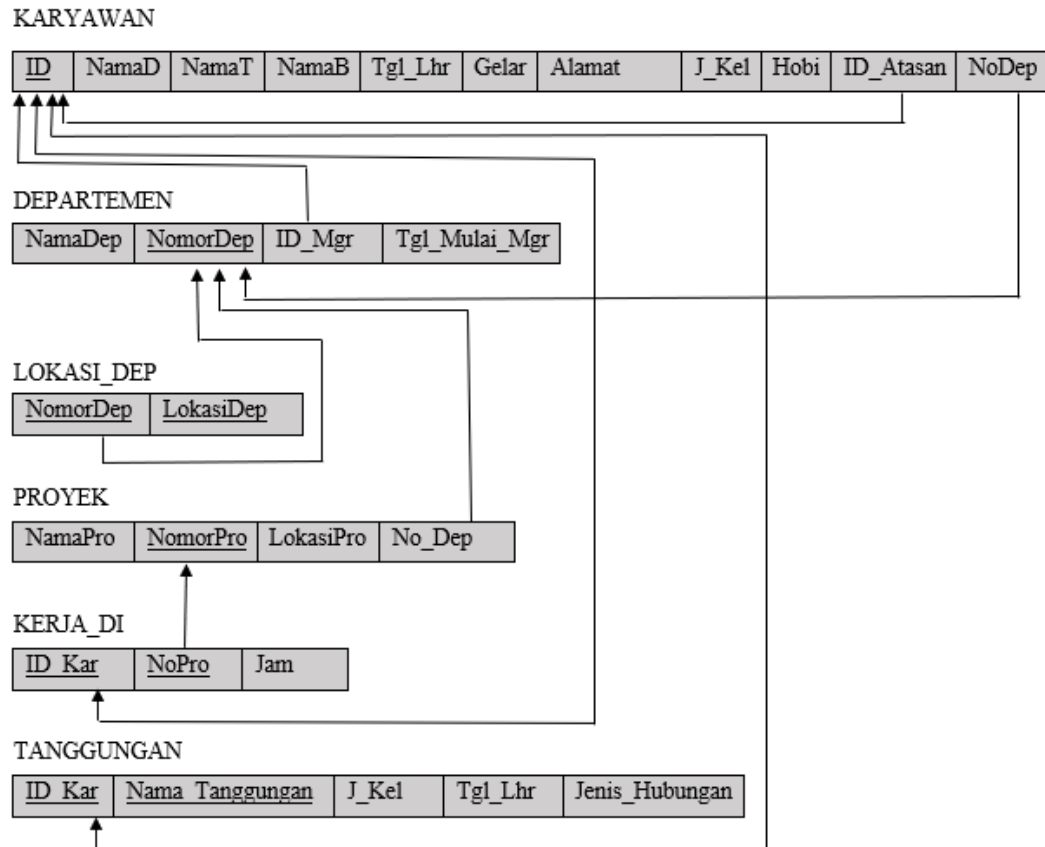
PROYEK

NamaPro	NomorPro	LokasiPro	No_Dep
Sistem XX	1	Surabaya	6
Sistem XY	5	Jakarta Pusat	6
Game ZZ	8	Denpasar	6
Sistem Payroll	9	Jakarta Pusat	6
Komputerisasi	12	Jakarta Pusat	5
Reorganisasi	20	Jakarta Pusat	2
Perkantoran	25	Denpasar	5

KERJA_DI

ID_Kar	NomorPro	Jam
123456789	1	22,5
123456789	5	6,5
707080801	8	30,0
777677768	5	20,0
777677768	1	10,0
122557721	5	10,0
122557721	8	23,5
122557721	12	15,0
122557721	20	10,0
987654321	25	12,5

Gambar 5.6 Pernyataan Basis Data yang Mungkin untuk Skema Basis Data Relasional PERUSAHAAN



Gambar 5.7 *Referential integrity constraint* yang ditampilkan pada Sekma Basis Data Relasional PERUSAHAAN

E. Rangkuman

Pada bab ini dibahas mengenai konsep pemodelan, struktur data, dan batasan yang disediakan oleh model data relasional. Dimulai dengan memperkenalkan konsep domain, atribut, dan tupel. Kemudian, definisi skema relasi sebagai daftar atribut yang menggambarkan struktur suatu relasi. Relasi, atau status relasi, adalah sekumpulan tupel yang sesuai dengan skema.

Ada beberapa karakteristik yang membedakan relasi dari tabel atau file biasa. Yang pertama adalah bahwa suatu relasi tidak sensitif terhadap urutan tupel. Yang kedua melibatkan pengurutan atribut dalam skema relasi dan pengurutan nilai yang sesuai dalam sebuah tupel. Di bagian ini juga diberikan definisi alternatif dari relasi yang tidak memerlukan pengurutan atribut, akan tetapi definisi pertama yang tetap digunakan, yang membutuhkan atribut dan nilai tupel untuk diurutkan, untuk kenyamanan. Lebih lanjut, dibahas nilai dalam tupel dan memperkenalkan nilai NULL untuk mewakili informasi yang hilang atau tidak diketahui. Dalam hal ini, diekankan bahwa nilai NULL harus dihindari sebisa mungkin.

Pada bab ini juga dibahas pengklasifikasian batasan terhadap basis data yang menjadi batasan berbasis model yang melekat (*inherent model-based constraint*), batasan berbasis skema (*schema-based constraints*). Dilanjutkan dengan membahas batasan skema yang berkaitan dengan model relasional, dimulai dengan batasan domain (*domain constraint*), kemudian batasan kunci (termasuk konsep *superkey*, *key*, dan *primary key*), dan batasan NOT NULL pada atribut. Pada bagian ini didefinisikan juga berkaitan basis data relasional dan skema basis data relasional.

F. Latihan Soal

1. Definisikan istilah-istilah berikut yang berlaku untuk model relasional data, yaitu domain, atribut, n-tupel, skema relasi, status relasi, derajat relasi, skema basis data relasional, dan status basis data relasional.
2. Mengapa tupel dalam suatu relasi tidak terurut?
3. Mengapa tupel duplikat tidak diizinkan dalam suatu relasi?
4. Apa perbedaan antara *key* dan *superkey*?
5. Mengapa kita menunjuk salah satu *candidate key* dari suatu relasi sebagai *primary key*?
6. Diskusikan karakteristik relasi yang membuatnya berbeda dari tabel dan file biasa!
7. Bahaslah berbagai alasan yang menyebabkan terjadinya nilai NULL dalam relasi!
8. Diskusikan integritas entitas dan batasan integritas referensial. Mengapa masing-masing dianggap penting?
9. Menentukan *foreign key*. Untuk apa konsep ini digunakan?

G. Referensi

- Bagui, Sikha Saha & Richard Walsh Earp. 2023. Database Design Using Entity-Relationship Diagrams. Third Edition. New York: CRC Press
- Connolly, Thomas M., & Carolyn E. Begg. 2015. Database Systems: A Practical Approach to Design, Implementation, and Management. England: Pearson.
- Date, C. J. 2019. Database Design and Relational Theory: Normal Forms and All That Jazz. Second Edition. USA: Apress
- Elmasri, Ramez & Shamkant B. Navathe. 2016. Fundamental of Database Systems. Seventh Edition. United State: Pearson.
- Silberschatz, A., Hendry F. Korth, & S. Sudarshan. 2020. Database System Concepts. Seventh Edition. New York: McGraw-Hill Education.