CHAPTER 8

Pemodelan Data dan Analisis

# Pratinjau Bab dan Tujuan

Dalam bab ini Anda akan belajar cara menggunakan alat pemodelan data populer, hubungan rntity diagmms, untuk mendokumentasikan data yang harus ditangkap at1d disimpan oleh sistem, secara independen untuk menunjukkan bagaimana data itu akan atau akan - yaitu, terlepas dari spesifik inpu5, output, dan pemrosesan. Anda juga akan belajar tentang teknik analisis data yang disebut nomJOliwtion yang digunakan untuk memastikan bahwa model data adalah model data yang "baik". Kamu akan tahu pemodelan data dan analisis data sebagai alat dan teknik analisis sistem saat kamu bisa:

* Menentukan pemodelan sistem dan membedakan antara sistem logis dan fisik model.
* Tentukan pemodelan data dan jelaskan manfaatnya.
* Mengenali dan memahami konsep dasar dan konstruksi model data.
* Baca dan tafsirkan model data hubungan entitas.
* Jelaskan kapan model data dibangun selama proyek dan di mana model disimpan.
* Entitas dan hubungan llicover.
* Buat diagram konteks hubungan entitas .
* temukan atau ciptakan kunci untuk entitas dan buat diagram berbasis kunci.
* Bangun diagram hubungan entitas yang sepenuhnya dikaitkan dan gambarkan semua struktur data dan atribut ke repositori atau ensiklopedia
* Normalisasi model data logis untuk menghilangkan kotoran yang dapat membuat database tidak stabil, tidak fleksibel, dan tidak dapat diskalakan.
* Jelaskan alat yang berguna untuk memetakan persyaratan data ke lokasi operasi bisnis.

# Pengantar

Ketika proyek sistem Layanan Anggota SoundStage bergerak dari analisis persyaratan menjadi desain logis, ttsk pertama menurut metodologi mereka adalah menganalisis persyaratan data untuk saya sistem baru. Bob Martinez ingat seorang profesor favorit di perguruan tinggi yang selalu sald, "Dapatkan data dengan benar dan sistem akan dapat mendukung semua persyaratan Anda saat ini dengan elegan dan bahkan persyaratan yang belum ditetapkan pengguna; mendapatkan data yang salah dan akan merepotkan di leher untuk memenuhi persyaratan hari ini, besok, dan selamanya."

Bob menikmati kelas basis datanya di kampus dan selalu berhasil dengan baik di dalamnya. Tentu saja, sistem pelayanan anggota lebih besar dan lebih detail dari setiap proyek data yang dia lakukan di sekolah. Untungnya, ia memiliki database dari versi awal systern untuk memulai, ditambah formulir dan laporan dari sistem sebelumnya, ditambah catatan dari pandangan pengguna, ditambah narasi kasus penggunaan yang dibuat selama fase analisis persyaratan. Sandra telah meminta Bob untuk mengambil bidikan pertama dalam menarik semuanya menjadi model data logis. Dia bertekad untuk membuatnya terkesan.

# Apa itu Pemodelan Data?

Model sistem memainkan peran penting dalam pengembangan sistem. bab ini akan menyajikan pemodelan data sebagai teknik untuk mendefinisikan persyaratan bisnis untuk suatu basis data. Pemodelan data kadang-kadang disebut pemodelan basis data karena model data akhirnya diimplementasikan sebagai basis data.

Gambar 8-1 Adalah contoh model data sederhana yang disebut diagram hubungan entitas, atau ERD. Diagram ini membuat pernyataan bisnis berikut:

* Kita perlu Menyimpan data tentang PELANGGAN, PESANAN, dan PRODUK PERSEDIAAN.
* Nilai PELANGGAN NUMBER unik hanya satu dan hanya satu PELANGGAN. Nilai ORDER NUMBER tidak cukup mengidentifikasi satu dan hanya satu ORDER. Nilai NOMOR PRODUK tidak berubah dari satu dan hanya satu PRODUK PERSEDIAAN.
* Untuk PELANGGAN kita perlu mengetahui NAMA PELANGGAN, ALAMAT PENGIRIMAN, ALAMAT PENAGIHAN, dan SALDO AKIBAT. Untuk ORDER kita perlu tahu ORDER DATE dan MEMESAN. TOTAL BIAYA. Untuk PRODUK INVENTARIS kita perlu mengetahui NAMA PRODUK, UNIT UKURAN PRU, dan UNIT PIUCE PRODUK.
* PELANGGAN telah menempatkan nol, satu, atau PESANAN saat ini atau lebih BARU
* Semua ORDER ditempatkan tepat oleh satu PELANGGAN. Nilai CUSTOMS NUMBER. (sebagaimana dicatat dalam ORDER) mengidentifikasi PELANGGAN itu.
* ORDER menjual satu atau lebih PRODUK ORDER. Dengan demikian, ORDER harus mengandung setidaknya satu PRODUK ORDER.
* PRODUK INVENTORY mungkin telah terjual nol, satu, atau lebih PRODUK PESANAN.
* Semua PRODUK ORDER mengidentifikasi PRODUK INVENTORY tunggal pada ORDER tunggal. PESANAN NOMOR (untuk PRODUK PESAN) lfentfle ORDERED, dan NOMOR PRODUK. (untuk PRODUK ORDER) lfentlfles PRODUK INVENTORY. Bersama-sama, mereka mengidentifikasi satu dan hanya satu PRODUK ORDER.
* Untuk setiap PRODUK PESANAN kita perlu tahu KUANTITAS MEMESAN dan UNIT WAKTU HARGA PESANAN.

Setelah mempelajari bab ini, Anda akan dapat membaca model data dan membangunnya.

# Konsep Sistem untuk Pemodelan Data

ada beberapa notasi untuk pemodelan data. ‘Model aktual ts sering disebut entitas relatlonship diagram (ERD) karena itu mengurai data dalam hal entitas dan hubungan yang dideskripsikan oleh data. Ada beberapa notasi untuk ERD. Sebagian besar dinamai menurut penemunya (mis., A1et1, Martin, Bachman, Merlse) atau setelah dipublikasikan standar (mis., IDEFIX). "Bahasa" pemodelan data ini umumnya mendukung konsep dan konstruksi dasar yang sama. Kami telah mengadopsi notasi Martin (rekayasa informasi) karena penggunaannya yang luas dan dukungan alat CASE. mari kita jelajahi beberapa konsep dasar yang mendasari semua model data.

# Entitas

Semua Sistem berisi data-biasanya banyak data! Dara menjelaskan "thhngs." Pertimbangkan sistem sekolah. Sistem sekolah memasukkan data yang menggambarkan hal-hal seperti MAHASISWA, GURU, KURSUS, dan KELAS. Untuk hal-hal ini, tidak sulit untuk dibayangkan beberapa data yang menjelaskan setiap Instance yang diberikan. Misalnya, data yang menggambarkan siswa tertentu mungkin termasuk NAMA, ALAMAT, NOMOR TELEPON, TANGGAL KELAHIRAN, GENDER, RACE, MAJOR, dan RATA-RATA POINT KELAS, untuk beberapa nama.

Kita perlu konsep untuk secara abstrak mewakili semua Mesin Virtual dari kelompok yang serupa. Kami menyebut konsep ini sebagai entitas. Entitas adalah sesuatu yang dibutuhkan bisnis untuk menyimpan data. Dalam pemodelan sistem, kami merasa berguna untuk menetapkan setiap konsep abstrak ke bentuk. Dalam buku ini, suatu entitas akan digambarkan sebagai persegi panjang dengan sudut bulat (lihat margin). Bentuk ini mewakili semua instance dari entitas bernama) '. Sebagai contoh. entitas SISWA mewakili semua siswa dalam sistem. Jadi. entitas mengidentifikasi kelas entitas tertentu dan dapat dibedakan dari entitas lainnya.

Kategori kegiatan (dan contoh) meliputi:

**Orang**:

BADAN, KONTRAKTOR, PELANGGAN, DEPARTEMEN, DIVISI, KARYAWAN, INSTRUKTUR, PEMASOK SISWA. Perhatikan bahwa kelas entitas seseorang dapat mewakili individu, grup, atau organisasi.

**Tempat**:

DAERAH PENJUALAN, BANGUNAN, KAMAR, KANTOR CABANG, KAMPUS.

**Benda**:

BUKU, MESIN, BAGIAN, PRODUK, BAHAN BAKU, LISENSI PERANGKAT LUNAK ,, PAKET PERANGKAT LUNAK, ALAT, MODEL KENDARAAN, KENDARAAN dan entitas objek dapat mewakili objek aktual (seperti lisensi perangkat lunak tertentu) atau spedllcations untuk jenis objek (seperti spek untuk paket perangkat lunak yang berbeda).

**Acara**:

APPUKASI, PENGHARGAAN, PEMBATALAN, KELAS, PENERBANGAN, FAKTUR, PESANAN, PENDAFTARAN, PEMBARUAN, PERSYARATAN, PEMESANAN, PENJUALAN, PENJUALAN, PERJALANAN.

**Konsep:**

AKUN, BLOK WAKTU, OBLIGASI, KURSUS, DANA, KUALIFIKASI, SAHAM.

**entitas entitas a**

**terjadinya suatu entitas.**

Sangat penting untuk membedakan antara entitas dan insrances mereka. Entitas blStaoce adalah kejadian tunggal entitas. Sebagai contoh. entitas SISWA dapat memiliki beberapa Instance: Mary, Joe, mark, Susan, Cheryl dan sebagainya. Dalam pemodelan data, kami tidak memusatkan diri dengan sruden indtvidual karena kami menyadari bahwa setiap siswa dijelaskan oleh potongan data yang serupa.

# Atribut

Jika suatu entitas adalah sesuatu yang ingin kita simpan dara, maka kita perlu mengidentifikasi potongan bendungan tertentu yang ingin kita simpan tentang setiap instance dari entitas yang diberikan. Kami menyebutnya potongan atribut data ini. Seperti disebutkan di awal bagian ini, masing-masing untuk. pendirian entitas SISWA mungkin dijelaskan oleh atribut berikut: NAME, ALAMAT, NOMOR TELEPON, TANGGAL KELAHIRAN, GENDER, UTAMA, RATA-RATA POIN RATA-RATA, dan Lainnya.

Kita sekarang dapat memperluas abstraksi grafis entitas untuk lndude atribut dengan merekam atribut-atribut di dalam bentuk etttity bersama dengan nama (lihat margin).

Beberapa atribut dapat secara logis dikelompokkan menjadi superattrlbutes dengan atribut compolutd. Misalnya, NAMA siswa sebenarnya adalah atribut majemuk yang terdiri dari NAMA TERAKHIR, NAMA PERTAMA, dan MIDDLE INTIAL.di margin, kami menunjukkan satu kemungkinan notasi untuk atribut gabungan. Perhatikan bahwa suatu periode ditempatkan pada awal setiap atribut primitif yang termasuk dalam atribut komposit.

Domain Saat menganalisis suatu sistem, kami harus mendefinisikan nilai-nilai untuk atribut yang sah atau yang masuk akal bisnis. Nilai-nilai untuk setiap atribut didefinisikan dalam tiga sifat: tipe data, domain, dan default. Tipe data untuk atribut menentukan tipe data apa yang dapat disimpan dalam atribut itu. Pengetikan data harus sesuai dengan Anda yang memiliki program komputer tertulis; jenis dedarlng untuk variabel adalah umum untuk sebagian besar bahasa pemrograman.

Untuk keperluan analisis sistem dan persyaratan bisnis definitlon, akan berguna untuk mendeklarasikan tipe data logis (nontedullcal) untuk atribut bisnis. Demi argumen, kita akan menggunakan tipe data logis yang ditunjukkan pada Tabel 8-1.

Tipe data attrlbure membatasi domainnya. Domain atribute menentukan nilai apa yang dapat diambil oleh atribut. Bahkan, perancang sistem harus menggunakan teknologi untuk menegakkan domain bisnis dari semua atribut. Tabel 8-2 menunjukkan

bagaimana domain logis dapat diekspresikan untuk setiap tipe data.



Akhirnya, setiap atribut harus memiliki nilai default logis yang mewakili nilai atribut Jika nilainya tidak ditentukan oleh pengguna. Tabble 8.3 menunjukkan kemungkinan atribut default untuk suatu atribut. Perhatikan bahwa NOT NULL adalah cara untuk menentukan bahwa setiap insrance dari atribut harus memiliki nilai, sementara NULL adalah cara untuk menentukan bahwa beberapa contoh atribut mungkin opsional, atau tidak memiliki nilai.