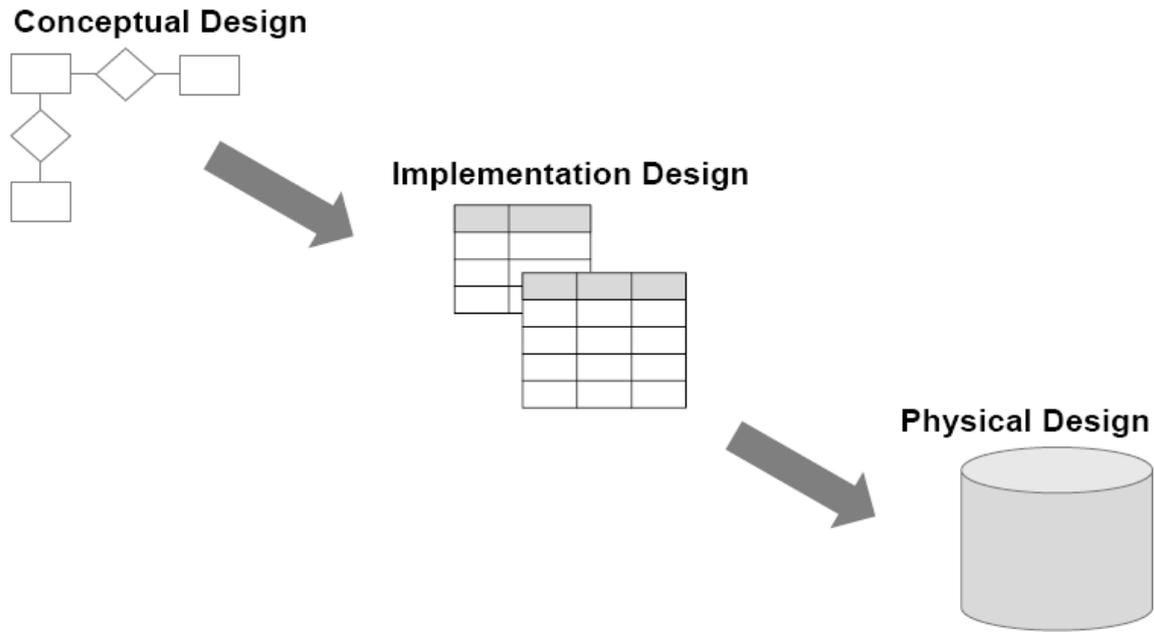


Tutorial objek database

BAGIAN 1 PENGANTAR OBJEK DATABASE

Database Design



Sistem manajemen basis data

Konseptual pemodelan akses Data dan representasi

Client Interface Layer

E/R
SQL, JDBC, ODBC

Data dan semantik operasional

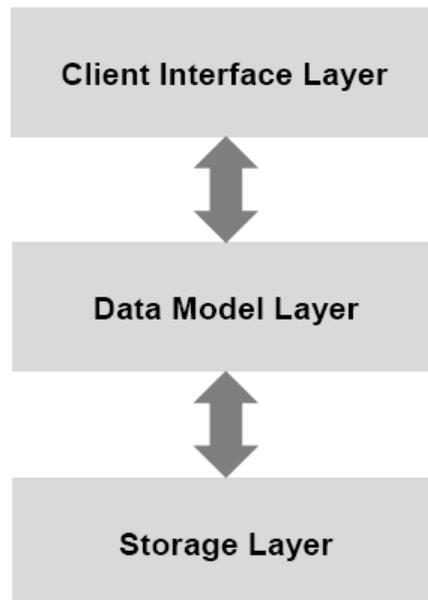
Data Model Layer

Relational Model

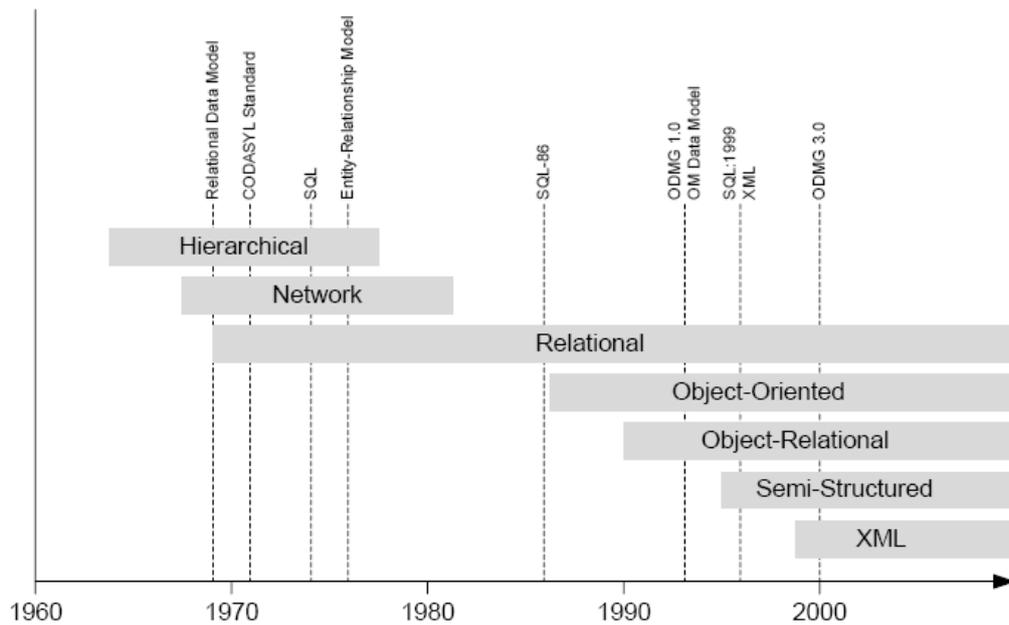
Persistence ACID Distribution

Storage Layer

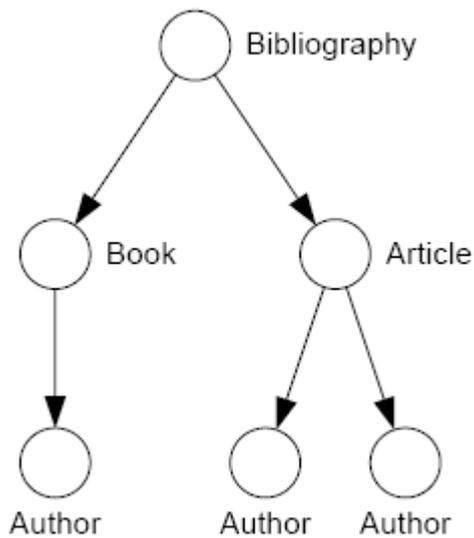
RDBMS



Evolusi dan sejarah



Database hierarki



- Data ditampilkan seperti pohon
 - seperti pohon yang memiliki banyak cabang
 - setiap cabang mempunyai satu pohon
- Records dijelaskan oleh jenis badan
- Hubungan 1:N (satu-ke-banyak)
- Query oleh jalur navigasi

Contoh :

File sistem

LDAP

Windows registri dan Active Directory

XML dokumen dan XQuery

Jaringan database

Data diselenggarakan di grafik (lattice)

- seperti pohon yang memiliki banyak cabang
- setiap cabang mempunyai banyak pohon

Diagram Bachmann

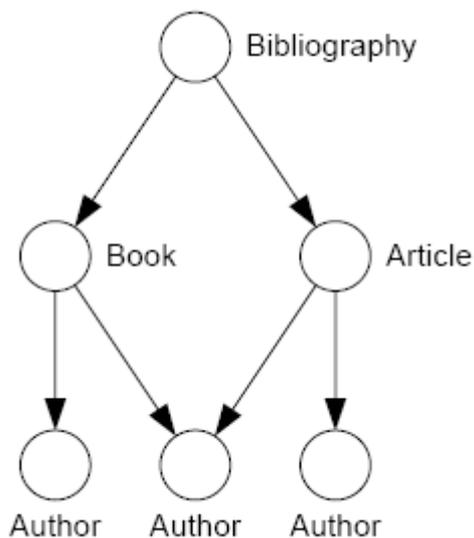
- Record mendefinisikan jenis properties

Set tipe relasi

- Parent-anak, (double) linked list...

Query oleh grafik navigasi

Contoh : CODASY



Database relational

- Data sebagai tupel dalam hubungan
- Link antara data tupel
- Primary dan kunci asing
- Relational aljabar
- Project, pilih, bergabung
- Relational Bentuk normal
- Declarative Bahasa
- Data definisi, konsistensi, manipulasi dan query
- Contoh : Oracle 11 g, Microsoft SQL Server, IBM DB2, PostgreSQL, MySQL

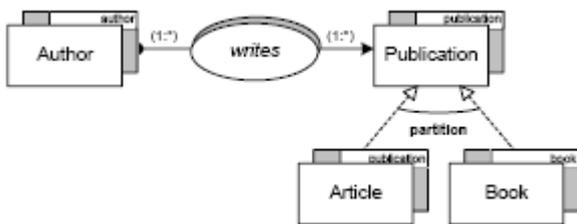
Object-Relational Impedance Mismatch

Pengembangan aplikasi berorientasi objek dan manajemen data relasional hasil dalam bentrok dua model-model yang kompatibel

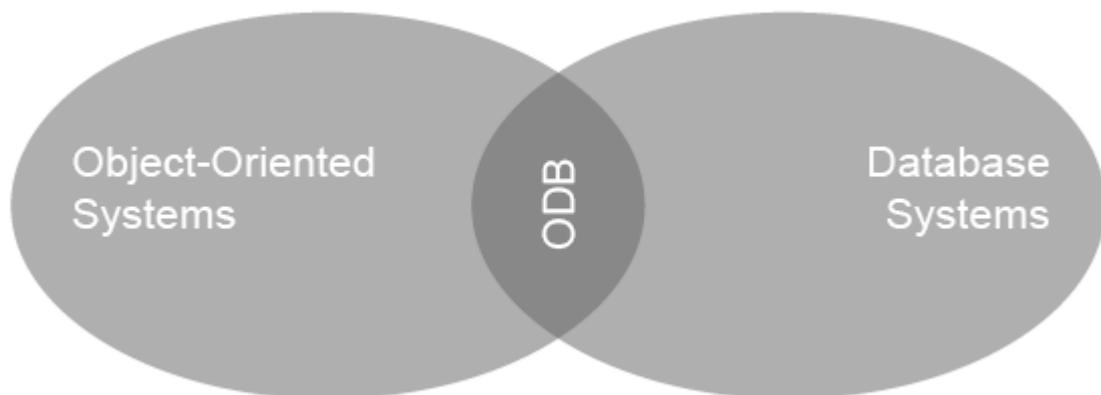
Code untuk memetakan antara model-model adalah atas kepala cukup besar, mahal dan sulit untuk mempertahankan

Objek database

- Data digambarkan sebagai objek
- Identitas Object
- Attributes dan metode
- References, hubungan, Asosiasi
- Extensible jenis hirarki
- user jenis, tipe data abstrak, Didefinisikan
- Single atau pewarisan berganda
- overloading, mengganti, terlambat mengikat
- Declarative bahasa untuk tujuan ad hoc
- Binding untuk bahasa pemrograman berorientasi objek



Object Databases



Menghindari object-relational impedansi ketidakcocokan

- Provide model data seragam
- Combine fitur dan sifat
- Object dan bahasa sistem yang berorientasi
- Database sistem manajemen

Mendefinisikan objek database

- Diverse fokus sistem database berorientasi objek.
- Making bahasa berorientasi objek pemrograman gigih.
- Managing dan menyimpan objek data.
- Many upaya untuk mendefinisikan berorientasi objek database.
- The Manifesto berorientasi objek database
- 13 Fitur wajib
- 5 Karakteristik opsional
- 4 membuka pilihan.
- several dari kamp relasional. Bantahan-bantahan tersebut
- several sifat-sifat penting yang tidak dibahas.

Manifesto Database berorientasi objek

1. Kompleks objek
2. Objek identitas
3. Enkapsulasi
4. Jenis dan kelas
5. Jenis dan kelas hierarki
6. Overriding, overloading dan terlambat mengikat
7. Komputasi kelengkapan
8. Diperpanjang
9. Ketekunan
10. Efisiensi
11. Concurrency
12. Keandalan
13. Deklaratif query language

Objek

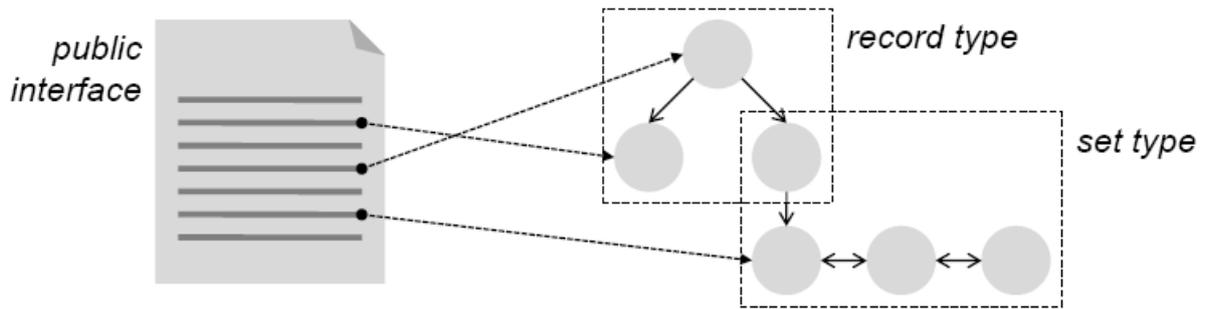
Complex objek

- Complex objek yang terbentuk dari lebih sederhana oleh konstruktor.
- Record, set, tas, daftar dan array kompleks objek konstruktor.
- Constructor orthogonality.
- Object identitas dan kesetaraan.
- Every objek yang memiliki pengidentifikasi objek yang unik dan kekal (OID).
- Sharing objek melalui referensi.
- Two objek identik jika mereka mempunyai OID yang sama.
- Two objek sama jika mereka memiliki keadaan yang sama.
- Shallow dan mendalam kesetaraan.

Encapsulation

- Object terdiri dari antarmuka dan pelaksanaan.
- Interface mendefinisikan tanda tangan dari metode umum.
- Implementation termasuk objek data dan metode.
- Object negara hanya diubah melalui metode umum.

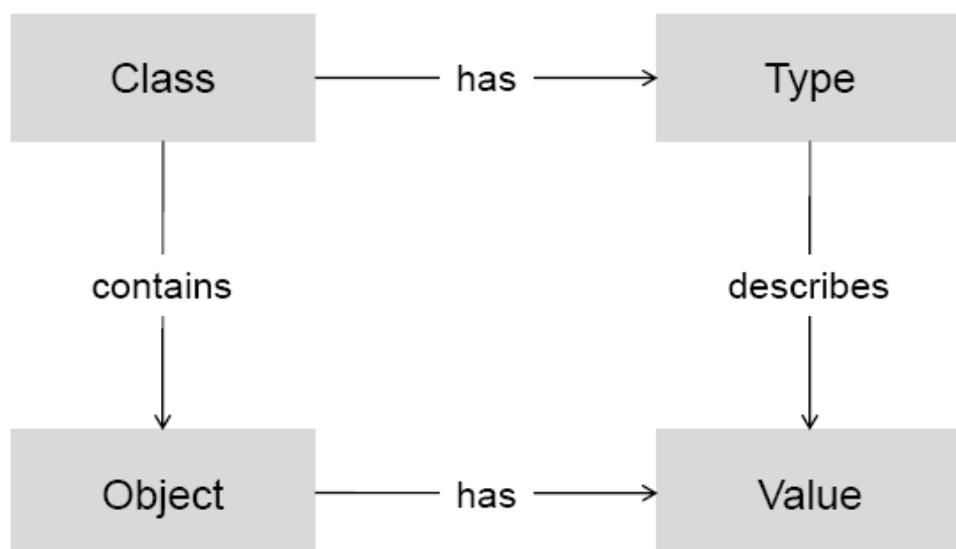
- Object mungkin terkena deklaratif kueri struktur data



Jenis dan kelas

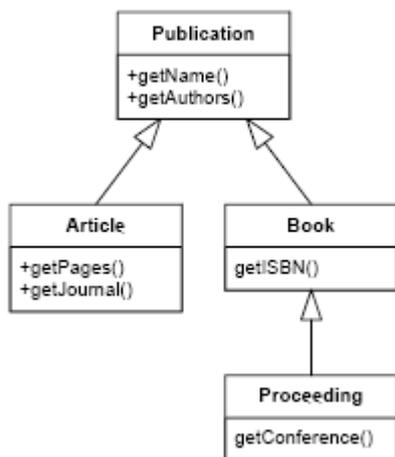
- Jenis data
- Definisi properti objek
- Bagian static menjelaskan struktur objek
- Bagian dynamic menjelaskan perilaku objek.
- Separation antarmuka dan pelaksanaan.
- Used untuk memeriksa kebenaran program pada waktu kompilasi.
- Object kelas.
- Container untuk objek dari jenis yang sama.
- Objects dapat ditambahkan dan dihapus.
- Used untuk membuat dan memanipulasi objek saat menjalankan

Types and Classes



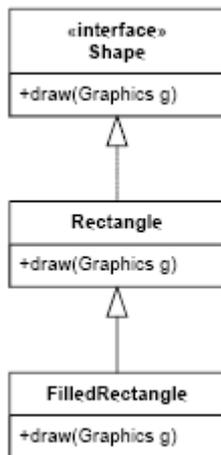
Hierarki-hierarki generalisasi

- ✓ Advantages.
- ✓ Alat pemodelan yang powerful.
- ✓ Guarantee semantik kompleksitas.
- ✓ Reuse spesifikasi dan pelaksanaan.
- ✓ Inheritance.
- ✓ Objects subkelas otomatis milik superclass.
- ✓ Attributes dan metode diwariskan dari superclass.
- ✓ subclass dapat memperkenalkan atribut baru dan metode.
- ✓ Migration antara kelas.
- ✓ MOVE objek antara tingkat hirarki.
- ✓ Object spesialisasi (↓) dan generalisasi (↑).
- ✓ Class contoh versus anggota kelas.



Override, Overloading dan terlambat mengikat

- Mengganti method.
- Method didefinisikan ulang di subtype.
- Guarantees spesialisasi metode.
- Preserves seragam metode antarmuka.
- Method overloading.
- Effect yang disebabkan oleh metode override.
- Various versi metode dapat eksis secara paralel.
- Mengikat late.
- Appropriate versi kelebihan beban metode yang dipilih saat menjalankan.
- Also dikenal sebagai virtual metode pengiriman.



Komputasi kelengkapan dan diperpanjang

- Computational.kelengkapan
- Requirement untuk metode pelaksanaan bahasa.
- Any multisektoral yang dapat dikomputasi fungsi dapat dinyatakan.
- Can direalisasikan melalui koneksi dengan bahasa yang ada.
- Extensibility.
- Database memiliki seperangkat standar jenis.
- Developers dapat menentukan jenis baru sesuai kebutuhan.
- No penggunaan perbedaan antara jenis sistem dan pengguna.

Ketekunan

- Data telah hidup lebih lama pelaksanaan program.
- Implicit dan ketekunan eksplisit.
- Transparent ketekunan.
- Persistence ortogonal untuk mengetik.
- Any objek mungkin gigih atau sementara.
- No perlu untuk menulis kode untuk ketekunan.
- Persistence strategi.
- Dari kemampuan ketekunan mewarisi dari kelas gigih telah ditetapkan. Warisan
- Dari Instansiasi: objek membuat gigih dan mendapatkan kapabilitas ketekunan berdasarkan Instansiasi.
- Dari reachability: objek dibuat terus-menerus jika dicapai dari objek lain terus-menerus.

Efisiensi, kontrol concurrency, dan pemulihan

- Secondary manajemen Penyimpanan.
- Manajemen index, data clustering dan buffering.
- Access jalan seleksi dan permintaan optimasi.
- Concurrency.
- Management dari beberapa pengguna secara bersamaan berinteraksi.
- Atomicity, konsistensi, isolasi dan daya tahan.
- Serialisability operasi.
- Reliability.
- Resiliency untuk pengguna, perangkat lunak dan perangkat keras kegagalan.

- Transactions dapat dilakukan atau dibatalkan.
- Restore keadaan koheren sebelumnya data.
- Redoing dan melepas transaksi.
- Logging operasi.

Bahasa kueri deklaratif

- ✓ High-tingkat bahasa.
- ✓ Express non-sepele queries ringkas.
- ✓ Text atau berbasis grafis antarmuka.
- ✓ Declarative.
- ✓ Efficient pelaksanaan.
- ✓ Possibility untuk query optimisation.
- ✓ Application independen.
- ✓ Work pada database setiap kemungkinan.
- ✓ No perlu untuk metode tambahan pada jenis ditetapkan pengguna

Opsional karakteristik dan pilihan yang terbuka

- Karakteristik optional.
- Pusaka Multiple.
- Type memeriksa dan kesimpulan.
- Distribution.
- Design transaksi, panjang transaksi, transaksi yang bersarang.
- Versions.
- Pilihan Open.
- Paradigma Programming.
- Sistem representation.
- Sistem Type.
- Uniformity.

Luar Manifesto

- Database administrasi utilitas.
- View definisi dan turunan data.
- Object peran.
- Objects memiliki peran selain jenis.
- Roles dapat memperoleh dan kehilangan secara dinamis.
- Database evolusi.
- Schema dan data telah anggun berevolusi dari waktu ke waktu.
- Constraints.
- Integrity, semantik dan evolusi kendala.
- Definition, manajemen dan penegakan kendala.