MODUL 11 :
Data Management Design

Contents

11.1 Pengantar

Setiap sistem informasi memerlukan persistent data yaitu data yang selalu aktiv walaupun sistem sudah tidak aktif. Persistent data dapat dimodelkan dengan menggunakan UML. Pesistent data dapat disimpan dalam bentuk file atau DBMS atau juga dalam bentuk objek database sehingga mempengararuhi rancangan dari manajemen data.

Beberapa sistem berorientasi objek dibatasi oleh organisasi dalam hal investasi hardware dan software yang digunakan dalam menggunakan relasional database (RDBMS) untuk menyimpan data. Normalisasi dapat digunakan untuk merancang tabel dalam database relasional. ada beberapa aturan dalam melakukan konversi satu kelas diagram kedalam tabel yang sesuai.

Rancanangan sebuah objek DBMS juga memberikan dampak yang berbeda dalam merancang model data. Hal ini tergantung dari objek DBMS nya. Beberapa sistem memerlukan data dalam bentuk terdistribusi dengan database yang berbeda. Beberapa mekanisme seperti CORBA (Common Object Request Broker Architecture), RMI (Remote Methode Invocation) atau EJB (enterprise Java Bean) dapat digunakan untuk objek persistant dari logika bisnis pada application layer dan interface objects. CORBA dan EJB daoat menyediakan infrastruktur untuk membangun database terdistribusi.

11.2 Mapping Class ke Table

Dibawah ini adalah yang digunakan oleh Rumbaugh (1991) dan Brown and Whitenack (1996) untuk melakukan mapping dari class ke dalam bentuk tabel :

1. Class dengan struktur yang sederhana maka kelas tersebut akan menjadi sebuah tabel.
2. Identifier dari objek akan menjadi primary key.
3. Kelas yang berisi kelas lain sebagai atribut dipisahkan menjadi kelas yang berbeda dan identifier dari kelas yang berada dalam kelas lain akan menjadi foreign key pada kelas yang memuat kelas tersebut.
4. Untuk koleksi kelas maka dibuattdalam dua tabel, satu untuk objek dalam koleksi, yang lain untuk memegang Obyek ID dari obyek yang berisi dan obyek yang terkandung.
5. asosiasi Satu-ke-banyak dapat diperlakukan seperti koleksi

6. asosiasi Banyak-ke-banyak menjadi dua tabel terpisah untuk objek dan tabel untuk menempatkan ID Obyek pasangannya
7. asosiasi Satu-ke-satu diimplementasikan sebagai atribut foreign key -masing-masing kelas memperoleh atribut ekstra untuk ID Obyek yang lain
8. untuk mengimplementasikan inheritance
    a. hanya mengimplemetasikan superclass sebagai tabel termasuk semua atribut subclass
    b. hanya mengimplementasikan subclasses sebagai tabel, ada duplikasi superclass pada setiap atribut
    c. implementasi superclass dan subclasses sebagai tabel akan menshare primary keys

11.3 Design for Object Database

ODBMS memiliki keuntungan bahwa obyek dapat disimpan secara langsung. Object Data Management Group (ODMG) menjadi standar. Tidak semua database objek sesuai dengan standar. Obyek database terkait erat dengan bahasa pemrograman dengan cara menavigasi melalui database.

Contoh ObjectStore dalam C++

```
IntCampaign * CreativeStaff::findIntCampaign
                              ( string campaignCode )
{
   IntCampaign * intCampaignPointer;
   intCampaignPointer =
   staffIntCampaignList.getValue().query_pick(
        "IntCampaign*",
        "campaignCode == this->campaignCode",
        os_database::of(this));
   return intCampaignPointer;
}
```

Beberapa transparan akan 'terwujud' objek dari database ketika mereka disebut. Transaksi pembaruan perlu diberi tanda kurung dengan start dan finish metode transaksi. Operasi masih diimplementasikan dalam bahasa berorientasi objek

Referensi

1. Simon Bennet, Steve McRobb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML,* Edisi 3. ; McGraw Hill, 2006. (SB)