STT TERPADU
NURUL FIKRI

Basis Data 2

Sirojul Munir S.Si,M.Kom
rojulman@nurulfikri.ac.id | rojulman
dev.xbata.com

Transactions

Transaksi ?


ONLINE
Download from
Dreamstime.com
29539405
iqoncept | Dreamstime.com

Pengertian Transaction

- **Transaction :** Proses eksekusi sebuah program yang dilakukan pada akses basis data dimana didalamnya terdapat rangkain perintah perubahan data (DML).
- **DBMS** harus menjamin bahwa setiap transaksi harus dapat dikerjakan secara utuh atau tidak sama sekali, Tidak boleh ada transaksi yang hanya dikerjakan sebagian, karena dapat menyebabkan inkonsistensi data

Transaction: Commit & Rollback

- Sebuah Transaksi dapat menghasilkan 2 kemungkinan:

- 1. Jika dilaksanakan lengkap seluruhnya, transaksi tersebut telah di commit dan basis data mencapai keadaan konsisten baru.
- 2. Jika transaksi tidak sukses, maka transaksi dibatalkan dan basis data dikembalikan ke keadaan konsisten sebelumnya (rollback)

Transaction – Integritas data (1)

Agar integritas data terjamin, maka transaksi harus memenuhi:

- **Atomik,** dimana semua operasi dalam transaksi dapat dikerjakan seluruhnya atau tidak sama sekali.
- **Konsisten,** dimana eksekusi transaksi secara tunggal harus dapat menjamin data tetap konsisten setelah transaksi berakhir.

Transaction – Integritas data (2)

- **Terisolasi,** jika pada sebuah sistem basis data terdapat sejumlah transaksi yang dilaksanakan secara bersamaan, maka semua transaksi yang dilaksanakan pada saat yang bersamaan tersebut harus dapat dimulai dan bisa berakhir.
- **Bertahan (Persistence),** dimana perubahan data yang terjadi setelah sebuah transaksi berakhir dengan baik, harus dapat bertahan bahkan jika seandainya sistem menjadi mati.

Perintah Transaction

```
dbkoperasi=$ BEGIN;
dbkoperasi=$ // perintah DML
dbkoperasi=$ COMMIT/ROLLBACK;
```

Contoh: Insert data-batalkan-kembali ke kondisi awal

```
dbkoperasi=$ BEGIN
dbkoperasi=$ INSERT INTO jenis_produk VALUES(10,'Assesories');
dbkoperasi=$ INSERT INTO jenis_produk VALUES (11,'Komputer');
dbkoperasi=$ SELECT * FROM jenis_produk ;
dbkoperasi=$ ROLLBACK;
dbkopearsi=$ SELECT * FROM jenis_produk;
```

Point Time Recovery - SAVEPOINT

- **SAVEPOINT :** Berfungsi untuk memberi tanda (check-point) pada sebuah rangkaian transaksi eksekusi perintah DML. Dengan SAVEPOINT transaksi dapat dikembalikan ke sebuah posisi tertentu (tidak perlu dari awal lagi).

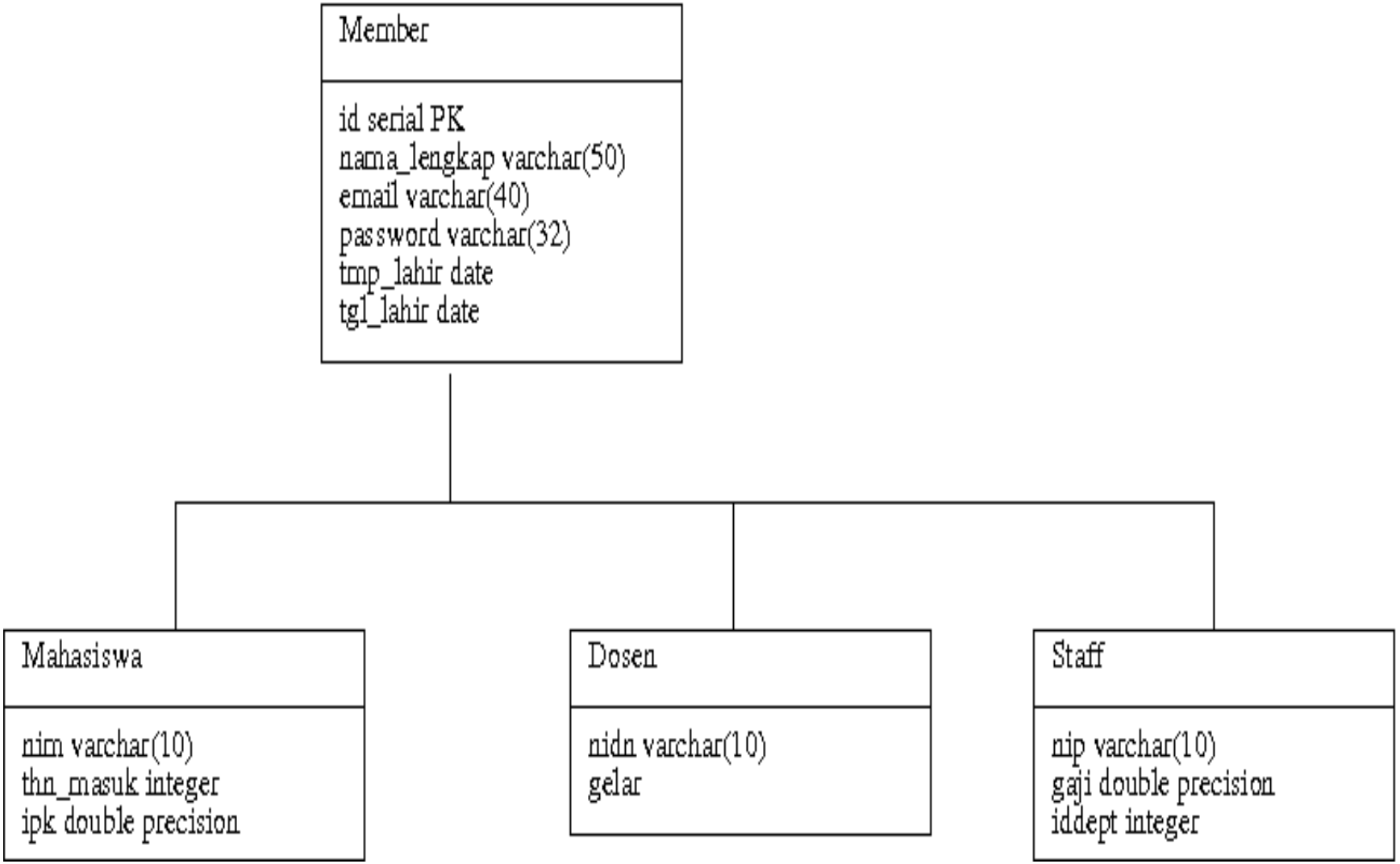
```
dbkoperasi=$ BEGIN;
dbkoperasi=$ DELETE FROM jenis_produk WHERE id=10 ;
dbkoperasi=$ SAVEPOINT savepoin1 ;
dbkoperasi=$ DELETE FROM jenis_produk WHERE id=11;
dbkoperasi=$ ROLLBACK TO SAVEPOINT savepoin1;
dbkoperasi=$ COMMIT ;
```

Inheretance

Table Inheritance

- PostgreSQL adalah database ORDBMS : Object Relational Database Management Systems
- PostgreSQL mensupport sifat penurunan (inheritance) dari satu table ke table lain
- Tabel turunan mewarisi kolom-kolom dari tabel induknya, selain itu tabel turunan dapat memiliki kolom-kolomnya sendiri yang tidak ada dalam tabel induk.
- Record-record pada tabel turunan dapat diakses dari tabel induk.

Table Inheritance - Member



Command : INHERITS

Implementasi inheretenace pada table

- **INHERITS:** kata kunci yang digunakan untuk membuat object table turunan dari table lain
- **Command:** Tabel tabel2 adalah turunan dari table1

```
CREATE TABLE tabel2
(
    -- command DDL
) INHERITS ( tabel1 );
```

Contoh: Tabel member adalah parent dari tabel mahasiswa dan dosen

```
dbkoperasi=$ CREATE TABLE member (
             id serial primary key,nama_lengkap varchar(50),
             email varchar(40), password varchar(32),
             tmp_lahir varchar(30),tgl_lahir date );

dbkoperasi=$ CREATE TABLE mahasiswa (
             nim varchar(10) unique, thn_masuk integer,
             ipk double precision
             ) INHERITS ( member );
dbkoperasi=$ CREATE TABLE dosen (
             nidn varchar(10) unique,
             gelar varchar(20)
             ) INHERITS ( member );
```

Contoh: Skema Tabel member

```
dbkoperasi=# \d member
                                  Table "public.member"
    Column     |          Type          |                       Modifiers
---------------+------------------------+-------------------------------------------------
 id            | integer                | not null default nextval('member_id_seq'::regclass)
 nama_lengkap | character varying(50)  |
 gender        | character(1)           |
 email         | character varying(40)  |
 password      | character varying(32)  |
 tmp_lahir     | character varying(30)  |
 tgl_lahir     | date                   |
Indexes:
    "member_pkey" PRIMARY KEY, btree (id)
Number of child tables: 2 (Use \d+ to list them.)
```

Contoh: Skema Tabel member dan childs

```
dbkoperasi=# \d+ member

                    Table "public.member"
    Column    |         Type          |  Modifiers   | Storage  | Stats target | Descri
--------------+-----------------------+--------------+----------+--------------+----------
 id           | integer               |   xxxxx      | plain    |              |
 nama_lengkap | character varying(50) |              | extended |              |
 gender       | character(1)          |              | extended |              |
 email        | character varying(40) |              | extended |              |
 password     | character varying(32) |              | extended |              |
 tmp_lahir    | character varying(30) |              | extended |              |
 tgl_lahir    | date                  |              | plain    |              |
Indexes:
    "member_pkey" PRIMARY KEY, btree (id)
Child tables: dosen,
              mahasiswa
Has OIDs: no
```

Contoh: Skema Tabel mahasiswa

```
dbkoperasi=# \d mahasiswa
                                  Table "public.mahasiswa"
    Column     |          Type          |                        Modifiers
---------------+------------------------+-----------------------------------------------------
 id            | integer                | not null default nextval('member_id_seq'::regclass)
 nama_lengkap | character varying(50) |
 gender        | character(1)           |
 email         | character varying(40) |
 password      | character varying(32) |
 tmp_lahir     | character varying(30) |
 tgl_lahir     | date                   |
 nim           | character varying(10) |
 thn_masuk     | integer                |
 ipk           | double precision       |
Indexes:
    "mahasiswa_nim_key" UNIQUE CONSTRAINT, btree (nim)
Inherits: member
```

Contoh: Skema Tabel dosen

```
dbkoperasi=# \d dosen
                                    Table "public.dosen"
    Column     |          Type           |                        Modifiers
---------------+-------------------------+-----------------------------------------------------
 id            | integer                 | not null default nextval('member_id_seq'::regclass)
 nama_lengkap | character varying(50)  |
 gender        | character(1)            |
 email         | character varying(40)  |
 password      | character varying(32)  |
 tmp_lahir     | character varying(30)  |
 tgl_lahir     | date                    |
 nidn          | character varying(10)  |
 gelar         | character varying(20)  |
Indexes:
    "dosen_nidn_key" UNIQUE CONSTRAINT, btree (nidn)
Inherits: member
```

Contoh: Data tabel member dan childsnya

```
dbkoperasi=# select id,nama_lengkap,email from member;
 id | nama_lengkap |      email
----+--------------+------------------
  1 | glagah putih | glagah@gmail.com
  2 | sambungsari  | ssari@gmail.com
  3 | agung sedayu | agung@gmail.com
(3 rows)

dbkoperasi=# select id,nama_lengkap,email from mahasiswa ;
 id | nama_lengkap |      email
----+--------------+-----------------
  2 | sambungsari  | ssari@gmail.com
(1 row)

dbkoperasi=# select id,nama_lengkap,email from dosen ;
 id | nama_lengkap |      email
----+--------------+-----------------
  3 | agung sedayu | agung@gmail.com
```