

## Tugas Manajemen Basis Data

### Materi Transaction

#### Membuat Query Tabel Account

```
CREATE TABLE Account(  
  ID SERIAL NOT NULL PRIMARY KEY,  
  NAME VARCHAR(50) NOT NULL,  
  BALANCE INTEGER NOT NULL,  
  LAST_UPDATE TIMESTAMP DEFAULT NOW() NOT NULL  
)
```

#### Membuat Insert Data Database

```
INSERT INTO Account (NAME,BALANCE) VALUES  
( 'Jasmine',1500000),  
( 'Ariel',500000),  
( 'Belle',300000)
```

### Non-Repeatable Read

#### Non-Repeatable Read using Read Committed

The screenshot displays a PostgreSQL database interface with a table named 'Account'. The table has four columns: ID (SERIAL), NAME (VARCHAR(50)), BALANCE (INTEGER), and LAST\_UPDATE (TIMESTAMP). The table contains three rows of data: Jasmine (1500000), Ariel (500000), and Belle (300000). The interface also shows a query history and a table of transaction isolation levels.

|                  | Dirty Reads | Non-Repeatable Reads | Phantom Reads | Serialization Anomaly |
|------------------|-------------|----------------------|---------------|-----------------------|
| Read Uncommitted | ✗           | ✓                    | ✓             | ✓                     |
| Read Committed   | ✗           |                      |               |                       |
| Repeatable Read  | ✗           |                      |               |                       |
| Serializable     | ✗           |                      |               |                       |

The screenshot also shows a Windows PowerShell window with the following SQL commands:

```
Tugas13=# begin;  
BEGIN  
Tugas13=# set transaction isolation level read committed;  
SET  
Tugas13=# update account set balance = balance + 10000 where id = 2;  
UPDATE 1  
Tugas13=# commit;  
COMMIT  
Tugas13=#
```

The screenshot also shows a Windows PowerShell window with the following SQL commands:

```
Tugas13=# begin;  
BEGIN  
Tugas13=# set transaction isolation level read committed;  
SET  
Tugas13=# select * from account order by id;  
id | name | balance | last_update  
-----  
1 | Jasmine | 1500000 | 2023-05-25 08:51:57.272686  
2 | Ariel | 500000 | 2023-05-25 08:51:57.272686  
3 | Belle | 300000 | 2023-05-25 08:51:57.272686  
(3 rows)  
  
Tugas13=# select * from account order by id;  
id | name | balance | last_update  
-----  
1 | Jasmine | 1500000 | 2023-05-25 08:51:57.272686  
2 | Ariel | 500000 | 2023-05-25 08:51:57.272686  
3 | Belle | 300000 | 2023-05-25 08:51:57.272686  
(3 rows)  
  
Tugas13=#
```

#### Non-Repeatable Reads using Read Repeatable Read





The screenshot displays a presentation slide, the pgAdmin database management tool, and a Windows PowerShell terminal window. The slide illustrates a sequence of SQL operations: an update on account balance, followed by a select query, and then a commit. The pgAdmin interface shows the 'Tugas13/postgres@PostgreSQL 15' database with a table 'account' containing three rows. The PowerShell terminal shows the execution of SQL commands: 'begin;', 'set transaction isolation level serializable;', 'update account set balance = balance + 10000 where id = 2;', 'commit;', and 'select \* from account where balance < 700000;'. The terminal output shows the updated balance for the account with id=2.

**Slide Content:**

```

UPDATE ACCOUNT SET BALANCE =
BALANCE + 100000 WHERE ID = 2;
commit transaction

SELECT * FROM ACCOUNT WHERE BALANCE < 700000
Get data

commit transaction
  
```

**pgAdmin Data Output:**

| id | name    | balance | last_update                |
|----|---------|---------|----------------------------|
| 1  | Jasmine | 1500000 | 2023-05-25 08:51:57.272686 |
| 2  | Ariel   | 530000  | 2023-05-25 08:51:57.272686 |
| 3  | Belle   | 300000  | 2023-05-25 08:51:57.272686 |

**PowerShell Output:**

```

Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level serializable;
SET
Tugas13=# update account set balance = balance + 10000 where id = 2;
UPDATE 1
Tugas13=# commit;
COMMIT
Tugas13=#

Tugas13=# select * from account where balance < 700000;
id | name | balance | last_update
---+---+---+---
3 | Belle | 300000 | 2023-05-25 08:51:57.272686
2 | Ariel | 520000 | 2023-05-25 08:51:57.272686
(2 rows)

Tugas13=# commit;
COMMIT
Tugas13=# select * from account where balance < 700000;
id | name | balance | last_update
---+---+---+---
3 | Belle | 300000 | 2023-05-25 08:51:57.272686
2 | Ariel | 520000 | 2023-05-25 08:51:57.272686
(2 rows)

Tugas13=#
  
```

## Serialization Anomaly

### Serialization Anomaly using Read Committed

The screenshot displays a presentation slide, the pgAdmin database management tool, and a Windows PowerShell terminal window. The slide illustrates a sequence of SQL operations: a select query, followed by an insert, and then a commit. The pgAdmin interface shows the 'Tugas13/postgres@PostgreSQL 15' database with a table 'account' containing five rows. The PowerShell terminal shows the execution of SQL commands: 'begin;', 'set transaction isolation level read committed;', 'select sum(balance) from account;', 'insert into account (name,balance) values ('sum',2380000);', 'commit;', and 'select sum(balance) from account;'. The terminal output shows the sum of balances before and after the insert.

**Slide Content:**

```

SELECT SUM(BALANCE) FROM ACCOUNT
Get data

INSERT INTO ACCOUNT data based on select
Insert data

commit transaction

SELECT * FROM ACCOUNT
  
```

**pgAdmin Data Output:**

| id | name  | balance | last_update                |
|----|-------|---------|----------------------------|
| 2  | Ariel | 500000  | 2023-05-25 08:51:57.272686 |
| 3  | Belle | 300000  | 2023-05-25 08:51:57.272686 |
| 4  | sum   | 2300000 | 2023-05-25 13:57:22.118198 |
| 5  | sum   | 2300000 | 2023-05-25 13:59:16.033449 |

**PowerShell Output:**

```

Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level read committed;
SET
Tugas13=# select sum(balance) from account;
sum
-----
2380000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',2380000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#

Tugas13=# select sum(balance) from account;
sum
-----
4660000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',2380000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#
  
```

### Serialization Anomaly using Repeatable Read

13. Tansaka - PowerPoint

Slide 24 of 20

Click to add notes

pgAdmin 4

Dashboard Properties SQL Statistics Dependencies Dependents Tugast3/postgre

Query Query History

```
1 select * from account order by id;
2
3 update account set balance = 1500000 where id = 1;
4
5
```

Data Output Messages Notifications

| id | [PK] integer | name    | character varying (50) | balance | integer | last_update                | timestamp without time zone |
|----|--------------|---------|------------------------|---------|---------|----------------------------|-----------------------------|
| 1  | 1            | Jasmine |                        | 1500000 |         | 2023-05-25 08:51:57.272686 |                             |
| 2  | 2            | Ariel   |                        | 500000  |         | 2023-05-25 08:51:57.272686 |                             |
| 3  | 3            | Bele    |                        | 300000  |         | 2023-05-25 08:51:57.272686 |                             |
| 4  | 4            | sum     |                        | 2300000 |         | 2023-05-25 13:57:22.118198 |                             |
| 5  | 5            | sum     |                        | 2300000 |         | 2023-05-25 13:59:16.033449 |                             |
| 6  | 7            | sum     |                        | 6900000 |         | 2023-05-25 14:00:35.182182 |                             |
| 7  | 8            | sum     |                        | 6900000 |         | 2023-05-25 14:00:38.928436 |                             |

Total rows: 7 of 7 Query complete 00:00:00.090 Ln 1, Col 1

Windows PowerShell

```
Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level repeatable read;
SET
Tugas13=# select sum(balance) from account;
sum
-----
6900000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',6900000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#
```

Windows PowerShell

```
Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level repeatable read;
SET
Tugas13=# select sum(balance) from account;
sum
-----
6900000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',6900000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#
```

## Serialization Anomaly using Serialization

13. Tansaka - PowerPoint

Slide 24 of 20

Click to add notes

pgAdmin 4

Dashboard Properties SQL Statistics Dependencies Dependents Tugast3/postgre

Query Query History

```
1 select * from account order by id;
2
3 update account set balance = 1500000 where id = 1;
4
5
```

Data Output Messages Notifications

| id | [PK] integer | name | character varying (50) | balance  | integer | last_update                | timestamp without time zone |
|----|--------------|------|------------------------|----------|---------|----------------------------|-----------------------------|
| 3  | 3            | Bele |                        | 300000   |         | 2023-05-25 08:51:57.272686 |                             |
| 4  | 4            | sum  |                        | 2300000  |         | 2023-05-25 13:57:22.118198 |                             |
| 5  | 5            | sum  |                        | 2300000  |         | 2023-05-25 13:59:16.033449 |                             |
| 6  | 7            | sum  |                        | 6900000  |         | 2023-05-25 14:00:35.182182 |                             |
| 7  | 8            | sum  |                        | 6900000  |         | 2023-05-25 14:00:38.928436 |                             |
| 8  | 10           | sum  |                        | 20700000 |         | 2023-05-25 14:03:31.297318 |                             |

Total rows: 8 of 8 Query complete 00:00:00.071 Ln 1, Col 1

Windows PowerShell

```
Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level serializable;
SET
Tugas13=# select sum(balance) from account;
sum
-----
20700000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',20700000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#
```

Windows PowerShell

```
Tugas13=# begin;
BEGIN
Tugas13=# set transaction isolation level serializable;
SET
Tugas13=# select sum(balance) from account;
sum
-----
20700000
(1 row)

Tugas13=# insert into account (name,balance) values ('sum',20700000);
INSERT 0 1
Tugas13=# commit;
COMMIT
Tugas13=#
```

## ISOLATION LEVEL

|                  | Dirty Reads | Non-Repeatable Reads | Phantom Reads | Serialization Anomaly |
|------------------|-------------|----------------------|---------------|-----------------------|
| Read Uncommitted | ✗           | ✓                    | ✓             | ✓                     |
| Read Committed   | ✗           | ✓                    | ✓             | ✓                     |
| Repeatable Read  | ✗           | ✗                    | ✓             | ✓                     |
| Serializable     | ✗           | ✗                    | ✗             | ✗                     |