Proyek Akhir Teknologi Basis Data

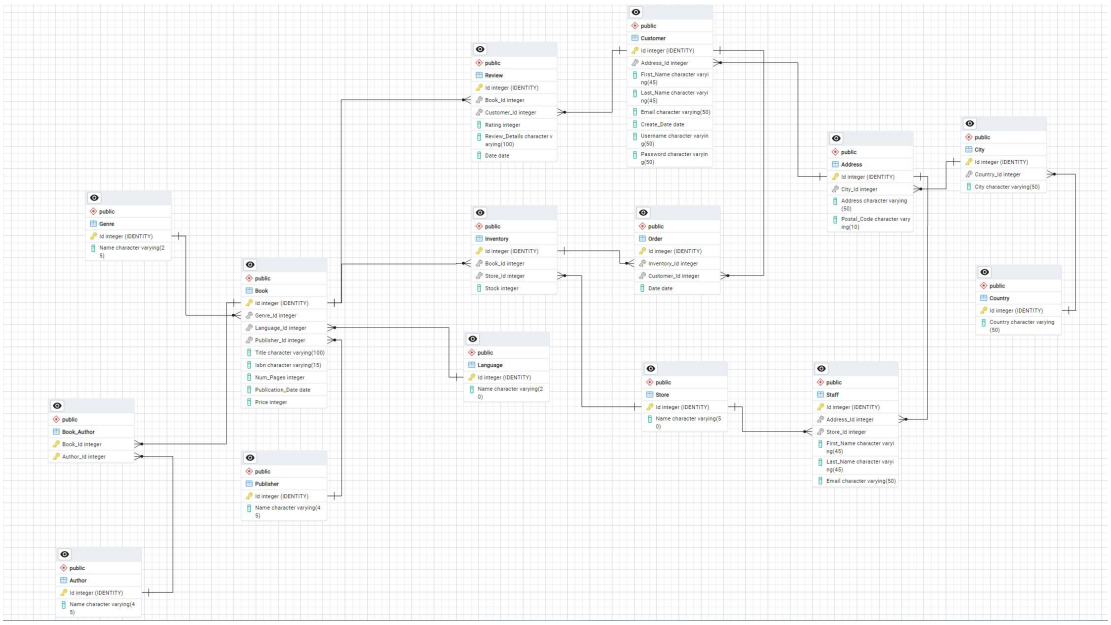
Task 1: DB Expansion and Implementation

Disusun oleh :

Melvin Waluyo

(22/492978/TK/53972)

- **ERD**



Relasi pada ERD

1. Book-Genre
   - Satu Book memiliki satu Genre
   - Satu Genre dapat termasuk dalam banyak Book

2. Book-Publisher
   - Satu Book diterbitkan oleh satu Publisher
   - Satu Publisher dapat menerbitkan banyak Book

3. Book-Book_Author
   - Satu Book termasuk dalam banyak Book_Author
   - Satu Book_Author hanya berisi satu Book

4. Book-Language
   - Satu Book ditulis dalam satu Language
   - Satu Language dapat termasuk dalam banyak Book

5. Book-Review
   - Satu Book dapat memiliki banyak Review
   - Satu Review hanya mencakup satu Book

6. Book-Inventory
   - Satu Book dapat termasuk dalam banyak Inventory
   - Satu Inventory hanya mencakup satu Book

7. Book_Author-Author
   - Satu Book_Author hanya berisi satu Author
   - Satu Author dapat termasuk dalam banyak Book_Author

8. Review-Customer
   - Satu Review hanya ditulis oleh satu customer
   - Satu Customer dapat menulis banyak Review

9. Inventory-Order
    - Satu Inventory dapat termasuk dalam banyak Order
    - Satu Order hanya mencakup satu Inventory

10. Inventory-Store
    - Satu Inventory terletak dalam satu Store
    - Satu Store dapat memiliki banyak Inventory

11. Customer-Order
    - Satu Customer dapat melakukan banyak Order
    - Satu Order hanya mencakup satu Customer

12. Customer-Address
    - Satu Customer memiliki satu Address
    - Satu Address dapat mencakup banyak Customer

13. Store-Staff
    - Satu Store dapat memiliki banyak Staff
    - Satu Staff hanya bekerja pada satu Store

14. Staff-Address
    - Satu Staff memiliki satu Address
    - Satu Address dapat mencakup banyak Staff

15. Address-City
    - Satu Address hanya berada dalam satu City
    - Satu City dapat memiliki banyak Address

16. City-Country
    - Satu City hanya berada dalam satu Country
    - Satu Country dapat memiliki banyak City

- **Database Schema**

```
-- This script was generated by the ERD tool in pgAdmin 4.
-- Please log an issue at https://github.com/pgadmin-
org/pgadmin4/issues/new/choose if you find any bugs, including
reproduction steps.
BEGIN;


CREATE TABLE IF NOT EXISTS public."Customer"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Address_Id" integer,
    "First_Name" character varying(45),
    "Last_Name" character varying(45),
    "Email" character varying(50),
    "Create_Date" date,
    "Username" character varying(50),
    "Password" character varying(50),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Publisher"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Name" character varying(45),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Author"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Name" character varying(45),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Order"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Inventory_Id" integer,
    "Customer_Id" integer,
    "Date" date,
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Book"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Genre_Id" integer,
    "Language_Id" integer,
    "Publisher_Id" integer,
```

```
    "Title" character varying(100),
    "Isbn" character varying(15),
    "Num_Pages" integer,
    "Publication_Date" date,
    "Price" integer,
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Genre"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Name" character varying(25),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Language"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Name" character varying(20),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Review"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Book_Id" integer,
    "Customer_Id" integer,
    "Rating" integer,
    "Review_Details" character varying(100),
    "Date" date,
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Book_Author"
(
    "Book_Id" integer,
    "Author_Id" integer,
    PRIMARY KEY ("Book_Id", "Author_Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Inventory"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Book_Id" integer,
    "Store_Id" integer,
    "Stock" integer,
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Store"
```

```
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Name" character varying(50),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Staff"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Address_Id" integer,
    "Store_Id" integer,
    "First_Name" character varying(45),
    "Last_Name" character varying(45),
    "Email" character varying(50),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Address"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "City_Id" integer,
    "Address" character varying(50),
    "Postal_Code" character varying(10),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."City"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Country_Id" integer,
    "City" character varying(50),
    PRIMARY KEY ("Id")
);
```

```
CREATE TABLE IF NOT EXISTS public."Country"
(
    "Id" integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 ),
    "Country" character varying(50),
    PRIMARY KEY ("Id")
);
```

```
ALTER TABLE IF EXISTS public."Customer"
    ADD CONSTRAINT "Customer_Address_Id_fk" FOREIGN KEY ("Address_Id")
    REFERENCES public."Address" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Order"
```

```
    ADD CONSTRAINT "Order_Inventory_Id_fk" FOREIGN KEY ("Inventory_Id")
    REFERENCES public."Inventory" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Order"
    ADD CONSTRAINT "Order_Customer_Id_fk" FOREIGN KEY ("Customer_Id")
    REFERENCES public."Customer" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "Book_Genre_Id_fk" FOREIGN KEY ("Genre_Id")
    REFERENCES public."Genre" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "Book_Language_Id_fk" FOREIGN KEY ("Language_Id")
    REFERENCES public."Language" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book"
    ADD CONSTRAINT "Book_Publisher_Id_fk" FOREIGN KEY ("Publisher_Id")
    REFERENCES public."Publisher" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Review"
    ADD CONSTRAINT "Review_Book_Id_fk" FOREIGN KEY ("Book_Id")
    REFERENCES public."Book" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Review"
    ADD CONSTRAINT "Review_Customer_Id_fk" FOREIGN KEY ("Customer_Id")
    REFERENCES public."Customer" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
```

```
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book_Author"
    ADD CONSTRAINT "Book_Author_Book_Id_fk" FOREIGN KEY ("Book_Id")
    REFERENCES public."Book" ("Id") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Book_Author"
    ADD CONSTRAINT "Book_Author_Author_Id_fk" FOREIGN KEY ("Author_Id")
    REFERENCES public."Author" ("Id") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Inventory"
    ADD CONSTRAINT "Inventory_Book_Id_fk" FOREIGN KEY ("Book_Id")
    REFERENCES public."Book" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Inventory"
    ADD CONSTRAINT "Inventory_Store_Id_fk" FOREIGN KEY ("Store_Id")
    REFERENCES public."Store" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Staff"
    ADD CONSTRAINT "Staff_Store_Id_fk" FOREIGN KEY ("Store_Id")
    REFERENCES public."Store" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Staff"
    ADD CONSTRAINT "Staff_Address_Id_fk" FOREIGN KEY ("Address_Id")
    REFERENCES public."Address" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."Address"
    ADD CONSTRAINT "Address_City_Id_fk" FOREIGN KEY ("City_Id")
    REFERENCES public."City" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
ALTER TABLE IF EXISTS public."City"
    ADD CONSTRAINT "City_Country_Id_fk" FOREIGN KEY ("Country_Id")
    REFERENCES public."Country" ("Id") MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE RESTRICT
    NOT VALID;
```

```
CREATE VIEW public."Book_List" AS
SELECT
    b."Id" AS "Book_Id",
    b."Title",
    b."Isbn",
    b."Num_Pages",
    b."Publication_Date",
    b."Price",
    g."Name" AS "Genre",
    l."Name" AS "Language",
    p."Name" AS "Publisher"
FROM
    public."Book" b
JOIN
    public."Genre" g ON b."Genre_Id" = g."Id"
JOIN
    public."Language" l ON b."Language_Id" = l."Id"
JOIN
    public."Publisher" p ON b."Publisher_Id" = p."Id";
```

```
CREATE VIEW public."Customer_List" AS
SELECT
    c."Id" AS "Customer_Id",
    c."First_Name",
    c."Last_Name",
    c."Email",
    c."Create_Date",
    a."Address",
    ci."City",
    co."Country"
FROM
    public."Customer" c
LEFT JOIN
```

```
    public."Address" a ON c."Address_Id" = a."Id"
LEFT JOIN
    public."City" ci ON a."City_Id" = ci."Id"
LEFT JOIN
    public."Country" co ON ci."Country_Id" = co."Id";
```

```
CREATE VIEW public."Staff_List" AS
SELECT
    s."Id" AS "Staff_Id",
    s."First_Name",
    s."Last_Name",
    s."Email",
    st."Name" AS "Store",
    a."Address",
    ci."City",
    co."Country"
FROM
    public."Staff" s
LEFT JOIN
    public."Store" st ON s."Store_Id" = st."Id"
LEFT JOIN
    public."Address" a ON s."Address_Id" = a."Id"
LEFT JOIN
    public."City" ci ON a."City_Id" = ci."Id"
LEFT JOIN
    public."Country" co ON ci."Country_Id" = co."Id";
```

```
CREATE VIEW public."Sales_by_Store" AS
SELECT
    st."Name" AS "Store",
    COUNT(o."Id") AS "Total_Sales",
    SUM(b."Price") AS "Total_Revenue"
FROM
    public."Order" o
JOIN
    public."Inventory" i ON o."Inventory_Id" = i."Id"
JOIN
    public."Book" b ON i."Book_Id" = b."Id"
JOIN
    public."Store" st ON i."Store_Id" = st."Id"
GROUP BY
    st."Name";
```

```
END;
```

- **SQL untuk Memasukkan Data**

```
BEGIN;

-- Insert data into Country
INSERT INTO public."Country" ("Country") VALUES
('USA'), ('Canada'), ('UK'), ('Germany'), ('France'),
('Australia'), ('India'), ('Japan'), ('China'), ('Brazil'),
('Italy'), ('Spain'), ('Russia'), ('South Korea'), ('Mexico'),
('South Africa'), ('Netherlands'), ('Sweden'), ('Norway'), ('Denmark');
```

```
-- Insert data into City
INSERT INTO public."City" ("Country_Id", "City") VALUES
(1, 'New York'), (2, 'Toronto'), (3, 'London'), (4, 'Berlin'), (5,
'Paris'),
(6, 'Sydney'), (7, 'Mumbai'), (8, 'Tokyo'), (9, 'Beijing'), (10, 'Sao
Paulo'),
(11, 'Rome'), (12, 'Madrid'), (13, 'Moscow'), (14, 'Seoul'), (15,
'Mexico City'),
(16, 'Cape Town'), (17, 'Amsterdam'), (18, 'Stockholm'), (19, 'Oslo'),
(20, 'Copenhagen');
```

```
-- Insert data into Address
INSERT INTO public."Address" ("City_Id", "Address", "Postal_Code")
VALUES
(1, '123 Main St', '10001'), (2, '456 Queen St', 'M5H 2N2'), (3, '789
King St', 'SW1A 1AA'),
(4, '321 First St', '10115'), (5, '654 Second St', '75001'), (6, '987
Third St', '2000'),
(7, '123 Fourth St', '400001'), (8, '456 Fifth St', '100-0001'), (9,
'789 Sixth St', '100000'),
(10, '321 Seventh St', '01000-000'), (11, '654 Eighth St', '00100'),
(12, '987 Ninth St', '28001'),
(13, '123 Tenth St', '101000'), (14, '456 Eleventh St', '04524'), (15,
'789 Twelfth St', '01000'),
(16, '321 Thirteenth St', '8001'), (17, '654 Fourteenth St', '1012'),
(18, '987 Fifteenth St', '100 44'),
(19, '123 Sixteenth St', '0101'), (20, '456 Seventeenth St', '1050');
```

```
-- Insert data into Customer
INSERT INTO public."Customer" ("Address_Id", "First_Name", "Last_Name",
"Email", "Create_Date", "Username", "Password") VALUES
(1, 'John', 'Doe', 'john.doe@example.com', '2021-01-01', 'johndoe',
'password1'), (2, 'Jane', 'Smith', 'jane.smith@example.com', '2021-01-
02', 'janesmith', 'password2'),
(3, 'Jim', 'Brown', 'jim.brown@example.com', '2021-01-03', 'jimbrown',
'password3'), (4, 'Jake', 'White', 'jake.white@example.com', '2021-01-
04', 'jakewhite', 'password4'),
```

```
(5, 'Jill', 'Green', 'jill.green@example.com', '2021-01-05',
'jillgreen', 'password5'), (6, 'Jack', 'Black',
'jack.black@example.com', '2021-01-06', 'jackblack', 'password6'),
(7, 'Jerry', 'Blue', 'jerry.blue@example.com', '2021-01-07',
'jerryblue', 'password7'), (8, 'Jessica', 'Yellow',
'jessica.yellow@example.com', '2021-01-08', 'jessicayellow',
'password8'),
(9, 'Jeremy', 'Purple', 'jeremy.purple@example.com', '2021-01-09',
'jeremypurple', 'password9'), (10, 'Joy', 'Orange',
'joy.orange@example.com', '2021-01-10', 'joyorange', 'password10'),
(11, 'Jordan', 'Pink', 'jordan.pink@example.com', '2021-01-11',
'jordanpink', 'password11'), (12, 'Jasmine', 'Gray',
'jasmine.gray@example.com', '2021-01-12', 'jasminegray', 'password12'),
(13, 'Jeff', 'Red', 'jeff.red@example.com', '2021-01-13', 'jeffred',
'password13'), (14, 'Julia', 'Brown', 'julia.brown@example.com', '2021-
01-14', 'juliabrown', 'password14'),
(15, 'Jon', 'White', 'jon.white@example.com', '2021-01-15', 'jonwhite',
'password15'), (16, 'Judy', 'Green', 'judy.green@example.com', '2021-
01-16', 'judygreen', 'password16'),
(17, 'Joe', 'Black', 'joe.black@example.com', '2021-01-17', 'joeblack',
'password17'), (18, 'Jenny', 'Blue', 'jenny.blue@example.com', '2021-
01-18', 'jennyblue', 'password18'),
(19, 'Justin', 'Yellow', 'justin.yellow@example.com', '2021-01-19',
'justinyellow', 'password19'), (20, 'Janet', 'Purple',
'janet.purple@example.com', '2021-01-20', 'janetpurple', 'password20');
```

```
-- Insert data into Publisher
INSERT INTO public."Publisher" ("Name") VALUES
('Penguin'), ('HarperCollins'), ('Macmillan'), ('Simon & Schuster'),
('Hachette'),
('Scholastic'), ('Pearson'), ('Random House'), ('Oxford University
Press'), ('Cambridge University Press'),
('Bloomsbury'), ('Wiley'), ('Cengage'), ('Springer'), ('Taylor &
Francis'),
('Sage'), ('McGraw-Hill'), ('Elsevier'), ('Thomson Reuters'),
('Routledge');
```

```
-- Insert data into Author
INSERT INTO public."Author" ("Name") VALUES
('Stephen King'), ('J.K. Rowling'), ('J.R.R. Tolkien'), ('George R.R.
Martin'), ('Agatha Christie'),
('James Patterson'), ('Mark Twain'), ('Ernest Hemingway'), ('Jane
Austen'), ('Charles Dickens'),
('Leo Tolstoy'), ('Fyodor Dostoevsky'), ('Gabriel Garcia Marquez'),
('Harper Lee'), ('F. Scott Fitzgerald'),
('H.G. Wells'), ('Arthur Conan Doyle'), ('Isaac Asimov'), ('Hermann
Hesse'), ('Franz Kafka');
```

```
-- Insert data into Genre
```

```
INSERT INTO public."Genre" ("Name") VALUES
('Fiction'), ('Non-Fiction'), ('Mystery'), ('Thriller'), ('Romance'),
('Science Fiction'), ('Fantasy'), ('Biography'), ('History'),
('Poetry'),
('Children'), ('Young Adult'), ('Horror'), ('Self-Help'), ('Health'),
('Travel'), ('Science'), ('Religion'), ('Humor'), ('Business');
```

```
-- Insert data into Language
INSERT INTO public."Language" ("Name") VALUES
('English'), ('French'), ('German'), ('Spanish'), ('Chinese'),
('Japanese'), ('Russian'), ('Portuguese'), ('Italian'), ('Dutch'),
('Swedish'), ('Korean'), ('Hindi'), ('Arabic'), ('Bengali'),
('Turkish'), ('Vietnamese'), ('Thai'), ('Greek'), ('Czech');
```

```
-- Insert data into Book
INSERT INTO public."Book" ("Genre_Id", "Language_Id", "Publisher_Id",
"Title", "Isbn", "Num_Pages", "Publication_Date", "Price") VALUES
(1, 1, 1, 'The Shining', '9780450040184', 447, '1977-01-28', 20), (2, 1,
2, 'Harry Potter and the Philosopher''s Stone', '9780747532699', 223,
'1997-06-26', 25),
(3, 1, 3, 'The Lord of the Rings', '9780261102385', 1216, '1954-07-29',
30), (4, 1, 4, 'A Game of Thrones', '9780553103540', 694, '1996-08-06',
35),
(5, 1, 5, 'Murder on the Orient Express', '9780062073501', 256, '1934-
01-01', 15), (6, 1, 6, 'Along Came a Spider', '9780316693646', 435,
'1993-01-01', 20),
(7, 1, 7, 'The Adventures of Tom Sawyer', '9780143039563', 274, '1876-
01-01', 10), (8, 1, 8, 'The Old Man and the Sea', '9780684830490', 132,
'1952-01-01', 15),
(9, 1, 9, 'Pride and Prejudice', '9780141439518', 279, '1813-01-28',
15), (10, 1, 10, 'A Tale of Two Cities', '9780141439600', 489, '1859-
01-01', 20),
(11, 1, 11, 'War and Peace', '9780140447934', 1225, '1869-01-01', 25),
(12, 1, 12, 'Crime and Punishment', '9780140449136', 671, '1866-01-01',
20),
(13, 1, 13, 'One Hundred Years of Solitude', '9780060883287', 417,
'1967-01-01', 18), (14, 1, 14, 'To Kill a Mockingbird', '9780061120084',
281, '1960-01-01', 18),
(15, 1, 15, 'The Great Gatsby', '9780743273565', 180, '1925-01-01', 10),
(16, 1, 16, 'The War of the Worlds', '9780141441030', 288, '1898-01-01',
15),
(17, 1, 17, 'The Hound of the Baskervilles', '9780141032431', 256,
'1902-01-01', 12), (18, 1, 18, 'Foundation', '9780553293357', 244,
'1951-01-01', 15),
(19, 1, 19, 'Steppenwolf', '9780142437186', 240, '1927-01-01', 14), (20,
1, 20, 'The Trial', '9780805209990', 304, '1925-01-01', 15);
```

```
-- Insert data into Book_Author
INSERT INTO public."Book_Author" ("Book_Id", "Author_Id") VALUES
```

```
(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8), (9, 9),
(10, 10),
(11, 11), (12, 12), (13, 13), (14, 14), (15, 15), (16, 16), (17, 17),
(18, 18), (19, 19), (20, 20);
```

```
-- Insert data into Store
INSERT INTO public."Store" ("Name") VALUES
('Store A'), ('Store B'), ('Store C'), ('Store D'), ('Store E'),
('Store F'), ('Store G'), ('Store H'), ('Store I'), ('Store J'),
('Store K'), ('Store L'), ('Store M'), ('Store N'), ('Store O'),
('Store P'), ('Store Q'), ('Store R'), ('Store S'), ('Store T');
```

```
-- Insert data into Inventory
INSERT INTO public."Inventory" ("Book_Id", "Store_Id", "Stock") VALUES
(1, 1, 50), (2, 2, 60), (3, 3, 40), (4, 4, 30), (5, 5, 70),
(6, 6, 20), (7, 7, 90), (8, 8, 10), (9, 9, 80), (10, 10, 55),
(11, 11, 65), (12, 12, 35), (13, 13, 45), (14, 14, 75), (15, 15, 25),
(16, 16, 95), (17, 17, 5), (18, 18, 85), (19, 19, 15), (20, 20, 50);
```

```
-- Insert data into Staff
INSERT INTO public."Staff" ("Address_Id", "Store_Id", "First_Name",
"Last_Name", "Email") VALUES
(1, 1, 'Alice', 'Anderson', 'alice.anderson@example.com'), (2, 2, 'Bob',
'Brown', 'bob.brown@example.com'),
(3, 3, 'Charlie', 'Clark', 'charlie.clark@example.com'), (4, 4, 'Daisy',
'Davis', 'daisy.davis@example.com'),
(5, 5, 'Edward', 'Evans', 'edward.evans@example.com'), (6, 6, 'Fiona',
'Fisher', 'fiona.fisher@example.com'),
(7, 7, 'George', 'Green', 'george.green@example.com'), (8, 8, 'Hannah',
'Hill', 'hannah.hill@example.com'),
(9, 9, 'Ian', 'Irvine', 'ian.irvine@example.com'), (10, 10, 'Jenny',
'Jones', 'jenny.jones@example.com'),
(11, 11, 'Kevin', 'King', 'kevin.king@example.com'), (12, 12, 'Laura',
'Lewis', 'laura.lewis@example.com'),
(13, 13, 'Michael', 'Martin', 'michael.martin@example.com'), (14, 14,
'Nancy', 'Nelson', 'nancy.nelson@example.com'),
(15, 15, 'Oscar', 'Owen', 'oscar.owen@example.com'), (16, 16, 'Paul',
'Parker', 'paul.parker@example.com'),
(17, 17, 'Quinn', 'Quinn', 'quinn.quinn@example.com'), (18, 18,
'Rachel', 'Roberts', 'rachel.roberts@example.com'),
(19, 19, 'Steve', 'Smith', 'steve.smith@example.com'), (20, 20, 'Tina',
'Turner', 'tina.turner@example.com');
```

```
-- Insert data into Order
INSERT INTO public."Order" ("Inventory_Id", "Customer_Id", "Date")
VALUES
(1, 1, '2021-02-01'), (2, 2, '2021-02-02'), (3, 3, '2021-02-03'), (4, 4,
'2021-02-04'), (5, 5, '2021-02-05'),
```

```
(6, 6, '2021-02-06'), (7, 7, '2021-02-07'), (8, 8, '2021-02-08'), (9, 9,
'2021-02-09'), (10, 10, '2021-02-10'),
(11, 11, '2021-02-11'), (12, 12, '2021-02-12'), (13, 13, '2021-02-13'),
(14, 14, '2021-02-14'), (15, 15, '2021-02-15'),
(16, 16, '2021-02-16'), (17, 17, '2021-02-17'), (18, 18, '2021-02-18'),
(19, 19, '2021-02-19'), (20, 20, '2021-02-20');
```

```
-- Insert data into Review
INSERT INTO public."Review" ("Book_Id", "Customer_Id", "Rating",
"Review_Details", "Date") VALUES
(1, 1, 5, 'Amazing book!', '2021-03-01'), (2, 2, 4, 'Great read.',
'2021-03-02'), (3, 3, 5, 'Loved it!', '2021-03-03'),
(4, 4, 3, 'Good but long.', '2021-03-04'), (5, 5, 4, 'Well written.',
'2021-03-05'), (6, 6, 5, 'Thrilling!', '2021-03-06'),
(7, 7, 4, 'Classic.', '2021-03-07'), (8, 8, 3, 'Too short.', '2021-03-
08'), (9, 9, 4, 'Engaging.', '2021-03-09'),
(10, 10, 5, 'A masterpiece.', '2021-03-10'), (11, 11, 4, 'Impressive.',
'2021-03-11'), (12, 12, 5, 'Thought-provoking.', '2021-03-12'),
(13, 13, 5, 'Magical realism at its best.', '2021-03-13'), (14, 14, 5,
'Heartfelt.', '2021-03-14'), (15, 15, 4, 'Captivating.', '2021-03-15'),
(16, 16, 3, 'Outdated.', '2021-03-16'), (17, 17, 4, 'Intriguing.',
'2021-03-17'), (18, 18, 5, 'Sci-fi classic.', '2021-03-18'),
(19, 19, 4, 'Deep and philosophical.', '2021-03-19'), (20, 20, 5,
'Kafkaesque.', '2021-03-20');
```

```
END;
```

- **Contoh Penggunaan TCL**

```
BEGIN TRANSACTION;

INSERT INTO public."Language" ("Name") VALUES ('Indonesian');
SELECT * FROM public."Language";
```

Id	Name
Filter...	Filter...
1	English
2	French
3	German
4	Spanish
5	Chinese
6	Japanese
7	Russian
8	Portuguese
9	Italian
10	Dutch
11	Swedish
12	Korean
13	Hindi
14	Arabic
15	Bengali
16	Turkish
17	Vietnamese
18	Thai
19	Greek
20	Czech
21	Indonesian

Menggunakan ROLLBACK

```
ROLLBACK;
SELECT * FROM public."Language";
```

Id	Name
Filter...	Filter...
1	English
2	French
3	German
4	Spanish
5	Chinese
6	Japanese
7	Russian
8	Portuguese
9	Italian
10	Dutch
11	Swedish
12	Korean
13	Hindi
14	Arabic
15	Bengali
16	Turkish
17	Vietnamese
18	Thai
19	Greek
20	Czech

Menyimpan Perubahan

```
COMMIT;
```