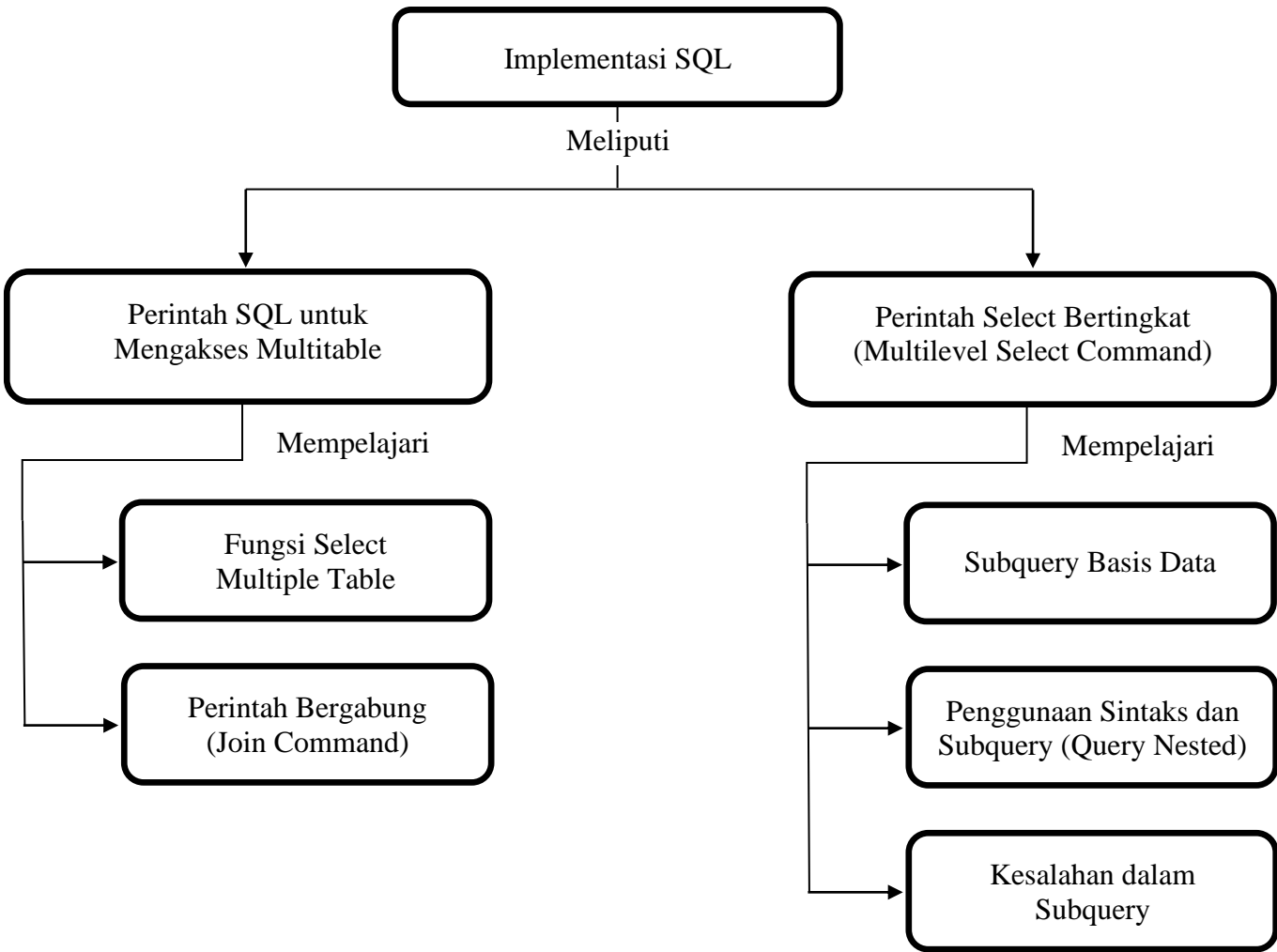


BAB I

IMPLEMENTASI SQL

Peta Konsep



Dengan adanya hubungan (*relationship*) antara satu tabel dengan tabel yang lain, maka perintah dasar MySQL akan terus berkembang sesuai dengan kebutuhan. Misalnya, menampilkan data menjadi hal yang sangat penting karena harus melihat dan menyeleksi data dalam tabel maupun antar tabel. Oleh sebab itu, seorang *Database Administrator* (DBA) membuat beberapa *query* ke *database* untuk prosedur pencocokan mengambil data pada tabel ke-1, ke-2, ke-3, ke-4, ke-5, dan seterusnya. Dengan acuan data pada tabel ke 6, maka dapat dibuat *query* secara berulang. Jika dilihat sekilas, metode tersebut terlihat kurang efektif karena terlalu banyak kode yang harus ditulis, meskipun bisa menggunakan satu *query* saja. Guna melihat data (*selection*) dapat menggunakan *query SELECT* diikuti beberapa pernyataan khusus yang berhubungan dengan tabel yang diseleksi.

A. Perintah SQL untuk Mengakses Multitable

Pada dasarnya, *SELECT* merupakan perintah DML (*Data Manipulation Language*) untuk menghasilkan *record set* dengan mengekstrak data dari *database*. Perintah *SELECT* sering dipakai dalam sebuah *database* RDBMS (*Relational Database Management System*) dan memiliki bentuk yang kompleks. Format perintah *SELECT* yang digunakan adalah sebagai berikut.

```
SELECT select_list [INTO new_table] FROM table_source [WHERE search_condition]
[GROUP BY group_by_expression] [HAVING search_condition] [ORDER BY
order_expression [ASC | DESC] ].
```

Misalnya, kita memiliki tabel **dt_mobil** dengan

B. Perintah Select Bertingkat (*Multilevel Select Command*)

- 1. Setelah di analisa tempalet FrontEnd (EShopper) dapat dibuat rancangan database Blue Print (Class Diagram) untuk BackEnd seperti Gambar berikut:

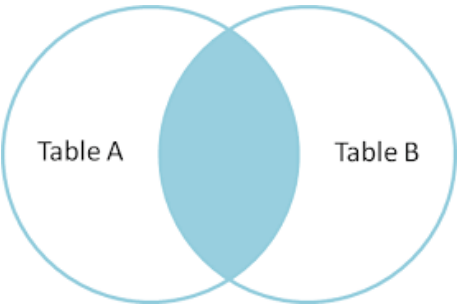
2. Perintah Join (*Join Command*)

Pada umumnya, mengambil data bisa dilakukan dengan cara mengakses beberapa table secara bersamaan. Salah satu metode yang digunakan untuk mengambil data dari beberapa tabel sekaligus dapat menggunakan perintah *join*. Perintah *join* (*join command*) pada SQL berfungsi menghubungkan tabel yang satu dengan tabel yang lain sehingga saling berhubungan atau berelasi satu sama lain. Hal ini bertujuan untuk menampilkan beberapa data dari tabel yang berbeda dengan menggunakan satu perintah. Perintah *join* diklasifikasikan menjadi sebagai berikut.

a. Perintah *Inner Join*

Data yang akan ditampilkan pada *inner join* hanya data yang memiliki pasangan saja, sedangkan data pada tabel yang tidak memiliki sebuah kesamaan maka data tersebut tidak akan ditampilkan. *Inner Join* identik dengan sebuah perintah yang digunakan untuk menampilkan sebuah data atau *record* dengan menghubungkan dua tabel atau lebih dalam satu perintah. Tabel akan digabungkan dua arah menggunakan *inner join*, sehingga tidak ada data yang *NULL* di satu sisi. Format penulisan yang digunakan adalah

```
SELECT column FROM TableP INNER JOIN TableQ ON P.columnName = Q.columnName;
```



Misalnya, menggabungkan tabel *dt_mobil* dan *dt_servis* dengan tujuan untuk menampilkan daftar *nm_pelanggan* yang pernah melakukan perbaikan (*servis*).

1) Penggabungan dengan *where*

Format penggabungan dengan *WHERE* menggunakan perintah:
SELECT tabelP.*, tabelQ.* FROM tabelP, tabelQ WHERE tabelP.PK=tabelQ.FK;
Misalnya, perintah SQL yang digunakan untuk menggabungkan tabel *dt_mobil* dan *dt_servis* sebagai berikut:

```
SELECT dt_mobil.nopol, dt_mobil.nm_pelanggan, dt_servis.id_srv,  
dt_servis.tgl_servis FROM dt_mobil, dt_servis WHERE  
dt_mobil.nopol=dt_servis.nopol;
```

Hasilnya sebagai berikut:

```
mysql> select * from dt_mobil;
+-----+-----+-----+
| nopol | nm_pelanggan | merek |
+-----+-----+-----+
| K 11 PWF | Oktyawati | Toyota |
| N 1 SA | Maheswari | Mercedes Benz |
| WA 2 N | Diprayogi LS | Mitsubishi |
| DW 1 WJT | Rangga Haris | Honda |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from dt_servis;
+-----+-----+-----+-----+-----+-----+
| id_srv | nopol | tgl_servis | model | tipe | warna |
+-----+-----+-----+-----+-----+-----+
| 1 | K 11 PWF | 2019-04-02 | FORTUNER | SUV | GREY |
| 2 | N 1 SA | 2019-04-05 | AMG GTR | SPORT | GREY |
| 3 | N 1 SA | 2019-04-10 | AMG GTR | SPORT | GREY |
| 4 | WA 2 N | 2019-01-20 | CITY | SEDAN | HITAM |
| 5 | DW 1 WJT | 2018-12-14 | FORTUNER | SUV | GREY |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT dt_mobil.nopol, dt_mobil.nm_pelanggan, dt_servis.id_srv, dt_servis
.tgl_servis FROM dt_mobil, dt_servis WHERE dt_mobil.nopol=dt_servis.nopol;
+-----+-----+-----+-----+
| nopol | nm_pelanggan | id_srv | tgl_servis |
+-----+-----+-----+-----+
| K 11 PWF | Oktyawati | 1 | 2019-04-02 |
| N 1 SA | Maheswari | 2 | 2019-04-05 |
| N 1 SA | Maheswari | 3 | 2019-04-10 |
| WA 2 N | Diprayogi LS | 4 | 2019-01-20 |
| DW 1 WJT | Rangga Haris | 5 | 2018-12-14 |
+-----+-----+-----+-----+
5 rows in set (0.37 sec)
```

Berdasarkan hasil query diatas, terdapat lima transaksi servis yang dilakukan oleh tiga mobil **dt_mobil**. Namun, terdapat satu mobil yaitu **dt_mobil WA 2 N** tidak ditampilkan, karena belum pernah melakukan transaksi servis.

2) Penggabungan dengan *Inner Join*

Format penggabungan dengan WHERE menggunakan perintah:
SELECT tabelP.*, tabelQ.* FROM tabelP INNER JOIN tabelQ ON tabelP.PK=tabelQ.FK;
Misalnya, perintah SQL yang digunakan untuk menggabungkan tabel **dt_mobil** dan **dt_servis** sebagai berikut:

```
SELECT dt_mobil.nopol, dt_mobil.nm_pelanggan, dt_servis.id_srv,
dt_servis.tgl_servis FROM dt_mobil INNER JOIN dt_servis ON
dt_mobil.nopol=dt_servis.nopol;
```

Hasilnya adalah sama seperti penggabungan dengan *WHERE*. Perhatikan gambar berikut:

```
mysql> SELECT dt_mobil.nopol, dt_mobil.nm_pelanggan, dt_servis.id_srv, dt_servis.tgl_servis FROM
dt_mobil INNER JOIN dt_servis ON dt_mobil.nopol=dt_servis.nopol;
+-----+-----+-----+-----+
| nopol | nm_pelanggan | id_srv | tgl_servis |
+-----+-----+-----+-----+
| K 11 PWF | Oktyawati | 1 | 2019-04-02 |
| N 1 SA | Maheswari | 2 | 2019-04-05 |
| N 1 SA | Maheswari | 3 | 2019-04-10 |
| WA 2 N | Diprayogi LS | 4 | 2019-01-20 |
| DW 1 WJT | Rangga Haris | 5 | 2018-12-14 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```