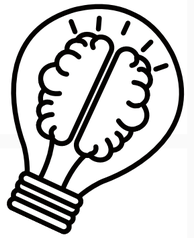
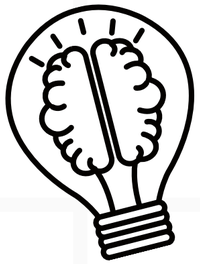
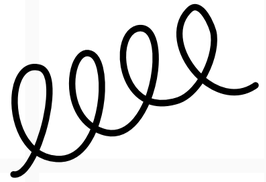


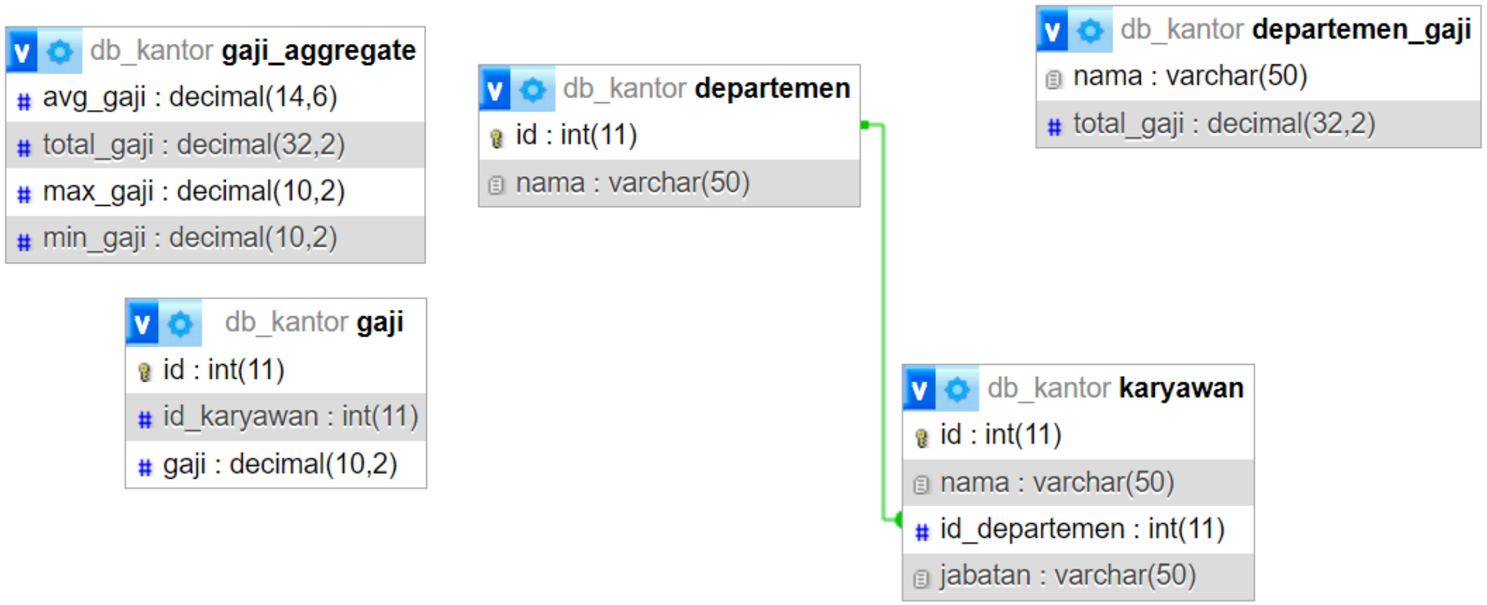
DATABASE

Gajih kantor





PEMBUATAN ENTITY DIAGRAM



PEMBUATAN TRIGGER DAN VIEW

```
    ->   WHERE id_karyawan = NEW.id;
    -> END//
Query OK, 0 rows affected (0.019 sec)

MariaDB [db_kantor]> DELIMITER ;
MariaDB [db_kantor]> CREATE VIEW departemen_gaji AS
    -> SELECT d.nama, SUM(g.gaji) AS total_gaji
    -> FROM departemen d
    -> JOIN karyawan k ON d.id = k.id_departemen
    -> JOIN gaji g ON k.id = g.id_karyawan
    -> GROUP BY d.nama;
Query OK, 0 rows affected (0.014 sec)

MariaDB [db_kantor]> SELECT * FROM view departemen_gaji;
ERROR 1146 (42S02): Table 'db_kantor.view' doesn't exist
MariaDB [db_kantor]> SELECT k.nama, d.nama AS nama_departemen
    -> FROM karyawan k
    -> INNER JOIN departemen d ON k.id_departemen = d.id;
+-------------+-----------------+
| nama        | nama_departemen |
+-------------+-----------------+
| John Doe    | IT              |
| Jane Smith  | Marketing       |
| Bob Johnson | IT              |
| Alice Brown | Riset           |
| Mike Davis  | Marketing       |
+-------------+-----------------+
5 rows in set (0.002 sec)
```

PEMBUATAN AGREGAT

```
MariaDB [db_kantor]> SELECT
    ->     d.nama AS nama_departemen,
    ->     COUNT(k.id) AS jumlah_karyawan,
    ->     SUM(g.gaji) AS total_gaji,
    ->     AVG(g.gaji) AS rata_rata_gaji,
    ->     MAX(g.gaji) AS gaji_maksimum,
    ->     MIN(g.gaji) AS gaji_minimum
    -> FROM
    ->     karyawan k, departemen d, gaji g
    -> WHERE
    ->     k.id_departemen = d.id AND
    ->     k.id = g.id_karyawan
    -> GROUP BY
    ->     d.nama;
+-----------------+-----------------+-----------+----------------+--------------+--------------+
| nama_departemen | jumlah_karyawan | total_gaji | rata_rata_gaji | gaji_maksimum | gaji_minimum |
+-----------------+-----------------+-----------+----------------+--------------+--------------+
| IT              |               2 |  170000.00 |   85000.000000 |      90000.00 |     80000.00 |
| Marketing       |               2 |  190000.00 |   95000.000000 |     120000.00 |     70000.00 |
| Riset           |               1 |  110000.00 |  110000.000000 |     110000.00 |    110000.00 |
+-----------------+-----------------+-----------+----------------+--------------+--------------+
3 rows in set (0.002 sec)
```

PEMBUATAN INNER JOIN

```
MariaDB [db_kantor]> SELECT
    ->     karyawan.id AS karyawan_id,
    ->     karyawan.nama AS nama_karyawan,
    ->     departemen.nama AS nama_departemen,
    ->     karyawan.jabatan AS jabatan,
    ->     gaji.gaji AS gaji
    -> FROM
    ->     karyawan
    -> INNER JOIN
    ->     departemen ON karyawan.id_departemen = departemen.id
    -> INNER JOIN
    ->     gaji ON karyawan.id = gaji.id_karyawan;
+-------------+---------------+-----------------+---------------------+-----------+
| karyawan_id | nama_karyawan | nama_departemen | jabatan             | gaji      |
+-------------+---------------+-----------------+---------------------+-----------+
|           1 | John Doe      | IT              | Software Engineer   |  80000.00 |
|           3 | Bob Johnson   | IT              | DevOps Engineer     |  90000.00 |
|           2 | Jane Smith    | Marketing       | Marketing Manager   | 120000.00 |
|           5 | Mike Davis    | Marketing       | Sales Representative |  70000.00 |
|           4 | Alice Brown   | Riset           | Data Scientist      | 110000.00 |
+-------------+---------------+-----------------+---------------------+-----------+
5 rows in set (0.001 sec)
```

PEMBUATAN LEFT JOIN

```
MariaDB [db_kantor]> SELECT k.nama, d.nama AS nama_departemen
    -> FROM karyawan k
    -> LEFT JOIN departemen d ON k.id_departemen = d.id;
+-------------+-----------------+
| nama        | nama_departemen |
+-------------+-----------------+
| John Doe    | IT              |
| Jane Smith  | Marketing       |
| Bob Johnson | IT              |
| Alice Brown | Riset           |
| Mike Davis  | Marketing       |
+-------------+-----------------+
5 rows in set (0.007 sec)
```

PEMBUATAN SUBQUERY

```
MariaDB [db_kantor]> SELECT
    ->     id,
    ->     nama,
    ->     jabatan
    -> FROM
    ->     karyawan
    -> WHERE
    ->     id_departemen = (
    ->         SELECT id
    ->         FROM departemen
    ->         WHERE nama = 'IT'  -- Ganti dengan nama departemen yang diinginkan
    ->     );
+----+-------------+------------------+
| id | nama        | jabatan          |
+----+-------------+------------------+
|  1 | John Doe    | Software Engineer |
|  3 | Bob Johnson | DevOps Engineer   |
+----+-------------+------------------+
2 rows in set (0.001 sec)
```

PEMBUATAN HAVING

```
MariaDB [db_kantor]> SELECT d.nama, SUM(g.gaji) AS total_gaji
    -> FROM departemen d
    -> JOIN karyawan k ON d.id = k.id_departemen
    -> JOIN gaji g ON k.id = g.id_karyawan
    -> GROUP BY d.nama
    -> HAVING SUM(g.gaji) > 200000000;
Empty set (0.007 sec)
```

PEMBUATAN WILDCRAD

```
MariaDB [db_kantor]> SELECT *
    -> FROM karyawan k
    -> WHERE k.id IN (
    ->   SELECT g.id_karyawan
    ->   FROM gaji g
    ->   WHERE g.gaji > (
    ->     SELECT AVG(gaji)
    ->     FROM gaji
    ->   )
    -> );
+----+-------------+---------------+------------------+
| id | nama        | id_departemen | jabatan          |
+----+-------------+---------------+------------------+
|  2 | Jane Smith  |             2 | Marketing Manager |
|  4 | Alice Brown |             3 | Data Scientist    |
+----+-------------+---------------+------------------+
2 rows in set (0.007 sec)
```

BACKUP DENGAN DUMP

mysqldump adalah alat baris perintah yang disediakan dengan server MariaDB atau MySQL untuk melakukan backup atau dumping data.

REPLIKASI DATA

Replikasi data adalah proses menduplikasi data dari satu database (atau beberapa database) ke database lain secara real-time atau berdasarkan jadwal tertentu. Tujuan utama dari replikasi data adalah meningkatkan ketersediaan data, meningkatkan kinerja sistem, dan memungkinkan redundansi untuk keamanan dan pemulihan bencana.