**LAPORAN DESAIN DATABASE SIAKAD BASIS DATA II**

**Dosen Pengampu :**

**Nama : Moch Kautsar Sophan, S.Kom, M.MT NIP : 197707132002121004**

****

|  |  |
| --- | --- |
| **Disusun Oleh :** |  |
| **Harits Putra Junaidi** | **230411100003** |
| **Achmad Lutfi Madhani** | **230411100059** |
| **Danendra Mahardhika** | **230411100086** |
| **Ahmad Ubaydir Rohman** | **230411100116** |
| **Moh Naufal Thariq** | **230411100142** |
| **Seinal Arifin** | **230411100149** |

**PRODI TEKNIK INFORMATIKA JURUSAN TEKNIK INFORMATIKA FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA 2025**

1. **. informasi server  
    . diagram database (PDM), pastikan gambar jelas   
    . hasil running sql untuk**
2. **.. menampilkan jumlah record mahasiswa  
    .. menampilkan jumlah record mahasiswa per tahun  
    .. menampilkan jumlah record krs  
    .. menampilkan jumlah record krs per semester  
    .. menampilkan jumlah record nilai  
    .. menampilkan jumlah record nilai per MK**
3. **jelaskan  
   . stored procedure yang di buat   
   . stored function yang dibuat  
   . trigger yang di buat­**
   * **Stored procedure untuk generate krs\_detail**

BEGIN

DECLARE kelas\_id INT; DECLARE sks\_mata\_kuliah INT;

DECLARE total\_sks\_temp INT DEFAULT 0; DECLARE max\_sks INT DEFAULT 24;

DECLARE target\_sks INT DEFAULT 20; DECLARE done INT DEFAULT FALSE;

DECLARE kelas\_cursor CURSOR FOR SELECT k.id\_kelas, mk.sks

FROM kelas k

JOIN mata\_kuliah mk ON k.id\_mata\_kuliah = mk.id\_mata\_kuliah WHERE k.id\_tahun\_akademik = ta\_id

AND mk.id\_prodi = prodi\_id AND mk.semester = semester\_mhs AND mk.status = 'Aktif'

AND k.status = 'Aktif' ORDER BY RAND();

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN kelas\_cursor; kelas\_loop: LOOP

FETCH kelas\_cursor INTO kelas\_id, sks\_mata\_kuliah; IF done THEN

LEAVE kelas\_loop; END IF;

IF (total\_sks\_temp + sks\_mata\_kuliah) <= max\_sks THEN INSERT IGNORE INTO krs\_detail (

id\_krs, id\_kelas, status

) VALUES (

krs\_id, kelas\_id, 'Aktif'

);

IF ROW\_COUNT() > 0 THEN

SET total\_sks\_temp = total\_sks\_temp + sks\_mata\_kuliah; END IF;

END IF;

IF total\_sks\_temp >= target\_sks THEN LEAVE kelas\_loop;

END IF; END LOOP;

CLOSE kelas\_cursor;

UPDATE krs SET total\_sks = total\_sks\_temp WHERE id\_krs = krs\_id;

END

* **UAmbil kelas aktif dari prodi dan semester yang sesuai.**
* **Tambahkan kelas ke detail KRS sampai jumlah SKS mendekati target (20 SKS, maksimal 24 SKS).**
* **Pastikan total SKS tidak melebihi batas.**
  + **Stored procedure untuk generate mahasiswa**

# BEGIN

DECLARE tahun INT; DECLARE prodi INT; DECLARE i INT;

DECLARE counter INT DEFAULT 1;

DECLARE dosen\_id INT; DECLARE min\_dosen\_id INT; DECLARE max\_dosen\_id INT; DECLARE dosen\_count INT;

SET tahun = 2019; WHILE tahun <= 2024 DO

SET prodi = 1; WHILE prodi <= 5 DO

-- Cek apakah ada dosen untuk prodi ini

SELECT COUNT(\*) INTO dosen\_count FROM dosen WHERE id\_prodi = prodi;

IF dosen\_count > 0 THEN

-- Dapatkan ID dosen minimum dan maksimum untuk prodi ini

SELECT MIN(id\_dosen) INTO min\_dosen\_id FROM dosen WHERE id\_prodi = prodi; SELECT MAX(id\_dosen) INTO max\_dosen\_id FROM dosen WHERE id\_prodi = prodi;

SET i = 1;

WHILE i <= 1000 DO

-- Hitung ID dosen dengan cara yang lebih sederhana

-- Gunakan formula sederhana untuk distribusi merata SET dosen\_id = min\_dosen\_id + ((i - 1) % dosen\_count);

-- Verifikasi apakah dosen\_id valid

SELECT COUNT(\*) INTO @dosen\_exists FROM dosen WHERE id\_dosen = dosen\_id AND id\_prodi = prodi;

-- Jika tidak valid, gunakan min\_dosen\_id sebagai fallback IF @dosen\_exists = 0 THEN

SET dosen\_id = min\_dosen\_id; END IF;

INSERT INTO mahasiswa ( nim, nama\_lengkap, id\_prodi, tahun\_masuk, email,

status, jenis\_kelamin, tanggal\_lahir, dosen\_wali

# ) VALUES (

CONCAT(tahun, LPAD(prodi, 2, '0'), LPAD(i, 4, '0')),

CONCAT('Mahasiswa\_', tahun, '\_', counter), prodi,

tahun,

CONCAT('mahasiswa', LPAD(counter, 5, '0'), '@example.com'), 'Aktif',

IF(MOD(counter, 2) = 0, 'P', 'L'),

# DATE\_SUB(CURDATE(), INTERVAL (18 + FLOOR(RAND() \* 5)) YEAR),

dosen\_id

);

SET i = i + 1;

SET counter = counter + 1; END WHILE;

# END IF;

SET prodi = prodi + 1; END WHILE;

SET tahun = tahun + 1; END WHILE;

END

* **Loop untuk beberapa tahun dan prodi.**
* **Untuk setiap prodi, cek dosen aktif.**
* **Generate data mahasiswa (NIM, nama, email, jenis kelamin, tanggal lahir).**
* **Assign dosen wali secara merata menggunakan mod dan rentang ID dosen.**
  + **Stored procedure untuk generate mata kuliah**

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE v\_id\_prodi INT;

DECLARE v\_kode\_prodi VARCHAR(10); DECLARE v\_jumlah\_semester INT DEFAULT 8; DECLARE i INT;

DECLARE j INT;

-- Cursor untuk semua prodi aktif DECLARE cur CURSOR FOR

SELECT id\_prodi, kode\_prodi FROM program\_studi WHERE status = 'Aktif';

-- Handler jika tidak ada data lagi

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur; read\_loop: LOOP

FETCH cur INTO v\_id\_prodi, v\_kode\_prodi; IF done THEN

LEAVE read\_loop; END IF;

SET i = 1;

WHILE i <= v\_jumlah\_semester DO SET j = 1;

WHILE j <= 6 DO

INSERT INTO mata\_kuliah ( id\_prodi, kode\_mata\_kuliah, nama\_mata\_kuliah, sks,

semester, jenis, status

) VALUES (

v\_id\_prodi,

CONCAT(v\_kode\_prodi, LPAD(((i - 1) \* 6 + j), 3, '0')),

CONCAT('Mata Kuliah ', ((i - 1) \* 6 + j)), 3,

i, 'Wajib', 'Aktif'

);

SET j = j + 1; END WHILE; SET i = i + 1;

END WHILE; END LOOP;

CLOSE cur;

END

* **Untuk setiap prodi aktif, buat mata kuliah sebanyak 6 per semester.**
* **Total 8 semester, sehingga total mata kuliah yang dibuat 48 per prodi.**
* **Kode mata kuliah dan nama disusun otomatis sesuai nomor urut.**
  + **Stored procedure untuk generate nilai**

# BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE krs\_detail\_id INT; DECLARE t DECIMAL(5,2); DECLARE uts DECIMAL(5,2); DECLARE uas DECIMAL(5,2); DECLARE akhir DECIMAL(5,2); DECLARE g VARCHAR(2);

DECLARE cur CURSOR FOR

SELECT id\_krs\_detail FROM krs\_detail;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur; krs\_loop: LOOP

FETCH cur INTO krs\_detail\_id; IF done THEN

LEAVE krs\_loop; END IF;

SET t = ROUND(RAND() \* 25 + 75, 2); -- tugas: 75–100

SET uts = ROUND(RAND() \* 30 + 60, 2); -- uts: 60–90

SET uas = ROUND(RAND() \* 30 + 60, 2); -- uas: 60–90

SET akhir = ROUND((t \* 0.3 + uts \* 0.3 + uas \* 0.4), 2);

-- Penentuan grade

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SET g | = CASE |  | | | |
|  | WHEN akhir | >= | 85 | THEN | 'A' |
|  | WHEN akhir | >= | 80 | THEN | 'A-' |
|  | WHEN akhir | >= | 75 | THEN | 'B+' |
|  | WHEN akhir | >= | 70 | THEN | 'B' |
|  | WHEN akhir | >= | 65 | THEN | 'B-' |
|  | WHEN akhir | >= | 60 | THEN | 'C+' |
|  | WHEN akhir | >= | 50 | THEN | 'C' |
|  | WHEN akhir | >= | 40 | THEN | 'D' |
|  | ELSE 'E' |  |  |  |  |
| END; |  |  |  |  |  |

-- Insert nilai INSERT INTO nilai (

id\_krs\_detail, nilai\_tugas, nilai\_uts, nilai\_uas, nilai\_akhir, grade

# ) VALUES (

krs\_detail\_id,

t, uts, uas, akhir, g

);

# END LOOP;

CLOSE cur;

END

* **Ambil semua id\_krs\_detail dari tabel krs\_detail.**
* **Untuk tiap data:**
* **Buat nilai acak:**
* **Tugas: antara 75–100**
* **UTS & UAS: antara 60–90**
* **Hitung nilai akhir:**

**tugas 30% + UTS 30% + UAS 40%**

* **Tentukan grade berdasarkan nilai akhir (A, B+, C, dst).**
* **Simpan semua nilai ke tabel nilai.**
  + **Stored procedure untuk generate ruangan**

BEGIN

DECLARE prodi\_id INT; DECLARE ruang\_counter INT;

DECLARE done INT DEFAULT FALSE;

-- Hapus string concatenation dan LIKE operator

-- untuk menghindari masalah collation SET ruang\_counter = 1;

WHILE ruang\_counter <= 200 DO INSERT IGNORE INTO ruangan (

nama\_ruangan, gedung, lantai, kapasitas, status

) VALUES (

CONCAT('Ruang-', ruang\_counter), CONCAT('Gedung-', CEIL(ruang\_counter / 50)),

1 + (ruang\_counter % 4), CASE

WHEN RAND() < 0.3 THEN 30

WHEN RAND() < 0.7 THEN 40

ELSE 50 END,

'Aktif'

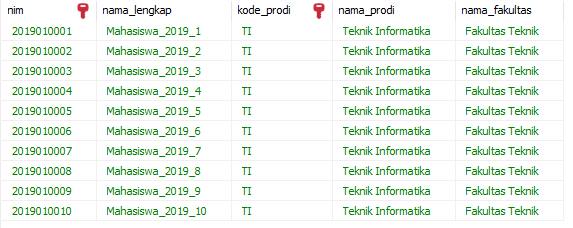
);

SET ruang\_counter = ruang\_counter + 1; END WHILE;

END

* **Buat 200 ruangan dengan nama dan gedung yang berurutan.**
* **Kapasitas ruangan dipilih secara acak (30, 40, atau 50).**
* **Lantai ruangan berulang dari 1 sampai 4.**

1. **Hasil Query Join Sederhana**
   * JOIN Data Mahasiswa dengan Program Studi



SELECT

m.nim, m.nama\_lengkap, p.kode\_prodi, p.nama\_prodi, f.nama\_fakultas

FROM mahasiswa m

JOIN program\_studi p ON m.id\_prodi = p.id\_prodi JOIN fakultas f ON p.id\_fakultas = f.id\_fakultas ORDER BY m.nim

LIMIT 10;

* **JOIN Data Kelas dengan Mata Kuliah dan Dosen**

SELECT

k.kode\_kelas, mk.nama\_mata\_kuliah, mk.sks,

d.nama\_lengkap AS nama\_dosen, ta.tahun,

ta.semester FROM kelas k

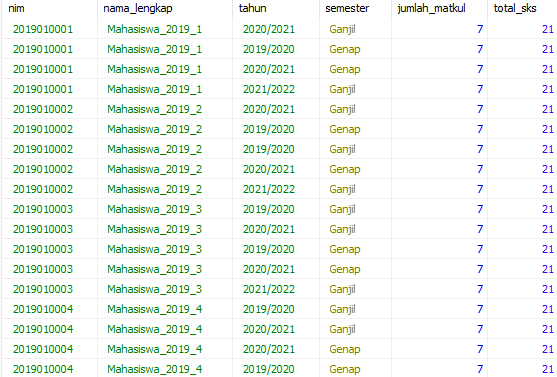
JOIN mata\_kuliah mk ON k.id\_mata\_kuliah = mk.id\_mata\_kuliah JOIN dosen d ON k.id\_dosen = d.id\_dosen

JOIN tahun\_akademik ta ON k.id\_tahun\_akademik = ta.id\_tahun\_akademik WHERE ta.status = 'Aktif'

ORDER BY mk.nama\_mata\_kuliah;



* **JOIN Data KRS Mahasiswa dengan Detail Mata Kuliah**

****

SELECT

m.nim, m.nama\_lengkap, ta.tahun, ta.semester,

COUNT(kd.id\_krs\_detail) AS jumlah\_matkul, SUM(mk.sks) AS total\_sks

FROM krs kr

JOIN mahasiswa m ON kr.id\_mahasiswa = m.id\_mahasiswa

JOIN tahun\_akademik ta ON kr.id\_tahun\_akademik = ta.id\_tahun\_akademik JOIN krs\_detail kd ON kr.id\_krs = kd.id\_krs

JOIN kelas k ON kd.id\_kelas = k.id\_kelas

JOIN mata\_kuliah mk ON k.id\_mata\_kuliah = mk.id\_mata\_kuliah WHERE ta.status = 'Aktif' AND kd.status = 'Aktif'

GROUP BY m.id\_mahasiswa, ta.id\_tahun\_akademik ORDER BY m.nim;