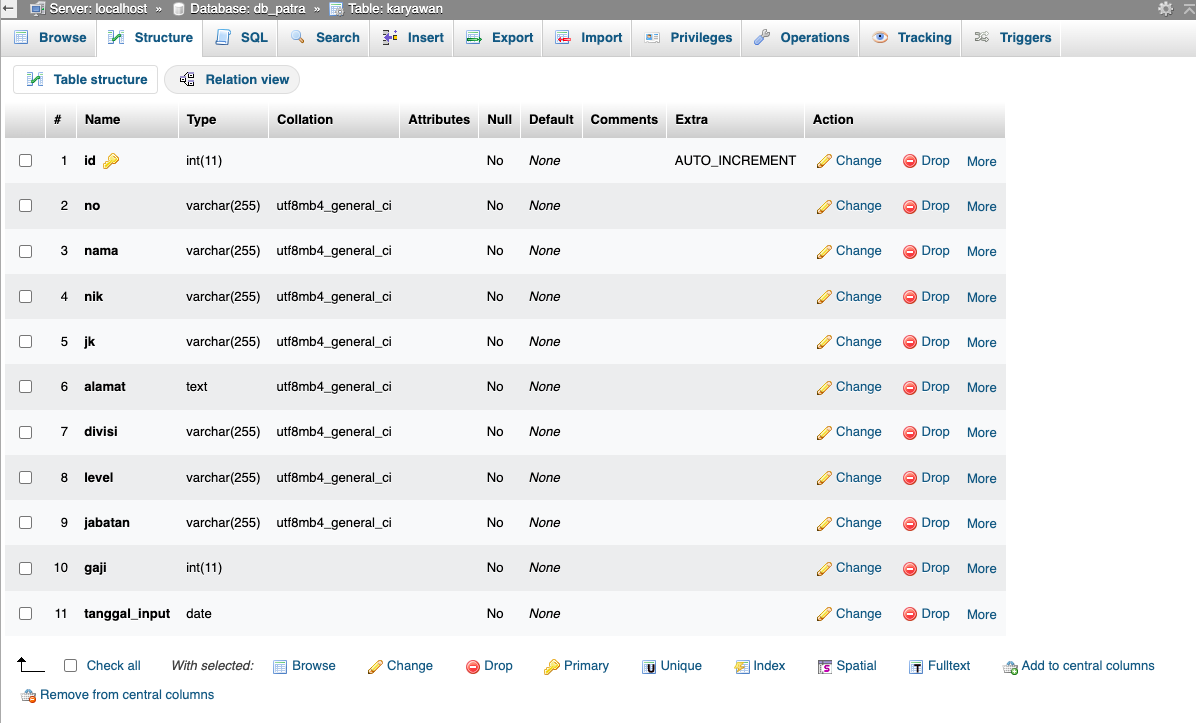
**STRUKTUR QUERY LANGUAGE**

**TYPE DATA SQL (GLOBAL TYPE)**

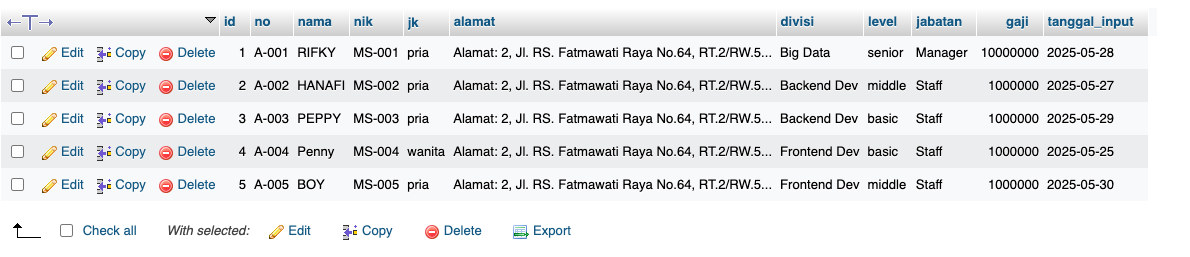
| **Kategori** | **Tipe Data** | **Deskripsi Singkat** |
| --- | --- | --- |
| **Angka (Numerik)** | **INT** | **Bilangan bulat (biasanya 4 byte), sangat umum untuk ID atau hitungan** |
|  | **TINYINT** | **Bilangan bulat kecil (1 byte), cocok untuk status (0/1) atau boolean** |
|  | **BIGINT** | **Bilangan bulat sangat besar (8 byte), cocok untuk jumlah besar atau waktu Unix** |
|  | **DECIMAL(p,s) / NUMERIC** | **Bilangan desimal tetap, cocok untuk data keuangan** |
|  | **FLOAT / DOUBLE** | **Bilangan pecahan desimal (floating point), cocok untuk data ilmiah atau statistik** |
| **Teks (String)** | **VARCHAR(n)** | **Teks dengan panjang variabel, paling umum dipakai untuk nama, email, dll** |
|  | **CHAR(n)** | **Teks dengan panjang tetap, cocok untuk kode tetap (seperti kode pos, kode negara)** |
|  | **TEXT** | **Teks panjang, cocok untuk deskripsi, komentar, atau artikel** |
| **Tanggal/Waktu** | **DATE** | **Format tanggal (YYYY-MM-DD)** |
|  | **DATETIME** | **Format tanggal dan waktu (YYYY-MM-DD HH:MM:SS), sangat umum untuk mencatat waktu** |
|  | **TIMESTAMP** | **Mirip DATETIME tapi mengikuti zona waktu dan auto-update pada perubahan data** |
| **Boolean** | **BOOLEAN / TINYINT(1)** | **Representasi nilai true/false (biasanya disimpan sebagai 0/1)** |
| **Lainnya** | **ENUM('a','b')** | **Pilihan tetap dari beberapa string, cocok untuk status atau kategori tertentu** |
|  | **BLOB** | **Data biner besar, cocok untuk menyimpan file atau gambar dalam database (meskipun tidak disarankan untuk performa)** |

1 = Case Study Data Karyawan

Struktur :



Data:



Latihan

1. Buatlah database patra dan tabel karyawan
2. Buatlah filter data berdasarkan nama depan yang berawalan huruf “R”
3. Buatlah filter data berdasarkan nama belakang yang berawalan huruf “Y”
4. Buatlah urutan data berdasarkan a - z
5. Buatlah urutan data berdasarkan z - a
6. Buatlah query data antara 1 dan 3
7. Buat Konversi Tanggal yang dari angka menjadi huruf supaya lebih dipahami
8. Buatlah Konversi bulan menjadi huruf supaya lebih mudah di pahami
9. Buat tanggal\_input menjadi tanggal indonesia
10. Buatlah selisih antara 2 tanggal
11. Buatlah query untuk menghilangkan penggandaan pada data level
12. Buat lah query mencari gaji terbesar
13. Buat lah query mencari gaji terkecil
14. Buat lah query mencari gaji rata-rata
15. Buat lag query mencari jumlah data gaji
16. Buat lah query untuk menampilkan jika wanita mendapatkan cuti hamil sedangkan pria tidak
17. Buatlah query untuk mengubah nama menjadi huruf besar
18. Buatlah query untuk mengubah nama menjadi huruf kecil
19. Buatlah query untuk memotong nama antara 1 dan 3 huruf
20. Buatlah query untuk menggabungkan huruf

Jawaban Soal No.2

2. SELECT \* FROM `karyawan` where nama LIKE 'R%';

3. SELECT \* FROM `karyawan` where nama LIKE '%Y';

4. SELECT \* FROM `karyawan` ORDER BY ID ASC

5. SELECT \* FROM `karyawan` ORDER BY ID DESC

6. SELECT \* FROM `karyawan` WHERE id BETWEEN 1 AND 3

7.SELECT DAYNAME(tanggal\_input) as tanggal\_huruf FROM `karyawan`

8.SELECT MONTHNAME(tanggal\_input) as bulan\_huruf FROM `karyawan`;

9.SELECT DATE\_FORMAT(tanggal\_input,'%d-%M-%Y') as tanggal\_full\_huruf FROM `karyawan`;

10.SELECT DATEDIFF(tanggal\_input,NOW()) as tanggal\_selesih FROM `karyawan`;

11. SELECT DISTINCT(level) as hilangkan\_penggandaan FROM `karyawan`;

12.SELECT MAX(gaji) as gaji\_teratas FROM `karyawan`

13.SELECT MIN(gaji) as gaji\_terbawah FROM `karyawan`

14.SELECT AVG(gaji) as gaji\_rata\_rata FROM `karyawan`

15.SELECT COUNT(gaji) as jumlah\_gaji FROM `karyawan`

16.SELECT IF(jk = 'wanita', "cuti hamil", "tidak dapat cuti") as cuti FROM karyawan;

17.SELECT UPPER(nama) as nama\_besar FROM `karyawan`

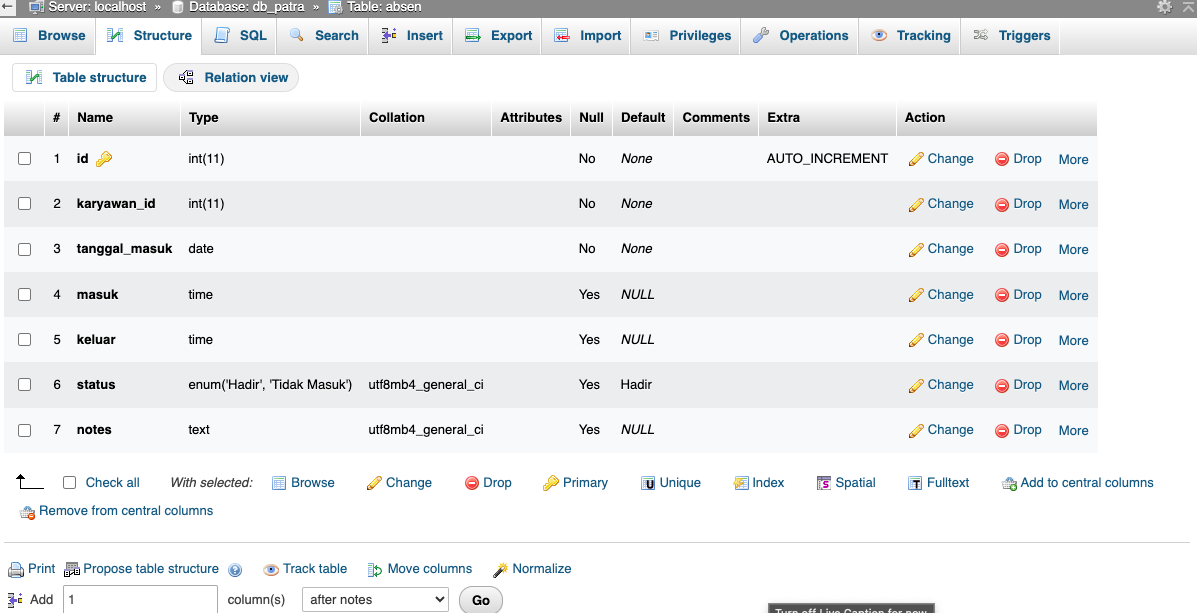
18.SELECT LOWER(nama) as nama\_kecil FROM `karyawan`

19.SELECT SUBSTR(nama,1,2) as potong\_text FROM `karyawan`

20.SELECT CONCAT(nama, "-", nik) AS gabung\_text FROM karyawan;

2 = Case Study Data Karyawan Dan Absen

Join Data Antar Tabel



Latihan

1. Buatlah tabel di atas
2. Gabungkan lah 2 tabel antara karyawan dan absen untuk mendapatkan hasil karyawan yang masuk saja gambaran seperti berikut :
   1. Nama karyawan, Tanggal Masuk, Masuk , Keluar dan Status Kehadiran.
3. Gabungkan lah 2 tabel antara karyawan dan absen untuk mendapatkan hasil karyawan yang tidak masuk saja gambaran seperti berikut :
   1. Nama karyawan, Tanggal Masuk, Masuk , Keluar dan Status Kehadiran.
4. Gabungkan lah 2 tabel untuk mendapatkan jumlah karyawan yang masuk dalam waktu sebulan
5. Gabungkan lah 2 tabel untuk mendapatkan jumlah karyawan yang tidak masuk dalam waktu sebulan
6. Gabungkan lah 2 tabel untuk di jadikan sebagai data grafik untuk menampilkan data absen karyawan di bulan yang sedang berjalan

Jawaban

2. SELECT \* FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'hadir' GROUP BY karyawan.id;

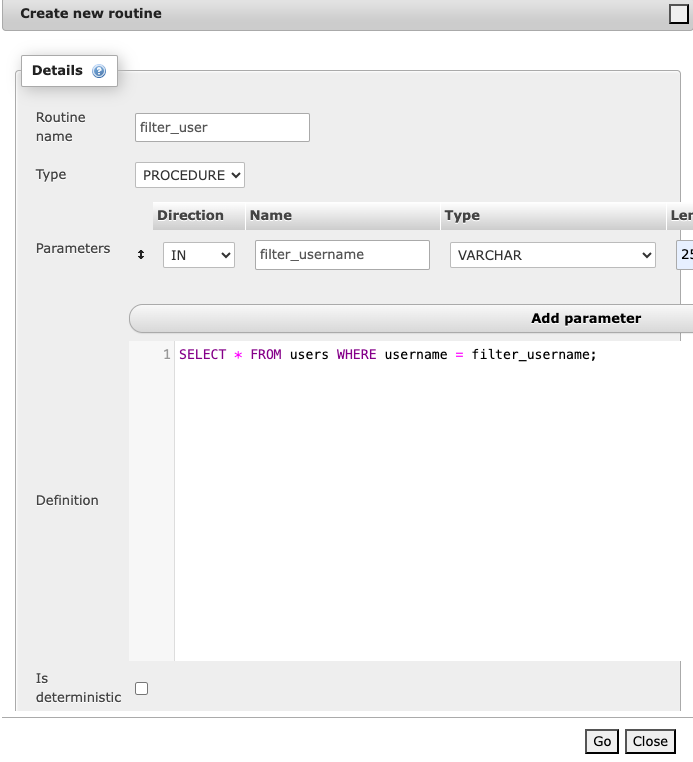
3. SELECT \* FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'tidak\_hadir' GROUP BY karyawan.id;

4. SELECT \*,COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'hadir' AND MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;

5.SELECT \*,COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE absen.status = 'tidak\_hadir' AND MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;

6.SELECT karyawan.id,karyawan.nama, absen.status, karyawan.tanggal\_input, absen.masuk, absen.keluar, COUNT(absen.status) as jml FROM `karyawan` LEFT JOIN absen ON karyawan.id = absen.karyawan\_id WHERE MONTH(karyawan.tanggal\_input) = MONTH(NOW()) GROUP BY karyawan.id;

3 = Case Study Stored Prosedure



Selanjutnya kita bisa exceute dengan memanggil fungsi tersebut dengan command, CALL, seperti contoh berikut :

Call filter\_user()

Cara Lain Untuk Mengolah data stored prosedure ini bisa dengan menggunakan full command line dengan mengetikkan perintan seperti ini :

DELIMITER //

CREATE PROCEDURE GetKaryawanByID(IN karyawanID INT)

BEGIN

SELECT \* FROM karyawan WHERE id\_karyawan = karyawanID;

END //

DELIMITER ;

Selanjutnya sama seperti di atas dengan memanggil fungsi tersebut dengan call , “call GetKaryawanByID(1)”;

Latihan

1. Buatlah Stored Prosedured Diatas dengan menggunakan command line sesuai dari code diatas.

2. Panggilah dengan nama fungsi , bisa disesuaikan dengan nama fungsi diatas

SOAL SOAL SQL LANJUTAN

#### **1. Total Gaji per Departemen**

Tampilkan nama departemen dan total gaji yang dibayarkan ke semua karyawan di masing-masing departemen, urutkan dari yang paling besar.

#### **2. Karyawan Berdasarkan Lama Kerja**

Tampilkan nama karyawan, nama departemen, dan jumlah tahun kerja (dibulatkan ke bawah) bagi karyawan yang sudah bekerja lebih dari 5 tahun.

#### **3. Rata-rata Gaji Berdasarkan Jenis Kelamin**

Hitung rata-rata gaji karyawan berdasarkan jenis kelamin (L atau P), dan tampilkan dalam urutan rata-rata tertinggi.

#### **4. Karyawan Tertua di Setiap Departemen**

Tampilkan satu karyawan tertua dari masing-masing departemen beserta nama dan tanggal lahirnya.

#### **5. Jumlah Karyawan yang Ulang Tahun Bulan Ini**

Hitung berapa jumlah karyawan yang berulang tahun di bulan yang sama dengan bulan saat ini (MONTH(NOW())).

Jawaban

1. SELECT d.nama\_departemen, SUM(k.gaji) AS total\_gaji FROM karyawan k LEFT JOIN departemen d ON k.id\_departemen = d.id\_departemen GROUP BY d.nama\_departemen

ORDER BY total\_gaji DESC;

2.SELECT k.nama, d.nama\_departemen,

FLOOR(DATEDIFF(CURDATE(), k.tanggal\_masuk) / 365) AS tahun\_kerja

FROM karyawan k

JOIN departemen d ON k.id\_departemen = d.id\_departemen

WHERE DATEDIFF(CURDATE(), k.tanggal\_masuk) / 365 > 5;

3. SELECT jk, ROUND(AVG(gaji), 2) AS rata\_gaji

FROM karyawan

GROUP BY jk

ORDER BY rata\_gaji DESC;

4. SELECT k.id, k.nama, d.nama\_departemen, k.tanggal\_lahir

FROM karyawan k

JOIN departemen d ON k.id\_departemen = d.id\_departemen

WHERE k.tanggal\_lahir = (

SELECT MIN(k2.tanggal\_lahir)

FROM karyawan k2

WHERE k2.id\_departemen = k.id\_departemen

);

5. SELECT COUNT(\*) AS jumlah\_ulang\_tahun

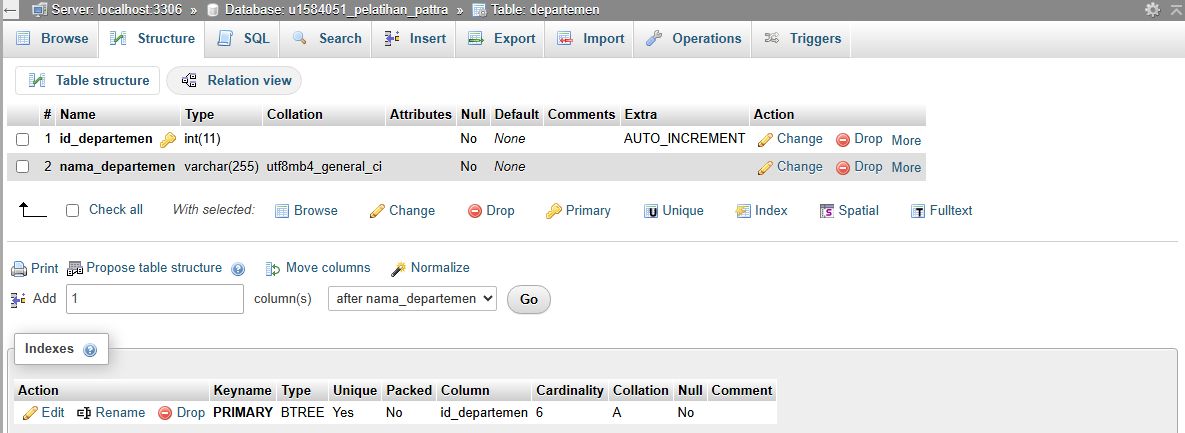
FROM karyawan

WHERE MONTH(tanggal\_lahir) = MONTH(CURDATE());

Operational CRUD

(Create, Read, Update, Delete)

Struktur dan Data



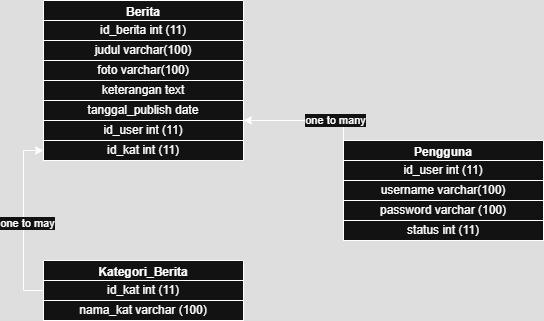
1. Insert / Create
   1. Buatlah Query Insert data kedalam tabel departemen
2. Update /
   1. Buatlah Query Update data kedalam tabel departemen
3. Delete
   1. Buatlah Query Delete data di dalam tabel departemen
4. Select/ Read
   1. Buatlah Query Select All di dalam tabel departemen

Jawaban

1. Insert into departemen(nama\_departemen)values(‘Departemen Baru’);
2. Update departemen SET nama\_departemen =’Departemen Baru’ WHERE id\_departemen = 2;
3. Delete FROM departemen WHERE id\_departemen = 2;
4. SELECT \* FROM departemen;

CLASS DIAGRAM / RELATIONAL DATABASE / ENTITY RELATIONAL DATABASE

Case Study News Portal Apps



Keterangan

Table terdiri dari 3 yaitu

* Berita
* Kategori Berita
* Pengguna

Relasi antar tabel menggunakan relasi one to many dan setiap tabel di wajibkan mempunyai trigger untuk menghubungkan antar masing masing tabel.

Next().

UJIAN FINAL SQL QUERY

**Case Studi, E-Commerce.**

1. Normalization
   1. Database e-commerce
      1. Table product
      2. Table orders
      3. Table Users
      4. Product\_category , next
2. Database Relation Table
3. Xampp Implementation
4. Examination

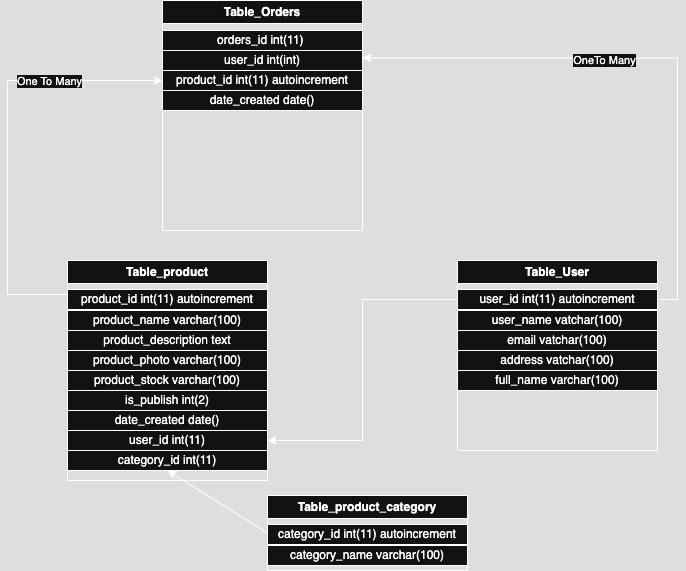
1. Normalization

Adalah Pembuatan Relasi tabel sehingga tidak ada lagi redundan data atau duplikasi tabel antar tabel

Latihan :

1. Buatkanlah normalisasi dari database e-commerce
2. Buatkan Tabel Beserta Relasi Antar Field nya
3. Design Normalisasi menggunakan =”[https://app.diagrams.net/”=](https://app.diagrams.net/%E2%80%9D=)
4. Jika Ada kekurangan tabel bisa ditambahkan atau di modify
5. next

2. Class Diagram



Latihan

1. Buat lah class diagram diatas dengan menggunakan sql
2. Setelah di buat tolong tampilkan semua list produk
3. Tampilkan produk berdasarkan kategorinya
4. Tampilkan produk yang paling banyak di beli
5. Tampilkan produk yang paling sedikit di beli
6. Tampilkan pengguna yang sering melakukan pemesanan
7. Tampilkan list pengguna secara keseluruhan
8. Create data per masing masing table minimal 5 data dengan validasi dari pengajar
9. Update data per masing masing table dengan validasi dari pengajar
10. Delete data satuan table dengan validasi dari pengajar

**Latihan Kelompok**

Kelompok Terdiri dari 5 Orang, Untuk Point 50, dan Masing Masing Orang 10 Point untuk 3 Kelompok.

Setiap Kelompok Akan Di wajibkan Memilih Salah Satu Dari 5 Point Berikut :

1. Koleksi buku Apps
2. Koleksi foto Apps
3. Koleksi video Apps
4. Diary Apps
5. Note Apps
6. Tugas Per-Masing Masing Kelompok Adalah Membuat Class Diagram Relational Database dengan Menentukan Database dan Tabel dari Masing Masing Kasus.
7. Memberikan Keterangan di setiap tabel beserta atribut nya. Semakin jelas keterangan , maka akan semakin tinggi nilai pointnya.
8. Melakukan Operasi CRUD(Create, Read, Update, Delete) dalam kasus kali ini,
9. Command Line dari CRUD Sebelumnya Harus di tulis dan akan dinilai oleh pemberi materi
10. Setiap Kelompok harus memilih 1 orang untuk mewakili tim nya untuk mempresentasikan hasil dari analisa tim nya.