**PROJECT BASIS DATA**

**PADA SISTEM INFORMASI (APLIKASI)**

**BENGKEL KOMPUTER SMKN 1 KARANG BARU**

****

**OLEH:**

**AHMADI MUSLIM**

**NISN. 91287491261**

**REKAYASA PERANGKAT LUNAK**

**SMK NEGERI 1 KARANG BARU**

**PEMERINTAH PROVINSI ACEH**

**2023**

**Langkah-langkah Membuat ERD**

1. Menentukan entitas
2. Menentukan atribut termasuk atribut kunci (Primary key)
3. Identifikasi relasi
4. Menentukan kardinalitas

**Menentukan Entitas**

Berdasarkan aturan-aturan yang di definisikan di atas dapat kita tentukan jumlah entitas ada sebanyak 4 yakni:

1. Petugas
2. Pembeli
3. Layanan
4. Transaksi

**Menentukan Atribut**

Selanjutnya dari Keempat entitas tersebut kita jabarkan atribut-atribut yang melekat pada masing-masing entitas. Atribut yang bersifat unik akan di jadikan sebagai atribut kunci (*primary key*).

**1. Pelanggan (S1)**

* id\_pelanggan int 11 not null primarykey auto increment
* nama\_pelanggan varchar 30
* desa\_pelanggan varchar 50
* kec\_pelanggan varchar 30
* hp\_pelanggan varchar 30

**2. Petugas (S2)**

* id\_petugas int 11 not null primarykey auto increment
* nama\_petugas varchar 30 not null
* desa\_petugas varchar 50 not null
* kec\_petugas varchar 30 not null
* hp\_petugas varchar 30 not null
* jabatan varchar 30 not null
* username varchar 6
* password varchar 5

**3. Layanan**

* id\_layanan int 11 not null primarykey auto increment
* nama\_layanan varchar 50 not null
* harga int 11 not null

**4. Transaksi**

* id\_transaksi int 11 not null primarykey auto increment
* id\_pelanggan int 11 not null foreignkey
* id\_petugas int 11 not null foreignkey
* id\_barang int 11 not null foreignkey
* tanggal date not null
* jumlah int 11 not null

*Atribut dengan kode (PK) akan menjadi atribut kunci (primary key) pada masing-masing entitas.*

**Menentukan Relasi & Kardinalitasnya**

**1. Pelanggan Beli Layanan**

* Merupakan relasi antara entitas petugas dan layanan yang berarti setiap petugas boleh menjual layanan apapun yang ingin dijual.
* Kardinalitas relasi antara petugas dan layanan adalah ***one to many*** karena seorang pembeli dapat membeli banyak layanan atau banyak layanan dapat dibeli oleh seorang pembeli.

**1. Petugas Jual Layanan**

* Merupakan relasi antara entitas petugas dan layanan yang berarti setiap petugas boleh menjual layanan apapun yang ingin dijual.
* Kardinalitas relasi antara petugas dan layanan adalah***one to many*** karena seorang petugas dapat menjual banyak layanan atau banyak layanan dapat dijual oleh seorang petugas.

**2. Petugas Rekap Transaksi**

* Merupakan relasi antara entitas petugas dan transaksi yang berarti petugas dapat melakukan rekap transaksi dari interaksi yang terjadi.
* Kardinalitas relasi antara petugas dan transaksi adalah ***one to one*** karena seorang petugas dapat merekap satu layanan.

**Hasil ERD Penjualan Layanan Bengkel komputer**

Dari tahap-tahap di atas maka dapat di buat rancangan ERD petugasan layanan Bengkel komputer adalah sebagai berikut:

**ERD Bengkel komputer Sebelum Normalisasi**



**ERD Bengkel komputer Setelah Normalisasi**



**m**

**1**

**1**

**1**

**m**

**1**

**DESAIN LOGIKAL**

Desain logikal yaitu proses pembuatan model dari informasi yang digunakan perusahaan berdasarkan model dan data spesifik. Deskripsi implementasi *database* berdasarkan hasil desain logikal dengan *Entity Relationship Diagram* (ERD) pada *Database Management System* (DBMS) menghasilkan ERT sebagai berikut



**DESAIN FISIKAL & SOURCE SQL**

**Menggambarkan Rancangan Entitas Pada Basisdata Secara Fisikal (Physical Data Disaign) serta Membuat Source SQL pembuatan Masing-Masing Tabel/Entitas**

**Tabel pelanggan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_pelanggan** | int | 11 | not null | primarykey | auto increment |
| 2 | nama\_peanggan | varchar | 30 |  |  |  |
| 3 | desa\_pelanggan | varchar | 50 |  |  |  |
| 4 | kec\_pelanggan | varchar | 30 |  |  |  |
| 5 | hp\_pelanggan | varchar | 30 |  |  |  |

create table pelanggan (

id\_pelanggan int(11)not null auto\_increment,

nama\_pelanggan varchar (30) not null,

desa\_pelanggan varchar (50),

kec\_pelanggan varchar (30),

hp\_pelanggan varchar (14),

primary key (id\_pelanggan)

);

**Tabel petugas**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
|  | **id\_petugas** | int | 11 | not null | primarykey | auto increment |
|  | nama\_petugas | varchar | 30 | not null |  |  |
|  | username | varchar | 10 | not null |  |  |
|  | password | varchar | 6 | not null |  |  |
|  | desa\_petugas | varchar | 50 | not null |  |  |
|  | kec\_petugas | varchar | 30 | not null |  |  |
|  | hp\_petugas | varchar | 30 | not null |  |  |
|  | jabatan | varchar | 30 | not null |  |  |

create table petugas (

id\_petugas int(11)not null auto\_increment,

nama\_petugas varchar (30)not null,

username varchar (10)not null,

password varchar (6)not null,

desa\_petugas varchar (50)not null,

kec\_petugas varchar (30)not null,

hp\_petugas varchar (14)not null,

jabatan varchar (50)not null,

primary key (id\_petugas)

);

**Tabel layanan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_layanan** | int | 11 | not null | primarykey | auto increment |
| 2 | nama\_layanan | varchar | 50 | not null |  |  |
| 3 | harga | int | 20 | not null |  |  |

create table layanan (

id\_layanan int(11)not null primary key auto\_increment,

nama\_layanan varchar (50) not null,

harga int(20) not null

);

**Tabel transaksi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| 1 | **id\_transaksi** | int | 11 | not null | primarykey | auto increment |
| 2 | *id\_pelanggan* | int | 11 | not null | foreignkey |  |
| 3 | *id\_petugas* | int | 11 | not null | foreignkey |  |
| 4 | tanggal | date |  | not null |  |  |

create table transaksi (

id\_transaksi int(11)not null auto\_increment,

id\_petugas int(11)not null,

id\_pelanggan int(11)not null,

tanggal date not null,

primary key(id\_transaksi),

constraint id\_petugas foreign key (id\_petugas) references petugas (id\_petugas),

constraint id\_pelanggan foreign key (id\_pelanggan) references pelanggan (id\_pelanggan)

);

**Tabel detiltransaksi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama kolom** | **Tipe data** | **Lebar** | **Null** | **kunci** | **keterangan** |
| **1** | **id\_detiltransaksi** | **int** | **11** | **not null** | **primarykey** | **auto increment** |
| 1 | *id\_transaksi* | int | 11 | not null | foreignkey |  |
| 2 | *id\_layanan* | int | 11 | not null | foreignkey |  |
| 3 | jumlah | int | 20 | not null |  |  |
| 4 | subtotal | int | 20 | not null |  |  |

create table detiltransaksi (

id\_detiltransaksi int(11)not null,

id\_transaksi int(11)not null,

id\_layanan int(11)not null,

jumlah int (20) not null,

subtotal int(20) not null,

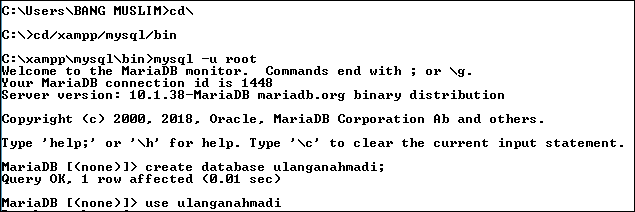
constraint id\_transaksi foreign key (id\_transaksi) references transaksi (id\_transaksi),

constraint id\_layanan foreign key (id\_layanan) references layanan (id\_layanan)

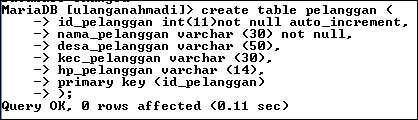
);

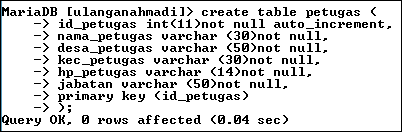
**Implementasi Syntax Sql Database Melalui Cmd**

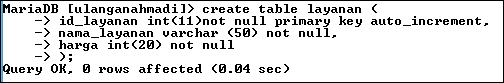
**Membuat Basisdata**

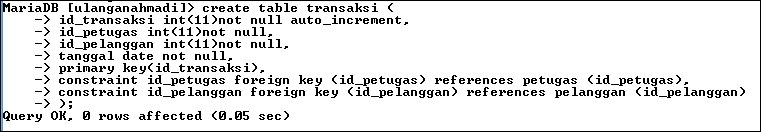


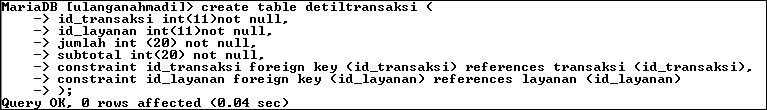
**Membuat Tabel**



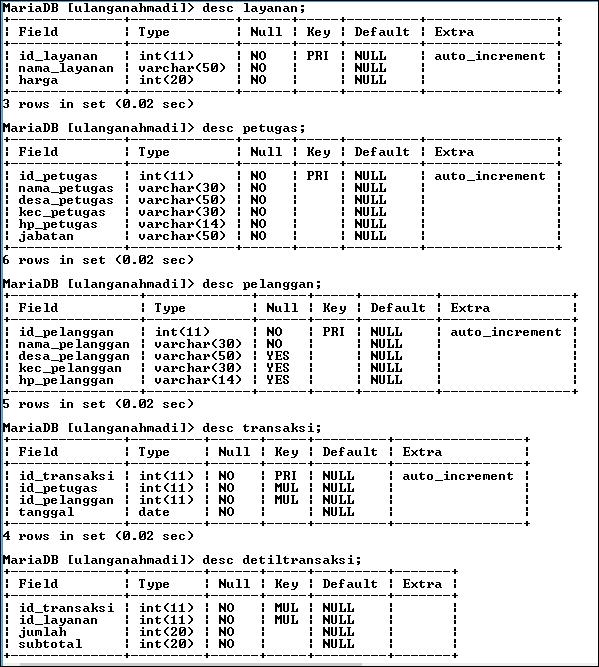




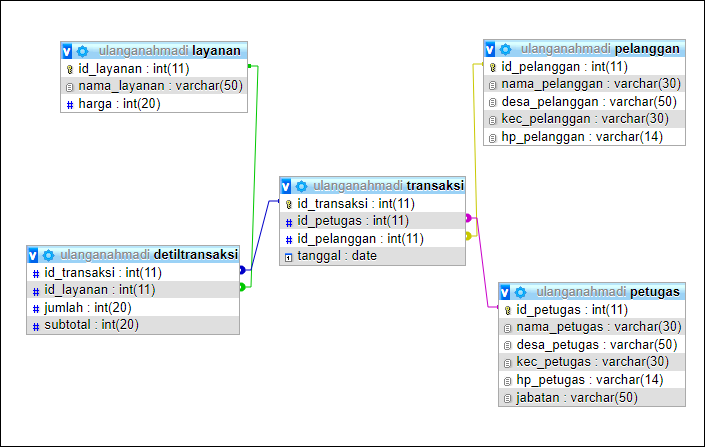




**Medeskripsikan tabel**



**Hasil Desain Konseptual Skema Relasi Database Di Phpmyadmin**



**MENGISI DATA RECORD**

**Cara Manual Dengan Source Code**

Memasukkan data dalam jumlah banyak

Insert into namatabel1

Values

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’),

(‘variable1’,’variable2’,’variabel3’,’dst’);

Insert into layanan

Values

(' ','Instal Ulang (Paket Standar)','50000'),

(' ','Instal Ulang (Paket Komplit)','80000'),

(' ','Hang / Freeze','50000'),

(' ','Layar Biru / Bluescreen','50000'),

(' ','Lambat / Lemot','35000'),

(' ','Sering Mati Tiba Tiba','30000'),

(' ','Restart Tiba Tiba','25000'),

(' ','Blank / Tidak Ada Tampilan','40000'),

(' ','Ada Bunyi Tiiit','25000'),

(' ','Mati Total','200000'),

(' ','Kadang Nyala Kadang Tidak','80000'),

(' ','Ada Bunyi Berderik','25000'),

(' ','Laptop/Pc Kepanasan','25000'),

(' ','Tampilan Laptop Bergaris Horisontal/Vertical','100000'),

(' ','Gagal Instal Ulang','100000'),

(' ','Tidak Keluar Suara','100000'),

(' ','Tampilan Layar Besar Besar / Pecah Pecah','25000'),

(' ','Tidak Bisa Masuk Windows / Loading Terus','25000'),

(' ','Kena Virus (Pembersihan Virus)','25000'),

(' ','Tidak Bisa Koneksi Internet','25000'),

(' ','Hardware Tidak Terdeteksi','25000');



Insert into transaksi

Values

(' ','2','2','2021-01-22'),

(' ','3','3','2021-01-23'),

(' ','5','1','2021-01-24'),

(' ','4','4','2021-01-25'),

(' ','6','1','2021-01-26'),

(' ','4','2','2021-01-27'),

(' ','5','3','2021-01-28'),

(' ','5','1','2021-01-29'),

(' ','2','4','2021-01-30'),

(' ','5','1','2021-01-31'),

(' ','2','2','2021-02-01'),

(' ','5','3','2021-02-02'),

(' ','5','1','2021-02-03'),

(' ','2','4','2021-02-04'),

(' ','5','1','2021-02-05'),

(' ','2','2','2021-02-06'),

(' ','5','3','2021-02-07'),

(' ','2','1','2021-02-08'),

(' ','5','4','2021-02-09'),

(' ','2','1','2021-02-10');

Insert into detiltransaksi

Values

('2','2','2','100000'),

('3','3','3','100000'),

('5','5','5','200000'),

('2','4','4','100000'),

('3','7','7','100000'),

('5','4','4','100000'),

('5','5','5','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

('5','5','5','200000'),

('2','2','2','100000'),

('2','2','2','100000'),

('2','3','3','200000'),

('3','5','5','100000'),

('5','4','4','100000'),

('7','7','7','200000'),

('4','4','4','100000'),

('5','5','5','100000'),

('5','5','5','100000'),

('2','2','2','200000'),

('5','5','5','100000'),

('2','2','2','100000');

('2','2','2','100000'),

('3','3','3','100000'),

('5','5','5','200000'),

('2','4','4','100000'),

('3','7','7','100000'),

('5','4','4','100000'),

('5','5','5','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

('5','5','5','200000'),

('2','2','2','100000'),

('2','2','2','100000'),

('2','3','3','200000'),

('3','5','5','100000'),

('5','4','4','100000'),

('7','7','7','200000'),

('4','4','4','100000'),

('5','5','5','100000'),

('5','5','5','100000'),

('2','2','2','200000'),

('5','5','5','100000'),

('2','2','2','100000'),

Melihat isi data / record yang telah dimasukkan

Select \*from namatabel1;



**Cara IMPORT Dari Excel**

Buka file yang sudah dieksport tadi



Import record yang sudah diisikan ke file csv tadi dengan cara: Masuk ke database dan buka tabel yang akan diimport datanya



Lalu cari file yang sudah diisi datanya tadi dan ubah format filenya menjadi csv Using Load Data



Jika sudah tekan go/kirim



Jika berhasil, maka akan tampil semua data record dari file csv yang sudah diisi tadi

Dan lakukan ke semua tabel yang ada



**SELECT:**

Bentuk umum.

Select \*from namatabel1

**Output :**

Memilih kolom tertentu.

SELECT nama\_kolom1, nama\_kolom2 FROM nama\_tabel;

**Output :**

Memilih kolom dengan mengurutkan dari terkecil.

SELECT \* FROM nama\_tabel order by kolom\_dipilih ASC;

**Output :**

Memilih kolom dengan mengurutkan dari terbesar.

SELECT \* FROM nama\_tabel order by kolom\_dipilih DESC;

**Output :**

**UPDATE**

Update sebaris data record.

UPDATE nama\_tabel SET kolom1=data1, kolom2=data2,... WHERE kolom=data;

**Output :**

**DELETE**

Delete sebaris data record.

DELETE FROM nama\_tabel WHERE kolom=data;

**Output :**

Delete sebuah tabel.

DELETE FROM nama\_tabel;

**Output :**

**SELECT SUBQUERY:**

SELECT nama\_produk FROM ms\_produk LIMIT 3;

**Output :**

SELECT nama\_layanan FROM layanan LIMIT 3;

SELECT ms\_produk.nama\_produk AS nama from ms\_produk;

SELECT layanan.nama\_layanan AS layanan\_yang\_tersedia from layanan;

**Output :**

SELECT \* FROM ms\_produk AS t1;

## SELECT \* FROM petugas AS P1;

**Output :**

SELECT t1.kode\_produk, t1.nama\_produk FROM ms\_produk AS t1;

SELECT t1.kode\_layanan, t1.nama\_layanan FROM layanan AS t1;

**Output :**

SELECT \* FROM ms\_produk WHERE nama\_produk = 'Gantungan Kunci DQLab';

SELECT \* FROM layanan WHERE nama\_layanan = ‘ganti Hardisk';

**Output :**

SELECT \* FROM ms\_produk WHERE harga < 50000;

SELECT \* FROM layanan WHERE harga < 50000;

**Output :**

SELECT \* FROM ms\_produk WHERE nama\_produk = 'Gantungan Kunci DQLab' AND harga < 50000;

SELECT \* FROM layanan WHERE nama\_layanan = 'Lambat / Lemot ' AND harga\_layanan < 50000;

Output :

**SELECT nama\_layanan, COUNT(id\_layanan) as jumlah FROM layanan WHERE harga\_layanan=’50000’ GROUP BY harga\_layanan HAVING COUNT(id\_layanan)>2;**

**Output :**

**SELECT JOINT MULTITABLE:**

SELECT pelanggan.nama\_pelanggan, pembayaran.total\_bayar

FROM pelanggan INNER JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;  
**Output :**

SELECT petugas.nama\_petugas, layanan.nama\_layanan, transaksi.jumlah

FROM petugas

INNER JOIN transaksi ON petugas.id\_petugas = transaksi.id\_petugas

INNER JOIN layanan ON transaksi.id\_layanan = layanan.id\_layanan;

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan LEFT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

SELECT pelanggan.id\_pelanggan, pelanggan.nama\_pelanggan, pembayaran.id\_bayar, pembayaran.tanggal\_bayar FROM pelanggan RIGHT JOIN pembayaran ON pelanggan.id\_pelanggan=pembayaran.id\_pelanggan;

**Output :**

select a.\*, b.\*

from pelanggan a

join pembayaran b using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan);

**Output :**

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

select a.\*, b.\*

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.\*, b.\*

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

**Output :**

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

right join pembayaran b

using (id\_pelanggan)

union all

select a.nama\_pelanggan, b.total\_bayar

from pelanggan a

left join pembayaran b

using (id\_pelanggan);

select a.nama\_pelanggan, b.tanggal

from pelanggan a

right join transaksi b

using (id\_pelanggan)

union all

select a.nama\_pelanggan, b.tanggal

from pelanggan a

left join transaksi b

using (id\_pelanggan);

**Output :**