

IF2140 Permodelan Basis Data
Tugas Besar
ER-Modelling



Kelompok 2
Afkar Dhiya Ulhaq 18220006
I Putu Andika Bagus Jiwanta 18220053
William Gunawan 18220077
Zafran Divac Azzahra 18220079
Muhamad Fikri Nurohman 18220097

Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

DAFTAR ISI

Deskripsi Permasalahan	2
Diagram Relasional	4
Daftar Tabel Relasi	5
Relasi Employee	5
Relasi EmployeeContact	5
Relasi Technician	5
Relasi Fix	6
Relasi FlightCrew	6
Relasi FlightAttendant	7
Relasi Pilot	7
Relasi Airline	7
Relasi Airplane	8
Relasi AirplaneIssues	8
Relasi Customer	9
Relasi Ticket	9
Relasi Model	10
Relasi Flight	10
Relasi Fly	11
Pembagian Tugas	12
Referensi	12
Moments of Meeting	13
1. Deskripsi Asistensi	13
2. Hasil Asistensi	13
3. Foto Bersama	14

Deskripsi Permasalahan

Bandar Udara adalah kawasan di daratan dan/atau perairan yang digunakan sebagai tempat pesawat udara mendarat dan lepas landas. Suatu bandar udara yang paling sederhana minimal memiliki sebuah landasan pacu, sedangkan untuk bandara-bandara besar biasanya dilengkapi berbagai fasilitas lain, baik untuk operator layanan penerbangan maupun bagi penggunaannya seperti bangunan terminal dan hanggar. Menurut Annex 14 dari ICAO (International Civil Aviation Organization): Bandar udara adalah area tertentu di daratan atau perairan (termasuk bangunan, instalasi dan peralatan) yang diperuntukkan baik secara keseluruhan atau sebagian untuk kedatangan, keberangkatan dan pergerakan pesawat.

Mengingat pentingnya bandara bagi keberlangsungan mobilitas masyarakat di Indonesia, kami sebagai mahasiswa Sistem dan Teknologi Informasi (STI) ITB yang memperoleh ilmu mengenai Pemodelan Basis Data tertarik untuk menganalisis sistem yang digunakan bandara dalam perancangan dan pengembangan database yang digunakan di sebuah bandara. *Database* yang kami analisis adalah data maskapai, pegawai bandara, tiket pesawat, pelanggan, model pesawat, pesawat, masalah / isu yang terdapat pada sebuah pesawat, serta penerbangan. Tujuan dari proyek ini adalah untuk mendesain sistem database untuk bandara dan penerbangannya.

Terdapat beberapa maskapai penerbangan yang beroperasi di sebuah bandara. Pada *database* yang kami rancang, setiap maskapai penerbangan akan dicatat id/kode maskapai dan nama maskapai tersebut.

Sistem juga mencatat model-model pesawat. Setiap model pesawat dicatat nomor model yang bersifat unik, nama model, kapasitas, dan berat dari model tersebut. Setiap pesawat memiliki nomor registrasi yang bersifat unik. Pesawat dan model membentuk relasi *Models_of* yang bersifat *total participation*. Artinya, setiap pesawat pasti memiliki model yang tercatat pada sistem dan setiap model tercatat minimal memiliki satu pesawat yang teregistrasi.

Sistem juga mencatat Airline yang merupakan maskapai yang terdaftar di *database* ini. Entity Airline memiliki atribut *airline_id* yang bersifat unik dan *airline_name*. Beberapa FlightAttendant dan Pilot yang terdaftar pada sistem bekerja pada Airline yang didaftarkan. Airline memiliki beberapa airplane. Dalam hal ini, setiap Airplane yang teregistrasi akan terdaftar ke salah satu Airline. Namun, Airline dimungkinkan tidak memiliki Airplane.

Setiap *employee* dicatat SSN (Social Security Number atau ID kependudukan), nama, tanggal_lahir, gaji, alamat (no_jalan, kota, provinsi, negara), usia yang merupakan derivat dari tanggal lahir, dan no telepon yang bisa bernilai lebih dari satu untuk setiap *employee*. Setiap *Employee* pasti dikategorikan menjadi seorang teknisi dan *flight crew*. Untuk seorang *flight crew* dicatat *medical_examination_date* yang isinya merupakan tanggal tes medis yang dilakukannya. Seorang teknisi dapat mempunyai beberapa lisensi AME(Aircraft Maintenance Engineer) dan bertanggung jawab untuk melakukan *maintenance* terhadap *issue* yang ada pada pesawat. Selain teknisi, sistem juga membagi *employee* menjadi *flight crew*. *Flight crew* juga pasti dikategorikan menjadi seorang *flight attendant* dan *pilot*. Pada data *flight attendant*, dicatat lisensi, tinggi

badan, dan berat badan. Pada data Pilot, dicatat lisensi pilot yang terdiri atas PPL (private pilot license), CPL (commercial pilot license), IR (instrument rating), dan MER (multi-engine-rating).

Setiap *Issue* (permasalahan teknis) yang dialami oleh pesawat akan dicatat tanggal perbaikannya(*date*), tipe kerusakannya(*type*), deskripsi kerusakan (*description*), dan tanggal saat perbaikan selesai(*resolved_date*). Dalam satu waktu, pesawat mungkin dapat memiliki beberapa isu.

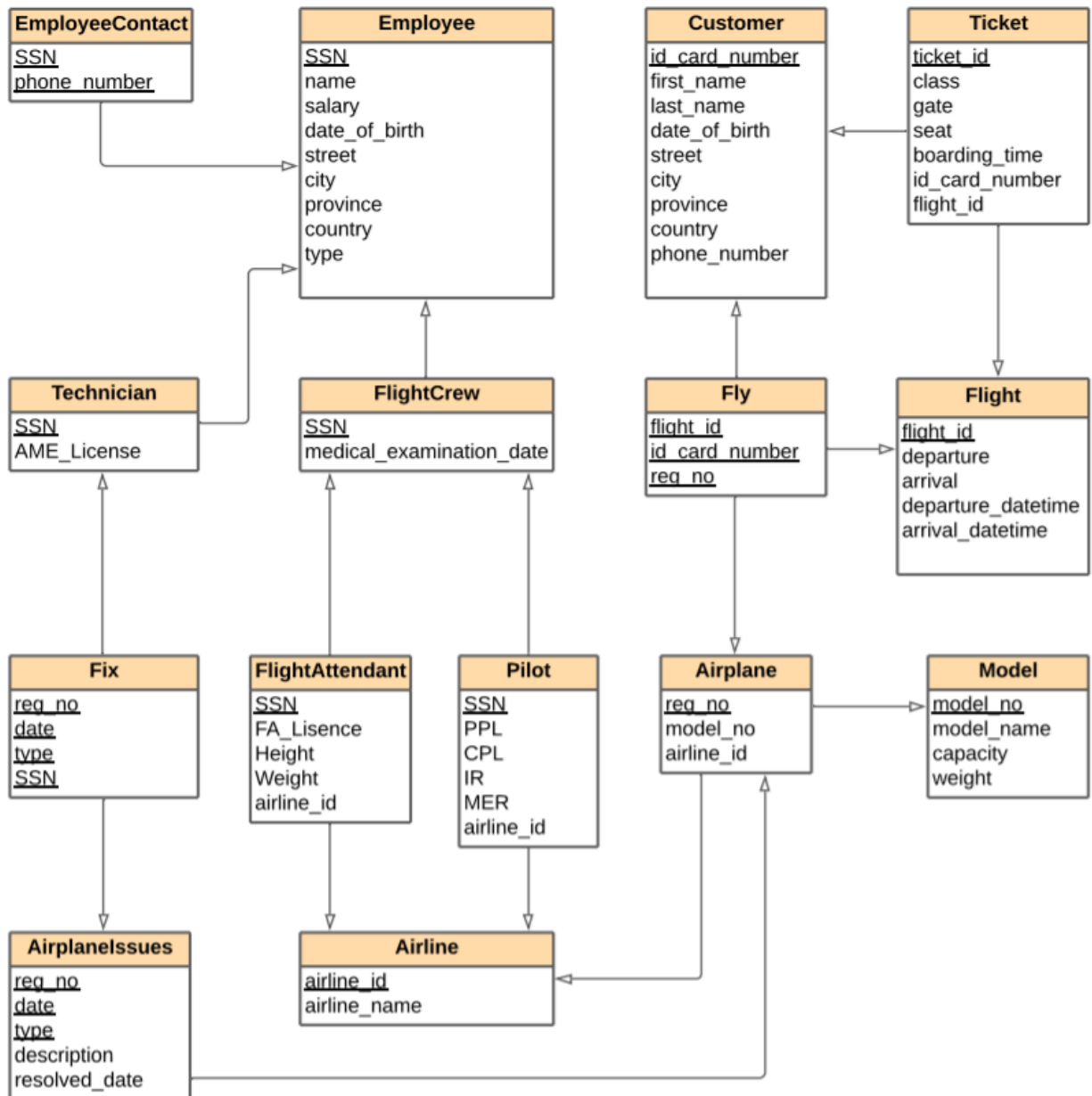
Setiap *customer* akan mempunyai nomor(*ID*) *customer* yang unik. Data yang dicatat dari *customer* berupa nama, tanggal lahir, nomor telepon, alamat *customer*, dan umur. Nama *customer* terdiri atas nama depan dan nama belakang, alamat *customer* terdiri atas nomor jalan, kota, provinsi, dan negara. Umur *customer* akan dihitung secara kontinyu berdasarkan tanggal lahir dari *customer*.

Data *ticket customer* disimpan dalam *entity* ticket. Pada *entity* ticket dicatat kode *ticket* yang unik dan sudah mengandung id penerbangan di dalamnya secara otomatis, *class*(kelas penumpang), *gate* (gerbang masuk pesawat), *seat*(tempat duduk penumpang), dan *boarding time* (waktu calon penumpang dipersilahkan memasuki pesawat).

Setiap penerbangan(*flight*) akan mempunyai kode penerbangan(*flight_id*) yang unik. Data penerbangan yang disimpan, yaitu *departure*(keberangkatan penumpang), *arrival*(kedatangan penumpang), *departure_datetime*(waktu penumpang berangkat), *arrival_datetime*(waktu penumpang tiba), dan *flight_length*(durasi penerbangan). Durasi penerbangan akan dihitung berdasarkan selisih antara waktu penumpang tiba dan waktu penumpang berangkat.

Entity Ticket dan Flight *total participation* terhadap *relationship* “Refers to” karena tidak ada penerbangan yang tidak memiliki tiket (penerbangan pribadi) dan tidak ada tiket yang tidak merujuk ke penerbangan manapun. *Entity* Customer *total participation* terhadap *relationship* Buys karena airport hanya mencatat pelanggan yang membeli tiket saja. Sementara *Entity* Ticket *partial participation* terhadap Buys karena tidak semua tiket harus terjual habis.

Diagram Relasional



Daftar Tabel Relasi

1. Relasi Employee

Employee = (SSN, name, salary, date_of_birth, type, street_number, city, province, country)

Keterangan :

- *type* dapat berupa *Technician*, *Flight Attendant* atau *Pilot*
- *country* hanya berisi negara Indonesia karena kami melakukan asumsi employee berasal dari Indonesia

<u>SSN</u>	name	salary	date_of_birth	type	street	city	province	country
350-55-3085	Chitra Fachrully	\$7631.01	26/10/1986	Pilot	Jl. Komando Raya	Jakarta Selatan	DKI Jakarta	Indonesia

FD = {SSN → name date_of_birth type salary street city province country}

Candidate key = SSN

Relasi *Employee* termasuk ke dalam **BCNF** karena seluruh *functional dependencies* bergantung kepada *candidate key* yaitu SSN. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

2. Relasi EmployeeContact

phone_number = (SSN, phone_number)

FK :

- EmployeeContact(SSN) → Employee(SSN)

<u>SSN</u>	<u>phone_number</u>
350-55-3085	231 527 8321

FD = {SSN phone_number → SSN phone_number}

Candidate key = SSN phone_number

Relasi *EmployeeContact* termasuk ke dalam **BCNF** karena seluruh *functional dependency* bergantung kepada *superkey* atau *candidate key* yaitu phone_number. Dapat dilihat juga *functional dependency* yang terdapat pada relasi ini bersifat trivial. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

3. Relasi Technician

Technician = (SSN, AME_License)

FK :

- Technician(SSN) → Employee(SSN)

<u>SSN</u>	AME_License
758-61-5317	A1_67914883

FD = {SSN → AME_License}.

Technician sudah merupakan relasi **BCNF** karena telah memenuhi syarat ketergantungan langsung dari semua atribut *non key* dan semua LHS merupakan *key*.

4. Relasi Fix

Fix = (reg_id, date, type, SSN)

FK :

- Fix(SSN) → Employee(SSN)
- Fix(reg_id, date, type) → AirplaneIssues(reg_id, date, type)

Keterangan :

- *Type* dapat berupa kerusakan pada mesin atau kurangnya bahan bakar

<u>reg_no</u>	<u>date</u>	<u>type</u>	<u>SSN</u>
101	2021/27/01	Airframe Loads and Flutter	382-66-9302

FD = {SSN date → reg_no type}

Pada relasi ini, kami mengambil asumsi bahwa seorang teknisi maksimal hanya akan mengerjakan satu kerusakan pesawat dalam sehari.

Relasi *Fix* berada pada bentuk **BCNF** karena *Left Hand Side* dari semua *Functional Dependency* yang ada merupakan *superkey* dan setiap *attribute non key* berhubungan langsung dengan *key*.

5. Relasi FlightCrew

FlightCrew = (SSN, medical_examination_date)

FK :

- FlightCrew(SSN) → Employee(SSN)

<u>SSN</u>	medical_examination_date
782-78-0284	2021-04-30

Candidate key = SSN

FD = {SSN → medical_examination_date}

Relasi *FlightCrew* berada pada bentuk **BCNF** karena *Left Hand Side* dari semua *Functional Dependency* yang ada merupakan *superkey*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

6. Relasi FlightAttendant

FlightAttendant = (SSN, airline_id, FA_Lisence, height, weight)

FK :

- FlightAttendant(SSN) → Employee(SSN)
- FlightAttendant(airline_id) → Airline(airline_id)

Keterangan :

- *Height* minimal 160 cm
- *Weight* minimal 50 kg

<u>SSN</u>	airline_id	FA_Lisence	height	weight
782-78-0284	760	FA839	163	53

Candidate key = SSN

FD = {SSN → airline_id FA_Lisence height weight}

Relasi *FlightAttendant* berada pada bentuk **BCNF** karena *Left Hand Side* dari semua *Functional Dependency* yang ada merupakan *superkey*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

7. Relasi Pilot

Pilot = (SSN, airline_id, PPL, CPL, IR, MER)

FK :

- Pilot(SSN) → Employee(SSN)
- Pilot(airline_id) → Airline(airline_id)

<u>SSN</u>	airline_id	PPL	CPL	IR	MER
554-91-6965	124	PPL18844	CPL4795	IR78794	MER881

Candidate key = SSN

FD = {SSN → airline_id PPL CPL IR MER}

Relasi Pilot berada pada bentuk **BCNF** karena *Left Hand Side* dari semua *Functional Dependency* yang ada merupakan *superkey*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

8. Relasi Airline

Airline = (airline_id, airline_name)

<u>airline_id</u>	airline_name
124	Air Asia

Candidate Key = airline_id

FD={airline_id → airline_name}

Relasi *Airline* berada pada bentuk **BCNF** karena *Left Hand Side* dari semua *Functional Dependency* yang ada merupakan *superkey*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

9. Relasi Airplane

Airplane = (reg_no, airline_id, model_no)

FK :

- Airplane(airline_id) → Airline(airline_id)
- Airplane(model_no) → Models(model_no)

<u>reg_no</u>	<u>airline_id</u>	<u>model_no</u>
101	643	21

Candidate Key = reg_no

FD = {reg_no → airline_id model_no}

Relasi *Airplane* termasuk ke dalam **BCNF** karena seluruh *fungsiional dependencies* bergantung kepada *candidate key* yaitu reg_no. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

10. Relasi AirplaneIssues

AirplaneIssues = (reg_id, date, type, description, resolved_date)

FK :

- AirplaneIssues(reg_no) → Airplane(reg_no)

<u>reg_no</u>	<u>date</u>	<u>type</u>	description	resolved_date
112	2021/04/26	Mechanical System	"Oxygen Distribution System"	2021/04/29

FD = {reg_no, date, type → description, resolved_date}

Deskripsi yang terdapat pada relasi ini bukanlah sebuah *statement* yang mendeskripsikan *type*-nya. Deskripsi pada setiap tipe dimungkinkan berbeda-beda. Dengan demikian, tidak ada FD type → description.

Relasi *AirplaneIssues* merupakan relasi **BCNF** karena *Left Hand Side* pada tiap *Functional Dependency* merupakan *key* dan semua *non key attribute*-nya bergantung langsung pada *key*.

11. Relasi Customer

Customer=(id_card_number,first_name,last_name,date_of_birth,age,street_number,city,province,country,phone_number)

Keterangan:

- *phone_number* memiliki format xxx xxx xxxx

<u>id_card_number</u>	first_name	last_name	date_of_birth	street	city	province	country	phone_number
779-63-3136	Emma	Alice	28/09/1995	08192 Calian gt Pass	Kaiyuan	Stockholm	China	696 696 4564

Candidate Key = id_card_number

FD = {id_card_number → first_name date_of_birth street city province country phone_number}

Relasi *Customer* termasuk **BCNF** karena seluruh *functional dependencies* bergantung kepada *candidate key* yaitu *id_card_number*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

12. Relasi Ticket

Ticket=(ticket_id,class,gate,seat,boarding_time,id_card_number,flight_id)

FK:

- Ticket(id_card_number) → Customer(id_card_number)
- Ticket(flight_id) → Flight(flight_id)

Keterangan:

- *Ticket_id* mengandung *flight_id* dan *seat* dari penumpang
- *boarding_time* diasumsikan bernilai sama dengan waktu keberangkatan atau *departure_datetime*

<u>ticket_id</u>	class	gate	seat	boarding_time	id_card_number	flight_id
------------------	-------	------	------	---------------	----------------	-----------

124-0112D	Kelas Satu	5	2D	01/11/21 00:00:00	779-63-3136	124-011
-----------	------------	---	----	----------------------	-------------	---------

Candidate Key = ticket_id

FD = {ticket_id → class gate seat boarding_time id_card_number flight_id}

Relasi *Ticket* termasuk ke dalam **BCNF** karena seluruh *funksional dependencies* bergantung kepada *candidate key* yaitu *ticket_id*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

13. Relasi Model

Model = (model_no, model_name, capacity, weight)

<u>model_no</u>	model_name	capacity	weight
1	ATR 42	133	40028

Candidate Key = model_no

FD = {model_no → model_name capacity weight}

Relasi *Model* termasuk ke dalam **BCNF** karena seluruh *funksional dependencies* bergantung kepada *candidate key* yaitu *model_no*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

14. Relasi Flight

Flight = (flight_id, departure, arrival, departure_datetime, arrival_datetime, flight_length)

Keterangan :

- Flight_id mengandung data dari airline_id dari relasi Airline
- Flight_length merupakan selisih dari departure_datetime dan arrival_datetime

<u>flight_id</u>	departure	arrival	departure_datetime	arrival_datetime
124-011	Pyapon	Jakarta	01/11/21 00:00	01/11/21 01:00

Candidate Key = flight_id

FD = {flight_id → departure arrival departure_datetime arrival_datetime }

Relasi *Flight* termasuk **BCNF** karena seluruh *funksional dependencies* bergantung kepada *candidate key* yaitu *flight_id*. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

15. Relasi Fly

Fly = (flight_id, id_card_number, reg_no)

FK:

- Fly(flight_id) → Flight(flight_id)
- Fly(id_card_number) → Customer(id_card_number)
- Fly(reg_no) → Airplane(reg_no)

flight_id	id_card_number	reg_no
124-011	779-63-3136	101

FD = {id_card_number → flight_id reg_no}

Candidate key = id_card_number

Relasi *Fly* termasuk ke dalam **BCNF** karena seluruh *funksional dependencies* bergantung kepada *candidate key* yaitu id_card_number. Oleh karena itu, relasi ini tidak perlu diubah menjadi bentuk yang lebih tinggi.

Bonus

IP : 20.205.187.95

user : IF2140

port : 22

password : tugasbesarIF2140

user mysql : root

password mysql : toobes

Pembagian Tugas

Nama Lengkap - NIM	Deskripsi Tugas
Afkar Dhiya Ulhaq 18220006	Menentukan <i>functional dependencies</i> , <i>candidate key</i> , dan normalisasi dari relasi <i>flight</i> , <i>ticket</i> , dan <i>customer</i> .
I Putu Andika Bagus Jiwanta 18220053	Membuat <i>functional dependencies</i> , <i>candidate key</i> , dan bentuk normal untuk relasi <i>fix</i> , <i>airplaneissues</i> , dan <i>technician</i>
William Gunawan 18220077	Membuat <i>functional dependencies</i> , <i>candidate key</i> , dan bentuk normal untuk relasi <i>airline</i> , <i>airplane</i> , dan <i>model</i> .
Zafran Divac Azzahra 18220079	Menentukan <i>functional dependencies</i> , <i>candidate key</i> , dan normalisasi dari relasi <i>employee</i> , <i>employeecontact</i> , dan <i>fly</i> .
Muhamad Fikri Nurohman 18220097	Membuat <i>functional dependencies</i> , <i>candidate key</i> , dan bentuk normal untuk relasi <i>flightcrew</i> , <i>pilot</i> , <i>flightattendant</i> . Menambah dummy data pada <i>flightcrew</i> , <i>pilot</i> , <i>flightattendant</i> .

Referensi

Shabani Mtulia. (2021). An Airport Flights Database System Report. Academia.edu.

https://www.academia.edu/4143317/An_Airport_Flights_Database_System_Report

Leslie, A. (2020, August 6). ICAO vs. IATA Codes — What's the Difference? - Alexander Leslie - Medium. Medium; Medium.

<https://aejleslie.medium.com/icao-vs-iata-codes-whats-the-difference-867e6db59c50>

Problems in applying 3NF on ERD. (2014, May 18). Stack Overflow.

<https://stackoverflow.com/questions/23725152/problems-in-applying-3nf-on-erd>

Peraturan Direktur Jenderal Perhubungan Udara No. KP 241 Tahun 2019

Moments of Meeting

1. Deskripsi Asistensi

Nomor Kelompok	2
Anggota yang hadir saat wawancara	<ol style="list-style-type: none">1. Afkar Dhiya Ulhaq2. William Gunawan3. Zafran Divac Azzahra4. Muhamad Fikri Nurohman
Tempat, tanggal	21 November 2021
Waktu	20.00-20.30
Nama Asisten	Rahmat Wibowo

2. Hasil Asistensi

No.	Pertanyaan	Jawaban
1.	Kalo misal ada yang harus di normalisasi harus mengubah diagram ER ga?	Tidak perlu yang diubah hanya diagram relational
2.	Apakah database yang kemarin ada kesalahan?	Tidak tahu karena kak Bowo tidak memeriksa
3.	Apakah <i>funksional dependencies</i> harus berbentuk canonical cover?	Boleh iya boleh tidak, digunakan seperlunya saja
4.	<i>Multivalued attributes</i>	Itu ada di phone_number employee, dibikin relasi sendiri <i>employeecontact</i> PK nya SSN sama phone_number