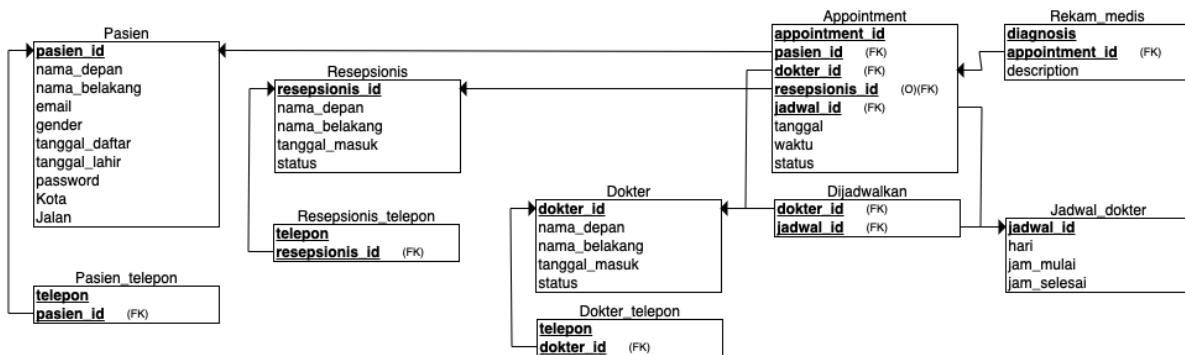


Penjelasan Normalization

Week 2 - Logical Schema & SQL Implementation

Kelas : KOM-A
Anggota : Aliya Khairun Nisa (24/543832/PA/23111)
Farsya Nabila Tori (24/543855/PA/23113)
Gracella Wiendy Koesnadi (24/541739/PA/22991)



Normalized (3NF) Schema

First Normal Form (1NF)

Relasi yang memenuhi 1NF adalah relasi yang setiap perpotongan baris dan kolomnya berisi satu dan hanya satu nilai. Maka dari itu, atribut yang *multivalued* akan didekomposisi menjadi tabel baru. Dalam *logical schema* kami, atribut yang di-set *multivalued* hanyalah atribut telepon yang ada di beberapa entitas. Sehingga atribut telepon pada tiap entitas (Pasien, Resepsonis, Dokter) dibuatkan tabel baru. Untuk atribut email dalam entitas pasien tidak *multivalued* karena kami berencana untuk menggunakan email tersebut untuk autentikasi akun pasien (pasien *login* ke akun dengan email). Jadi, email bukan atribut yang *multivalued*. Oleh karena itu, *database* kami sudah 1NF karena tidak ada atribut multivalued yang tidak ada grup atau kolom berulang dalam satu baris.

Selain itu, setiap tabel juga sudah memiliki *primary key* yang merupakan hal wajib pada aturan 1NF. Hal ini terutama terlihat pada relasi Rekam_medis yang termasuk *weak entity*, didefinisikan untuk memiliki *primary key* berupa gabungan diagnosis (*discriminator*) dan appointment_id (*primary key* dari *strong entity*).

Second Normal Form (2NF)

Relasi 2NF adalah relasi yang memenuhi 1NF dan setiap atribut yang bukan *primary key* memiliki ketergantungan fungsional penuh (*Full functional dependency*) pada *primary key*. Berikut adalah penjelasan 2NF pada *logical schema* kami.

Tabel	Penjelasan
Pasien	Semua atribut <i>non-key</i> (nama_depan, email, dll.) bergantung penuh pada pasien_id sebagai <i>primary key</i>
Pasien_telepon	Semua atribut <i>non-key</i> bergantung penuh ke Pasien melalui <i>foreign key</i>
Resepsionis	Semua atribut <i>non-key</i> (nama_depan, status, dll.) bergantung penuh pada resepsionis_id sebagai <i>primary key</i>
Resepsionis_telepon	Semua atribut <i>non-key</i> bergantung penuh ke resepsionis melalui <i>foreign key</i>
Dokter	Semua atribut <i>non-key</i> (nama_depan, tanggal_masuk, dll.) bergantung penuh pada dokter_id sebagai <i>primary key</i>
Dokter_telepon	Semua atribut bergantung penuh ke Dokter melalui <i>foreign key</i>
Jadwal_dokter	Semua atribut <i>non-key</i> (hari, jam_mulai, jam_selesai) bergantung penuh pada jadwal_id sebagai <i>primary key</i>
Dijadwalkan	Dokter_id dan jadwal_id berperan sebagai <i>composite primary key</i> , tetapi tidak ada atribut <i>non-key</i> tambahan (hanya <i>foreign key pair</i>)
Appointment	Semua atribut <i>non-key</i> (waktu, status, tanggal) bergantung penuh pada appointment_id sebagai <i>primary key</i>
Rekam_medis	Diagnosis dan description bergantung penuh pada appointment_id sebagai <i>primary key</i>

Database kami sudah 2NF karena setiap atribut yang bukan *primary key* pada tabel mempunyai ketergantungan penuh pada *primary key* (*no partial dependencies*).

Third Normal Form (3NF)

Relasi 3NF adalah relasi yang memenuhi 1NF, 2NF, dan atribut yang bukan *primary key* tidak memiliki ketergantungan transitif pada *primary key*. Dalam *logical schema* kami, atribut yang kemungkinan *transitive dependency* hanyalah diagnosis dan description pada

tabel Rekam_medis (appointment_id → diagnosis → description). Setelah ditelusuri lebih lanjut, kedua atribut ini tidak *transitive dependency* karena:

Diagnosis	Description
Flu	Batuk kering
Flu	Sakit tenggorokan
Flu	Pilek

Bisa dilihat bahwa nilai diagnosis sama, tetapi nilai description-nya dapat berbeda, ini membuktikan bahwa nilai description tidak bergantung pada diagnosis sehingga bukan *transitive dependency*. Dengan ini, *database* kami sudah 3NF karena nilai atomik, tidak ada *partial dependencies*, dan tidak ada *transitive dependencies*.

SQL Implementation

SQL

```
CREATE TABLE Pasien(
    pasien_id INT NOT NULL,
    nama_depan VARCHAR(50) NOT NULL,
    nama_belakang VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    gender enum('Perempuan','Laki-laki') NOT NULL,
    tanggal_daftar DATE NOT NULL,
    tanggal_lahir DATE NOT NULL,
    password VARCHAR(50) NOT NULL,
    Kota VARCHAR(50) NOT NULL,
    Jalan VARCHAR(255) NOT NULL,
    PRIMARY KEY (pasien_id)
);
```

```
CREATE TABLE Dokter(
    dokter_id INT NOT NULL,
```

```
nama_depan VARCHAR(50) NOT NULL,  
nama_belakang VARCHAR(50) NOT NULL,  
tanggal_masuk DATE NOT NULL,  
status enum('Aktif', 'Non Aktif') NOT NULL,  
PRIMARY KEY (dokter_id)  
);
```

```
CREATE TABLE Resepsionis(  
resepsionis_id INT NOT NULL,  
nama_depan VARCHAR(50) NOT NULL,  
nama_belakang VARCHAR(50) NOT NULL,  
tanggal_masuk DATE NOT NULL,  
status enum('Aktif', 'Non Aktif') NOT NULL,  
PRIMARY KEY (resepsionis_id)  
);
```

```
CREATE TABLE Jadwal_dokter(  
jadwal_id INT NOT NULL,  
hari VARCHAR(10) NOT NULL,  
jam_mulai CHAR(5) NOT NULL,  
jam_selesai CHAR(5) NOT NULL,  
PRIMARY KEY (jadwal_id)  
);
```

```
CREATE TABLE Dijadwalkan(  
dokter_id INT NOT NULL,  
jadwal_id INT NOT NULL,  
PRIMARY KEY (dokter_id, jadwal_id),
```

```
FOREIGN KEY (dokter_id) REFERENCES Dokter(dokter_id),
FOREIGN KEY (jadwal_id) REFERENCES Jadwal_dokter(jadwal_id)
);

CREATE TABLE Pasien_telepon(
telepon VARCHAR(50) NOT NULL,
pasien_id INT NOT NULL,
PRIMARY KEY (telepon, pasien_id),
FOREIGN KEY (pasien_id) REFERENCES Pasien(pasien_id)
);

CREATE TABLE Dokter_telepon(
telepon VARCHAR(50) NOT NULL,
dokter_id INT NOT NULL,
PRIMARY KEY (telepon, dokter_id),
FOREIGN KEY (dokter_id) REFERENCES Dokter(dokter_id)
);

CREATE TABLE Resepsionis_telepon(
telepon VARCHAR(50) NOT NULL,
resepsionis_id INT NOT NULL,
PRIMARY KEY (telepon, resepsionis_id),
FOREIGN KEY (resepsionis_id) REFERENCES Resepsionis(resepsionis_id)
);

CREATE TABLE Appointment(
appointment_id INT NOT NULL,
tanggal DATE NOT NULL,
```

```
waktu CHAR(13) NOT NULL,  
status enum('Belum Selesai', 'Selesai') NOT NULL,  
pasien_id INT NOT NULL,  
dokter_id INT NOT NULL,  
resepionis_id INT,  
jadwal_id INT NOT NULL,  
PRIMARY KEY (appointment_id, pasien_id, dokter_id, resepionis_id,  
jadwal_id),  
FOREIGN KEY (pasien_id) REFERENCES Pasien(pasien_id),  
FOREIGN KEY (dokter_id) REFERENCES Dokter(dokter_id),  
FOREIGN KEY (resepionis_id) REFERENCES Resepionis(resepionis_id),  
FOREIGN KEY (jadwal_id) REFERENCES Jadwal_dokter(jadwal_id)  
);
```

```
CREATE TABLE Rekam_medis(  
diagnosis VARCHAR(100) NOT NULL,  
description VARCHAR(255) NOT NULL,  
appointment_id INT NOT NULL,  
PRIMARY KEY (appointment_id),  
FOREIGN KEY (appointment_id) REFERENCES Appointment(appointment_id)  
);
```