# **COVER**

**SISTEM DATABASE SANDYAGA HOSPITAL**

LAPORAN PROYEK AKHIR

MATA KULIAH ISYS6028 – DATABASE SYSTEMS

KELAS LA20

A picture containing schematic

Description automatically generated

Oleh:

2301930476 – BILLY MOSES

2301927916 – GUSTI SANDYAGA PUTRA WARDHANA

2301876612 – JONATHAN EVAN SAMPURNA

Semester Ganjil 2020

MALANG

# **LEMBAR PERSETUJUAN PROYEK AKHIR**

**SISTEM DATABASE SANDYAGA HOSPITAL**

MATA KULIAH ISYS6028 – DATABASE SYSTEMS

KELAS LA20

Semester Ganjil 2020

Laporan proyek akhir di atas adalah benar karya dari:

**JONATHAN EVAN S.**

**2301876612**

**GUSTI SANDYAGA**

**2301927916**

**BILLY MOSES**

**2301930476**

Malang, 19 Desember 2020

**WINA PERMANA SARI**

**D5975**

DAFTAR ISI

[COVER 1](#_Toc61612775)

[LEMBAR PERSETUJUAN PROYEK AKHIR 2](#_Toc61612776)

[DAFTAR TABEL 4](#_Toc61612777)

[DAFTAR GAMBAR 5](#_Toc61612778)

[BAB 1. PENDAHULUAN 6](#_Toc61612779)

[BAB 2. LANDASAN TEORI 8](#_Toc61612780)

[BAB 3. ANALISIS 10](#_Toc61612781)

[BAB 4. DESAIN 12](#_Toc61612782)

[LAMPIRAN 25](#_Toc61612783)

[LEMBAR PENILAIAN 44](#_Toc61612784)

# DAFTAR TABEL

Tabel 4.1. Entity Types……………………………………………………………………………………12

Tabel 4.2. Relationship Types……………………………………………………………………………..13

Tabel 4.3. Attributes……………………………………………………………………………………….15

Tabel 4.4. Candidate Keys, Primary Keys, and Alternate Keys…………………………………………..18

Tabel 4.5. Redundant Relationship………………………………………………………………………..19

Tabel 4.6. List of Index……………………………………………………………………………………22

Tabel 4.7. Analyze Transaction…………………………………………………………………………...23

# DAFTAR GAMBAR

Gambar 3.1. Use Case Diagram……………………………………………………………………………10

Gambar 4.1. ERD Conceptual……………………………………………………………………………..20

Gambar 4.2. Class Diagram………………………………………………………………………………..21

Gambar 4.3. User View, Procedure, and Function…………………………………………………………23

# BAB 1. PENDAHULUAN

1.1. Latar Belakang

Pada akhir tahun 2019, dunia digemparkan dengan kemunculan *2019 novel coronavirus* (*2019-ncov*) di Wuhan, China, dan dampaknya berujung pada terjadinya pandemi COVID-19 di seluruh dunia. Berdasarkan data dari Jhons Hopkins School, secara global kasus positif COVID-19 lebih dari 3.754.650 kasus positif dengan 1.246.184 kasus dinyatakan pulih dan 263.861 kasus meninggal dunia. Dan di Indonesia sendiri, kasus positif sudah mencapai 12.438 kasus dengan 2.317 kasus dinyatakan sembuh dan 895 kasus dinyatakan meninggal dunia. [1]

Dikarenakan pandemi berskala global ini, penerapan Pembatasan Sosial Berskala Besar (PSBB) atau *social distancing* diterapkan dimana-mana demi mengurangi penyebaran virus tersebut. Dan sebagai upaya untuk mencegah lebih banyak penularan, ada berbagai prosedur kegiatan yang mewajibkan seseorang untuk melakukan tes medis, seperti *rapid test* dan *swab test*. Dengan pengawasan pemerintah, banyak rumah sakit dan pos-pos kesehatan tertentu yang menyediakan layanan pemeriksaan atau tes medis. Ada metode yang disebut *contact tracing,* yaitu sebuah proses untuk mengidentifikasi orang-orang yang melakukan kontak dengan pasien yang terjangkit COVID-19 dalam jangka waktu dua minggu. Metode ini sering digunakan untuk menghubungi orang-orang yang melakukan kontak fisik dengan penderita penyakit tersebut agar dapat segera dilakukan pemeriksaan dan isolasi untuk mencegah penularan yang lebih luas.

Diperlukan sistem manajemen data yang baik untuk dapat menyimpan dan mengolah data-data yang dibutuhkan rumah sakit terkait penanganan pasien. Dengan perkembangan teknologi otomasi sebagai penunjang utama pembuatan keputusan dalam organisasi-organiasi modern, manajemen informasi data banyak diterapkan dengan teknologi komputer. Informasi-informasi berupa data diolah dengan sistem database yang memungkinkan pemakai untuk mengakses dan memanipulasi data-data yang disimpan dalam sistem komputer [2]. Karena itu, sistem database dapat mempermudah pengaturan data operasional rumah sakit dan mempermudah pencatatan data untuk keperluan *contact tracing*.

1.2. Rumusan Masalah

1. Bagaimana sistem database pada rumah sakit bekerja?
2. Bagaimana database berperan dalam membantu pengaturan data operasional rumah sakit?
3. Bagaimana database dapat membantu penerapan proses *contact tracing* untuk keperluan pendataan pasien?

1.3. Tujuan Penelitian

1. Mengetahui bagaimana sistem database bekerja pada rumah sakit.
2. Mengetahui bagaimana database berperan dalam membantu pengaturan data operasional rumah sakit.
3. Mengetahui bagaimana database dapat membantu penerapan proses *contact tracing* untuk keperluan pendataan pasien.

1.4. Manfaat Penelitian

1. Sebagai referensi bagi lembaga-lembaga kesehatan untuk menerapkan penggunaan sistem database sebagai alat bantu pengaturan data-data operasional rumah sakit dan membantu proses *contact tracing*.
2. Sebagai sumber referensi dan bahan masukan penulis lain untuk melakukan penelitian terkait penggunaan sistem database pada dunia kesehatan.
3. Sebagai penambah wawasan untuk mempelajari praktik dan penerapan sistem database pada dunia kesehatan.

1.5. Sistematika Penulisan

Laporan ini disusun dengan menggunakan sistematika penulisan sebagai berikut:

1. Bab 1 Pendahuluan, berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.
2. Bab 2 Landasan Teori, berisi tentang teori yang berkaitan dengan topik penelitian, yaitu tentang sistem database dan sistem *contact tracing* pada rumah sakit.
3. Bab 3 Analisis, berisi tentang pendefinisian sistem, identifikasi masalah, dan identifikasi kebutuhan pengguna.
4. Bab 4 Desain, berisi tentang desain model konseptual (entity type, relationship type, attributes, attribute domain, candidate-primary-alternate keys, redundant relationship, dan ERD conceptual), model logical (class diagram), dan model physical (analyze transaction dan user view-procedure-function)
5. Daftar Pustaka, berisi kumpulan referensi dan bahan rujukan untuk penulisan laporan.

# BAB 2. LANDASAN TEORI

2.1. Sistem Database

Database adalah sekumpulan data-data yang secara logis saling berelasi dan dibuat untuk menyediakan kebutuhan informasi pada keperluan-keperluan tertentu seperti suatu organisasi atau perusahaan [3]. Kemudian Database Management System (DBMS) adalah suatu perangkat lunak yang berinteraksi dengan program dan database, yang mampu mengatur database dan mengontrol akses ke database. Pada proses pengolahan database menggunakan DBMS, ada yang disebut dengan Data Definition Language (DDL), yaitu proses menspesifikasi tipe data dan strukturnya untuk dapat disimpan pada database, serta Data Manipulation Language (DML), yaitu proses untuk melakukan *insert*, *update*, *delete*, dan *retrieve* data dari database [3]. Kedua proses tadi menggunakan suatu *query language* untuk menjalankan perintah-perintah dan operasi di dalam pengaturan database.

Penyusunan database memiliki manfaat untuk mengatasi masalah-masalah pada penyusunan data. Ada banyak contoh masalah yang dapat terjadi pada proses penyusunan data seperti redudansi dan inkonsistensi data (pemborosan tempat penyimpanan data yang sama dan data-data yang perlu diperbaharui untuk jangka waktu tertentu), kesulitan pengaksesan data, isolasi data untuk standarisasi (penyebaran data dengan format yang berbeda), banyak pemakai, masalah keamanan, masalah integrasi, dan masalah data *independence* [4].

2.2. Sistem *Contact Tracing*

Sistem pelacakan kontak atau *contact tracing* adalah proses untuk mengidentifikasi, menilai, dan mengelola orang-orang yang berkontak erat dengan kasus positif COVID-19 untuk mencegah penularan selanjutnya [5]. Proses ini perlu untuk dilakukan karena mereka yang terkonfirmasi positif COVID-19 dapat menularkan penyakit sejak dua hari sebelum hinga empat belas hari sesudah timbulnya gejala. Tanpa sistem pelacakan, kasus penularan bisa terjadi lebih cepat karena tidak diketahui data orang-orang yang melakukan kontak dengan pasien sehingga tidak dapat ditindaklanjuti dengan cepat.

Pengidentifikasian dengan sistem *contact tracing* adalah sebagai berikut [5]: orang-orang yang berkontak dengan kasus konfirmasi atau *probable* sejak dua hari sebelum hingga empat belas hari setelah timbul gejala klinis yaitu:

1. Orang yang memiliki kontak fisik atau berada kurang dari satu meter selama lima belas menit atau lebih.
2. Orang yang berada di lingkungan tertutup yang sama untuk jangka waktu lama, seperti tinggal dalam satu rumah, rekan kerja, teman sekolah, hadir di pertemuan, atau menggunakan alat transportasi yang sama.
3. Orang yang mengunjungi kasus baik di rumah atau di fasilitas layanan kesehatan.
4. Orang yang berada di fasilitas umum yang dikunjungi kasus.
5. Petugas kesehatan yang kontak tanpa menggunakan APD standar.
6. Orang yang berkontak dengan jenazah kasus konfirmasi tanpa menggunakan APD yang sesuai.

# BAB 3. ANALISIS

* 1. System Definition

Diagram

Description automatically generated

Gambar 3.1. Use Case Diagram

* 1. Identifikasi Permasalahan

Berdasarkan latar belakang, terdapat beberapa identifikasi permasalahan yaitu:

1. Tingginya tingkat penyebaran COVID-19.

2. Pentingnya sistem *contact tracing* untuk mengurangi proses penyebaran COVID-19.

3. Pentingnya penerapan sistem database pada operasional rumah sakit.

* 1. Identifikasi Kebutuhan Pengguna

Untuk membantu proses operasional rumah sakit, perlu ada sistem database yang menyimpan data-data pasien yang meliputi nama, jenis kelamin, usia, alamat, dan nomor telepon. Kemudian juga diperlukan penyimpanan data-data staff dan dokter seperti nama, jenis kelamin, alamat, nomor telepon, dan gaji. Berkaitan dengan data dokter, perlu ada penyimpanan data spesialisasi dokter. Juga diperlukan penyimpanan data jenis perawatan (treatment) dan tipe perawatan (treatment type) yang berisikan perawatan-perawatan yang disediakan di rumah sakit tersebut. Pasien dapat melakukan transaksi dengan memilih jenis perawatan dan melakukan pembayaran. Data transaksi disimpan dan diolah juga pada detail transaksi. Setalah melakukan transaksi, akan dilakukan perawatan atau pemeriksaan yang disesuaikan dengan jadwal shift yang tersedia di rumah sakit. Kemudian data transaksi dan jadwal pemeriksaan akan disimpan dan digunakan pada hasil. Jika melakukan pemeriksaan terkait penyakit COVID-19 dan hasilnya adalah positif, maka pasien akan menginputkan data-data terkait *contact tracing* yaitu nama-nama beserta nomor telepon orang-orang yang melakukan kontak fisik dengan pasien tersebut. Staff dan dokter yang melakukan pemeriksaan terhadap pasien tersebut juga akan dimasukkan pada daftar *contact tracing*.

# BAB 4. DESAIN

4.1. Conceptual Model

4.1.1. Entity Types

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Description | Aliases | Occurence |
| Patient | Berisi data diri pasien yang terdaftar | - | Setiap pasien dapat memilih treatment yang diperlukan |
| Staff | Berisi data diri staff yang bertugas | - | Setiap staff dapat menangani pasien yang berbeda |
| Doctor | Berisi data diri dokter yang bertugas | - | Setiap dokter dapat menangani pasien yang berbeda |
| Specialization | Berisi spesialisasi dari entitas Doctor | - | Satu jenis spesialisasi dapat dimiliki oleh lebih dari 1 dokter |
| TreatmentType | Berisi jenis-jenis dari perawatan yang tersedia | - | Satu jenis treatment type dapat dimiliki oleh lebih dari 1 treatment |
| Treatment | Berisi macam perawatan yang tersedia | - | Setiap treatment dapat dipilih oleh pasien yang berbeda |
| DetailSchedule | Berisi tentang detail jadwal yang tersedia | - | Setiap detail schedule menjelaskan shift secara mendetail |
| Schedule | Berisi tentang jadwal dan transaksi | - | Setiap schedule dapat memiliki shift yang berbeda |
| DetailTransaction | Berisi tentang detail transaksi yang dibuat | - | Setiap detail transaction memiliki data transaksi dan pilihan treatment |
| Transaction | Berisi tentang trasaksi yang dibuat | - | Setiap transaction menjelaskan pelaksanaan treatment yang mendetail |
| Result | Berisi hasil dari perawatan pasien | - | Setiap result menjelaskan hasil treatment pada pasien yang sudah melakukan transaksi |
| Tracing | Berisi macam lacakan terhadap pasien yang positif | - | Tracing memiliki data nama dan nomer telepon terkait contact tracing |

Tabel 4.1. Entity Types

4.1.2 Relationship Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Multiplicity | Relationship | Multiplicity | Entity Name |
| Patient | 1..1  1..1 | Include  Include | 1..1  0..\* | Transaction  Tracing |
| Staff | 1..1 | Include | 1..1 | Transaction |
| Doctor | 1..1  1..1 | Include  Include | 1..1  1..1 | Transaction  Specialization |
| Specialization | 1..1 | Include | 1..1 | Doctor |
| TreatmentType | 1..1 | Include | 1..\* | Treatment |
| Treatment | 1..\*  1..1 | Include  Related | 1..1  1..1 | TreatmentType  DetailTransaction |
| DetailSchedule | 1..1 | Include | 1..1 | Schedule |
| Schedule | 1..1  1..1 | Include  Include | 1..1  1..1 | DetailSchedule  Transaction |
| DetailTransaction | 1..1  1..1 | Related  Related | 1..1  1..1 | Transaction  Treatment |
| Transaction | 1..1  1..1  1..1  1..1  1..1  1..1 | Include  Include  Include  Include  Related  Include | 1..1  1..1  1..1  1..1  1..1  1..1 | Patient  Staff  Doctor  Schedule  DetailTransaction  Result |
| Result | 1..1 | Include | 1..1 | Transaction |
| Tracing | 0..\* | Include | 1..1 | Patient |

Tabel 4.2. Relationship Types

4.1.3. Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity Name | Attribute | Description | Data Type & Length | Nulls | Multi-valued |
| Patient | PatientID  PatientName  PatientGender  PatientAge  PatientAddress  PatientPhone | Identifikasi pasien  Nama pasien  Jenis kelamin pasien  Usia pasien  Alamat pasien  Nomor telepon pasien | CHAR(5)  VARCHAR(50)  VARCHAR(10)  NUMERIC(3)  VARCHAR(100)  VARCHAR(20) | NO  NO  NO  NO  NO  NO | NO  NO  NO  NO  NO  NO |
| Staff | StaffID  Staff Name  Staff Gender  StaffAddress  StaffPhone  Salary | Identifikasi staff  Nama staff  Jenis kelamin staff  Alamat staff  Nomor telepon staff  Gaji staff | CHAR(5)  VARCHAR(50)  VARCHAR(10)  VARCHAR(100)  VARCHAR(20)  NUMERIC(11,2) | NO  NO  NO  NO  NO  NO | NO  NO  NO  NO  NO  NO |
| Doctor | DoctorID  DoctorName  DoctorGender  DoctorAddress  DoctorPhone  DoctorSalary  SpecialistID | Identifikasi dokter  Nama dokter  Jenis kelamin dokter  Alamat dokter  Nomor telepon dokter  Gaji dokter  Identifikasi spesialisasi | CHAR(5)  VARCHAR(50)  VARCHAR(10)  VARCHAR(100)  VARCHAR(20)  NUMERIC(11,2)  CHAR(5) | NO  NO  NO  NO  NO  NO  NO | NO  NO  NO  NO  NO  NO  NO |
| Specialization | SpecialistID  Category | Identifikasi spesialisasi  Kategori spesialisasi | CHAR(5)  VARCHAR(50) | NO  NO | NO  NO |
| TreatmentType | TreatmentTypeID  TreatmentTypeName | Identifikasi jenis treatment  Nama jenis treatment | CHAR(5)  VARCHAR(20) | NO  NO | NO  NO |
| Treatment | TreatmentID  TreatmentName  TreatmentPrice  TreatmentTypeID | Identifikasi treatment  Nama treatment  Harga treatment  Identifikasi jenis treatment | CHAR(5)  VARCHAR(50)  NUMERIC(11,2)  CHAR(5) | NO  NO  NO  NO | NO  NO  NO  NO |
| DetailSchedule | ShiftID  Day  TimeStart  TimeEnd | Identifikasi shift  Hari shift  Waktu shift mulai  Waktu shift berakhir | CHAR(5)  VARCHAR(10)  VARCHAR(10)  VARCHAR(10) | NO  NO  NO  NO | NO  NO  NO  NO |
| Schedule | ScheduleID  TransactionID  ShiftID  ScheduleDate | Identifikasi jadwal  Identifikasi transaksi  Identifikasi shift  Tanggal pemeriksaan | CHAR(5)  CHAR(5)  CHAR(5)  DATE | NO  NO  NO  NO | NO  NO  NO  NO |
| DetailTransaction | TransactionID  TreatmentID | Identifikasi transaksi  Identifikasi treatment | CHAR(5)  CHAR(5) | NO  NO | NO  NO |
| Transaction | TransactionID  PatientID  StaffID  DoctorID  TransactionDate  PaymentType | Identifikasi transaksi  Identifikasi pasien  Identifikasi staff  Identifikasi dokter  Tanggal transaksi  Tipe pembayaran | CHAR(5)  CHAR(5)  CHAR(5)  CHAR(5)  DATE  VARCHAR(10) | NO  NO  NO  NO  NO  NO | NO  NO  NO  NO  NO  NO |
| Result | ResultID  TransactionID  ResultDate  Result  Comment | Identifikasi hasil  Identifikasi transaksi  Tanggal keluar hasil  Hasil  Komentar | CHAR(5)  CHAR(5)  DATE  VARCHAR(50)  VARCHAR(100) | NO  NO  NO  NO  YES | NO  NO  NO  NO  NO |
| Tracing | PatientID  RelatedName  RelatedPhone | Identifikasi pasien  Nama berkaitan  Nomor telepon berkaitan | CHAR(5)  VARCHAR(50)  VARCHAR(20) | NO  NO  NO | NO  NO  NO |

Tabel 4.3. Attributes

4.1.4. Attribute Domain

* Attribute domain pada tabel Patient yang memenuhi PatientID adalah constraint yang bernama PatientID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “PT” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Patient yang memenuhi PatientGender adalah constraint yang bernama PatientGender dimana constraint ini hanya berisi string “Male” untuk laki-laki dan “Female” untuk perempuan.
* Attribute domain pada tabel Staff yang memenuhi StaffID adalah constraint yang bernama StaffID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “SF” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Staff yang memenuhi StaffGender adalah constraint yang bernama StaffGender dimana constraint ini hanya berisi string “Male” untuk laki-laki dan “Female” untuk perempuan.
* Attribute domain pada tabel Doctor yang memenuhi DoctorID adalah constraint yang bernama DoctorID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “DR” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Doctor yang memenuhi DoctorGender adalah constraint yang bernama DoctorGender dimana constraint ini hanya berisi string “Male” untuk laki-laki dan “Female” untuk perempuan.
* Attribut domain pada tabel Doctor yang memenuhi SpecialistID adalah constraint yang bernama SpecialistID dimana constraint ini menghubungkan SpecialistID yang berada pada tabel Doctor dengan SpecialistID yang ada pada tabel Transaction.
* Attribute domain pada tabel Specialization yang memenuhi SpecialistID adalah constraint yang bernama SpecialistID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “SP” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel TreatmentType yang memenuhi TreatmentTypeID adalah constraint yang bernama TreatmentTypeID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “TT” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Treatment yang memenuhi TreatmentID adalah constraint yang bernama TreatmentID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “TM” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribut domain pada tabel Treatment yang memenuhi TreatmentTypeID adalah constraint yang bernama TreatmentTypeID dimana constraint ini menghubungkan TreatmentTypeID yang berada pada tabel Treatment dengan TreatmentTypeID yang ada pada tabel TreatmentType.
* Attribute domain pada tabel Transaction yang memenuhi TransactionID adalah constraint yang bernama TransactionID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “TR” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribut domain pada tabel Transaction yang memenuhi PatientID adalah constraint yang bernama PatientID dimana constraint ini menghubungkan PatientID yang berada pada tabel Transaction dengan PatientID yang ada pada tabel Patient.
* Attribut domain pada tabel Transaction yang memenuhi StaffID adalah constraint yang bernama StaffID dimana constraint ini menghubungkan StaffID yang berada pada tabel Transaction dengan StaffID yang ada pada tabel Staff.
* Attribut domain pada tabel Transaction yang memenuhi DoctorID adalah constraint yang bernama DoctorID dimana constraint ini menghubungkan DoctorID yang berada pada tabel Transaction dengan DoctorID yang ada pada tabel Doctor.
* Attribute domain pada tabel DetailTransaction yang memenuhi TransactionID adalah constraint yang bernama TransactionID dimana constraint ini menghubungkan TransactionID yang berada pada tabel DetailTransaction dengan TransactionID yang ada pada tabel Transaction.
* Attribute domain pada tabel DetailTransaction yang memenuhi TreatmentID adalah constraint yang bernama TreatmentID dimana constraint ini menghubungkan TreatmentID yang berada pada tabel DetailTransaction dengan TreatmentID yang ada pada tabel Treatment.
* Attribute domain pada tabel DetailSchedule yang memenuhi ShiftID adalah constraint yang bernama ShiftID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “SH” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Schedule yang memenuhi ScheduleID adalah constraint yang bernama ScheduleID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “SC” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Schedule yang memenuhi TransactionID adalah constraint yang bernama TransactionID dimana constraint ini menghubungkan TransactionID yang berada pada tabel Schedule dengan TransactionID yang ada pada tabel Transaction.
* Attribute domain pada tabel Schedule yang memenuhi ShiftID adalah constraint yang bernama ShiftID dimana constraint ini menghubungkan ShiftID yang berada pada tabel Schedule dengan ShiftID yang ada pada tabel DetailSchedule.
* Attribute domain pada tabel Result yang memenuhi ResultID adalah constraint yang bernama ResultID dimana constraint ini memiliki panjang 5 karakter dimana karakter pertama dan kedua harus berupa huruf “RS” dan karakter selanjutnya berupa angka dari 0 hingga 999.
* Attribute domain pada tabel Result yang memenuhi TransactionID adalah constraint yang bernama TransactionID dimana constraint ini menghubungkan TransactionID yang berada pada tabel Result dengan TransactionID yang ada pada tabel Transaction.
* Attribute domain pada tabel Tracing yang memenuhi PatientID adalah constraint yang bernama PatientID dimana constraint ini menghubungkan PatientID yang berada pada tabel Tracing dengan PatientID yang ada pada tabel Patient.

4.1.5. Candidate Keys, Primary Keys, and Alternate Keys

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name | Candidate Key | Primary Key | Alternate Key |
| Patient | PatientID | PatientID | x |
| Staff | StaffID | StaffID | x |
| Doctor | DoctorID  SpecialistID | DoctorID | SpecialistID |
| Specialization | SpecialistID | SpecialistID | x |
| TreatmentType | TreatmentTypeID | TreatmentTypeID | x |
| Treatment | TreatmentID  TreatmentTypeID | TreatmentID | TreatmentTypeID |
| DetailSchedule | ShiftID | ShiftID | x |
| Schedule | ScheduleID  ShiftID  TransactionID | ScheduleID | ShiftID  TransactionID |
| DetailTransaction | TransactionID  TreatmentID | TransactionID  TreatmentID | x |
| Transactions | TransactionID  PatientID  StaffID  DoctorID | TransactionID | PatientID  StaffID  DoctorID |
| Result | ResultID  TransactionID | ResultID | TransactionID |
| Tracing | PatientID | x | PatientID |

Tabel 4.4. Candidate Keys, Primary Keys, and Alternate Keys

4.1.6 Redundant Relationship

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Index Name | Index | Type | |
| Clustered | Non-Clustered |
| Patient | [PK\_\_Patient\_\_970EC366A2E113B0] | PatientID | ✔️ |  |
| Staff | [PK\_\_Staff\_\_96D4AB173A8F01A8] | StaffID | ✔️ |  |
| Doctor | [PK\_\_Doctor\_\_2DC00EBF799058B2] | DoctorID | ✔️ |  |
| Specialization | [PK\_\_Speciali\_\_7092086E0F436AEA] | SpecialistID | ✔️ |  |
| TreatmentType | [PK\_\_Treatmen\_\_F3EDEE9990902D6B] | TreatmentTypeID | ✔️ |  |
| Treatment | [PK\_\_Treatmen\_\_1A57B7F13A707402] | TreatmentID | ✔️ |  |
| DetailSchedule | [PK\_\_DetailSc\_\_C0A83881E82F7501] | ShiftID | ✔️ |  |
| Schedule | [PK\_\_Schedule\_\_9C8A5B49B1B2FAE8] | ScheduleID | ✔️ |  |
| DetailTransaction | [PK\_\_DetailTr\_\_54E6411426E767FE] | TransactionID  TreatmentID | ✔️ |  |
| Transactions | [PK\_\_Transact\_\_55433A6B5615A68F] | TransactionID | ✔️ |  |
| Result | [PK\_\_Result\_\_97690208654489CF] | ResultID | ✔️ |  |
| Tracing | - | - |  | ✔️ |

Tabel 4.5. Redundant Relationship

4.1.7. ERD Conceptual

Diagram, schematic

Description automatically generated

Gambar 4.1. ERD Conceptual

4.2. Logical Model (Class Diagram)

Diagram, schematic

Description automatically generated

Gambar 4.2. Class Diagram

4.3 Physical Model

4.3.1. List of Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity Name | Index Name | Index | Type | |
| Clustered | Non-Clustered |
| Patient | [PK\_\_Patient\_\_970EC366A2E113B0] | PatientID | ✔️ |  |
| Staff | [PK\_\_Staff\_\_96D4AB173A8F01A8] | StaffID | ✔️ |  |
| Doctor | [PK\_\_Doctor\_\_2DC00EBF799058B2] | DoctorID | ✔️ |  |
| Specialization | [PK\_\_Speciali\_\_7092086E0F436AEA] | SpecialistID | ✔️ |  |
| TreatmentType | [PK\_\_Treatmen\_\_F3EDEE9990902D6B] | TreatmentTypeID | ✔️ |  |
| Treatment | [PK\_\_Treatmen\_\_1A57B7F13A707402] | TreatmentID | ✔️ |  |
| DetailSchedule | [PK\_\_DetailSc\_\_C0A83881E82F7501] | ShiftID | ✔️ |  |
| Schedule | [PK\_\_Schedule\_\_9C8A5B49B1B2FAE8] | ScheduleID | ✔️ |  |
| DetailTransaction | [PK\_\_DetailTr\_\_54E6411426E767FE] | TransactionID  TreatmentID | ✔️ |  |
| Transactions | [PK\_\_Transact\_\_55433A6B5615A68F] | TransactionID | ✔️ |  |
| Result | [PK\_\_Result\_\_97690208654489CF] | ResultID | ✔️ |  |
| Tracing | - | - |  | ✔️ |

Tabel 4.6. List of Index

4.3.2. Analyze Transaction

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Transaction / Relation | Patient | | | | Staff | | | | Doctor | | | | Admin | | | |
| R | U | D | I | R | U | D | I | R | U | D | I | R | U | D | I |
| Patient | v | v |  |  | v | v |  |  | v |  |  |  | v | v | v | v |
| Staff |  |  |  |  | v | v |  |  | v |  |  |  | v | v | v | v |
| Doctor |  |  |  |  | v |  |  |  | v | v |  |  | v | v | v | v |
| Specialization |  |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| TreatmentType | v |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| Treatment | v |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| DetailSchedule | v |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| Schedule | v |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| DetailTransaction |  |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| Transaction | v |  |  |  | v |  |  |  | v |  |  |  | v | v | v | v |
| Result | v |  |  |  | v |  |  |  | v | v |  | v | v | v | v | v |
| Tracing | v | v | v | v | v |  |  |  | v |  |  |  | v | v | v | v |

Tabel 4.7. Analyze Transaction (R = Read, U = Update, D = Delete, I = Insert)

4.3.3. User View, Procedure, and Function

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Gambar 4.3. User View, Procedure, and Function

DAFTAR PUSTAKA

[1] G. Tugas and P. Penanganan, “Penanganan Pandemi Covid-19,” vol. 19, p. 32, 2019, [Online]. Available: https://covid19.go.id/storage/app/media/Protokol/Protokol Percepatan Penanganan Pandemi Corona Virus Disease 2019.pdf.

[2] J. Desember and T. Andrasto, “Pengembangan Sistem Database Hasil Penelitian Dan Pengabdian Kepada Masyarakat Dosen Unnes,” *J. Tek. Elektro Unnes*, vol. 5, no. 2, 2013, doi: 10.15294/jte.v5i2.3556.

[3] T. Connolly and C. Begg, *Pearson.Database.Systems.A.Practical.Approach.to.Design.Implementation.and.Management.6th.Global.Edition.1292061189.pdf*. 2014.

[4] Ir. Harianto Kristanto, “Konsep & Perancangan Database,” *ANDI Yogyakarta*. 1993, [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=tW0MAPGoaVAC&oi=fnd&pg=PA1&dq=sistem+database&ots=Qip1TnLhYK&sig=3979lgaASVCLcRyG9mm1SEH8PSk&redir\_esc=y#v=onepage&q=sistem database&f=false.

[5] Kementerian Kesehatan RIDirektorat Jenderal Pencegahan dan Pengendalian Penyakit, “Panduan SingkatPelacakan Kontak (Contact Tracing) untukKasus COVID-19,” *Kementeri. Kesehat. RIDirektorat Jenderal Pencegah. dan Pengendali. Penyakit*, pp. 1–23, 2020, [Online]. Available: https://www.kemkes.go.id/resources/download/info-terkini/COVID-19/Contact Tracing mobile size revisi7.pdf.

# LAMPIRAN

1. Create Tables:

CREATE TABLE Patient

(

PatientId CHAR(5) PRIMARY KEY NOT NULL CHECK(PatientId LIKE 'PT[0-9][0-9][0-9]'),

PatientName VARCHAR(50) NOT NULL,

PatientGender VARCHAR(10) NOT NULL CHECK(PatientGender IN ('Male', 'Female')),

PatientAge NUMERIC(3) NOT NULL,

PatientAddress VARCHAR(100) NOT NULL,

PatientPhone VARCHAR(20) NOT NULL,

);

CREATE TABLE Staff

(

StaffId CHAR(5) PRIMARY KEY NOT NULL CHECK(StaffId LIKE 'SF[0-9][0-9][0-9]'),

StaffName VARCHAR(50) NOT NULL,

StaffGender VARCHAR(10) NOT NULL CHECK(StaffGender IN ('Male', 'Female')),

StaffAddress VARCHAR(100) NOT NULL,

StaffPhone VARCHAR(20) NOT NULL,

StaffSalary NUMERIC (11,2) NOT NULL,

);

CREATE TABLE Specialization

(

SpecialistId CHAR(5) PRIMARY KEY NOT NULL CHECK(SpecialistId LIKE 'SP[0-9][0-9][0-9]'),

Category VARCHAR(50) NOT NULL,

);

CREATE TABLE Doctor

(

DoctorId CHAR(5) PRIMARY KEY NOT NULL CHECK(DoctorId LIKE 'DR[0-9][0-9][0-9]'),

DoctorName VARCHAR(50) NOT NULL,

DoctorGender VARCHAR(10) NOT NULL CHECK(DoctorGender IN ('Male', 'Female')),

DoctorAddress VARCHAR(100) NOT NULL,

DoctorPhone VARCHAR(20) NOT NULL,

DoctorSalary NUMERIC (11,2) NOT NULL,

SpecialistId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Specialization(SpecialistId),

);

CREATE TABLE TreatmentType

(

TreatmentTypeId CHAR(5) PRIMARY KEY NOT NULL

CHECK(TreatmentTypeId LIKE 'TT[0-9][0-9][0-9]'),

TreatmentTypeName VARCHAR(20) NOT NULL,

);

CREATE TABLE Treatment

(

TreatmentId CHAR(5) PRIMARY KEY NOT NULL CHECK(TreatmentId LIKE 'TM[0-9][0-9][0-9]'),

TreatmentName VARCHAR(50) NOT NULL,

TreatmentPrice NUMERIC(11,2) NOT NULL,

TreatmentTypeId CHAR(5) NOT NULL FOREIGN KEY REFERENCES

TreatmentType(TreatmentTypeId) ON UPDATE CASCADE ON DELETE CASCADE,

);

CREATE TABLE Transactions

(

TransactionId CHAR(5) PRIMARY KEY NOT NULL CHECK (TransactionId LIKE 'TR[0-9][0-9][0-9]'),

PatientId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Patient(PatientId) ON UPDATE CASCADE ON DELETE CASCADE,

StaffId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Staff(StaffId) ON UPDATE CASCADE ON DELETE CASCADE,

DoctorId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Doctor(DoctorId) ON UPDATE CASCADE ON DELETE CASCADE,

TransactionDate DATE NOT NULL,

PaymentType VARCHAR(10) NOT NULL,

);

CREATE TABLE DetailTransaction

(

PRIMARY KEY (TransactionId, TreatmentId),

TransactionId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Transactions(TransactionId) ON UPDATE CASCADE ON DELETE CASCADE,

TreatmentId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Treatment(TreatmentId) ON UPDATE CASCADE ON DELETE CASCADE,

);

CREATE TABLE DetailSchedule

(

ShiftId CHAR(5) PRIMARY KEY NOT NULL CHECK(ShiftId LIKE 'SH[0-9][0-9][0-9]'),

Day VARCHAR(10) NOT NULL,

TimeStart VARCHAR(10) NOT NULL,

TimeEnd VARCHAR(10) NOT NULL,

);

CREATE TABLE Schedule

(

ScheduleId CHAR(5) PRIMARY KEY NOT NULL CHECK(ScheduleId LIKE 'SC[0-9][0-9][0-9]'),

TransactionId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Transactions(TransactionId) ON UPDATE CASCADE ON DELETE CASCADE,

ShiftId CHAR(5) NOT NULL FOREIGN KEY REFERENCES DetailSchedule(ShiftId) ON UPDATE CASCADE ON DELETE CASCADE,

ScheduleDate DATE NOT NULL,

);

CREATE TABLE Result

(

ResultId CHAR(5) PRIMARY KEY NOT NULL CHECK(ResultId LIKE 'RS[0-9][0-9][0-9]'),

TransactionId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Transactions(TransactionId) ON UPDATE CASCADE ON DELETE CASCADE,

ResultDate DATE NOT NULL,

TestResult VARCHAR(50) NOT NULL,

Comment VARCHAR(100),

);

CREATE TABLE Tracing

(

PatientId CHAR(5) NOT NULL FOREIGN KEY REFERENCES Patient(PatientId) ON UPDATE CASCADE ON DELETE CASCADE,

RelatedName VARCHAR(50) NOT NULL,

RelatedPhone VARCHAR(20) NOT NULL,

);

1. Create View

-- View All Related With Positive Patient

CREATE VIEW [ViewRelatedContact] AS

SELECT

Tracing.PatientId [Patient ID],

Patient.PatientName [Patient Name],

Patient.PatientPhone [Patient Phone],

Result.ResultDate [First Day Of Isolation],

DATEADD(day, 14, Result.ResultDate) [Until The Day After 2 Weeks],

Staff.StaffId [Staff ID],

Staff.StaffName [Staff Name],

Staff.StaffPhone [Staff Phone],

Doctor.DoctorId [Doctor ID],

Doctor.DoctorName [Doctor Name],

Doctor.DoctorPhone [Doctor Phone],

Tracing.RelatedName [Other Related Person],

Tracing.RelatedPhone [Other's Phone]

FROM Tracing

INNER JOIN Patient ON Tracing.PatientId = Patient.PatientId

INNER JOIN Transactions ON Patient.PatientId = Transactions.PatientId

INNER JOIN Staff ON Transactions.StaffId = Staff.StaffId

INNER JOIN Doctor ON Transactions.DoctorId = Doctor.DoctorId

INNER JOIN Result ON Transactions.TransactionId = Result.TransactionId

Table

Description automatically generated

-- View All Treatment Category, Name, and Price

CREATE VIEW [ViewTreatmentList] AS

SELECT

TreatmentType.TreatmentTypeName [Treatment Category],

Treatment.TreatmentName [Treatment Name],

Treatment.TreatmentPrice [Price]

FROM TreatmentType

INNER JOIN Treatment ON TreatmentType.TreatmentTypeId = Treatment.TreatmentTypeId

SELECT \* FROM ViewTreatmentList

Graphical user interface, text, application

Description automatically generated

-- View Treatment Taken By Patient

CREATE VIEW [ViewPatientTreatment] AS

SELECT

Patient.PatientId [Patient ID],

Patient.PatientName [Patient Name],

Treatment.TreatmentName [Treatment Name],

Treatment.TreatmentPrice [Price],

Transactions.TransactionDate [Transaction Date]

FROM Patient

INNER JOIN Transactions ON Patient.PatientId = Transactions.PatientId

INNER JOIN DetailTransaction ON Transactions.TransactionId = DetailTransaction.TransactionId

INNER JOIN Treatment ON DetailTransaction.TreatmentId = Treatment.TreatmentId

SELECT \* FROM ViewPatientTreatment

Graphical user interface, text, application

Description automatically generated

1. Create Procedure

-- Procedure 1

create procedure show\_nota @nama varchar(50)

as

begin

select

transactions.TransactionId,

patient.PatientId,

patient.PatientName,

transactions.TransactionDate,

treatment.TreatmentName,

treatment.TreatmentPrice,

schedule.ScheduleDate

from

Patient,

Transactions,

DetailTransaction,

Treatment,

Schedule

where

patient.PatientId = Transactions.PatientId and

transactions.TransactionId = DetailTransaction.TransactionId and

DetailTransaction.TreatmentId = Treatment.TreatmentId and

transactions.TransactionId = schedule.TransactionId and

patientname = @nama

end;

exec show\_nota @nama = 'Mahesa Hakim';

exec show\_nota @nama = 'Nyoman Pranowo';

Graphical user interface, text, application

Description automatically generated

-- Procedure 2

create procedure show\_result @nama2 varchar(50)

as

begin

select

patient.PatientId,

patient.PatientName,

result.ResultDate,

treatment.TreatmentName,

result.TestResult,

result.Comment,

doctor.DoctorName

from

Patient,

Transactions,

Result,

Doctor,

DetailTransaction,

Treatment

where

patient.PatientId = transactions.PatientId and

transactions.TransactionId = Result.TransactionId and

transactions.DoctorId = doctor.DoctorId and

transactions.TransactionId = DetailTransaction.TransactionId and

DetailTransaction.TreatmentId = Treatment.TreatmentId

end;

exec show\_result @nama2 = 'Mahesa Hakim'

Graphical user interface, text, application

Description automatically generated

-- PROCEDURE 3, liat pasien

CREATE PROCEDURE SelectConfirmedPatients @InputTTID CHAR(5) AS

SELECT pt.PatientId, PatientName, PatientGender, PatientAge, PatientAddress, PatientPhone, TreatmentName FROM Patient pt, Transactions tr, DetailTransaction dt, Treatment tm, TreatmentType tt

WHERE pt.PatientId = tr.PatientId AND tr.TransactionId = dt.TransactionId AND dt.TreatmentId = tm.TreatmentId AND tm.TreatmentTypeId = tt.TreatmentTypeId

AND tm.TreatmentTypeID = @InputTTID

GO

EXEC SelectConfirmedPatients @InputTTID = 'TT007'

Graphical user interface, text, application

Description automatically generated

-- PROCEDURE 4, liat dokter dan spesialisasinya

CREATE PROCEDURE SelectSpecializedDoctor @InputCategory VARCHAR(50) AS

SELECT DoctorId, DoctorName, DoctorGender, DoctorAddress, DoctorPhone, DoctorSalary, Category

FROM Doctor dc, Specialization sp

WHERE dc.SpecialistId = sp.SpecialistId AND sp.Category = @InputCategory

GO

EXEC SelectSpecializedDoctor @InputCategory = 'Penyakit Dalam'

Graphical user interface, text, application

Description automatically generated

1. Create Trigger

-- Trigger 1

CREATE TRIGGER TrigTrans ON Transactions

AFTER INSERT AS

BEGIN

SELECT TOP 1 \*

FROM Transactions ORDER BY TransactionID DESC

END

BEGIN TRAN

INSERT INTO Transactions

VALUES ('TR005', 'PT004', 'SF005', 'DR006', '2021/01/07', 'Cash')

ROLLBACK

SELECT \* FROM Transactions

Graphical user interface, text, application

Description automatically generated

-- Trigger 2: detail transaction

CREATE TRIGGER TrigDetTrans ON DetailTransaction

AFTER INSERT AS

BEGIN

SELECT TOP 1 \*

FROM DetailTransaction ORDER BY TransactionID DESC

END

BEGIN TRAN

INSERT INTO DetailTransaction

VALUES ('TR005', 'TM018')

ROLLBACK

SELECT \* FROM DetailTransaction

Graphical user interface, text, application

Description automatically generated

-- Trigger 3: schedule

CREATE TRIGGER TrigSch ON Schedule

AFTER INSERT AS

BEGIN

SELECT TOP 1 \*

FROM Schedule

ORDER BY ScheduleId DESC

END

BEGIN TRAN

INSERT INTO Schedule

VALUES ('SC005', 'TR005', 'SH005', '2021/01/07')

ROLLBACK

SELECT \* FROM Schedule

Graphical user interface, text, application

Description automatically generated

--4

-- result

CREATE TRIGGER TrigRes ON Result

AFTER INSERT AS

BEGIN

SELECT TOP 1 \* FROM Result

END

BEGIN TRAN

INSERT INTO Result

VALUES ('RS005', 'TR005', '2021/01/08', 'Negative', 'Mohon dijaga kesehatannya')

ROLLBACK

SELECT \* FROM Result

Graphical user interface, application

Description automatically generated

1. Function

-- Function 1

select

patient.PatientName,

tracing.RelatedName,

tracing.RelatedPhone,

doctor.DoctorName,

staff.StaffName,

result.TestResult,

result.Comment

from tracing, patient, transactions, Doctor, Staff, Result

where

tracing.PatientId = patient.PatientId and

patient.PatientId = transactions.PatientId and

transactions.DoctorId = Doctor.DoctorId and

Transactions.StaffId = Staff.StaffId and

Transactions.TransactionId = Result.TransactionId

Graphical user interface, text, application

Description automatically generated

-- Function 2

SELECT

TreatmentType.TreatmentTypeName [Treatment Type],

Treatment.TreatmentName [Treatment Name],

Treatment.TreatmentPrice [Price]

FROM TreatmentType

INNER JOIN Treatment ON TreatmentType.TreatmentTypeId = Treatment.TreatmentTypeId

WHERE Treatment.TreatmentPrice < 1000000

Graphical user interface, application, Word

Description automatically generated

-- Function 3

SELECT

Transactions.TransactionId [Transaction ID],

Patient.PatientName [Patient Name],

Transactions.TransactionDate [Transaction Date],

Transactions.PaymentType [Payment Type]

FROM Transactions

INNER JOIN Patient ON Transactions.PatientId = Patient.PatientId

WHERE PaymentType LIKE 'Cash'

Graphical user interface, text, application

Description automatically generated

# **LEMBAR PENILAIAN**

**JUDUL**

MATA KULIAH ISYS6028 – DATABASE SYSTEMS

KELAS LA20

Semester Ganjil 2020

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DAFTAR MAHASISWA** | **NILAI** | | | | **BOBOT** | | | | **KREDIT** | | | | **TOTAL KREDIT** |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
|  |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
|  |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
|  |  |  |  |  | 20% | 20% | 24% | 36% |  |  |  |  |  |
| **TOTAL** | | | | | | | | | | | | |  |

**KETERANGAN :**

* **Skala Penilaian : 0 sd 100**
* **Komponen**

1 : Laporan

2 : Produk

3 : Pengetahuan

4 : Solusi