**Laporan Akhir Project Basis Data**



**Data Kelompok**

|  |  |
| --- | --- |
| **Nama Lengkap** | **NIM** |
| Diah Ayu Puspasari | 0110223052 |
| Nur fadillah | 0110223075 |
| Fatiyah Labibah |  |
| Muhammad Fajar |  |
| Muhammad Zen Alby |  |

merancang database Rumah sakit

**STT - TERPADU NURUL FIKRI**

**TAHUN AJARAN 2023**

**semester GENAP**

**DAFTAR ISI**

[**Latar Belakang** 3](#_Toc169558311)

[**Deskripsi Tugas Akhir** 3](#_Toc169558312)

[**Desain Database dengan mysql workbench** 3](#_Toc169558313)

[**Struktur Database** 4](#_Toc169558314)

[**1.** **Tabel Divisi** 4](#_Toc169558315)

[**2.** **Tabel Pegawai** 4](#_Toc169558316)

[**3.** **Tabel Pasien** 4](#_Toc169558317)

[**4.** **Tabel Dokter** 5](#_Toc169558318)

[**5.** **Tabel Pendaftaran** 5](#_Toc169558319)

[**6.** **Tabel Periksa** 6](#_Toc169558320)

[**7.** **Tabel Ruangan** 6](#_Toc169558321)

[**8.** **Tabel Rawat Inap** 6](#_Toc169558322)

[**9.** **Tabel Obat** 7](#_Toc169558323)

[**10.** **Tabel Pembayaran** 7](#_Toc169558324)

[**Relasi Table** 8](#_Toc169558325)

[**Tabel divisi** 8](#_Toc169558326)

[**Tabel Pegawai** 8](#_Toc169558327)

[**Tabel Pasien** 8](#_Toc169558328)

[**Tabel Dokter** 8](#_Toc169558329)

[**Tabel Periksa** 8](#_Toc169558330)

[**Tabel Pendaftaran** 9](#_Toc169558331)

[**Tabel Ruangan** 9](#_Toc169558332)

[**Tabel Rawat Inap** 9](#_Toc169558333)

[**Tabel Obat** 9](#_Toc169558334)

[**Tabel Pembayaran** 9](#_Toc169558335)

# **Latar Belakang**

Kesehatan merupakan salah satu aspek terpenting dalam kehidupan manusia. Rumah sakit sebagai fasilitas kesehatan memainkan peran vital dalam memberikan layanan medis kepada masyarakat. Dalam era digital saat ini, pengelolaan data yang efisien dan akurat menjadi sangat krusial untuk meningkatkan kualitas pelayanan dan operasional rumah sakit. Salah satu cara untuk mencapai efisiensi tersebut adalah dengan menerapkan sistem database yang komprehensif dan terstruktur dengan baik.

Sistem database rumah sakit mencakup berbagai komponen penting seperti data pasien, data dokter, catatan pemeriksaan medis, informasi divisi dan pegawai, manajemen ruangan, catatan rawat inap, informasi obat-obatan, serta sistem pembayaran. Dengan adanya sistem database yang terintegrasi, rumah sakit dapat mengelola data dengan lebih baik, mempercepat proses pengambilan keputusan, dan meningkatkan koordinasi antar departemen.

# **Deskripsi Tugas Akhir**

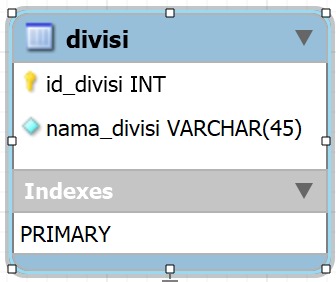
Rumah sakit sebagai lembaga pelayanan kesehatan memiliki tanggung jawab besar dalam memberikan layanan medis kepada masyarakat. Dengan perkembangan teknologi informasi, penggunaan sistem database yang efektif dapat membantu rumah sakit dalam mengelola data pasien, dokter, pemeriksaan, obat, dan berbagai informasi penting lainnya. Database yang baik dapat meningkatkan efisiensi operasional, mengurangi kesalahan manusia, dan mempercepat pengambilan keputusan.

Tujuan dari proyek ini adalah untuk merancang dan mengimplementasikan sistem database yang terintegrasi dan efisien untuk mendukung operasional rumah sakit. Sistem ini diharapkan dapat mengelola berbagai jenis data yang ada di rumah sakit secara efektif dan memberikan kemudahan dalam pengolahan serta akses data.

# **Desain Database dengan mysql workbench**

# **Struktur Database**

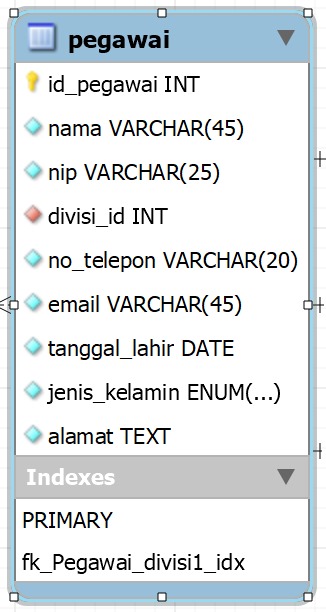
## **Tabel Divisi**

Kolom :

* id\_divisi : Tipe data integer, primary key, auto increatment, not null.
* nama\_divisi : Tipe data varchar untuk menyimpan nama divisi.

Deskripsi : Untuk menyimpan nama pegawai yang ada di rumah sakit.

## **Tabel Pegawai**

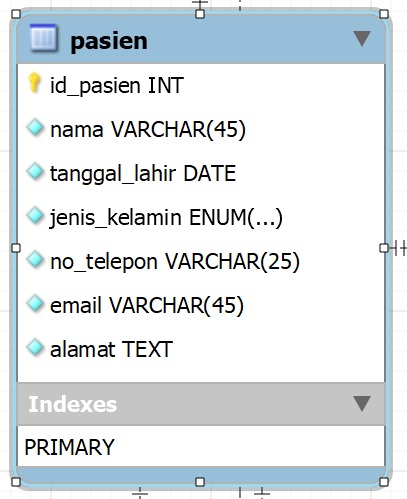
Kolom :

* id\_pegawai : Tipe data integer, primary key, auto increatment
* nama : Tipe data varchar untuk menyimpan nama pegawai.
* nip : Tipe data varchar untuk menyimpan nip pegawai.
* divisi\_id : Tipe data integer untuk memanggil dan menyimpan id divisi pegawai yang sudah ditentukan.
* no\_telepon : Tipe data varchar untuk menyimpan nomer telepon pegawai.
* email : Tipe data varchar untuk menyimpan email pegawai.
* tanggal\_lahir : Tipe data date untuk menyimpan tanggal lahir pegawai.
* jenis\_kelamin : Tipe data enum (‘Laki-laki’ ‘Perempuan) untuk memilih jenis kelamin pegawai dan menyimpannya.
* alamat : Tipe data text untuk menyimpan alamat pegawai.

Deskripsi :

## **Tabel Pasien**

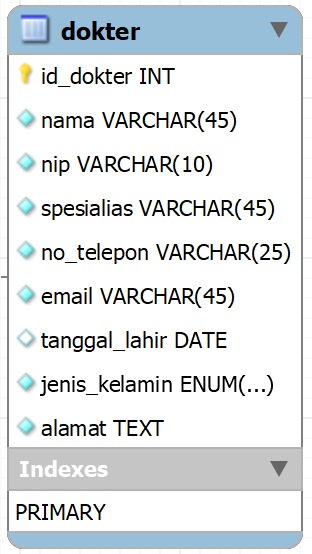
Kolom :

* id\_pasien Tipe data integer, primary key, auto increatment, not null.
* Nama : Tipe data Varchar untuk menyimpan nama pasien
* Tanggal\_lahir : Tipe data Date untuk menyimpan tanggal lahir pasien
* jenis\_kelamin : Tipe data enum (‘Laki-laki’ ‘Perempuan’) untuk memilih jenis kelamin pegawai dan menyimpannya.
* no\_telepon : Tipe data Varchar untuk menyimpan nomor telepon pasien
* email : Tipe data Varchar untuk menyimpan data email pasien
* alamat : Tipe data Text untuk menyimpan alamat pasien

Deskripsi : Untuk menyimpan data pasien yang hendak berobat atau di rawat dirumah sakit

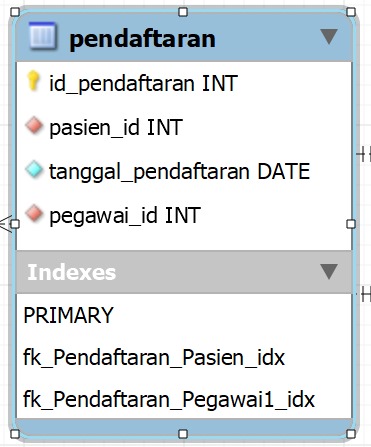
## **Tabel Dokter**

Kolom :

* ****id\_dokter : Tipe data integer, primary key, auto increatment
* nama : Tipe data varchar untuk menyimpan nama dokter.
* nip : Tipe data varchar untuk menyimpan nip dokter.
* spesialias : Tipe data integer untuk menyimpan bagian speasialiasa dokter yang sudah ditentukan.
* no\_telepon : Tipe data varchar untuk menyimpan nomer telepon dokter.
* email : Tipe data varchar untuk menyimpan email dokter.
* tanggal\_lahir : Tipe data date untuk menyimpan tanggal lahir dokter.
* jenis\_kelamin : Tipe data enum (‘Laki-laki’ ‘Perempuan’) untuk memilih jenis kelamin dokter dan menyimpannya.
* alamat : Tipe data text untuk menyimpan alamat dokter.

Deskripsi : Untuk menyimpan data dokter yang melakukan pemeriksaan terhadap pasien yang sudah melakukan pendaftaran

## **Tabel Pendaftaran**

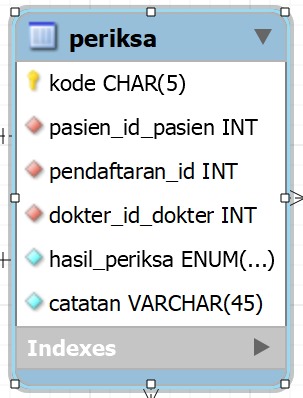
Kolom:

* id\_pendaftaran : Tipe data integer, primary key, auto increatment
* pasien\_id : Tipe data Integer untuk memanggil dan menyimpan id pasien
* tanggal\_pendaftaran : Tipe data Date untuk menyimpan tanggal pendaftaran pasien
* pegawai\_id : Tipe data Integer utnuk memanggil dan menyimpan id pegawai

Deskripsi : Untuk menyimpan data pasien sebelum melakukan periksa kedokter

## **Tabel Periksa**

Kolom :

* Kode : Tipe data char untuk menyimpan kode periksa.
* pasien\_id\_pasien : Tipe data integer untuk memanggil dan menyimpan id pasien.
* pendaftaran\_id :  Tipe data integer untuk memanggil dan menyimpan id pendaftaran.
* dokter\_id :  Tipe data integer untuk memanggil dan menyimpan id dokter.
* hasil\_periksa : Tipe data enum (‘Tidak dirawat’ ‘Rawat’) untuk memilih dan menyimpan hasil periksa dari pasien tersebut.
* catatan : Tipe data varchar untuk menyimpan catatan atau keterangan terkait hasil pemeriksaan.

Deskripsi : Untuk menyimpan data hasil periksa pasien yang sudah diperiksa oleh dokter

## **Tabel Ruangan**

Kolom :

* id\_ruangan :  Tipe data integer, primary key, auto increatment.
* nama\_ruangan : Tipe data varchar untuk menyimpan nama ruangan.
* lokasi : Tipe data varchar untuk menyimpan lokasi ruangan.
* jumlah\_kamar : Tipe data varchar untuk menyimpan jumlah kamar yang tersedia yang ada di ruangan tersebut.

Deskripsi : Untuk menyimpan jumlah data kamar yang tersedia 

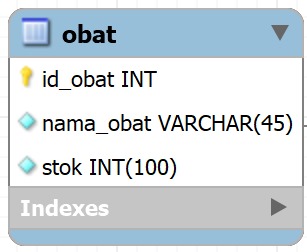
## **Tabel Rawat Inap**

Kolom:

* id\_rawatinap : Tipe data integer, primary key, auto increatment.
* pasien\_id : Tipe data Integer untuk memanggil dan menyimpan id pasien.
* ruangan\_id : Tipe data Integer untuk memanggil dan menyimpan id ruangan.
* tanggal\_masuk : Tipe data Date untuk menyimpan data tanggal masuk pasien yang sudah ditentukan oleh dokter
* tanggal\_keluar : Tipe data Date untuk menyimpan data tanggal keluar pasien yang sudah ditentukan oleh dokter
* catatan : Tipe data Text untuk menyimpan data catatan penyakit pasien
* pegawai\_id : Tipe data Integer utnuk memanggil dan menyimpan id pegawai

Deskripsi : Untuk menyimpan informasi data pasien yang hendak dirawat

## **Tabel Obat**

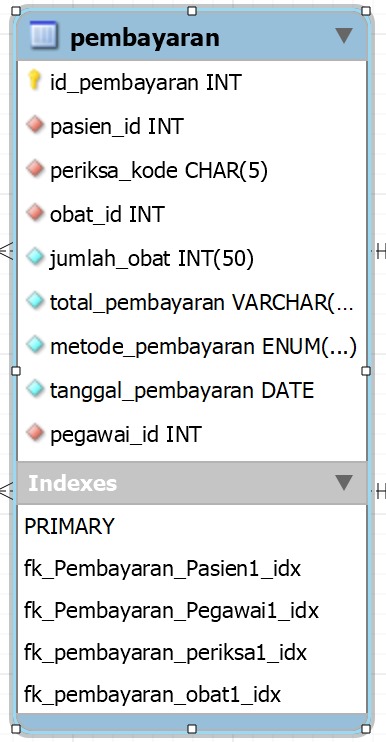
Kolom**:**

* id\_obat : Tipe data integer, primary key, auto increatment.
* nama\_obat : Tipe data varchar untuk menyimpan nama obat.
* stok : Tipe data integer untuk menyimpan data stok obat yang tersedia di Rumah sakit.

Deskripsi : Untuk menyimpan informasi data obat yaitu nama obat dan stoknya yang ada di Rumah sakit.

## **Tabel Pembayaran**

Kolom :

* ****id\_pembayaran : Tipe data integer, primary key, auto increatment.
* pasien\_id : Tipe data Integer untuk memanggil dan menyimpan id pasien.
* periksa\_kode : Tipe data Integer untuk memanggil dan menyimpan kode periksa pasien.
* obat\_id : Tipe data Integer untuk memanggil dan menyimpan id obat yang di peroleh dari hasil periksa
* jumlah\_obat : Tipe data Integer untuk menyimpan data jumlah obat yang dibutuhkan
* total\_pembayaran : Tipe data Varchar untuk menyimpan data pembayaran pasien yang sudah melakukan periksa di rawat ataupun tidak dirawat
* metode\_pembayaran : Tipe data Enum (‘Tunai’, ‘Non-Tunai’) untuk memilih dan menyimpan data pembayaran yang dilakukan oleh pasien
* tanggal\_pembayaran : Tipe data Date untuk menyimpan data tanggal pembayaran yang dilakukan oleh pasien
* pegawai\_id : Tipe data Integer utnuk memanggil dan menyimpan id pegawai

Deskripsi : Untuk menyimpan informasi data pembayaran yang berisikan data pasien dan obat.

# **Relasi Table**

## **Tabel divisi**

Hubungan one-to-many (satu ke banyak) antara divisi dan pegawai menggambarkan bahwa satu divisi dapat memiliki banyak pegawai, tetapi setiap pegawai hanya dapat berada di satu divisi. Ini adalah konsep umum dalam struktur organisasi di mana divisi atau departemen bertanggung jawab atas sejumlah pegawai.

## **Tabel Pegawai**

Hubungan one-to-many (satu ke banyak) antara pegawai dan entitas lain seperti pembayaran, pendaftaran dan rawat inap berarti bahwa satu pegawai dapat memiliki banyak entri pembayaran, pendaftaran dan rawat inap, tetapi setiap entri pembayaran, pendaftaran dan rawat inap  hanya terkait dengan satu pegawai.

Berikut penjelasan lebih lanjut:

* **Pendaftaran:** Setiap pegawai bisa memiliki banyak entri pendaftaran, namun setiap entri pendaftaran hanya terkait dengan satu pegawai.
* **Pembayaran:** Setiap pegawai bisa memiliki banyak entri pembayaran, namun setiap entri pembayaran  hanya terkait dengan satu pegawai.
* **Rawat Inap:** Setiap pegawai bisa memiliki banyak entri rawat inap, namun setiap entri  rawat inap hanya terkait dengan satu pegawai

## **Tabel Pasien**

Hubungan one-to-one antara pasien dan entitas lain seperti pembayaran, pendaftaran, dan rawat inap memastikan bahwa setiap pasien memiliki satu catatan terkait, dan sebaliknya. Ini berarti:

* **Pembayaran:** Setiap pasien memiliki satu entri pembayaran terkait, memastikan pembayaran dicatat sekali untuk setiap pasien.
* **Pendaftaran:** Setiap pasien memiliki satu entri pendaftaran terkait, memastikan pendaftaran dicatat khusus untuk setiap pasien.
* **Rawat Inap:** Setiap pasien memiliki satu entri rawat inap terkait, memastikan rawat inap dicatat satu kali untuk setiap pasien.

Hubungan many-to-many antara pasien dan dokter memungkinkan fleksibilitas dalam pengelolaan data medis, di mana setiap pasien bisa ditangani oleh banyak dokter dan setiap dokter bisa menangani banyak pasien. Tabel penghubung "Periksa" digunakan untuk mencatat detail setiap interaksi, memastikan data tersimpan dengan baik dan dapat diakses dengan mudah.

## **Tabel Dokter**

Hubungan many-to-many (banyak ke banyak) antara dokter dan pasien menggambarkan bahwa setiap dokter dapat merawat banyak pasien, dan setiap pasien dapat dirawat oleh banyak dokter. Untuk mengimplementasikan hubungan ini dalam basis data, digunakan tabel penghubung (junction table) yang disebut "Periksa".

## **Tabel Periksa**

Hubungan one-to-one (satu ke satu) antara "Periksa" dan entitas lain seperti "Pembayaran" dan "Pendaftaran" menggambarkan bahwa setiap entri dalam tabel "Periksa" hanya memiliki satu entri terkait dalam tabel "Pembayaran" dan satu entri terkait dalam tabel "Pendaftaran", dan sebaliknya. Tabel "Periksa" sendiri adalah hasil dari hubungan many-to-many antara "Pasien" dan "Dokter".

* **Pembayaran:** Setiap entri pemeriksaan (Periksa) memiliki satu entri pembayaran yang terkait, dan setiap entri pembayaran hanya terkait dengan satu pemeriksaan.
* **Pendaftaran:** Setiap entri pemeriksaan (Periksa) memiliki satu entri pendaftaran yang terkait, dan setiap entri pendaftaran hanya terkait dengan satu pemeriksaan.

## **Tabel Pendaftaran**

Hubungan one-to-one (satu ke satu) antara "Pendaftaran" dan entitas lain seperti "Periksa" dan "Pasien" menggambarkan bahwa setiap entri dalam tabel "Pendaftaran" hanya memiliki satu entri terkait dalam tabel "periksa" dan satu entri terkait dalam tabel "Pasien", dan sebaliknya.

* **Periksa:** setiap pendaftaran terkait dengan satu pemeriksaan, dan setiap pemeriksaan terkait dengan satu pendaftaran.
* **Pasien:** setiap pendaftaran terkait dengan satu pasien, dan setiap pasien terkait dengan satu pendaftaran.

## **Tabel Ruangan**

Hubungan one-to-one antara tabel "Ruangan" dan tabel "Rawat Inap" berarti setiap entri dalam tabel "Ruangan" hanya terkait dengan satu entri dalam tabel "Rawat Inap", dan sebaliknya. Ini mengindikasikan bahwa setiap ruangan hanya digunakan untuk satu pasien dalam satu periode waktu tertentu, dan setiap pasien hanya ditempatkan di satu ruangan. Dalam konteks perawatan medis, hal ini menunjukkan bahwa setiap pasien dirawat di satu ruangan khusus.

## **Tabel Rawat Inap**

Hubungan one-two-one antara tabel “Pasien” dan tabel “Ruangan” berikut penjelasannya :

* **Rawat Inap ke Pasien:** Setiap entri dalam tabel "Rawat Inap" hanya terkait dengan satu entri dalam tabel "Pasien", dan sebaliknya. Ini menunjukkan bahwa setiap rawat inap hanya terkait dengan satu pasien, dan setiap pasien hanya memiliki satu rawat inap.
* **Rawat Inap ke Ruangan**: Setiap entri dalam tabel "Rawat Inap" hanya terkait dengan satu entri dalam tabel "Ruangan", dan sebaliknya. Ini menunjukkan bahwa setiap rawat inap terkait dengan satu ruangan, dan setiap ruangan hanya digunakan untuk satu rawat inap pada suatu waktu tertentu.

## **Tabel Obat**

Hubungan one-to-many (satu ke banyak) antara tabel "Obat" dan tabel "Pembayaran" berarti Setiap entri dalam tabel "Obat" dapat terkait dengan banyak entri dalam tabel "Pembayaran". Namun, setiap entri dalam tabel "Pembayaran" hanya terkait dengan satu entri dalam tabel "Obat".

Ini menunjukkan bahwa satu transaksi pembayaran dapat melibatkan pembelian banyak jenis obat yang berbeda, tetapi setiap jenis obat hanya terkait dengan satu transaksi pembayaran.

## **Tabel Pembayaran**

Hubungan one-to-one (satu ke satu) antara tabel "Pembayaran" dan tabel "Pasien", serta tabel "Pembayaran" dan tabel "Periksa" berarti:

* Pembayaran ke Pasien: Setiap entri dalam tabel "Pembayaran" hanya terkait dengan satu entri dalam tabel "Pasien", dan sebaliknya. Ini menunjukkan bahwa setiap pembayaran hanya terkait dengan satu pasien, dan setiap pasien hanya memiliki satu entri pembayaran.
* Pembayaran ke Periksa: Setiap entri dalam tabel "Pembayaran" hanya terkait dengan satu entri dalam tabel "Periksa", dan sebaliknya. Ini menunjukkan bahwa setiap pembayaran hanya terkait dengan satu entri pemeriksaan, dan setiap pemeriksaan hanya memiliki satu entri pembayaran.