

TUGAS UTS ANALISIS DESAIN DAN SISTEM

Nama: Suci Asri Novariane

NIM: 12030123120011

Kelas: C

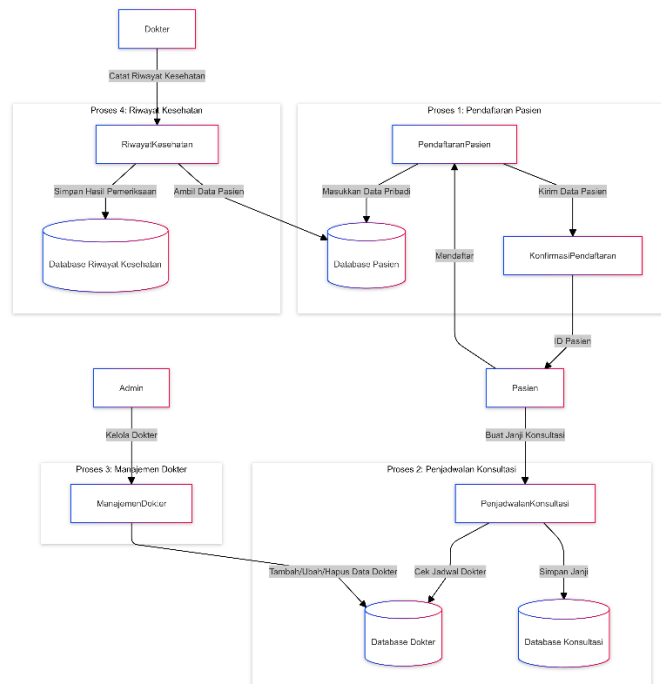
Contoh Kasus:

Kasus Sistem Informasi Rumah Sakit

Sebuah rumah sakit ingin mengembangkan sistem informasi untuk mengelola data pasien, jadwal konsultasi dengan dokter, serta riwayat kesehatan pasien. Sistem ini memiliki beberapa fungsi utama:

- Pendaftaran pasien: Pasien baru dapat mendaftar dengan mengisi data pribadi.
- Penjadwalan konsultasi: Pasien dapat membuat janji temu dengan dokter melalui sistem, memilih dokter dan waktu konsultasi.
- Manajemen dokter: Admin rumah sakit dapat menambah, mengubah, dan menghapus data dokter.
- Riwayat kesehatan: Dokter dapat mencatat hasil pemeriksaan pasien ke dalam sistem.

1. DFD (Data Flow Diagram)



Kode Diagram:

graph TD

%% Entitas Eksternal

Pasien -->|Mendaftar| PendaftaranPasien

Pasien -->|Buat Janji Konsultasi| PenjadwalanKonsultasi

Dokter -->|Catat Riwayat Kesehatan| RiwayatKesehatan

Admin -->|Kelola Dokter| ManajemenDokter

%% Proses Pendaftaran Pasien

subgraph Proses 1: Pendaftaran Pasien

PendaftaranPasien -->|Masukkan Data Pribadi| DatabasePasien

PendaftaranPasien -->|Kirim Data Pasien| KonfirmasiPendaftaran

end

%% Proses Penjadwalan Konsultasi

subgraph Proses 2: Penjadwalan Konsultasi

PenjadwalanKonsultasi -->|Cek Jadwal Dokter| DatabaseDokter

PenjadwalanKonsultasi -->|Simpan Janji| DatabaseKonsultasi

end

%% Proses Manajemen Dokter

subgraph Proses 3: Manajemen Dokter

ManajemenDokter -->|Tambah/Ubah/Hapus Data Dokter| DatabaseDokter

end

%% Proses Riwayat Kesehatan

subgraph Proses 4: Riwayat Kesehatan

RiwayatKesehatan -->|Simpan Hasil Pemeriksaan| DatabaseRiwayat

RiwayatKesehatan -->|Ambil Data Pasien| DatabasePasien

end

%% Database

DatabasePasien[(Database Pasien)]

DatabaseDokter[(Database Dokter)]

DatabaseKonsultasi[(Database Konsultasi)]

DatabaseRiwayat[(Database Riwayat Kesehatan)]

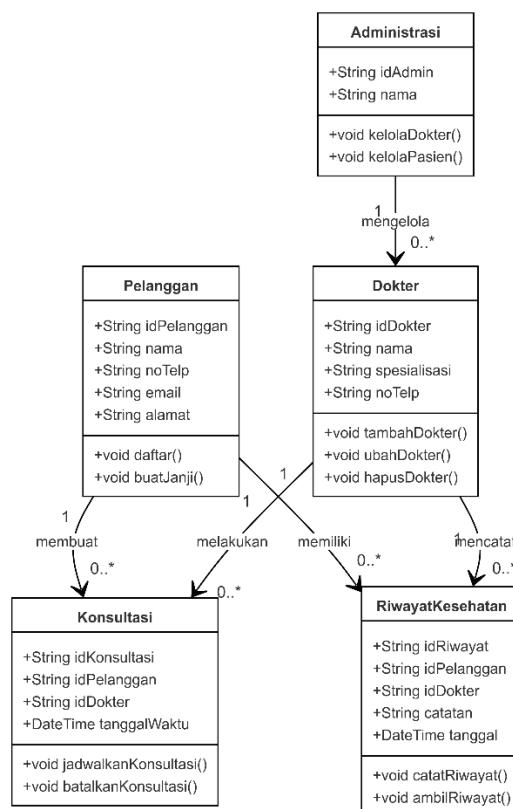
%% Output

KonfirmasiPendaftaran -->|ID Pasien| Pasien

Penjelasan Diagram:

- Pendaftaran Pasien: Pasien baru mengisi data pribadi untuk mendaftar. Data pasien disimpan di Database Pasien dan sistem mengirimkan ID pasien sebagai konfirmasi.
- Penjadwalan Konsultasi: Pasien membuat janji konsultasi dengan memilih dokter dan waktu yang tersedia. Sistem mengecek jadwal dokter dari Database Dokter dan menyimpan janji ke Database Konsultasi.
- Manajemen Dokter: Admin rumah sakit dapat menambah, mengubah, dan menghapus data dokter yang akan disimpan di Database Dokter.
- Riwayat Kesehatan: Dokter mencatat hasil pemeriksaan pasien. Data riwayat kesehatan disimpan di Database Riwayat Kesehatan dengan mengambil data pasien dari Database Pasien.

2. Class Diagram



Kode Diagram:

classDiagram

%% Kelas Pelanggan

```
class Pelanggan {  
    +String idPelanggan  
    +String nama  
    +String noTelp  
    +String email  
    +String alamat  
    +void daftar()  
    +void buatJanji()  
}
```

%% Kelas Dokter

```
class Dokter {  
    +String idDokter  
    +String nama  
    +String spesialisasi  
    +String noTelp  
    +void tambahDokter()  
    +void ubahDokter()  
    +void hapusDokter()  
}
```

%% Kelas Konsultasi

```
class Konsultasi {  
    +String idKonsultasi  
    +String idPelanggan  
    +String idDokter  
    +DateTime tanggalWaktu  
    +void jadwalkanKonsultasi()  
    +void batalkanKonsultasi()
```

```
}
```

%% Kelas Riwayat Kesehatan

```
class RiwayatKesehatan {  
    +String idRiwayat  
    +String idPelanggan  
    +String idDokter  
    +String catatan  
    +DateTime tanggal  
    +void catatRiwayat()  
    +void ambilRiwayat()  
}
```

%% Kelas Administrasi

```
class Administrasi {  
    +String idAdmin  
    +String nama  
    +void kelolaDokter()  
    +void kelolaPasien()  
}
```

%% Relasi

Pelanggan "1" --> "0..*" Konsultasi : membuat

Dokter "1" --> "0..*" Konsultasi : melakukan

Pelanggan "1" --> "0..*" RiwayatKesehatan : memiliki

Dokter "1" --> "0..*" RiwayatKesehatan : mencatat

Administrasi "1" --> "0..*" Dokter : mengelola

Penjelasan Diagram Kelas:

- Pelanggan:

Atribut: idPelanggan, nama, noTelp, email, alamat.

Method: daftar() untuk mendaftar, buatJanji() untuk membuat janji temu.

- Dokter:

Atribut: idDokter, nama, spesialisasi, noTelp.

Method: tambahDokter(), ubahDokter(), dan hapusDokter() untuk manajemen data dokter.

- Konsultasi:

Atribut: idKonsultasi, idPelanggan, idDokter, tanggalWaktu.

Method: jadwalkanKonsultasi() untuk menjadwalkan dan batalkanKonsultasi() untuk membatalkan.

Riwayat Kesehatan:

Atribut: idRiwayat, idPelanggan, idDokter, catatan, tanggal.

Method: catatRiwayat() untuk mencatat hasil pemeriksaan dan ambilRiwayat() untuk mengambil data riwayat kesehatan.

- Administrasi:

Atribut: idAdmin, nama.

Method: kelolaDokter() untuk mengelola data dokter dan kelolaPasien() untuk mengelola data pasien.

Relasi:

Pelanggan dapat memiliki banyak Konsultasi.

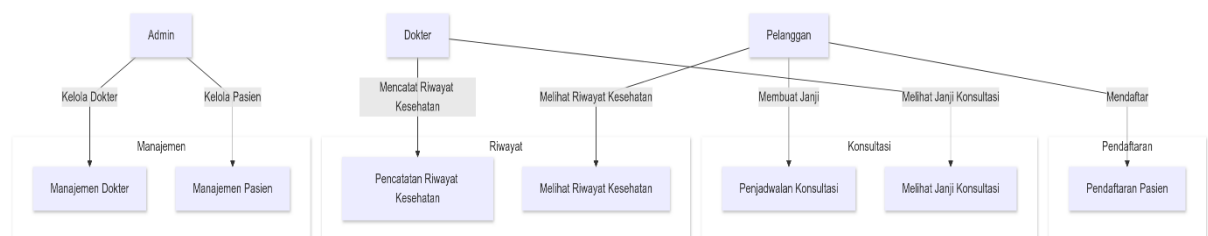
Dokter dapat melakukan banyak Konsultasi.

Pelanggan memiliki banyak Riwayat Kesehatan.

Dokter mencatat banyak Riwayat Kesehatan.

Administrasi dapat mengelola banyak Dokter.

3. Use Case



Kode Diagram:

```
%%{ init : { "theme" : "default" } }%%
```

```
graph TD
```

```
%% Aktor
```

Pelanggan -->|Mendaftar| PendaftaranPasien

Pelanggan -->|Membuat Janji| PenjadwalanKonsultasi

Pelanggan -->|Melihat Riwayat Kesehatan| MelihatRiwayatKesehatan

Dokter -->|Mencatat Riwayat Kesehatan| PencatatanRiwayatKesehatan

Dokter -->|Melihat Janji Konsultasi| MelihatJanjiKonsultasi

Admin -->|Kelola Dokter| ManajemenDokter

Admin -->|Kelola Pasien| ManajemenPasien

%% Proses

subgraph Pendaftaran

PendaftaranPasien["Pendaftaran Pasien"]

end

subgraph Konsultasi

PenjadwalanKonsultasi["Penjadwalan Konsultasi"]

MelihatJanjiKonsultasi["Melihat Janji Konsultasi"]

end

subgraph Riwayat

PencatatanRiwayatKesehatan["Pencatatan Riwayat Kesehatan"]

MelihatRiwayatKesehatan["Melihat Riwayat Kesehatan"]

end

subgraph Manajemen

ManajemenDokter["Manajemen Dokter"]

ManajemenPasien["Manajemen Pasien"]

End

Penjelasan Use Case Diagram:

Aktor:

Pelanggan: Mendaftar sebagai pasien, membuat janji konsultasi, dan melihat riwayat kesehatan.

Dokter: Mencatat riwayat kesehatan pasien dan melihat jadwal janji konsultasi.

Admin: Mengelola data dokter dan pasien.

Proses:

Pendaftaran Pasien: Proses pendaftaran pasien baru.

Penjadwalan Konsultasi: Proses pasien membuat janji untuk berkonsultasi dengan dokter.

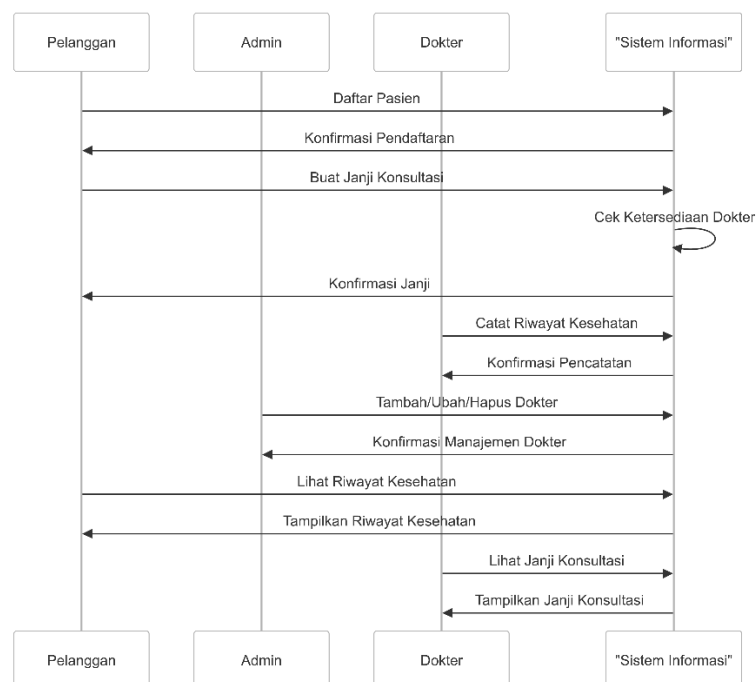
Pencatatan Riwayat Kesehatan: Proses dokter mencatat hasil pemeriksaan pasien.

Melihat Riwayat Kesehatan: Proses pasien melihat riwayat kesehatan mereka.

Manajemen Dokter: Proses admin untuk menambah, mengubah, atau menghapus data dokter.

Manajemen Pasien: Proses admin untuk menambah, mengubah, atau menghapus data pasien.

4. Sequence Diagram



Kode Diagram:

sequenceDiagram

%% Aktor

participant Pelanggan

participant Admin

participant Dokter

participant Sistem as "Sistem Informasi"

%% Pendaftaran Pasien

Pelanggan->>Sistem: Daftar Pasien

Sistem->>Pelanggan: Konfirmasi Pendaftaran

%% Penjadwalan Konsultasi

Pelanggan->>Sistem: Buat Janji Konsultasi

Sistem->>Sistem: Cek Ketersediaan Dokter

Sistem->>Pelanggan: Konfirmasi Janji

%% Mencatat Riwayat Kesehatan

Dokter->>Sistem: Catat Riwayat Kesehatan

Sistem->>Dokter: Konfirmasi Pencatatan

%% Manajemen Dokter (oleh Admin)

Admin->>Sistem: Tambah/Ubah/Hapus Dokter

Sistem->>Admin: Konfirmasi Manajemen Dokter

%% Melihat Riwayat Kesehatan

Pelanggan->>Sistem: Lihat Riwayat Kesehatan

Sistem->>Pelanggan: Tampilkan Riwayat Kesehatan

%% Melihat Janji Konsultasi (oleh Dokter)

Dokter->>Sistem: Lihat Janji Konsultasi

Sistem->>Dokter: Tampilkan Janji Konsultasi

Penjelasan Sequence Diagram:

- Pendaftaran Pasien:

Pelanggan mengirimkan permintaan untuk mendaftar sebagai pasien ke Sistem Informasi.

Sistem mengkonfirmasi pendaftaran kepada pelanggan.

- Penjadwalan Konsultasi:

Pelanggan membuat janji konsultasi dengan mengirimkan permintaan ke sistem.

Sistem memeriksa ketersediaan dokter dan mengkonfirmasi janji kepada pelanggan.

- Mencatat Riwayat Kesehatan:

Dokter mencatat riwayat kesehatan pasien melalui sistem.

Sistem mengkonfirmasi pencatatan riwayat kesehatan kepada dokter.

- Manajemen Dokter:

Admin melakukan tindakan menambah, mengubah, atau menghapus data dokter melalui sistem.

Sistem mengkonfirmasi tindakan manajemen dokter kepada admin.

- Melihat Riwayat Kesehatan:

Pelanggan meminta untuk melihat riwayat kesehatan mereka.

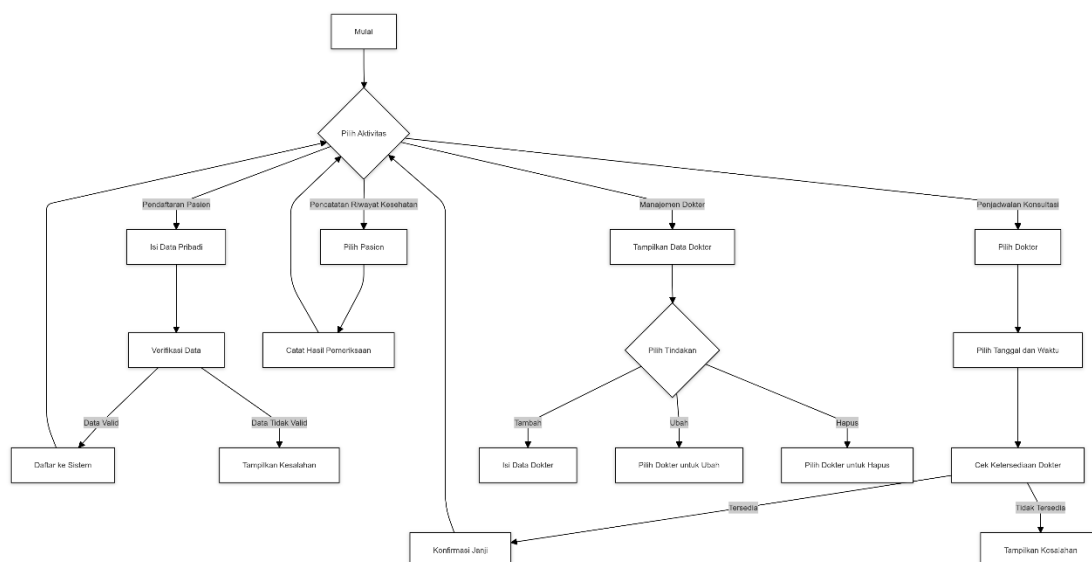
Sistem menampilkan riwayat kesehatan kepada pelanggan.

- Melihat Janji Konsultasi:

Dokter meminta untuk melihat jadwal janji konsultasi.

Sistem menampilkan jadwal konsultasi kepada dokter.

5. Activity Diagram



Kode Diagram:

graph TD

%% Awal

A[Mulai] --> B{Pilih Aktivitas}

%% Aktivitas Pendaftaran Pasien

B -->|Pendaftaran Pasien| C[Isi Data Pribadi]

C --> D[Verifikasi Data]

D -->|Data Valid| E[Daftar ke Sistem]

D -->|Data Tidak Valid| F[Tampilkan Kesalahan]

%% Aktivitas Penjadwalan Konsultasi

B -->|Penjadwalan Konsultasi| G[Pilih Dokter]

G --> H[Pilih Tanggal dan Waktu]

H --> I[Cek Ketersediaan Dokter]

I -->|Tersedia| J[Konfirmasi Janji]

I -->|Tidak Tersedia| K[Tampilkan Kesalahan]

%% Aktivitas Pencatatan Riwayat Kesehatan

B -->|Pencatatan Riwayat Kesehatan| L[Pilih Pasien]

L --> M[Catat Hasil Pemeriksaan]

%% Aktivitas Manajemen Dokter

B -->|Manajemen Dokter| N[Tampilkan Data Dokter]

N --> O{Pilih Tindakan}

O -->|Tambah| P[Isi Data Dokter]

O -->|Ubah| Q[Pilih Dokter untuk Ubah]

O -->|Hapus| R[Pilih Dokter untuk Hapus]

%% Akhir

E --> B

J --> B

M --> B

P --> B

Q --> B

R --> B

F --> B

K --> B

Penjelasan:

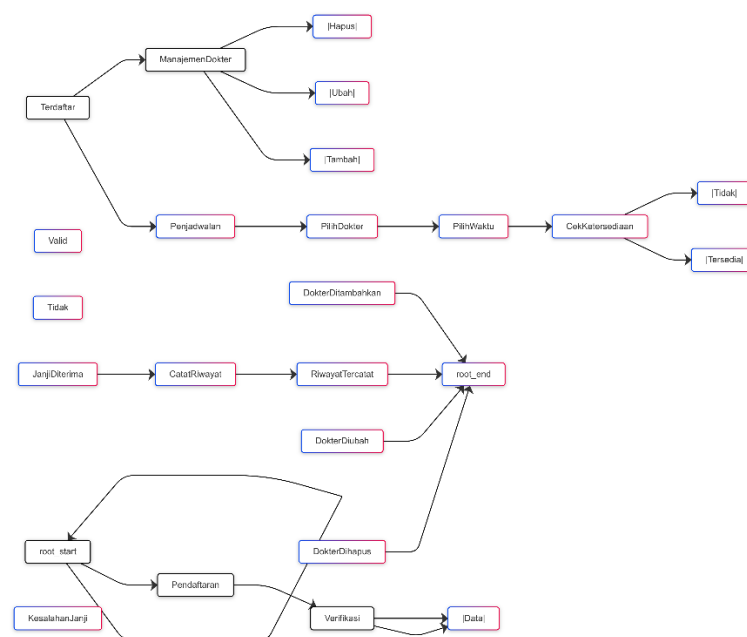
Mulai: Proses dimulai.

Pilih Aktivitas: Pengguna memilih aktivitas yang ingin dilakukan:

- Pendaftaran Pasien
- Penjadwalan Konsultasi
- Pencatatan Riwayat Kesehatan
- Manajemen Dokter

- Pendaftaran Pasien: Mengisi data pribadi dan melakukan verifikasi. Jika data valid, pasien terdaftar ke dalam sistem. Jika tidak valid, kesalahan ditampilkan.
- Penjadwalan Konsultasi: Memilih dokter dan tanggal/waktu. Lalu Sistem mengecek ketersediaan dokter. Jika tersedia, janji dikonfirmasi; jika tidak, kesalahan ditampilkan.
- Pencatatan Riwayat Kesehatan: Memilih pasien dan mencatat hasil pemeriksaan.
- Manajemen Dokter: Menampilkan data dokter dan memilih tindakan (tambah, ubah, atau hapus). Akhir: Proses kembali ke pilihan aktivitas setelah setiap tindakan selesai.

6. State Diagram



Kode Diagram:

stateDiagram-v2

[*] --> Pendaftaran

Pendaftaran --> Verifikasi

Verifikasi -->|Data Valid| Terdaftar

Verifikasi -->|Data Tidak Valid| Kesalahan

Kesalahan --> Pendaftaran

Terdaftar --> PenjadwalanKonsultasi

PenjadwalanKonsultasi --> CekKetersediaan

CekKetersediaan -->|Tersedia| JanjiDiterima

CekKetersediaan -->|Tidak Tersedia| KesalahanKonsultasi

KesalahanKonsultasi --> PenjadwalanKonsultasi

JanjiDiterima --> PencatatanRiwayat

PencatatanRiwayat --> RiwayatTercatat

Terdaftar --> ManajemenDokter

ManajemenDokter --> DokterDiubah

RiwayatTercatat --> [*]

DokterDiubah --> [*]

Penjelasan:

Pendaftaran: Proses dimulai dengan pendaftaran pasien.

Verifikasi: Setelah pendaftaran, data diverifikasi.

Data Valid: Jika data valid, pasien terdaftar.

Data Tidak Valid: Jika tidak valid, kesalahan ditampilkan dan kembali ke pendaftaran.

Terdaftar: Setelah terdaftar, pasien dapat melanjutkan ke penjadwalan konsultasi.

Penjadwalan Konsultasi: Proses penjadwalan dimulai dengan cek ketersediaan dokter.

Tersedia: Jika dokter tersedia, janji diterima.

Tidak Tersedia: Jika tidak tersedia, kesalahan ditampilkan dan kembali ke penjadwalan.

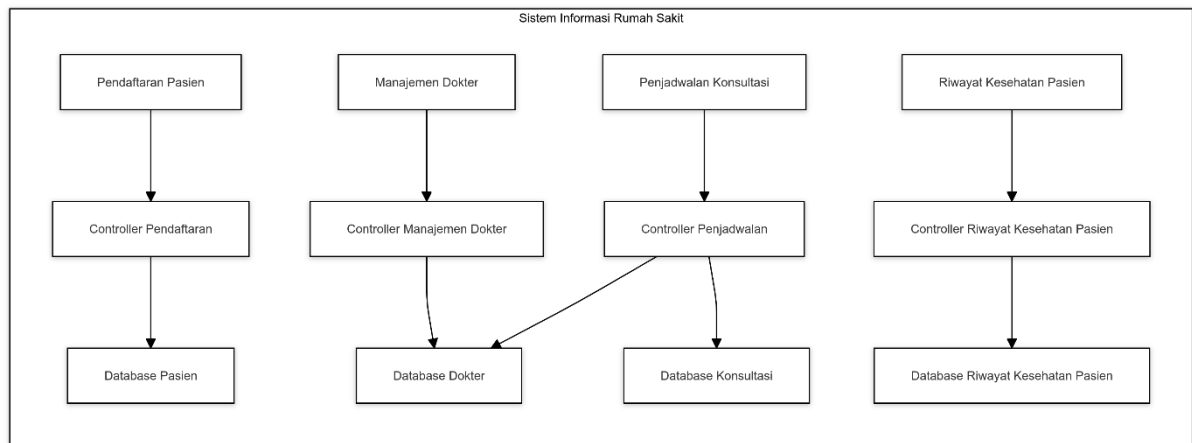
Pencatatan Riwayat: Setelah janji diterima, dokter dapat mencatat riwayat kesehatan pasien.

Riwayat Tercatat: Setelah pencatatan, proses selesai.

Manajemen Dokter: Admin dapat mengelola data dokter yang ada. Setelah perubahan, proses selesai.

Akhir: Proses berakhir setelah semua aktivitas selesai.

7. Component Diagram



Kode Diagram:

graph TD

%% Komponen Utama

subgraph "Sistem Informasi Rumah Sakit"

A[UI Pendaftaran Pasien]

B[UI Penjadwalan Konsultasi]

C[UI Manajemen Dokter]

D[UI Riwayat Kesehatan]

E[Controller Pendaftaran]

F[Controller Penjadwalan]

G[Controller Manajemen Dokter]

H[Controller Riwayat Kesehatan]

I[Database Pasien]

J[Database Dokter]

K[Database Konsultasi]

L[Database Riwayat Kesehatan]

end

%% Hubungan Antar Komponen

A --> E

B --> F

C --> G

D --> H

E --> I

F --> J

F --> K

H --> L

G --> J

Penjelasan:

Komponen Antarmuka Pengguna (UI):

- UI Pendaftaran Pasien: Antarmuka untuk pasien baru yang ingin mendaftar.
- UI Penjadwalan Konsultasi: Antarmuka untuk pasien yang ingin menjadwalkan konsultasi.
- UI Manajemen Dokter: Antarmuka untuk admin dalam mengelola data dokter.
- UI Riwayat Kesehatan: Antarmuka untuk dokter dalam mencatat hasil pemeriksaan pasien.

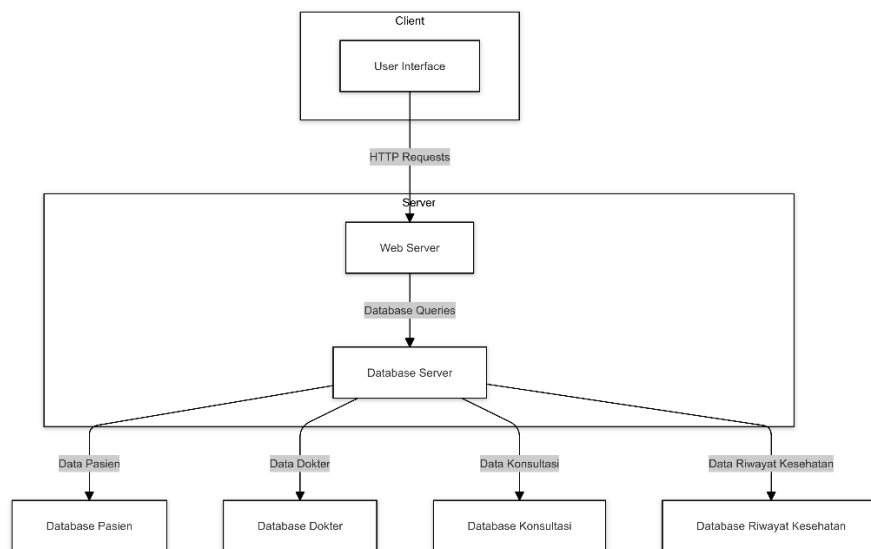
Controller:

- Controller Pendaftaran: Mengelola logika pendaftaran pasien dan berinteraksi dengan database.
- Controller Penjadwalan: Mengelola logika penjadwalan konsultasi dan berinteraksi dengan database pasien dan konsultasi.
- Controller Manajemen Dokter: Mengelola logika untuk menambah, mengubah, dan menghapus data dokter.
- Controller Riwayat Kesehatan: Mengelola pencatatan riwayat kesehatan pasien.

Database:

- Database Pasien: Menyimpan data pasien.
- Database Dokter: Menyimpan data dokter.
- Database Konsultasi: Menyimpan data janji konsultasi.
- Database Riwayat Kesehatan: Menyimpan catatan riwayat kesehatan pasien

8. Deployment



Kode Diagram:

graph TD

%% Nodes

subgraph "Server"

A[Web Server]

B[Database Server]

end

subgraph "Client"

C[User Interface]

end

%% Deployment

C -->|HTTP Requests| A

A -->|Database Queries| B

%% Database Instances

B -->|Data Pasien| D[Database Pasien]

B -->|Data Dokter| E[Database Dokter]

B -->|Data Konsultasi| F[Database Konsultasi]

B -->|Data Riwayat Kesehatan| G[Database Riwayat Kesehatan]

Penjelasan:

Penjelasan Deployment Diagram:

- Nodes:

Server: Merupakan bagian dari sistem yang menjalankan server web dan database.

Web Server: Menangani permintaan dari antarmuka pengguna dan mengelola logika aplikasi.

Database Server: Menyimpan dan mengelola semua data yang diperlukan untuk aplikasi.

Client: Merupakan antarmuka pengguna untuk pasien, dokter, dan admin.

User Interface: Menyediakan tampilan dan interaksi untuk pengguna akhir, seperti pendaftaran, penjadwalan, dan manajemen dokter.

- Deployment:

HTTP Requests: Antarmuka pengguna (client) berkomunikasi dengan server web melalui permintaan HTTP.

Database Queries: Server web mengirimkan kueri ke database server untuk mengambil atau menyimpan data.

- Database Instances:

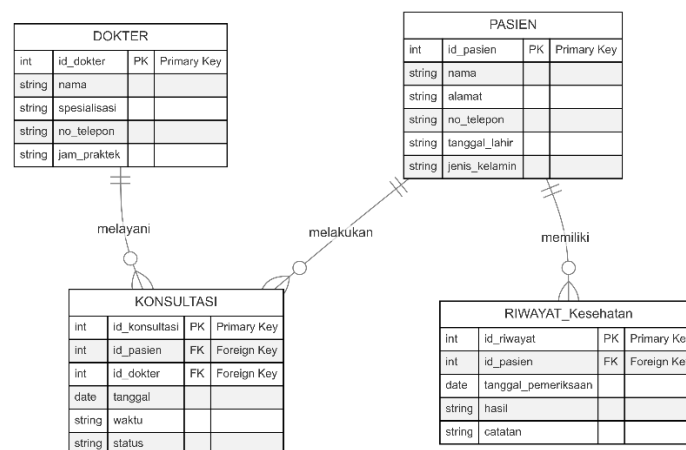
Database Pasien: Menyimpan data pasien yang terdaftar.

Database Dokter: Menyimpan data dokter yang tersedia.

Database Konsultasi: Menyimpan data tentang janji temu yang telah dijadwalkan.

Database Riwayat Kesehatan: Menyimpan catatan kesehatan pasien

9. ERD PLUS



Kode Diagram:

erDiagram

%% Entitas

PASIEN {

int id_pasien PK "Primary Key"

string nama

string alamat

string no_telepon

string tanggal_lahir

string jenis_kelamin

}

DOKTER {

int id_dokter PK "Primary Key"

string nama

string spesialisasi

string no_telepon

string jam_praktek

}

KONSULTASI {

int id_konsultasi PK "Primary Key"

int id_pasien FK "Foreign Key"

int id_dokter FK "Foreign Key"

date tanggal

string waktu

string status

}

RIWAYAT_Kesehatan {

int id_riwayat PK "Primary Key"

int id_pasien FK "Foreign Key"

```
    date tanggal_pemeriksaan
    string hasil
    string catatan
}
```

%% Relasi

PASIEN ||--o{ KONSULTASI : "melakukan"

DOKTER ||--o{ KONSULTASI : "melayani"

PASIEN ||--o{ RIWAYAT_Kesehatan : "memiliki"

Penjelasan:

- Entitas:

PASIEN: Menyimpan informasi tentang pasien, termasuk ID, nama, alamat, nomor telepon, tanggal lahir, dan jenis kelamin.

DOKTER: Menyimpan informasi tentang dokter, termasuk ID, nama, spesialisasi, nomor telepon, dan jam praktik.

KONSULTASI: Menyimpan informasi tentang janji konsultasi, termasuk ID konsultasi, ID pasien, ID dokter, tanggal, waktu, dan status janji.

RIWAYAT_Kesehatan: Menyimpan informasi tentang riwayat kesehatan pasien, termasuk ID riwayat, ID pasien, tanggal pemeriksaan, hasil, dan catatan.

- Relasi:

PASIEN dan KONSULTASI: Hubungan "melakukan", di mana satu pasien dapat melakukan banyak konsultasi (one-to-many).

DOKTER dan KONSULTASI: Hubungan "melayani", di mana satu dokter dapat melayani banyak konsultasi (one-to-many).

PASIEN dan RIWAYAT_Kesehatan: Hubungan "memiliki", di mana satu pasien dapat memiliki banyak catatan riwayat kesehatan (one-to-many).