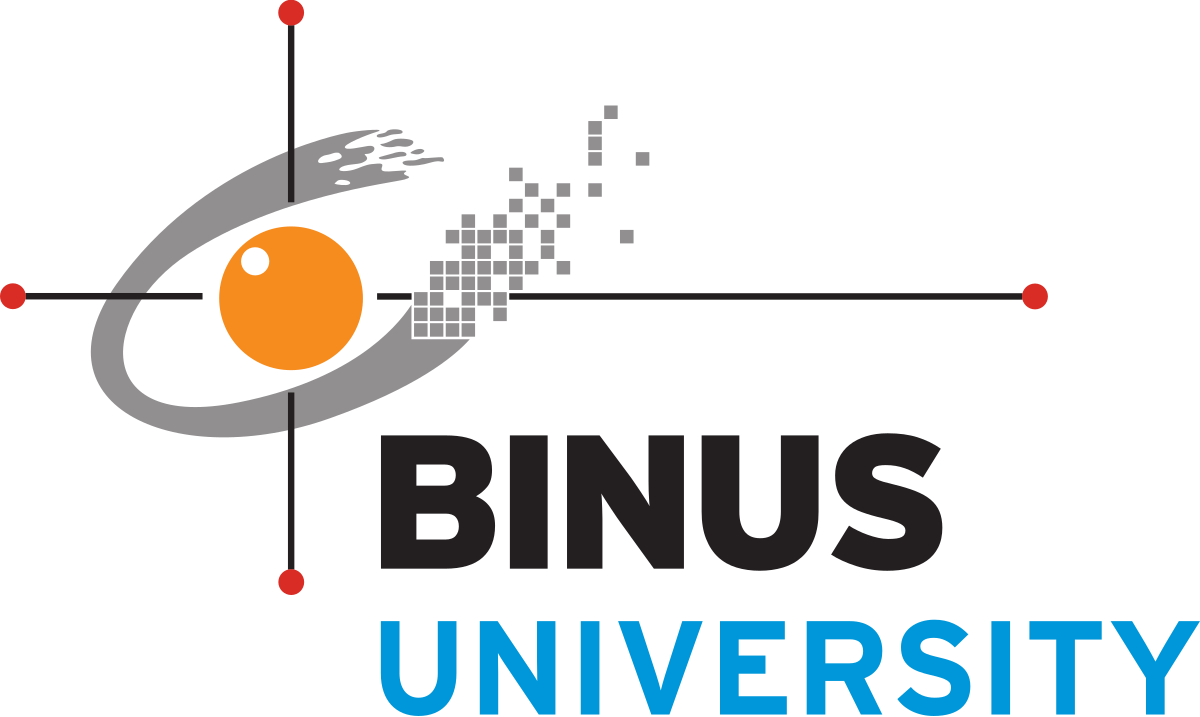
**LAPORAN AKHIR AOL DATABASE TECHNOLOGY**

**COMP6799001**

****

Oleh Grup 2:

Hans Ardianta - 2702249663

Gervasius Russell - 2702257450

Gabriella Clairine - 2702302183

Adrian Marc Fedier Purnama - 2702247216

Jonathan William Gunawan - 2702251794

Shyra Alexandria - 2702291356

Universitas Bina Nusantara

2024

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Data adalah sebuah informasi atau fakta yang masih mentah, data dapat berupa simbol, kalimat, citra, dan angka. Secara etimologis, istilah “data” berasal dari kata Latin jamak “datum” yang berarti “sesuatu yang diberikan”. Dalam penggunaan sehari-hari, data mengacu pada fakta tentang objek yang diamati, baik angka maupun kata. Menurut Kamus Besar Bahasa Indonesia (KBBI) mengartikan data sebagai kumpulan fakta atau informasi yang diperoleh melalui pengamatan, pengukuran, atau penelitian, biasanya dalam bentuk angka, teks, atau gambar, dan dapat diolah serta dianalisis untuk mendapatkan pemahaman atau pengetahuan lebih lanjut. Kuswandi dan E. Mutiara mendeskripsikan data sebagai sekumpulan informasi yang diperoleh dari pengamatan, dapat berupa simbol, angka, dan properti.

Perlu disadari bahwa data bersifat mentah artinya data perlu diolah dahulu untuk mendapatkan informasi sebenarnya atau seutuhnya. Untuk mempermudah mengolah data agar mendapatkan informasi yang diinginkan, diperlukan tempat untuk menyimpan data yang disebut dengan database. Database, adalah kumpulan data yang disimpan secara sistematis dalam suatu sistem komputer sehingga dapat diakses, diolah, dan dikelola dengan mudah dan efisien. Database biasanya diatur dalam struktur tertentu, seperti tabel, yang memungkinkan data diorganisasikan dan dihubungkan antara satu dengan yang lain. Database memiliki komponen penting, seperti *Database Management System* (DBMS), yang berfungsi sebagai perangkat lunak untuk memudahkan pembuatan, pemeliharaan, dan pengolahan database.

Dalam pengelolaan data yang semakin kompleks, khususnya pada data kesehatan yang melibatkan informasi rumah sakit, pasien, dokter, dan lainnya yang sangat penting untuk dipastikan bahwa data tersimpan dengan terstruktur dan efisien. Dengan adanya berbagai jenis data yang berkaitan, mulai dari nama rumah sakit hingga rincian pemeriksaan medis pasien, seringkali terjadi redundansi data yang dapat menghambat proses pengambilan informasi serta mengurangi efisiensi penyimpanan. Redundansi data juga meningkatkan risiko inkonsistensi data, yang dapat mempengaruhi validitas informasi yang diperoleh. Hal ini dapat diatasi dengan melakukan normalisasi data.

**1.2 Tujuan**

1. Mempercepat proses pengecekan data dan mengefisiensikan penyimpanan data rumah sakit.
2. Mempermudah pembaruan data jika terdapat sebuah masalah.

**BAB II**

**Sampel Data (UNF)**

Unnormalized Form (UNF) adalah bentuk data mentah yang belum mengalami proses normalisasi dalam database. Pada tahap ini, data sering kali disimpan dalam format tabel tunggal dengan penggabungan informasi yang kompleks, seperti *attribute multivalued*, *nested*, atau data yang berulang. UNF biasanya tidak efisien untuk pengelolaan karena adanya redundansi data, sulitnya pencarian informasi spesifik, dan potensi anomali data (seperti insertion, update, atau deletion anomalies).

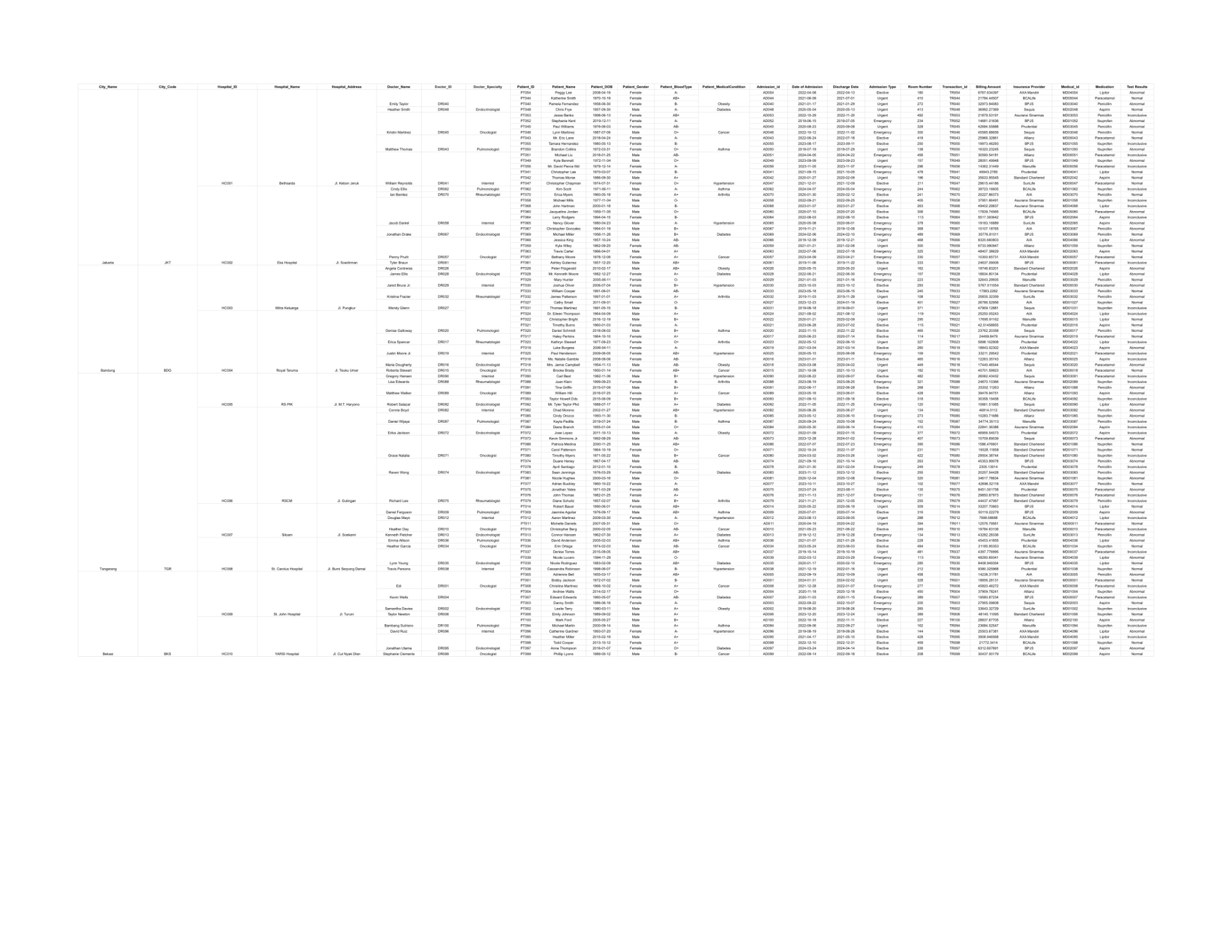
Karakteristik UNF:

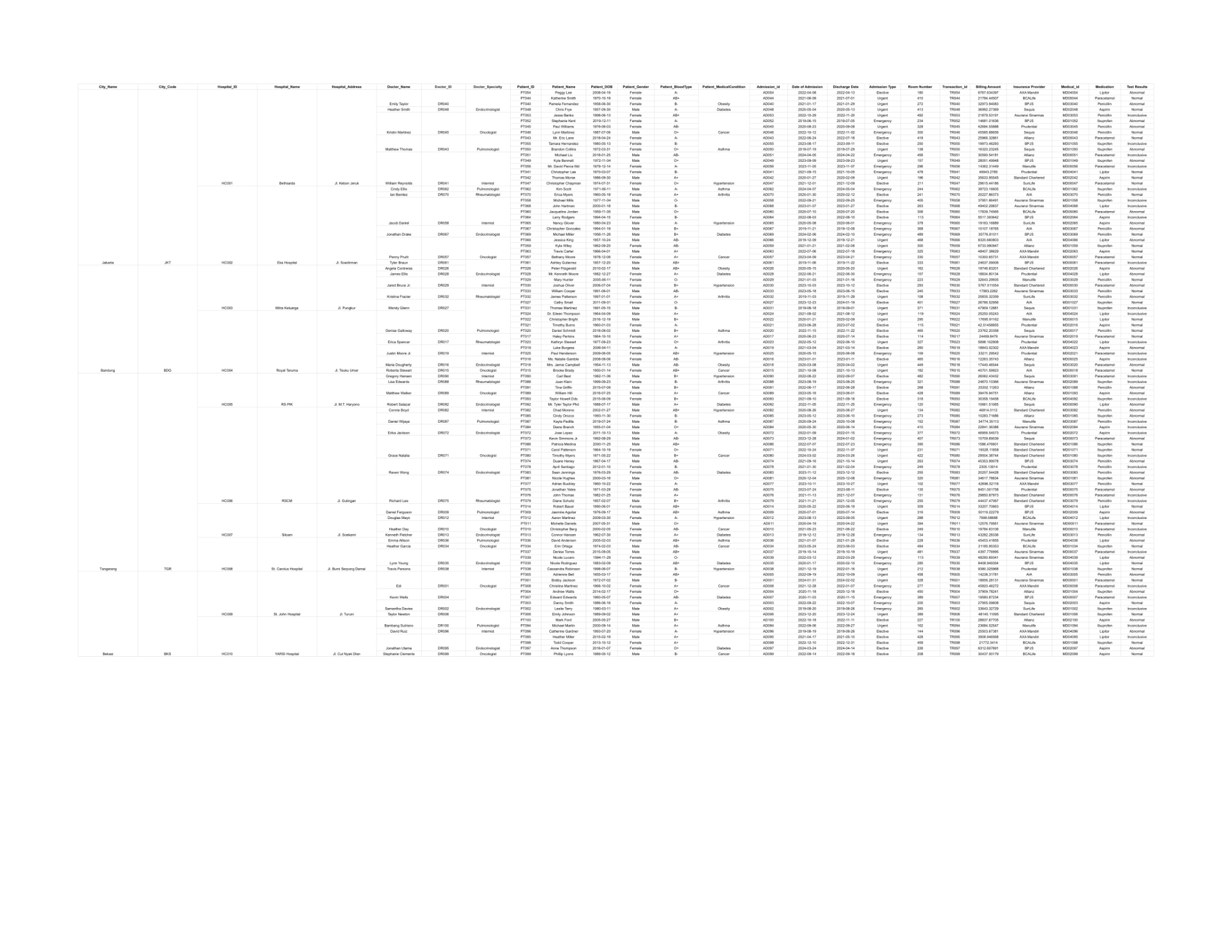
* Data Multivalued: Beberapa atribut memiliki lebih dari satu nilai dalam satu kolom, seperti ‘City\_Code’ pada contoh data kami yang mencantumkan banyak nomor dalam satu sel.
* Redundansi Data: Informasi yang sama berulang di beberapa baris, misalnya nama pelanggan tercantum berulang pada setiap transaksi mereka.
* Kesulitan Manipulasi Data: Operasi database seperti query menjadi tidak efisien karena data tidak terstruktur dengan baik.

Data dalam UNF sering kali digunakan dalam tahap awal pengumpulan data. Namun, jika dibiarkan tanpa normalisasi, akan menimbulkan masalah seperti pemborosan ruang penyimpanan, inkonsistensi data, dan anomali. Oleh karena itu, normalisasi menjadi langkah penting untuk memastikan bahwa data lebih terstruktur, efisien, dan dapat digunakan secara optimal.

Kita mendapatkan sampel data UNF “Kesehatan” berasal dari kaggle dengan link: <https://www.kaggle.com/datasets/prasad22/healthcare-dataset>. Dataset ini memiliki 55500 baris dan 15 *attributes* yang terdiri atas :

1. **Name** : Nama pasien
2. **Age** : Umur pasien dalam tahun
3. **Gender** : Jenis kelamin pasien seperti “Male” dan “Female”
4. **Blood Type** : Golongan darah pasien bersama rhesusnya (“A+”, “B+”, dan lain - lain)
5. **Medical Condition** : Kondisi medis pasien yang didiagnosa dokter
6. **Date of Admission** : Tanggal pasien masuk rumah sakit
7. **Doctor:** Nama doktor yang merawat pasien
8. **Hospital:** Nama rumah sakit pasien dirawat
9. **Insurance Provider:** Asuransi pasien
10. **Billing Amount:** Jumlah tagihan pasien
11. **Room Number:** Nomor kamar pasien
12. **Admission Type:** Menentukan jenis admisi, yang dapat berupa "Darurat", "Pilihan", atau "Mendesak", yang mencerminkan keadaan pasien tersebut.
13. **Discharge Date:** Tanggal pasien keluar dari fasilitas kesehatan
14. **Medication:** Obat yang diresepkan atau diberikan kepada pasien selama di rumah sakit. Contohnya termasuk “Aspirin”, “Ibuprofen”, “Penicillin”, “Paracetamol”, dan “Lipitor”.
15. **Test Results:** Menjelaskan hasil tes kesehatan yang dilakukan selama pasien masuk. Contohnya termasuk “Normal”, “Abnormal”, “Inconclusive”.

Sample Data UNF: 



Pada awalnya dataset yang kami temukan memiliki beberapa kelemahan seperti :

1. Tidak ada “Unique Identifier” untuk “Primary Key”
2. Kurangnya variasi relasi antar kolom

Untuk mengatasi kurangnya “Unique Identifier” tersebut kami menambahkan *attribute* “Transaction\_id”, “Doctor\_id”, “Patient\_id”, “Admission\_id”, “Medical\_id”. Lalu setelah menambahkannya, ternyata relasi antar kolom masih kurang sehingga kita menambah lagi *attributes* “City\_Code”, “City\_Name”, “Hospital\_id”, “Hospital\_name”, “Hospital\_address”. Semua yang ber “id” seperti “Transaction\_id”, “Patient\_id”, “Admission\_id”, kami membuatnya mengikuti nomor baris, misalnya untuk baris 1 (TR001, PT001, AD001), buat “Medical\_id” kami membuatnya mengikuti nomor baris dan jenis obatnya, misalnya untuk tipe obat “Paracetamol” jadi MD00, “Ibuprofen” jadi MD01, “Aspirin” jadi MD02, “Penicillin” jadi MD03, “Lipitor” jadi MD04, untuk obat lainnya jadi MD05. Lalu kami tambahkan buat urutan barisnya, misalnya untuk baris pertama dengan tipe obatnya “Paracetamol” maka Medical\_id nya menjadi MD00001, jika baris ke 90 dan tipe obatnya “Ibuprofen” maka Medical\_id nya jadi MD01090.

Lalu Kami membuat “City\_Code” berdasarkan singkatan dari “City\_Name” misalnya kalau “City\_Name” nya Jakarta, maka “City\_Code” nya menjadi JKT. Untuk “Hospital\_address” berdasarkan “Hospital\_name”, dan “Hospital\_name” berdasarkan “Hospital\_id”. Disini, untuk kasus “City\_name”, “City\_Code”, “Hospital\_id”, “Hospital\_name”, “Hospital\_address” kami buatnya menjadi tiap “City\_name” dan “City\_Code” terdapat 2 “Hospital\_id”, “Hospital\_name” dan “Hospital\_address”. Lalu masing - masing “Hospital\_address” memiliki doktor yang sama untuk “Doctor\_Specialty” yang sama. Jika “Doctor\_Specialty” nya sama tetapi “Hospital\_Address” nya berbeda maka doktor yang dijadwalkan juga berbeda. Lalu untuk semuanya yang berhubungan dengan pasien kita ubah nama kolomnya menjadi “Patient\_(nama kolom sebelumnya)”. Jadi misalnya “Name” menjelaskan nama dari pasien kita ubah nama kolomnya menjadi “Patient\_Name”. Lalu untuk umur dari pasien sendiri kita ubah tipe datanya yang sebelumnya integer menjadi tipe data date.

Dari dataset yang berasal dari kaggle yang awalnya adalah 10.000 baris dan 15 kolom, kami memanipulasinya menjadi 100 baris dan 25 kolom.

**BAB III**

**Anomali yang Terjadi**

**3.1 Skenario dan Anomali**

**3.1.1** **Perubahan Kondisi Medis Pasien Menjadi Kritis**

Dalam situasi di mana seorang pasien mengalami perubahan kondisi medis yang mengakibatkan kondisi menjadi kritis, langkah pertama yang diambil adalah admin segera menghubungi suster untuk memindahkan pasien ke ruang IGD, agar mendapatkan penanganan yang lebih intensif. Sementara itu, rumah sakit melakukan pemeriksaan untuk mengetahui apakah dokter yang menangani pasien tersebut sedang berada di rumah sakit yang sama atau tidak. Jika dokter tidak berada di rumah sakit, admin akan mencari dokter lain yang sedang bertugas dan berada di rumah sakit tersebut, sambil tetap menghubungi dokter yang ditugaskan untuk menangani pasien guna mendapatkan informasi terkini tentang kondisi pasien.

Setelah dokter melakukan pemeriksaan menyeluruh terhadap kondisi pasien, dia akan mengevaluasi situasi dan memutuskan apakah tindakan operasi diperlukan. Jika tindakan operasi dianggap perlu, admin akan segera melakukan pengecekan untuk memastikan apakah kamar operasi tersedia. Jika kamar operasi tersedia, rumah sakit akan segera menghubungi keluarga pasien untuk meminta persetujuan sebelum melanjutkan ke langkah berikutnya. Sebaliknya, jika kamar operasi tidak tersedia, rumah sakit akan mengambil langkah proaktif dengan menghubungi rumah sakit lain untuk meminta bantuan, sekaligus meminta persetujuan dari keluarga pasien untuk melakukan operasi dan transfer antar rumah sakit jika diperlukan.

Apabila kamar operasi tersedia dan keluarga pasien memberikan persetujuan yang diperlukan, pasien akan segera dipindahkan ke kamar operasi untuk menjalani tindakan yang diperlukan secepat mungkin. Namun, jika setelah pemeriksaan dokter dinyatakan bahwa tidak diperlukan tindakan operasi, pasien akan tetap berada di ruang IGD untuk mendapatkan perawatan intensif yang dibutuhkan untuk stabilisasi kondisi. Setelah kondisi pasien membaik dan tidak lagi dalam keadaan kritis, langkah selanjutnya adalah memindahkan pasien ke ruangan biasa, di mana dia dapat melanjutkan proses pemulihan dengan pengawasan yang sesuai. Proses ini dilakukan dengan penuh perhatian dan komunikasi yang baik antara pihak rumah sakit dan keluarga pasien untuk memastikan keselamatan dan kenyamanan pasien selama perawatan.

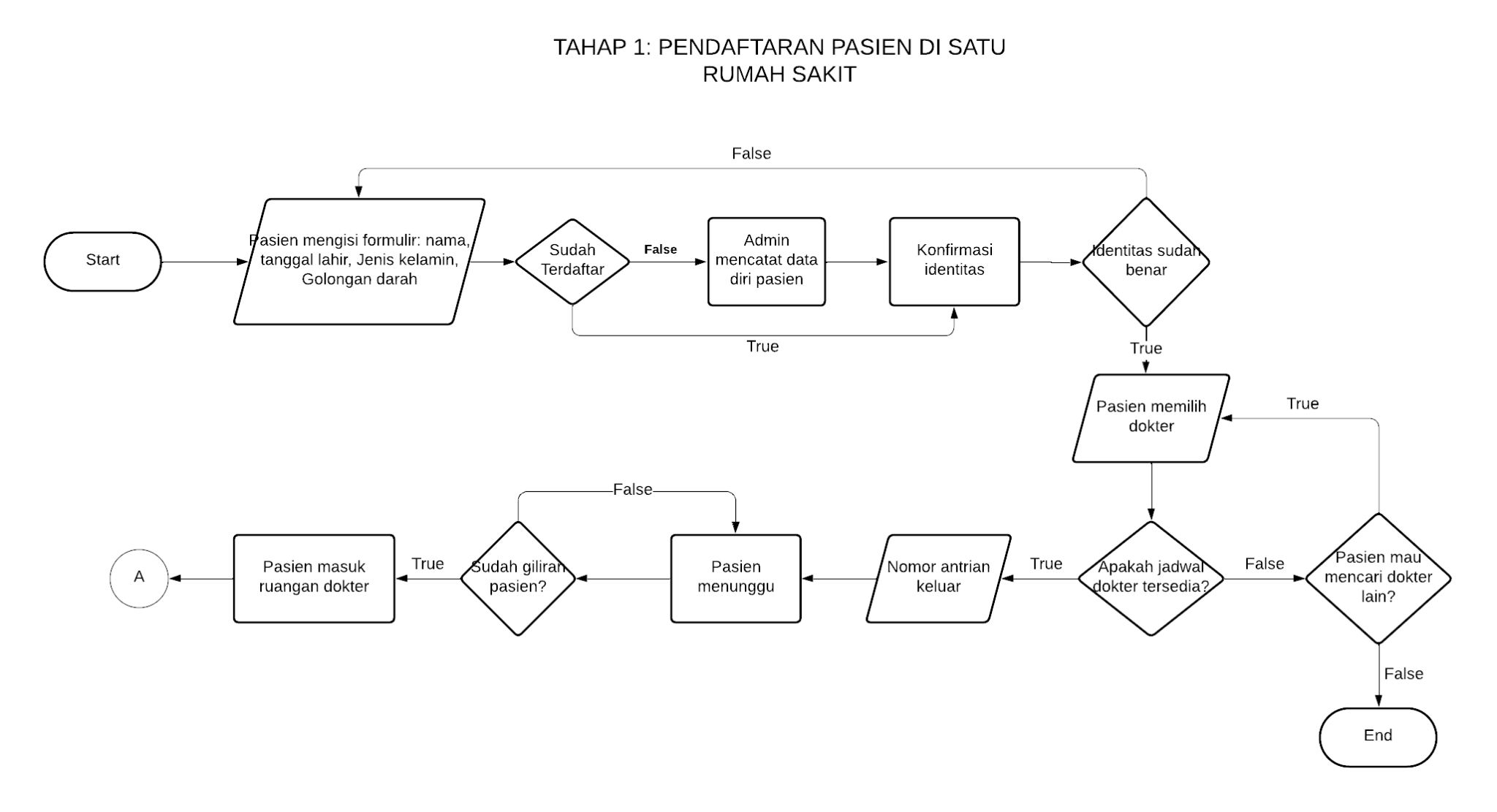
Dari skenario ini, terdapat anomali yang terjadi yaitu jika data tentang dokter yang bertugas dihapus secara tidak sengaja dari sistem sebelum pasien mendapatkan perawatan, sehingga menyulitkan identifikasi dokter yang seharusnya menangani pasien.

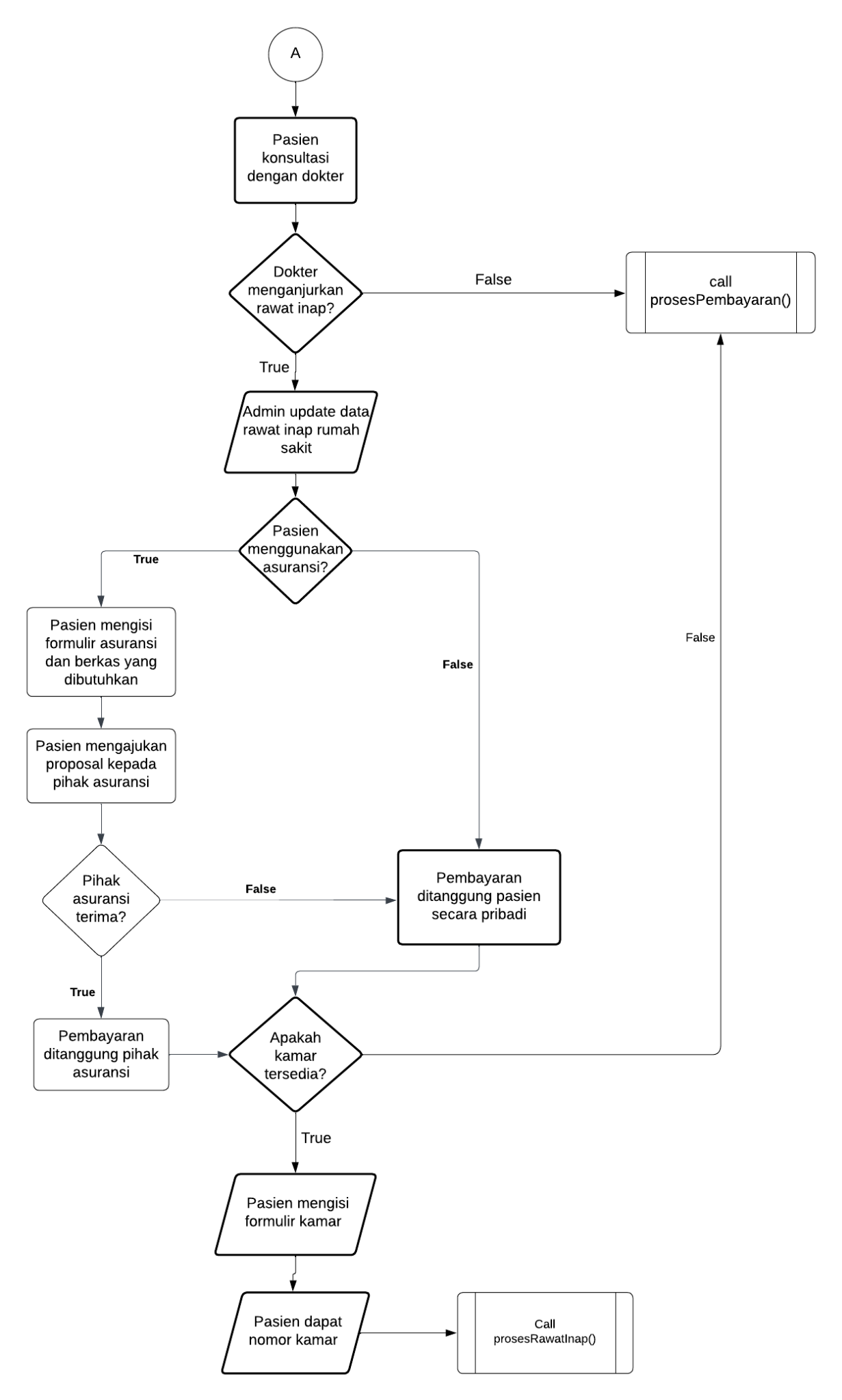
**3.1.2 Penambahan Hari untuk Rawat Inap Pasien**

Dalam skenario seorang pasien yang ingin menambah hari rawat inap, admin akan segera melakukan beberapa pembaruan penting; pertama, admin akan memperbarui data rawat inap rumah sakit, termasuk informasi mengenai kamar yang tersedia untuk memastikan bahwa pasien dapat diperpanjang masa perawatannya. Selanjutnya, admin akan memperbarui tanggal keluar pasien dari rumah sakit sesuai dengan permintaan penambahan hari rawat inap. Selain itu, admin juga akan memperbarui jadwal tim medis dan dokter yang berhubungan dengan pasien untuk memastikan bahwa semua pihak terkait siap memberikan perawatan yang dibutuhkan selama masa perpanjangan. Terakhir, admin akan memperbarui jadwal tagihan untuk mencerminkan perubahan durasi rawat inap pasien, sehingga semua aspek administrasi terkait perawatan pasien dapat dikelola dengan baik dan efisien.

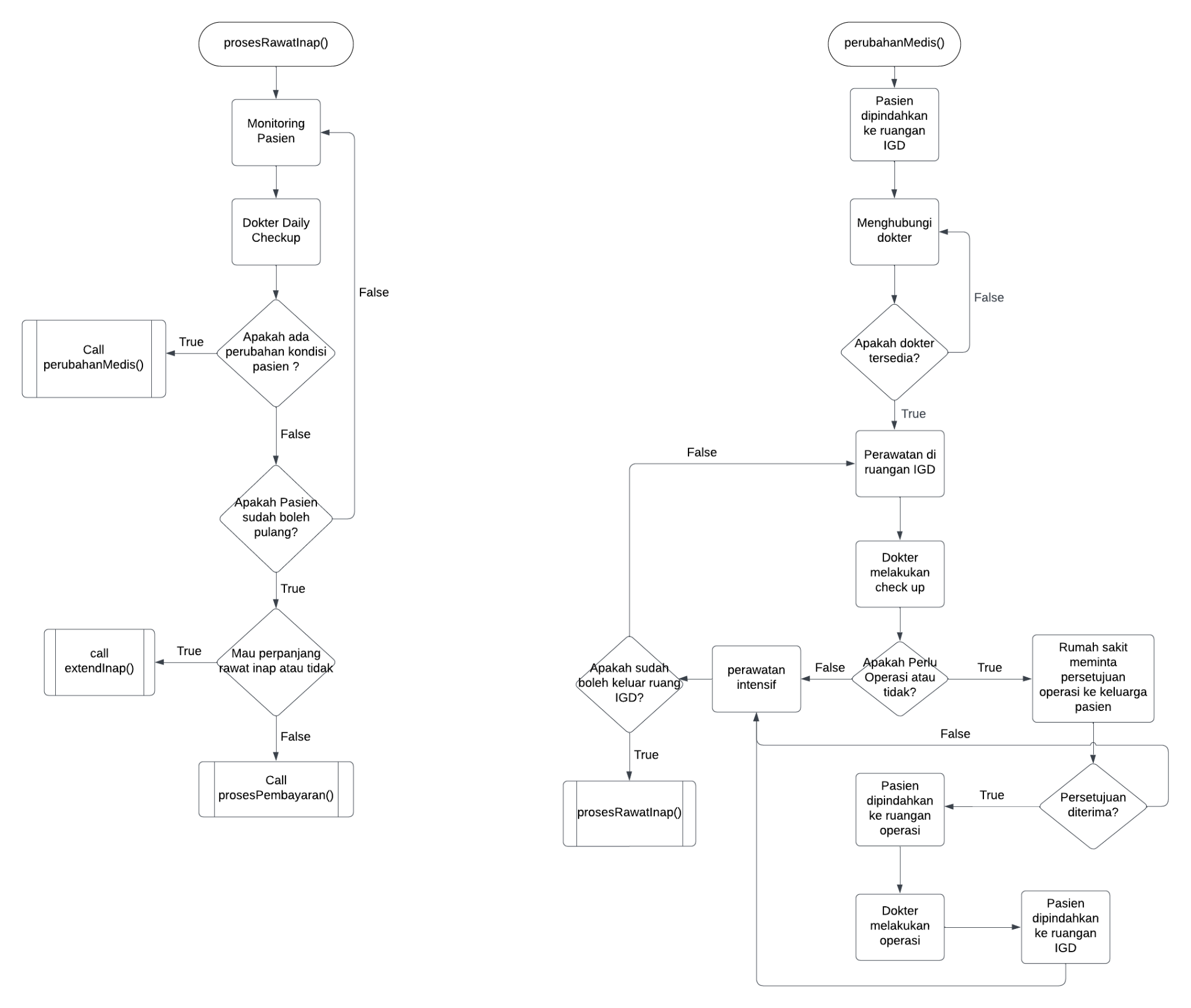
Dari skenario ini, terdapat anomali yang terjadi yaitu jika admin mengubah status kamar menjadi tersedia tanpa memperbarui data terkait pasien yang keluar, kamar tersebut bisa dianggap tersedia padahal masih ada pasien.

**3.2 Flowchart**

**3.2.1 TAHAP 1: PENDAFTARAN PASIEN DI SATU RUMAH SAKIT**

**3.2.2 TAHAP 2: PASIEN MELAKUKAN KONSULTASI**

**3.2.3 TAHAP 3: PASIEN RAWAT INAP ATAU PERUBAHAN MEDIS**

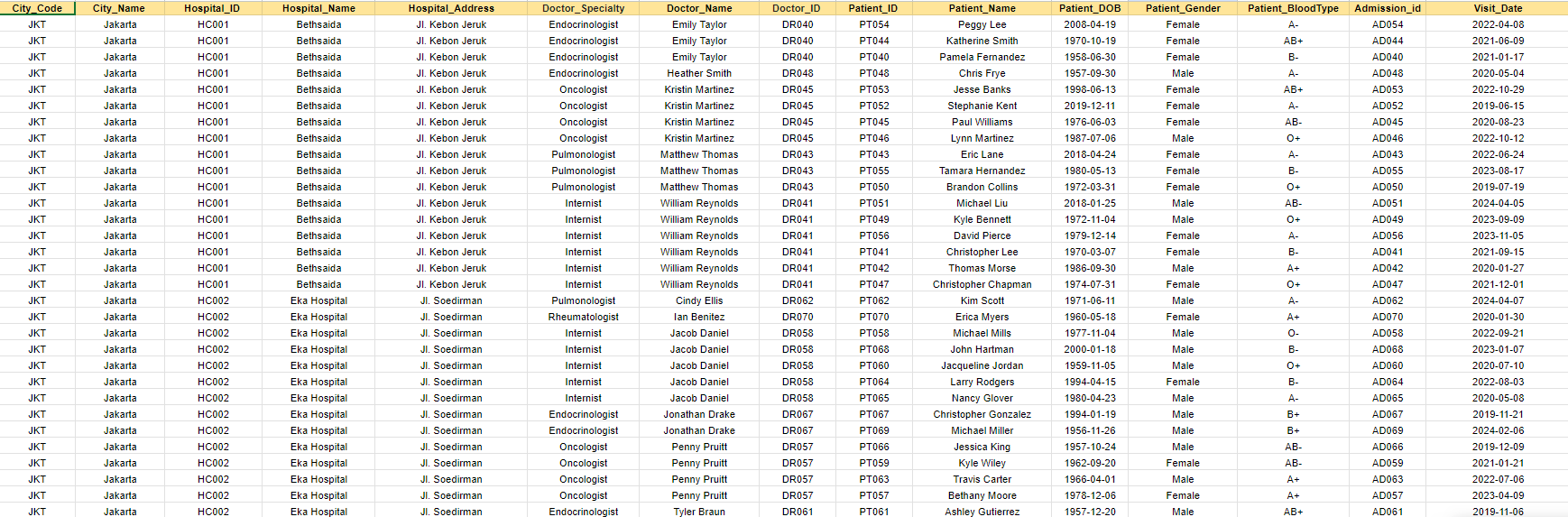


**BAB IV**

**Implementasi Normalisasi**

**First Normal Form (1NF)**: Pada 1NF, data dalam setiap kolom harus atomic, atau satu nilai saja tanpa pengulangan atau daftar nilai dalam satu kolom. Misalnya, dalam data kesehatan, informasi pasien tidak boleh menyimpan lebih dari satu hasil tes dalam satu kolom; setiap hasil harus ditempatkan dalam baris terpisah.

Berikut adalah hasil dari 1NF :

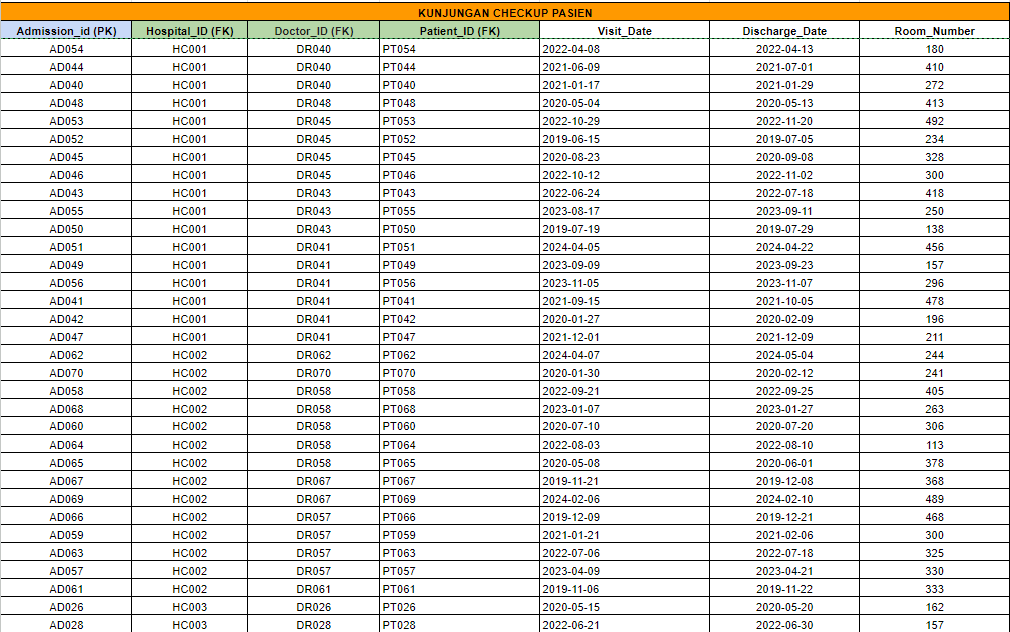


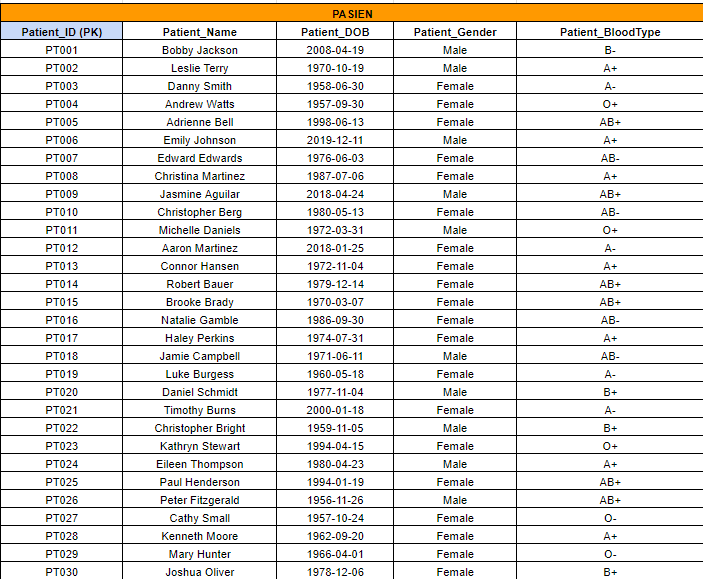
**Second Normal Form (2NF)**: Pada tahap Second Normal Form (2NF), sebuah tabel sudah harus memenuhi kriteria dari First Normal Form (1NF) terlebih dahulu. Artinya, data di dalam tabel sudah tidak memiliki grup berulang atau multivalued attributes, dan setiap kolom hanya menyimpan nilai atomik atau nilai yang tidak dapat dibagi lagi.

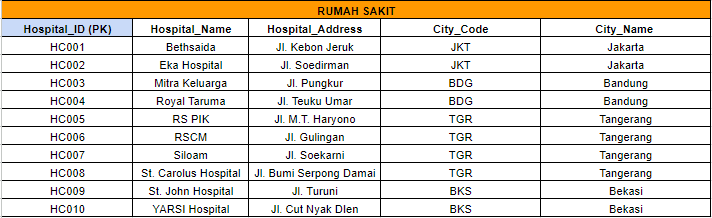
Setelah 1NF terpenuhi, langkah berikutnya dalam 2NF adalah menghilangkan ketergantungan parsial terhadap primary key. Ketergantungan parsial terjadi ketika atribut non-primer bergantung hanya pada sebagian dari primary key, bukan pada keseluruhan primary key. Hal ini biasanya ditemukan dalam tabel yang memiliki primary key gabungan (composite primary key), yaitu primary key yang terdiri dari lebih dari satu kolom. Misalnya, jika tabel "Kunjungan" memiliki primary key gabungan antara "Patient\_ID" dan "Visit\_Date," maka data seperti nama rumah sakit yang hanya bergantung pada "Patient\_ID" harus dipindahkan ke tabel lain.

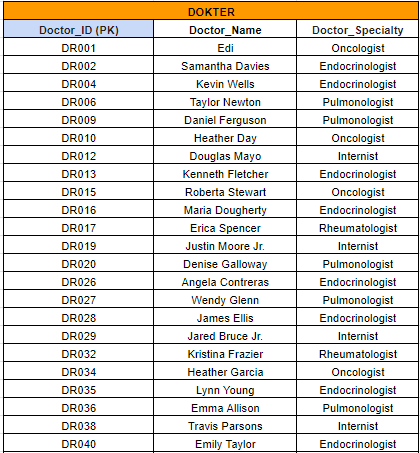
Kami membuat tabel tambahan seperti “PASIEN”, “RUMAH SAKIT”, “DOKTER” dan “ADMISI” karena atribut atribut mereka tidak semuanya bergantung ke unique key di tabel utama sehingga mereka harus dipisah menjadi tabel baru.

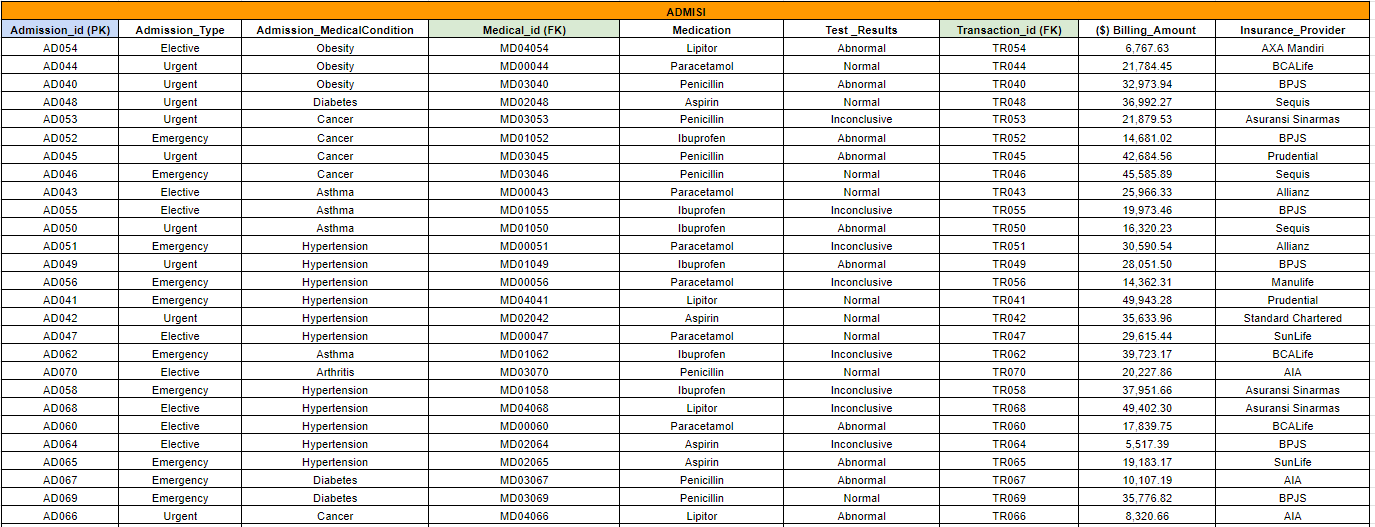
Berikut adalah hasil dari 2NF :











**Third Normal Form (3NF)**: Data dalam 3NF sudah memenuhi 2NF dan menghilangkan ketergantungan transitif. Ketergantungan transitif terjadi ketika atribut non-primer bergantung pada atribut non-primer lainnya. Sebagai contoh, dalam tabel pasien, jika kolom "Insurance\_Provider" bergantung pada "Patient\_ID" namun juga menentukan kolom "Insurance\_Contact," maka informasi kontak asuransi sebaiknya dipisahkan ke tabel lain.

Dari hasil 2NF ternyata masih ada ketergantungan transitif yang harus dibuang di tabel “ADMISI”, menurut kami, untuk mendapatkan “test\_result” dan “medication” kami harus mengetahui “medical\_id” terlebih dahulu dan untuk mengetahui medical\_id kita harus mengetahui “admission\_id” sehingga hal ini dianggap ketergantungan transitif. Hal ini berlaku juga untuk “insurance\_provide” dan “billing\_amount”. Untuk mendapatkan mereka kita harus mengetahui “transaction\_id” dan “transaction\_id” bergantung kepada “admission\_id” dan hal ini transitif. Oleh karena itu kita membuat tabel tambahan yaitu “ADMISI” , “PENGOBATAN”, “PEMBAYARAN”, dan “KOTA”.

Berikut adalah glosarium himpunan data setelah terbentuk menjadi 3NF:

| **KUNJUNGAN CHECKUP PASIEN** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Admission\_id | String | Identifikasi untuk setiap admisi dari pasien |
| Hospital\_ID | String | Identifikasi untuk setiap rumah sakit |
| Doctor\_ID | String | Identifikasi untuk setiap dokter |
| Patient\_ID | String | Identifikasi untuk setiap pasien |
| Visit\_Date | Date | Tanggal pasien masuk rumah sakit |
| Discharge\_Date | Date | Tanggal pasien keluar rumah sakit |
| Room Number | Integer | Nomor kamar pasien |

| **DOKTER** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Doctor\_ID | String | Identifikasi untuk setiap dokter |
| Doctor\_Name | String | Nama dari setiap dokter |
| Doctor\_Specialty | String | Bidang spesialis tiap dokter |

| **PASIEN** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Patient\_ID | String | Identifikasi untuk setiap pasien |
| Patient\_Name | String | Nama pasien |
| Patient\_DOB | Date | Tanggal lahir pasien |
| Patient\_Gender | String | Jenis kelamin pasien |
| Patient\_Bloodtype | String | Jenis golongan darah pasien |

| **KOTA** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| City\_Code | String | Kode singkatan untuk nama kota |
| City\_Name | String | Nama kota |

| **RUMAH SAKIT** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Hospital\_ID | String | Identifikasi untuk setiap rumah sakit |
| Hospital\_Name | String | Nama rumah sakit |
| Hospital\_Address | String | Alamat rumah sakit |

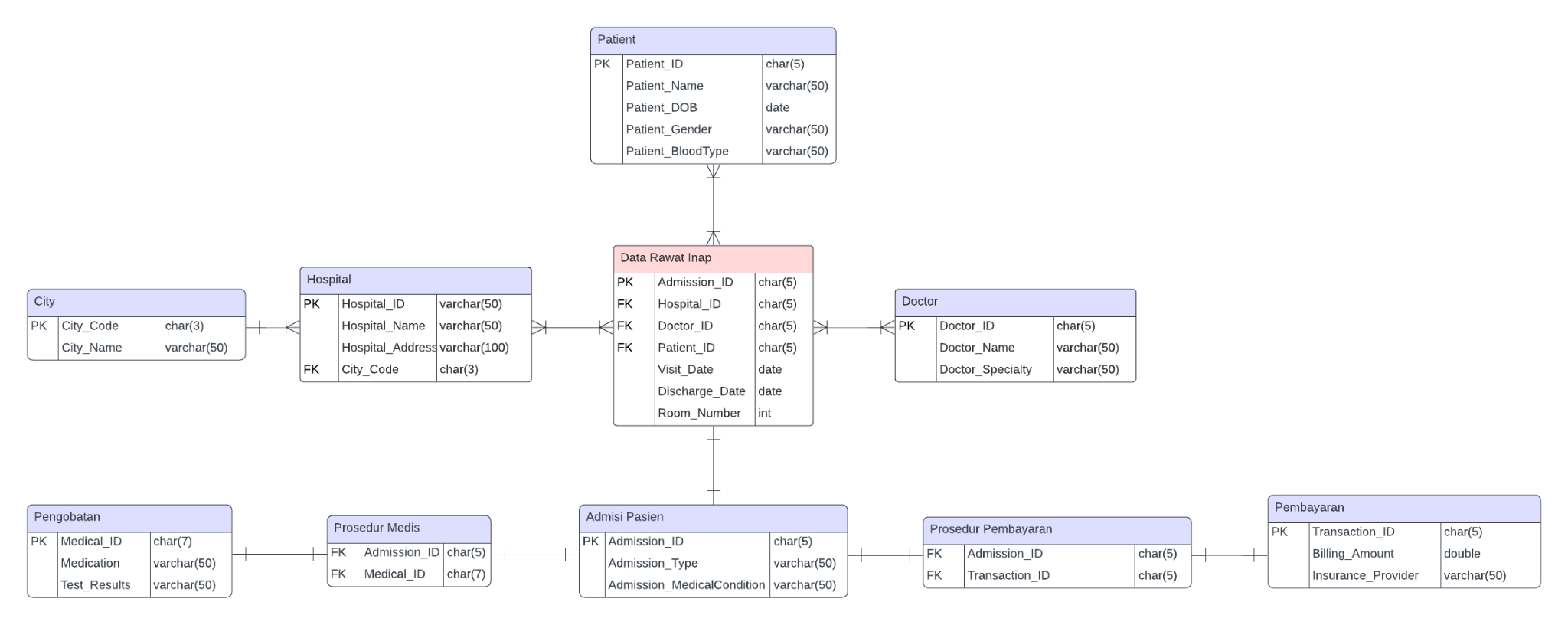
| **ADMISI** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Admission\_id | String | Identifikasi untuk setiap admisi dari pasien |
| Admisison\_Type | String | Jenis admisi dari pasien |
| Admission\_MedicalCondition | String | Kondisi medis pasien yang didiagnosa dokter |

| **PENGOBATAN** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Medical\_id | String | Identifikasi untuk setiap medis dari pasien |
| Medication | String | Obat yang diresepkan oleh dokter untuk pasien |
| Test\_Results | String | Hasil medis dari pasien |

| **PEMBAYARAN** | | |
| --- | --- | --- |
| **Atribut** | **Tipe Data** | **Deskripsi** |
| Transaction\_id | String | Identifikasi untuk setiap transaksi |
| Billing\_Amount | Integer | Jumlah tagihan dari pasien |
| Insurance\_Provider | String | Asuransi yang digunakan pasien |

**BAB V**

**ERD & Kesimpulan**

****

Dari pembahasan yang telah dilakukan dalam laporan ini, dapat disimpulkan bahwa pengelolaan data dalam sektor kesehatan merupakan aspek krusial yang tidak boleh diabaikan. Data yang beragam dan kompleks, seperti informasi pasien, dokter, rumah sakit, dan rincian medis, memerlukan sistem pengelolaan yang efisien dan terstruktur. Proses normalisasi data yang dijelaskan sebelumnya berfungsi untuk mengorganisasi informasi agar lebih teratur, mengurangi redundansi, serta meningkatkan konsistensi dan validitas data. Dengan menerapkan langkah-langkah normalisasi hingga 3NF, data kesehatan dapat dikelola dengan baik, sehingga mendukung kualitas pelayanan kesehatan yang optimal.

Pentingnya sistem database yang efisien juga terlihat pada dua skenario utama yang dibahas, yaitu perubahan kondisi medis pasien menjadi kritis dan penambahan hari untuk rawat inap. Pada kedua situasi tersebut, prosedur yang sistematis dan terintegrasi menjadi kunci untuk memastikan bahwa pasien mendapatkan penanganan yang cepat dan tepat. Pengelolaan data yang baik, mulai dari pengambilan keputusan mengenai tindakan medis hingga pembaruan administrasi, akan berdampak langsung pada pengalaman pasien dan keluarganya.

Kami menyadari bahwa masih banyak kekurangan pada laporan ini dan kami mengharapkan kritik dan saran agar kami dapat meningkatkan pengolahan sistem database yang lebih efisien. Semoga laporan ini dapat memberikan manfaat bagi pembaca. Terima kasih.

Tautan File .SQL: <https://drive.google.com/drive/folders/1UAg-ARDzzQzMCprjDmv7QvV3Fgf1l2E-?usp=sharing>

**BAB VI**

**DAFTAR PUSTAKA**

Alfia, N. E. (2020). Perancangan Aplikasi Retensi Data Pada Database MySQL (Studi Kasus: PT. Telkomsigma). *JUSIBI (Jurnal Sistem Informasi dan Bisnis)*, *2*(3), 364-374.

Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*.

Isnaini Amirotu. Pengertian Data, Fungsi, Jenis-Jenis, Manfaat, dan Contohnya. Telkom University. <https://telkomuniversity.ac.id/pengertian-data-fungsi-jenis-jenis-manfaat-dan-contohnya/>. [Diakses 25 Oktober 2024].