

LAPORAN TUGAS BESAR BASIS DATA

Administrasi Rumah Sakit

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah Basis Data Praktikum.



Disusun oleh:

Dzakira Fabillah	191524040
Putri Syalwa Abdullah	191524054
Rizka Auliarahmi	191524057

**PROGRAM STUDI D4 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG
2020**

DAFTAR ISI

LAPORAN TUGAS BESAR BASIS DATA	1
DAFTAR ISI	2
DAFTAR GAMBAR	5
DAFTAR TABEL	6
1. DESKRIPSI	8
2. RUANG LINGKUP	9
3. BUSINESS RULE	10
4. PERANCANGAN PENGOLAHAN DATA	13
4.1 Entity-Relationship Diagram (E-RD)	13
4.2 Logical Data Model	13
4.3 Relational Data Model	14
4.4 Kamus Data	14
4.4.1 Table	14
4.4.2 Sequence	19
4.4.3 Indexing	21
4.5 Function/ Procedure	22
4.6 Trigger	30
4.7 View	33
4.8 Package	34
5. IMPLEMENTASI PENGOLAHAN DATA	35
5.1 DDL	35
5.2 DML	42
5.3 PL/SQL	50
5.3.1 Function	50
5.3.2 Procedure	51
5.3.3 Trigger	61
5.4 Package	70
6. KESIMPULAN	76
7. LESSON LEARNED	77

DAFTAR GAMBAR

Gambar 1 Entity Relationship Diagram	13
Gambar 2 Logical Data Model Administrasi RS	13
Gambar 3 Relational Data Model Administrasi RS	14

DAFTAR TABEL

Tabel 1 Aturan Bisnis Pengolahan Data	9
Tabel 2 Tabel Kamus Data Dokter	13
Tabel 3 Kamus Data Pasien	14
Tabel 4 Kamus Data Pembayaran	15
Tabel 5 Kamus Data Tindakan	16
Tabel 6 Kamus Data Poliklinik	17
Tabel 7 Kamus Data Riwayat Pendaftaran	17
Tabel 8 Sequence	18
Tabel 9 Indexing	20
Tabel 10 Flowchart Function Pemasukan	21
Tabel 11 Flowchart Procedure Total Tarif	22
Tabel 12 Flowchart Procedure Jumlah Riwayat Pendaftaran	23
Tabel 13 Flowchart Procedure Insert Tindakan	23
Tabel 14 Flowchart Procedure Insert Riwayat Pendaftaran	24
Tabel 15 Flowchart Procedure Insert Poliklinik	25
Tabel 16 Flowchart Procedure Insert Pembayaran	26
Tabel 17 Flowchart Procedure Insert Pasien	27
Tabel 18 Flowchart Procedure Insert Dokter	28
Tabel 19 Flowchart Procedure Delete Dokter	29
Tabel 20 Flowchart Trigger Total Pembayaran	29
Tabel 21 Flowchart Trigger Jumlah Dokter	30
Tabel 22 Flow Chart Trigger Usia Pasien	31
Tabel 23 Rancangan Trigger Prevent Delete Dokter	31
Tabel 24 View Kunjungan Pasien Rawat Inap/Jalan	32
Tabel 25 View Data Dokter	32
Tabel 26 Rancangan Package pendaftaran_pasien	33
Tabel 27 DDL - Poliklinik	34
Tabel 28 DDL - Dokter	34
Tabel 29 DDL - Pasien	35
Tabel 30 DDL - Riwayat Pendaftaran	36
Tabel 31 DDL - Tindakan	38
Tabel 32 DDL - Pembayaran	39
Tabel 33 DDL - View Kunjungan Pasien	39
Tabel 34 DDL - View Data Dokter	40
Tabel 35 DML - Registrasi Poliklinik	41
Tabel 36 DML - Registrasi Dokter	42
Tabel 37 DML - Delete Dokter	43
Tabel 38 DML - Update Data Dokter	43
Tabel 39 DML - Registrasi Pasien	44
Tabel 40 DML - Update Data Pasien	45
Tabel 41 DML - Insert Riwayat Pendaftaran	45

Tabel 42 DML - Insert Pembayaran	46
Tabel 43 DML - Insert Tindakan	47
Tabel 44 PL/SQL – Function get income	49
Tabel 45 PL/SQL – Procedure total_tarif	50
Tabel 46 PL/SQL - Procedure jml_riwayat_pendaftaran	51
Tabel 47 PL/SQL - Procedure Insert Poliklinik	53
Tabel 48 PL/SQL - Procedure Insert Dokter	54
Tabel 49 PL/SQL - Procedure Insert Pasien	55
Tabel 50 PL/SQL - Procedure Insert Riwayat Pendaftaran	56
Tabel 51 PL/SQL - Procedure Insert Tindakan	57
Tabel 52 PL/SQL - Procedure Insert Pembayaran	58
Tabel 53 PL/SQL - Procedure Delete Dokter	59
Tabel 54 PL/SQL - Trigger total_pembayaran	60
Tabel 55 PL/SQL - Trigger jumlah_dokter	61
Tabel 56 PL/SQL - Trigger PREVENT_DOKTER_DEL	62
Tabel 57 PL/SQL - Trigger usia_pasien	63
Tabel 58 PL/SQL - Trigger POLI_ID	64
Tabel 59 PL/SQL - Trigger DOKTER_ID	65
Tabel 60 PL/SQL - Trigger REK_MEDIS_ID	66
Tabel 61 PL/SQL - Trigger REGISTRASI_NO	67
Tabel 62 PL/SQL - Trigger TINDAKAN_ID	67
Tabel 63 PL/SQL - Trigger PEMBAYARAN_ID	68
Tabel 64 Package - Package pendaftaran_pasien	69

1. DESKRIPSI

Suatu rumah sakit tentu memiliki banyak data yang harus dikelola, oleh karena itu dibuatlah suatu sistem untuk mengelola kebutuhan administrasi rumah sakit dengan deskripsi kebutuhan sebagai berikut.

Terdapat Rumah Sakit bernama Rumah Sakit DPR. Rumah sakit ini terdiri dari banyak poliklinik, seperti poliklinik anak, poliklinik bedah, poliklinik saraf, dan masih banyak lagi. Setiap poliklinik memiliki satu atau lebih dari satu dokter yang memiliki data berisi nama dokter, nomor izin praktek, spesialisasi, nomor hp, dan gaji. Pasien yang ingin berkonsultasi atau ingin memeriksakan diri akan mendaftar terlebih dahulu ke bagian pendaftaran. Di bagian pendaftaran, gejala, tindakan, usia (didapat dari tanggal lahir pasien), tipe perawatan, tanggal kedatangan, dan nomor registrasi orang tersebut akan dicatat. Jika orang tersebut baru pertama kali ke rumah sakit DPR, maka rumah sakit akan mencatat nama, tanggal lahir, jenis kelamin, dan orang tersebut akan mendapatkan no rekam medis. Namun jika orang tersebut sudah terdaftar sebagai pasien di rumah sakit DPR, maka hanya menyebutkan no rekam medis dan mendaftarkan pemeriksaan tanpa mendaftarkan data pasien kembali. Setiap tindakan pada pasien mempunyai tarif yang berbeda beda, total pembayaran yang harus dibayar pasien dihitung dari total tarif setiap tindakan yang diterimanya.

Tujuan dibuatnya sistem untuk mengelola kebutuhan administrasi rumah sakit ini yaitu agar mempermudah admin dalam mengelola data administrasi rumah sakit yang meliputi pengelolaan data poliklinik, pengelolaan data dokter, pengelolaan data pasien, pengelolaan data terkait pendaftaran pasien, dan dalam mendapatkan data pemasukan rumah sakit. Semua proses pengelolaan data ini akan dilakukan oleh admin.

2. RUANG LINGKUP

Ruang lingkup pada aplikasi ini diantaranya adalah :

1. Pengelolaan Data Poliklinik

Pada fitur ini, admin dapat mengelola data poliklinik yang terdiri dari id poliklinik, nama poliklinik, dan jumlah dokter. Poliklinik terdiri dari satu atau lebih dari satu dokter. Admin dapat menambahkan (**insert**) dan operasi **select** dalam menampilkan jumlah pendaftaran poliklinik dalam satu bulan. Admin tidak dapat mengupdate data secara manual, karena data hanya dapat di update secara otomatis, lebih spesifiknya jumlah dokter akan di update secara otomatis setiap kali ada proses insert/delete dokter. Suatu poliklinik minimal memiliki 1 dokter, sehingga dokter tidak dapat resign apabila poliklinik tempat ia praktek hanya tinggal memiliki 1 dokter. Informasi yang ditampilkan terkait Data Poliklinik ini diantaranya nama poliklinik, jumlah dokter, dan jumlah pendaftaran pada suatu poliklinik dalam setiap bulan.

2. Pengelolaan Data Dokter

Pada fitur ini, admin dapat melakukan **insert**, **delete**, **select**, dan **update** data dokter. Data dokter terdiri id dokter, nama, nomor izin praktek, nomor handphone, spesialisasi, gaji, dan id dari poliklinik dokter tersebut. Poliklinik dimana dokter tersebut terdaftar, haruslah poliklinik yang memang terdaftar di rumah sakit tersebut. Nomor izin praktek dokter bersifat unik. Data dokter akan dihapus ketika dokter tersebut sudah tidak lagi bekerja di rumah sakit tersebut. Informasi yang ditampilkan terkait Data Dokter ini diantaranya Id dokter, nama dokter, nomor izin praktek, poliklinik.

3. Pengelolaan Data Pasien

Pada fitur ini, admin dapat melakukan **insert**, **select** dan **update** data pasien. Data pasien terdiri dari nomor rekam medis, nama, tanggal lahir, dan jenis kelamin. Data pasien di insert/didaftarkan hanya pada saat kunjungan pertama pasien, sehingga untuk pendaftaran/kedatangan selanjutnya, hanya diperlukan data nomor rekam medis pasien. Informasi yang ditampilkan terkait Data Pasien ini diantaranya nomor rekam medis, nama, tanggal lahir, dan jenis kelamin.

4. Pengelolaan Data Terkait Proses Pendaftaran Pasien

Pada fitur ini, admin dapat melakukan **insert** data pendaftaran pasien ketika akan melakukan pemeriksaan/pengobatan. Data pendaftaran terdiri dari nomor rekam medis, symptom, diagnosa, tipe rawat, tanggal kedatangan, usia, dan nomor registrasi. Insert data akan dilakukan jika ada orang yang ingin berobat atau berkonsultasi baik orang yang baru pertama kali atau sudah lebih dari sekali berobat atau berkonsultasi ke Rumah Sakit DPR. Tipe rawat terbagi menjadi dua, yaitu rawat inap dan rawat jalan. Dalam satu kali pendaftaran/kunjungan, maka pasien dapat mendapatkan satu atau lebih dari satu tindakan. Tindakan-tindakan ini akan di **insert** oleh admin untuk mendapatkan jumlah biaya yang perlu dibayar pasien, berdasarkan tarif tarif setiap tindakan. Selain itu pada tindakan dilakukan juga operasi **select**. Admin Juga mengelola data pembayaran yang terdiri dari id pembayaran, total pembayaran, dan nomor registrasi. Satu pendaftaran maka akan menghasilkan satu proses pembayaran. Dalam rumah sakit ini, pembelian obat dilakukan secara terpisah sehingga tidak masuk ke proses pembayaran. Dalam proses pemeriksaan dokter hanya akan memberikan resep jika memang ada obat yang perlu dibeli. Informasi yang ditampilkan terkait Riwayat Pendaftaran Pasien ini diantaranya symptom, diagnosa, tipe rawat, tanggal kedatangan, usia, dan nomor registrasi. Sedangkan terkait pembayaran akan ditampilkan struk yang berisi id pembayaran, daftar tindakan yang dilakukan terhadap pasien, dan total biaya yang harus dibayarkan. Selain itu dapat dilihat juga informasi mengenai riwayat kunjungan pasien rawat inap/jalan diantaranya no rekam medis, nama pasien, nomor registrasi, diagnosa, total pembayaran.

3. BUSINESS RULE

Aturan bisnis yang terkait dengan proses bisnis (pengolahan data) yang akan melibatkan entitas dalam mengolah data ini yaitu:

Tabel 1 Aturan Bisnis Pengolahan Data

No.	Nama Proses Bisnis	Entitas	Aturan
1.	Pengelolaan Data Poliklinik	Poliklinik	Poliklinik memiliki id dengan aturan yaitu 5 digit pertama diisi prefix PLKNK dan 3 digit terakhir diisi nomor terurut berdasarkan kemunculan klinik di Rumah Sakit DPR yang didahului oleh padding angka 0
			Poliklinik harus memiliki nama
		Dokter	Suatu poliklinik terdiri dari satu atau lebih dokter. Dokter tidak dapat resign/di delete apabila poliklinik tempat ia praktek hanya tinggal memiliki 1 dokter.
2.	Pengelolaan Data Dokter	Dokter	Dokter harus memiliki id dengan aturan yaitu 4 digit pertama diisi prefix DKTR dan 4 digit terakhir diisi nomor terurut berdasarkan penerimaan dokter di Rumah Sakit DPR yang didahului oleh padding angka 0
			Dokter harus memiliki no izin praktek yang berisi 4 digit pertamanya yaitu bulan dan tahun dokter itu lulus izin praktek, dan 4 digit terakhir yaitu diisi nomor urutan dokter lulus izin praktek pada tahun tersebut yang didahului oleh padding angka 0.
			Dokter harus memiliki nama, spesialisasi, gaji, dan no hp

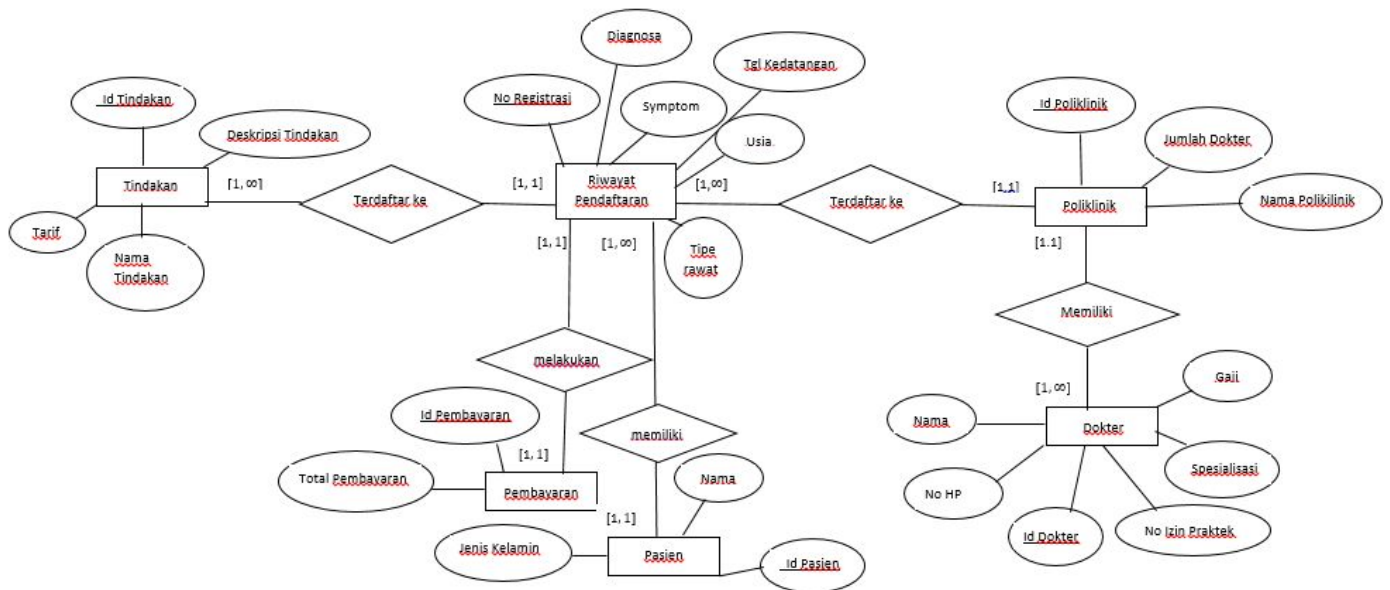
3.	Pendataan Pasien	Pasien	Attribute no rekam medis pada pasien mempunyai aturan: 1 digit pertama diisi P (diambil dari huruf pertama pada kata pasien). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0
			Satu orang pasien dapat memiliki lebih dari satu riwayat penyakit
			Setiap pasien memiliki satu no rekam medis masing-masing (dan bersifat tetap, pasien tidak perlu mendaftarkan data diri kembali jika sebelumnya sudah terdaftar)
4.	Pengelolaan Data Terkait Proses Pendaftaran Pasien	Pasien	Suatu pasien dapat mendaftarkan dirinya untuk pemeriksaan/pengobatan ke rumah sakit lebih dari satu kali
		Riwayat Pendaftaran	Diisi data mengenai symptom / gejala, diagnosa, tanggal kedatangan, tipe perawatan (inap atau jalan), dan juga usia (akan di insert otomatis berdasarkan tanggal lahir pasien)
			Riwayat Pendaftaran akan terdaftar ke poliklinik
			Attribute No registrasi pada pasien mempunyai aturan: 1 digit pertama diisi R (diambil dari huruf pertama pada kata registrasi). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0
		Pembayaran	Tarif pembayaran didasarkan pada tindakan-tindakan yang

			dilakukan dokter kepada pasien / Setiap pasien harus membayar biaya sesuai dengan tindakan yang diterima (yang tertera pada riwayat pendaftaran)
			Attribute id pembayaran memiliki ketentuan terdiri dari: 1 digit pertama diisi B (diambil dari salah satu huruf yaitu B pada kata pembayaran). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0
		Tindakan	<p>Tindakan menindaklanjuti attribute dari riwayat pendaftaran, dimana di riwayat pendaftaran terdapat symptom dan diagnosa, yang kemudian ditindaklanjuti dengan tindakan yang dilakukan dokter terhadap pasien. Misal: diberi suntikan, diberi infus, dll</p> <p>Attribute id tindakan mempunyai ketentuan terdiri dari: 1 digit pertama diisi T (diambil dari huruf pertama pada kata tindakan). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0</p> <p>Setiap tindakan memiliki tarif yang berbeda-beda</p>

4. PERANCANGAN PENGOLAHAN DATA

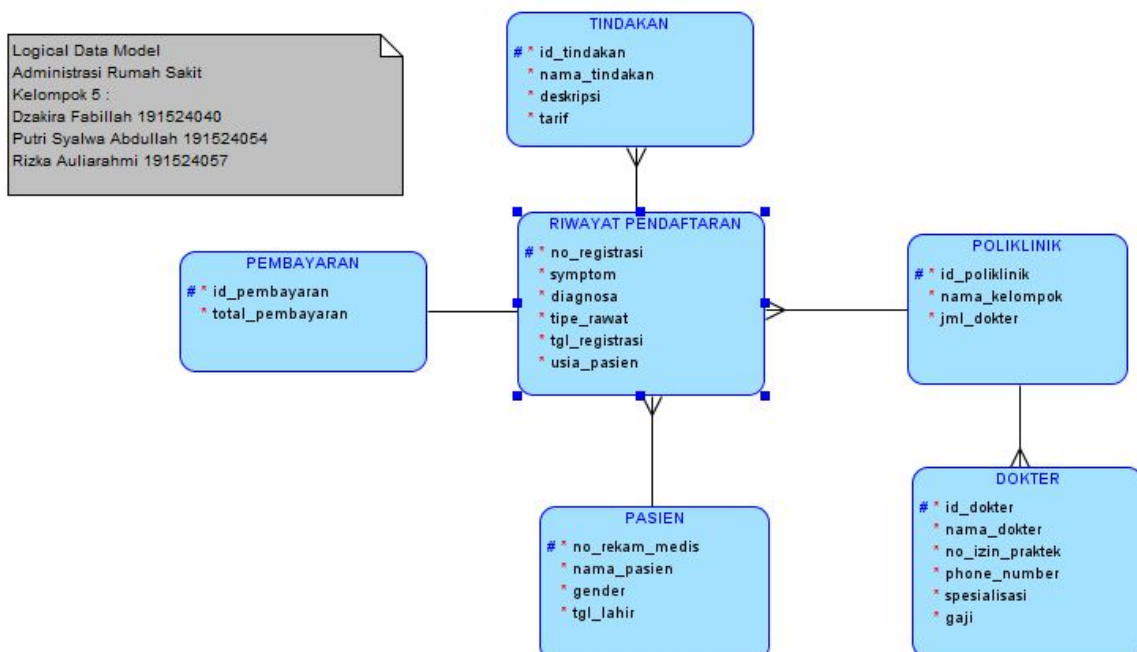
4.1 Entity-Relationship Diagram (E-RD)

Gambar 1 Entity Relationship Diagram



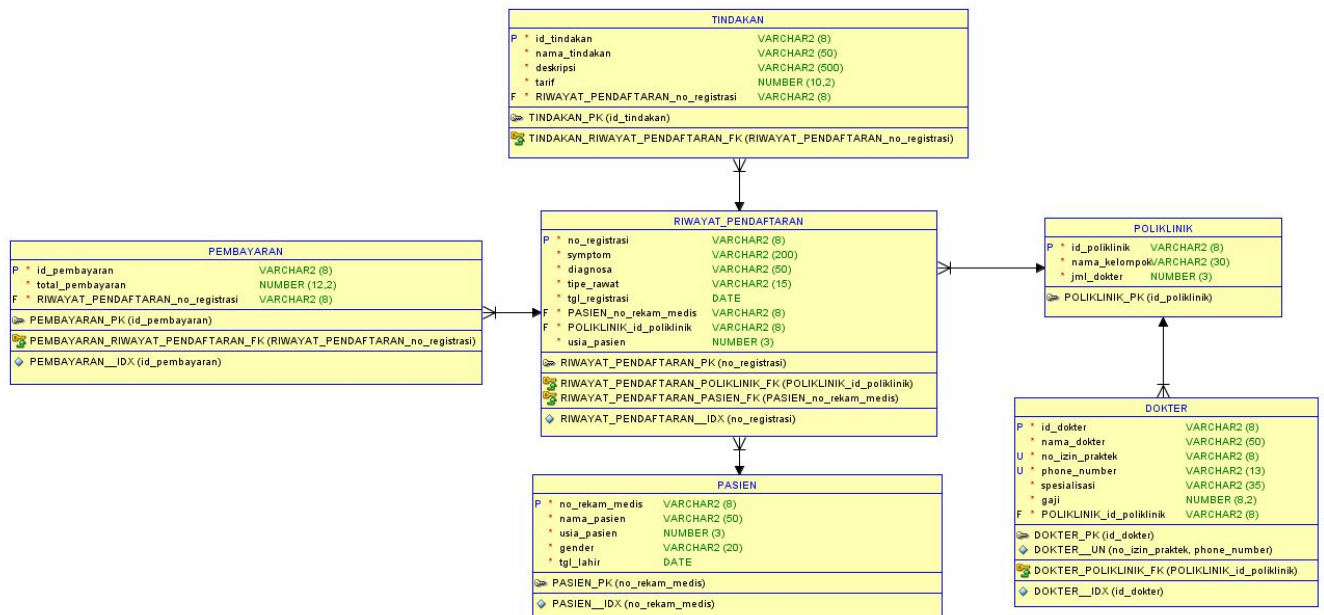
4.2 Logical Data Model

Gambar 2 Logical Data Model Administrasi RS



4.3 Relational Data Model

Gambar 3 Relational Data Model Administrasi RS



4.4 Kamus Data

4.4.1 Table

Tabel 2 Tabel Kamus Data Dokter

Nama		DOKTER			
Deskripsi		Data Dokter			
No.	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	id_dokter	varchar(8,0)	Primary Key	ID Dokter	DKTR0001 DKTR0002 DKTR0003
2.	nama_dokter	varchar(50)	NOT NULL	Nama Dokter	Rizka Auliarahmi Dzakira Fabillah Putri Syalwa

3.	no_izin_praktek	varchar(8,0)	UNIQUE	No Izin Praktek Dokter	07071642 11090633 03122407
4.	phone_number	varchar(13)	UNIQUE	Nomor Telepon Dokter	089789876543 087807895697 085721032000
5.	spesialisasi	varchar(35)	NOT NULL	Spesialisasi Dokter	Spesialis Anak Spesialis Paru Spesialis Jantung
6.	gaji	Number(8,2)	CHECK > 0	Gaji Dokter dalam 1 bulan	15.000.000 30.000.000 180.000.000
8.	id_poliklinik	varchar(8,0)	Foreign Key	ID Poliklinik	PLKNK001 PLKNK002 PLKNK003

Tabel 3 Kamus Data Pasien

Nama		PASIEN			
Deskripsi		Menyimpan Data Pasien Rumah Sakit			
No .	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	no_rekam_medis	varchar(8,0)	Primary Key	nomor rekam medis	P0000001 P0000002 P0000003
2.	nama_pasien	varchar(50)	NOT NULL	nama pasien	Ranti Fajar Vodka Bagus Setiawan
3.	tgl_lahir	DATE	NOT NULL	tanggal lahir pasien	31/12/2009

					14/02/2000 18/9/1977
4.	gender	varchar(20)	NOT NULL & CHECK	Jenis kelamin pasien, data bisa diisi antara Perempuan/ Laki-laki	Perempuan Laki-laki

Tabel 4 Kamus Data Pembayaran

Nama		PEMBAYARAN			
Deskripsi		Pembayaran dalam satu kali pendaftaran			
No	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	id_pembayaran	varchar(8)	Primary Key	kode pembayaran	B0000001 B0000002 B0000003
2.	total_pembayaran	Number(12,2)	NOT NULL	total pembayaran berdasarkan tindakan tindakan yang dilakukan dalam 1 kali kunjungan	5.00.000 1.400.000 40.000.000
3.	no_registrasi	varchar(8,0)	Foreign Key	Nomor Registrasi	R0000001 R0000002 R0000003

Tabel 5 Kamus Data Tindakan

Nama		TINDAKAN			
Deskripsi		Menyimpan Jenis jenis tindakan yang dapat dilakukan terhadap pasien			
No	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	id_tindakan	varchar(8)	Primary Key	ID tindakan	T0000001 T0000002 T0000003
2.	nama_tindakan	varchar(50)	NOT NULL	nama tindakan	Konsultasi Medis Pemeriksaan Radiologi Operasi Jantung
3.	deskripsi	varchar(500)	NOT NULL	Deskripsi tindakan	Rontgen Dada
4.	tarif	Number(10,2)	CHECK > 0	tarif tindakan	50.000 100.000 150.000
5.	no_registrasi	varchar(8,0)	Foreign Key	Nomor Registrasi	R0000001 R0000002

					R0000003
--	--	--	--	--	----------

Tabel 6 Kamus Data Poliklinik

Nama		POLIKLINIK			
Deskripsi		Menyimpan Data Poliklinik yang ada di Rumah Sakit			
No	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data
1.	id_poliklinik	varchar(8,0)	Primary Key	ID Poliklinik	PLKNK001 PLKNK002 PLKNK003
2.	nama_poliklinik	varchar(30)	NOT NULL	Nama Poliklinik	Poli Anak Poli Paru Poli Jantung/ Cardiologist
3.	jml_dokter	Integer(3)	NOT NULL	Jumlah Dokter yang ada dalam poliklinik tersebut	10 8 6

Tabel 7 Kamus Data Riwayat Pendaftaran

Nama		RIWAYAT PENDAFTARAN			
Deskripsi		Menyimpan Data Pendaftaran Suatu Pasien			
No	Nama Kolom	Tipe Data	Constraint	Deskripsi	Contoh Data

1.	no_registrasi	varchar(8,0)	Primary Key	Nomor Registrasi	R0000001 R0000002 R0000003
2.	symptom	varchar(200)	NOT NULL	Gejala atau keluhan yang dialami pasien	Batuk, Pilek, Nafas pendek, sakit tenggorokan
3.	diagnosa	varchar(50)	NOT NULL	Diagnosa penyakit pasien	Bronkitis Demam Berdarah Usus Buntu
4.	tipe_rawat	varchar(15)	NOT NULL,CHECK	Tipe rawat bisa diisi Rawat Inap/Jalan.	Rawat Jalan Rawat Inap
5.	tgl_registrasi	DATE	NOT NULL	Tanggal Pendaftaran	15 Desember 2018 19 Desember 2018 18 Oktober 2018
6.	no_rekam_medis	varchar(8,0)	Foreign Key	nomor rekam medis	P0000001 P0000002 P0000003
7.	id_poliklinik	varchar(8,0)	Foreign Key	ID Poliklinik	PLKNK001 PLKNK002 PLKNK003
8.	usia_pasien	Integer(3)	NOT NULL	usia pasien dalam hitungan tahun	11 20 45

4.4.2 Sequence

Tabel 8 Sequence

No.	Nama Tabel	Nama Kolom	Deskripsi Format	Contoh Data (Minimal 3)
1.	Dokter	id_dokter	Prefix DKTR dan nomor terurut yang didahului oleh padding angka 0	DKTR0001 DKTR0002 DKTR0003 Minimum : DKTR0001 Maksimum : DKTR9999
2.	Pasien	no_rekam_medis	1 digit pertama diisi P (diambil dari huruf pertama pada kata pasien). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0	P0000001 P0000002 P0000003 Minimum: P0000001 Maksimum: P9999999
3.	Poliklinik	id_poliklinik	Prefix PLKNK dan nomor terurut yang didahului oleh padding angka 0	PLKNK001 PLKNK002 PLKNK003 Minimum : PLKNK001 Maksimum : PLKNK999
4.	Pendaftaran	no_registrasi	1 digit pertama diisi R (diambil dari huruf pertama pada kata registrasi). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0	R0000001 R0000002 R0000003 Minimum: R0000001

				Maksimum: R9999999
5.	Tindakan	id_tindakan	1 digit pertama diisi T (diambil dari huruf pertama pada kata tindakan). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0	T0000001 T0000002 T0000003 Minimum: T0000001 Maksimum: T9999999
6.	Pembayaran	id_pembayaran	1 digit pertama diisi B (diambil dari salah satu huruf yaitu B pada kata pembayaran). 7 digit terakhir diisi nomor urutan kedatangan pasien yang didahului oleh padding angka 0	B0000001 B0000002 B0000003 Minimum: B0000001 Maksimum: B9999999

4.4.3 Indexing

Tabel 9 Indexing

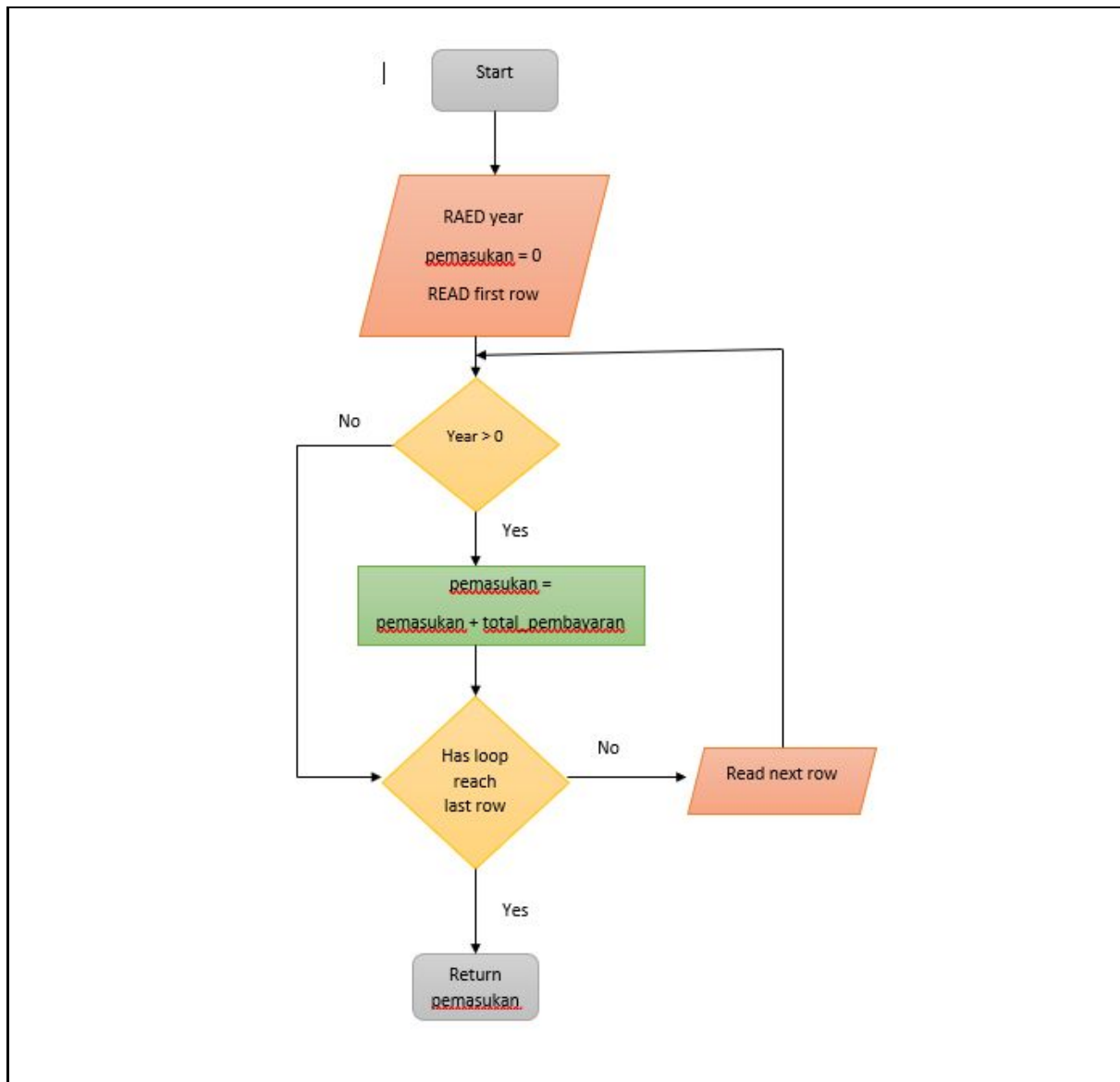
No.	Nama Tabel	Nama Index	Nama Kolom	Keterangan
1.	Dokter	Idx_Nama_Dokter	nama_dokter	CREATE INDEX Idx_Nama_Dokter ON Dokter(nama_dokter);
2.	Pasien	Idx_Nama_Pasien	nama_pasien	CREATE INDEX Idx_Nama_Pasien ON Pasien(nama_pasien);

3.	Poliklinik	Idx_Nama_Poliklinik	nama_poliklinik	CREATE INDEX Idx_Nama_Polikli nik ON Poliklinik(nama _Poliklinik);
----	------------	---------------------	-----------------	--

4.5 Function/ Procedure

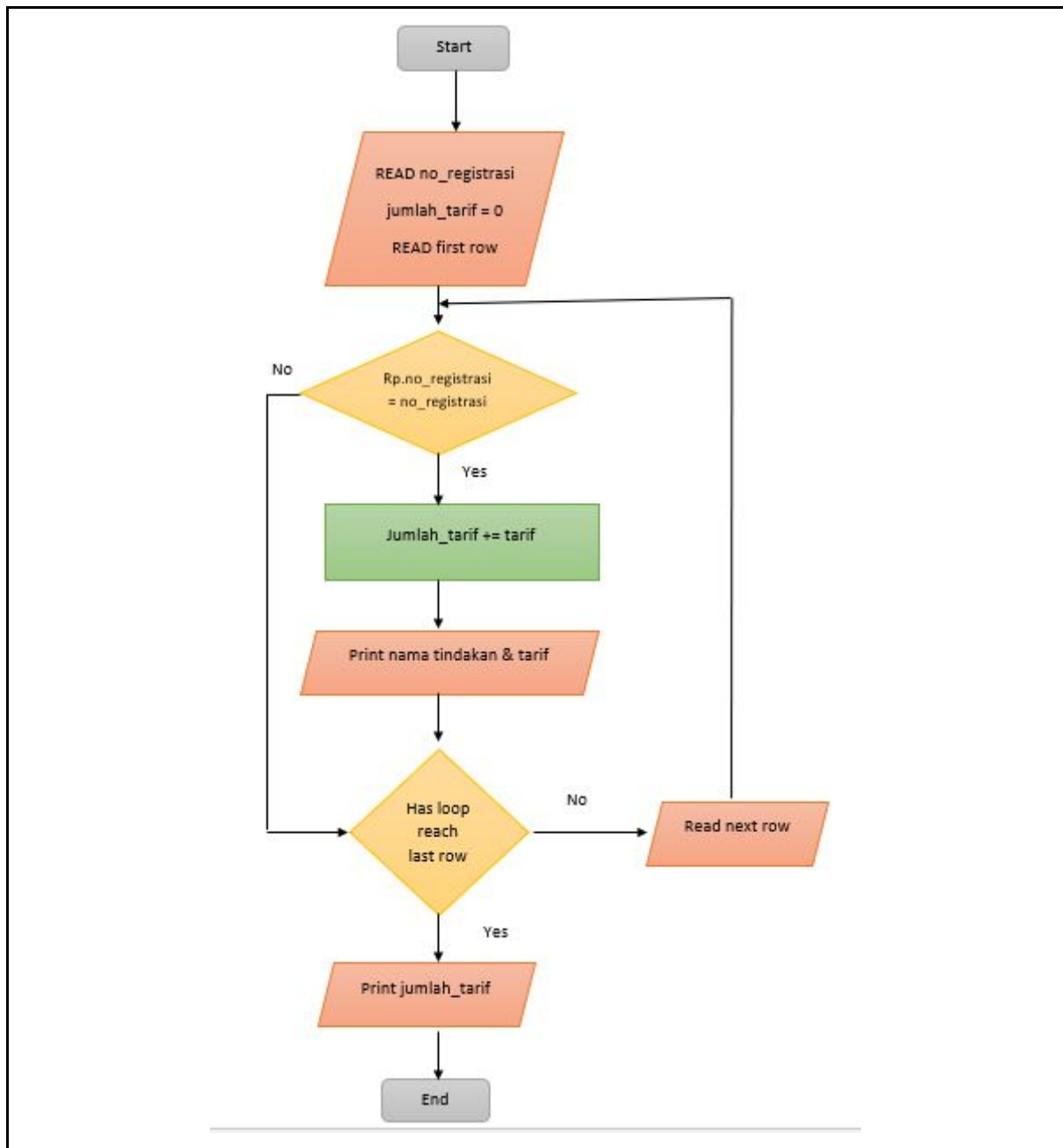
Tabel 10 Flowchart Function Pemasukan

Nama	get_income
Deskripsi	Menghitung total pemasukan RS dalam satu tahun. Diambil dari entity Riwayat_Pendaftaran dan entity Pembayaran
Parameter	in_year IN NUMBER
Flowchart	



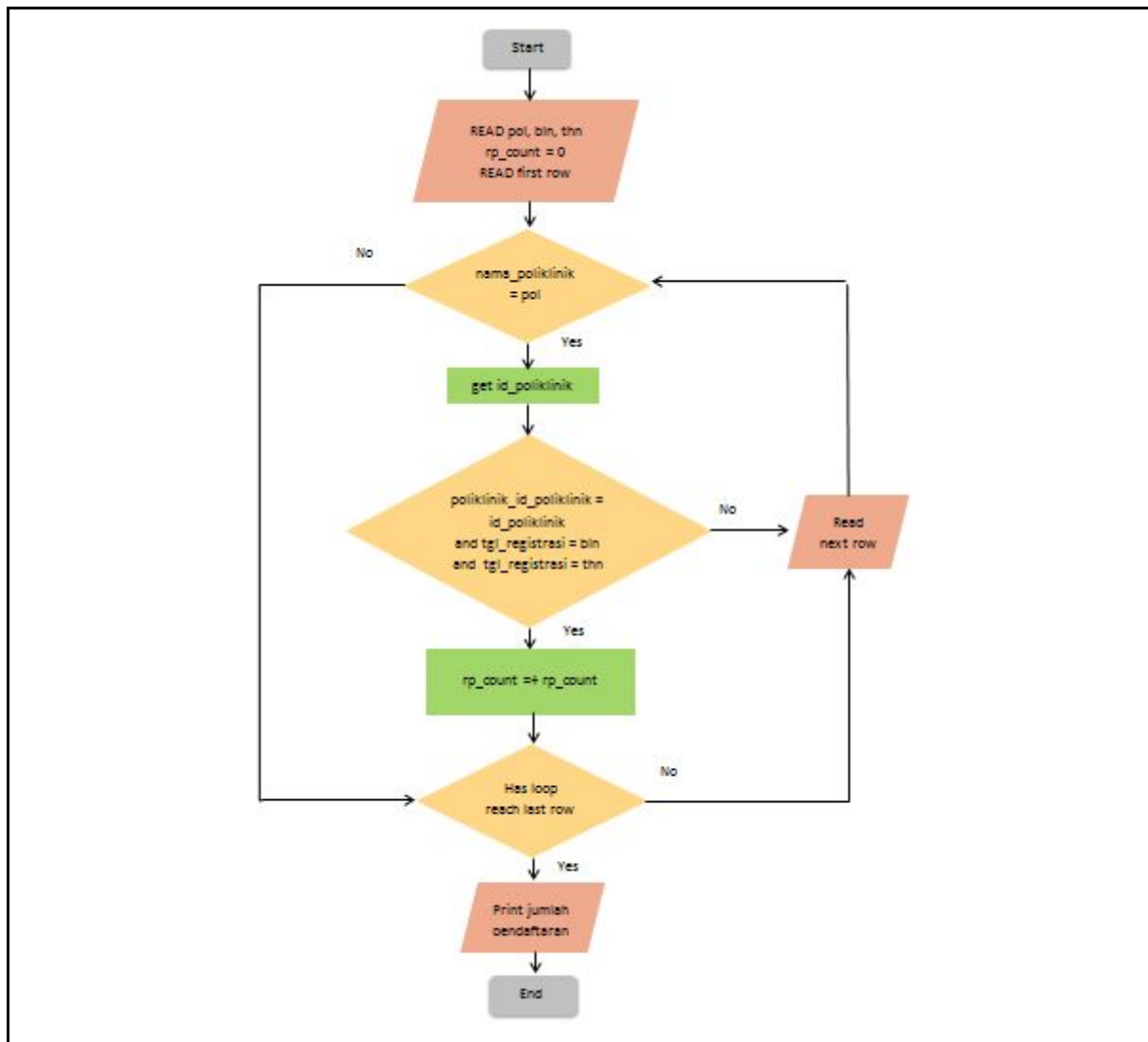
Tabel 11 Flowchart Procedure Total Tarif

Nama	Total_Tarif
Deskripsi	procedure total_tarif menampilkan struk pembayaran dari suatu pendaftaran, menampilkan nomor registrasi, tindakan apa saja yang dilakukan dan total tarif nya
Parameter	no_regis in VARCHAR2
FLOWCHART	



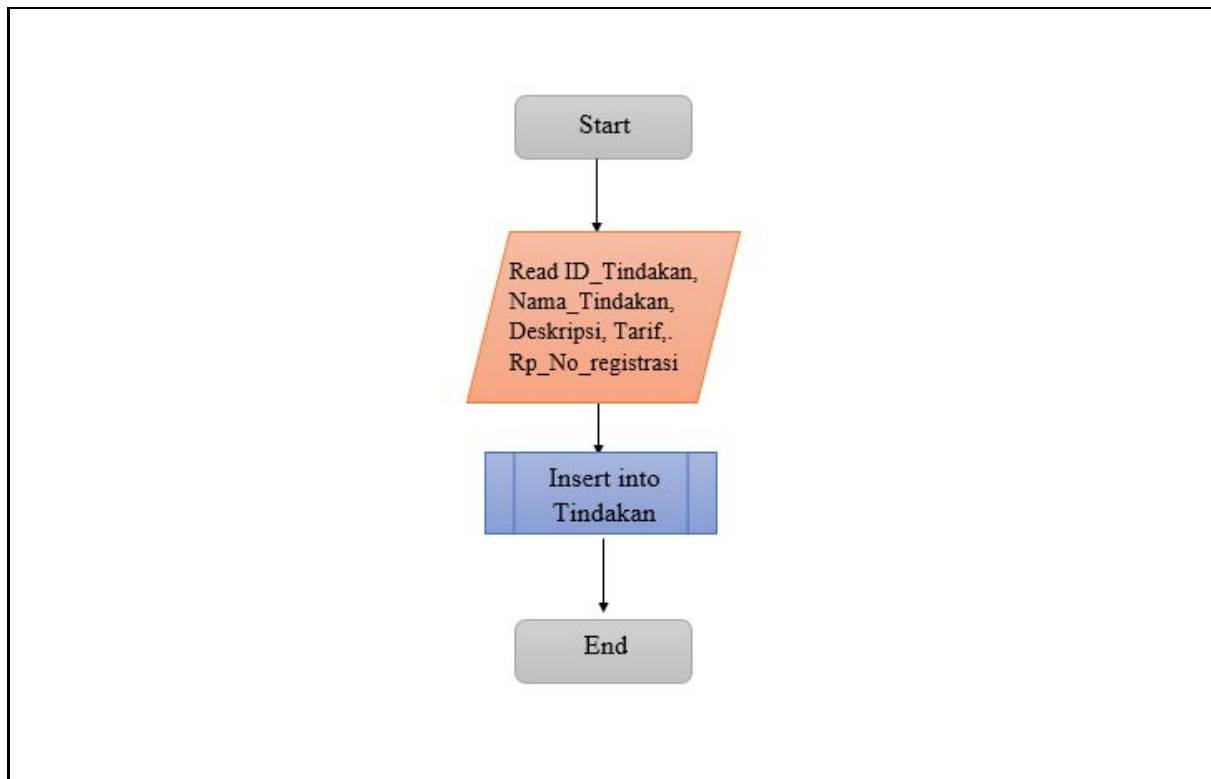
Tabel 12 Flowchart Procedure Jumlah Riwayat Pendaftaran

Nama	jml_riwayat_pendaftaran
Deskripsi	Menghitung jumlah pendaftaran berdasarkan nama poliklinik, bulan dan tahun yang dipilih.
Parameter	pol in VARCHAR2, bln in VARCHAR2, thn in VARCHAR2
FLOWCHART	



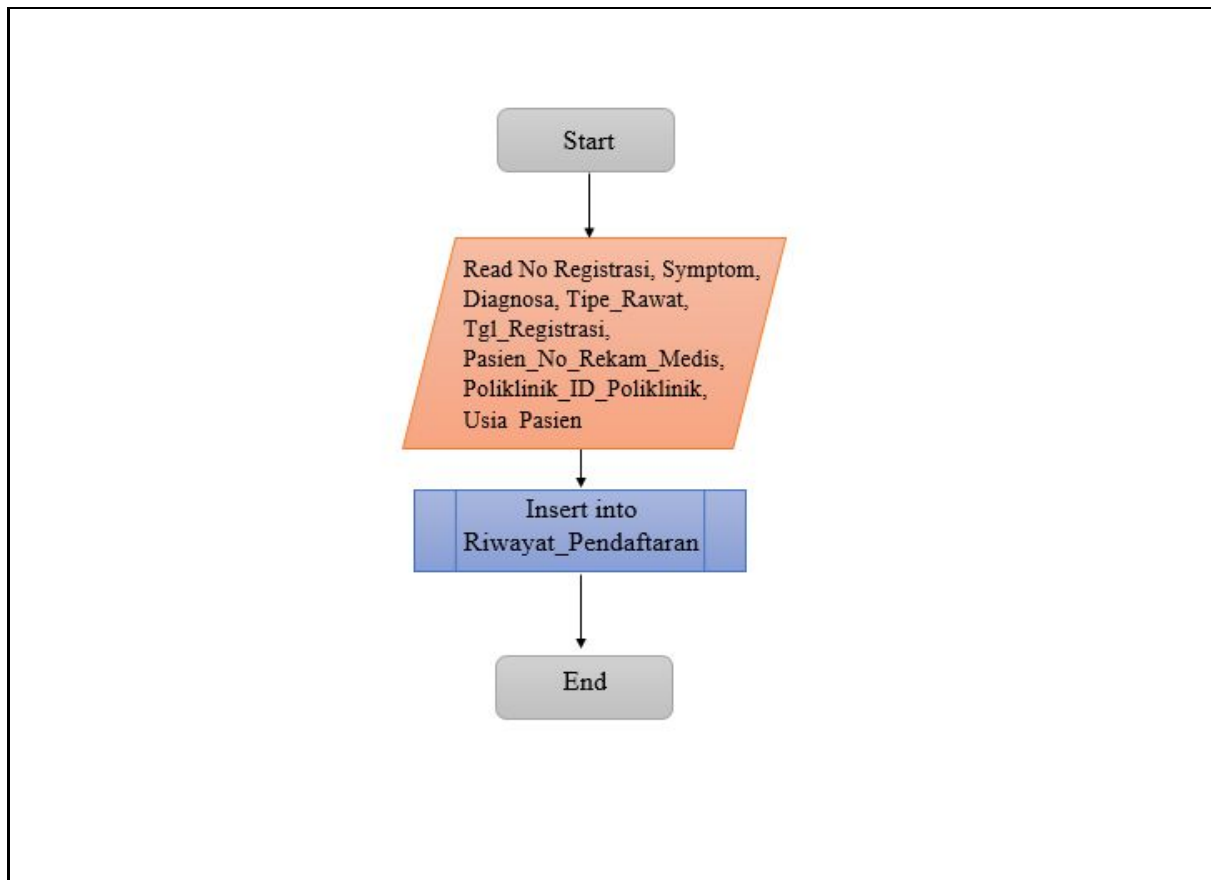
Tabel 13 Flowchart Procedure Insert Tindakan

Nama	TINDAKAN_INSERT
Deskripsi	Procedure untuk menambahkan pasien baru, id tindakan akan didapatkan secara otomatis karena adanya trigger
Parameter	ID_Tindakan in VARCHAR2, Nama_Tindakan in VARCHAR 2, Deskripsi in VARCHAR2, Tarif in VARCHAR2, RP_No_Registrasi n VARCHAR2
FLOWCHART	



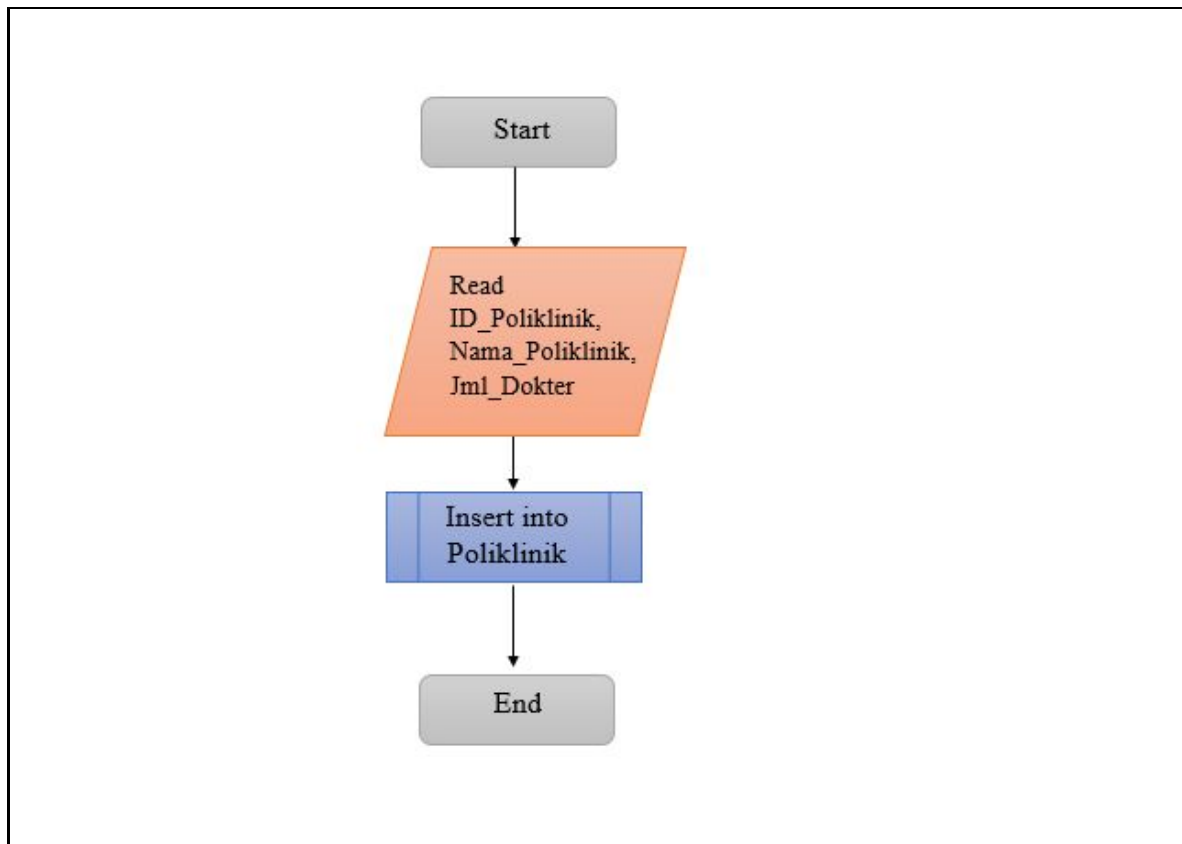
Tabel 14 Flowchart Procedure Insert Riwayat Pendaftaran

Nama	RIWAYAT_PENDAFTARAN_INSERT
Deskripsi	Procedure untuk menambahkan riwayat pendaftaran baru. no registrasi dan usia pasien akan didapatkan secara otomatis karena adanya trigger.
Parameter	Symptom in VARCHAR 2, Diagnosain VARCHAR2, Tipe_Rawat in VARCHAR2, Tgl_Registrasi in DATE, Pasien_No_Rekam_Medis in VARCHAR2, Poliklinik_ID_Poliklinik in VARCHAR2
FLOWCHART	



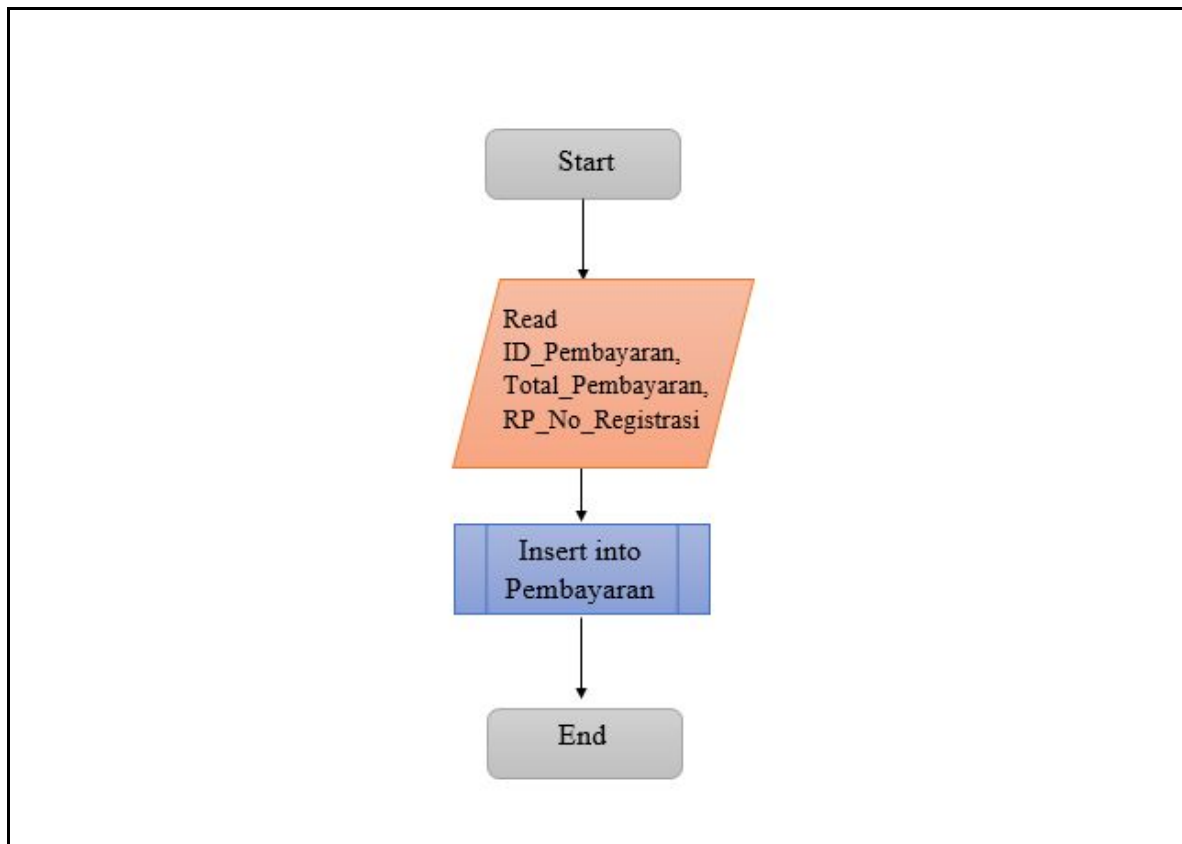
Tabel 15 Flowchart Procedure Insert Poliklinik

Nama	POLIKLINIK_INSERT
Deskripsi	Procedur untuk melakukan insert data poliklinik baru. ID poliklinik nilainya akan didapatkan secara otomatis karena adanya trigger, dan nilai jml_dokter akan di set sebagai 0.
Parameter	NAMA_POLIKLINIK in VARCHAR2
FLOWCHART	



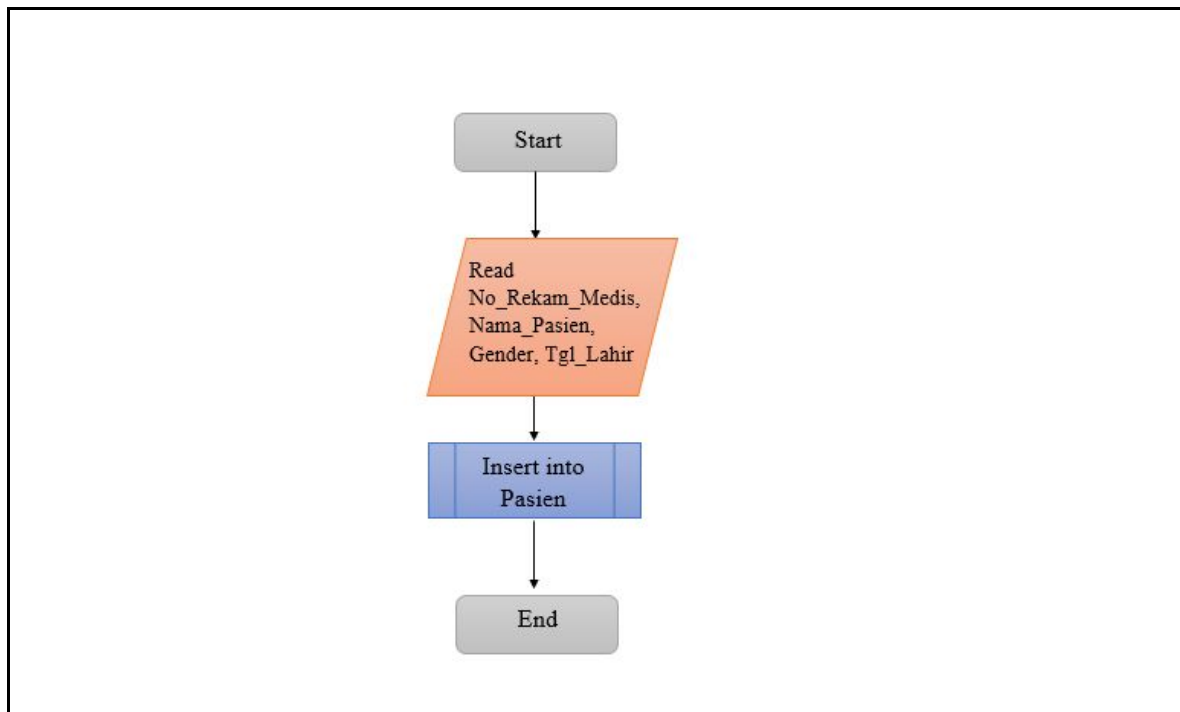
Tabel 16 Flowchart Procedure Insert Pembayaran

Nama	PEMBAYARAN_INSERT
Deskripsi	Procedure akan menambahkan data pembayaran baru. ID pembayaran dan total pembayaran akan bertambah secara otomatis dengan adanya trigger
Parameter	RP_No_Registrasi n VARCHAR2
FLOWCHART	



Tabel 17 Flowchart Procedure Insert Pasien

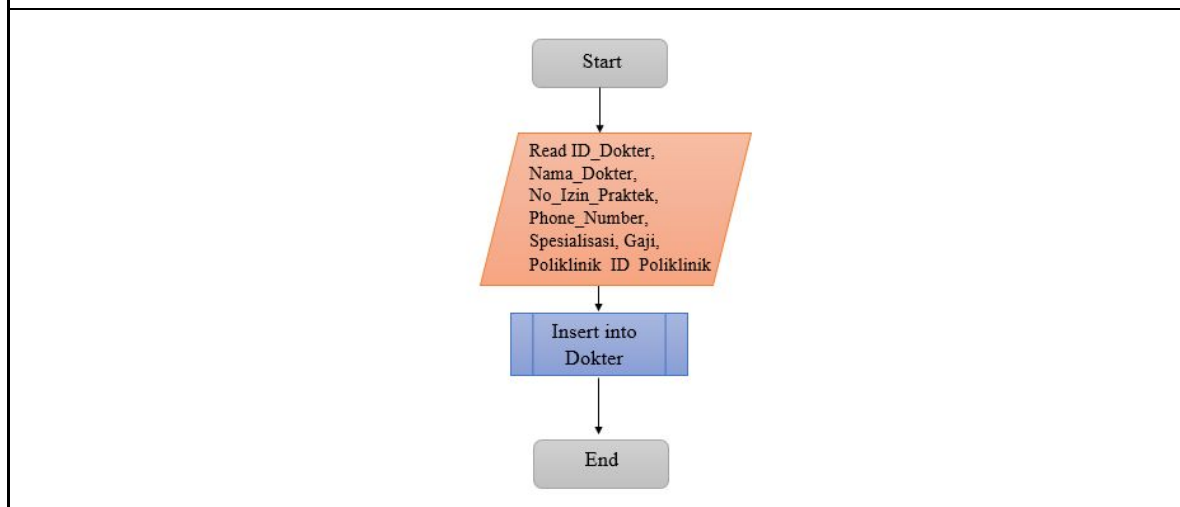
Nama	PASIEN_INSERT
Deskripsi	Procedure untuk menambahkan pasien baru, no rekam medis akan didapatkan secara otomatis karena adanya trigger
Parameter	Nama_Pasien in VARCHAR2, Gender in VARCHAR 2, Tgl_Lahir in DATE
FLOWCHART	



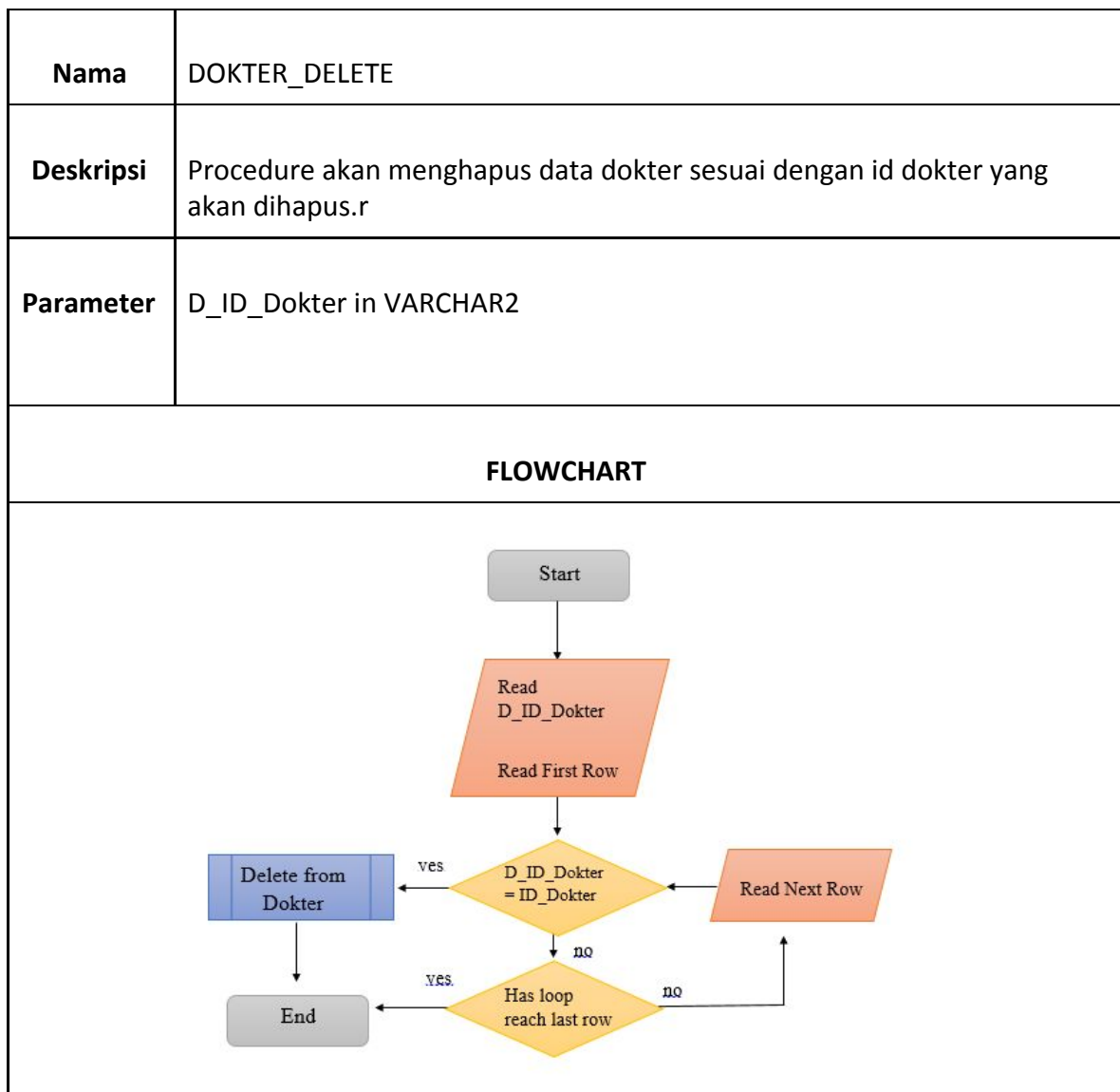
Tabel 18 Flowchart Procedure Insert Dokter

Nama	DOKTER_INSERT
Deskripsi	Procedur untuk melakukan insert data dokter baru. ID Dokter nilainya akan didapatkan secara otomatis karena adanya trigger
Parameter	Nama_Dokter in VARCHAR2, No_Izin_Praktek in VARCHAR 2, Phone_Number in VARCHAR2, Spesialisasi in VARCHAR2, Gaji in NUMBER, Poliklinik_ID_Poliklinik in VARCHAR2

FLOWCHART



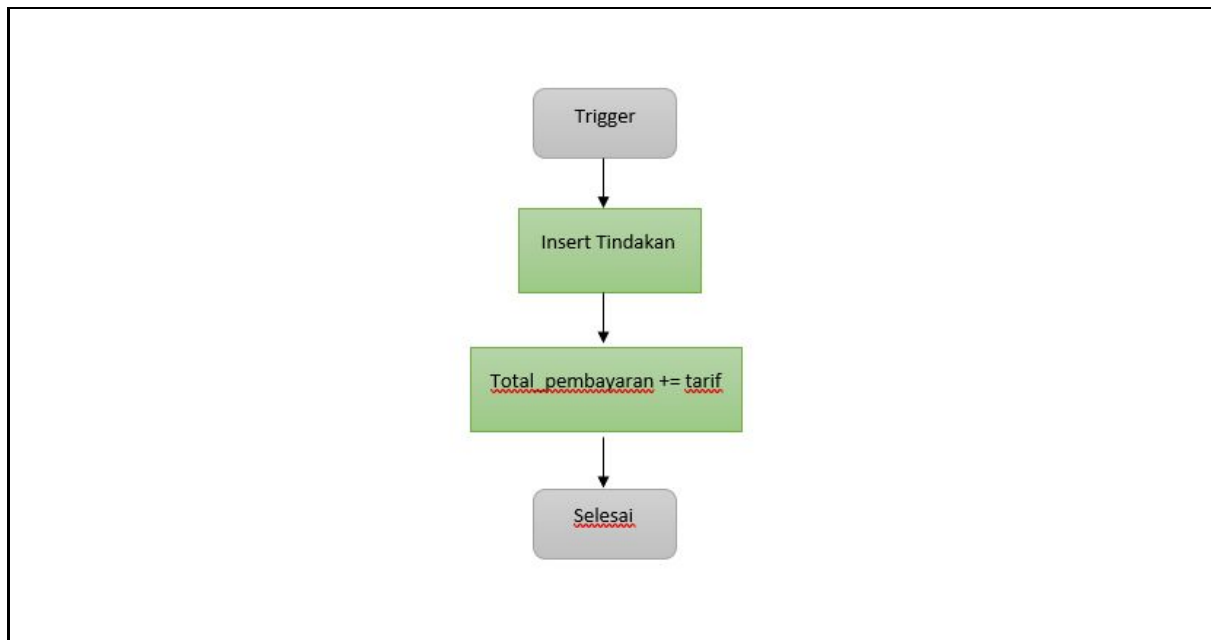
Tabel 19 Flowchart Procedure Delete Dokter



4.6 Trigger

Tabel 20 Flowchart Trigger Total Pembayaran

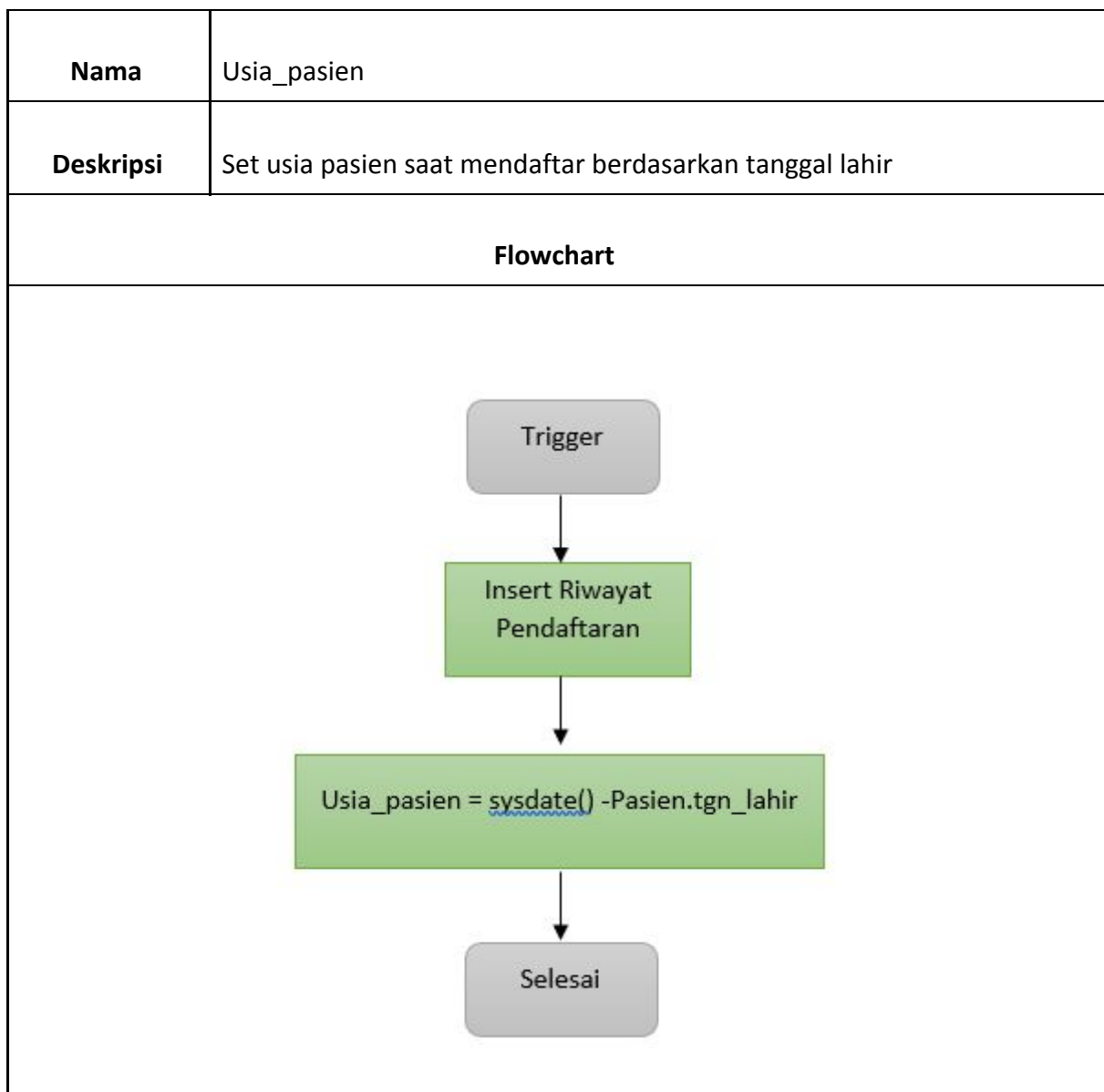
Nama	total_pembayaran
Deskripsi	Ketika suatu tindakan dengan tarif tertentu di insert, maka total pembayaran dari satu riwayat pendaftaran juga akan bertambah.
Flowchart	



Tabel 21 Flowchart Trigger Jumlah Dokter

Nama	jumlah_dokter
Deskripsi	Ketika ada dokter baru/dokter yang resign, maka jumlah_dokter di polilink tempat dokter tersebut melakukan praktek akan di update
Flowchart	
<pre> graph TD Trigger([Trigger]) --> Event{Event} Event --> InsertDokter[Insert Dokter] Event --> DeleteDokter[Delete Dokter] InsertDokter --> JmlDokterPlus["Jml dokter ++"] DeleteDokter --> JmlDokterMinus["Jml dokter --"] JmlDokterPlus --> Selesai([Selesai]) JmlDokterMinus --> Selesai </pre>	

Tabel 22 Flow Chart Trigger Usia Pasien



Tabel 23 Rancangan Trigger *Prevent Delete* Dokter

Nama	PREVENT_DOKTER_DEL
Deskripsi	Mencegah Penghapusan Data dokter, ketika poliklinik tempat dokter tersebut praktek tidak memiliki dokter lain lagi karena poliklinik minimal memiliki satu dokter.

4.7 View

Tabel 24 View Kunjungan Pasien Rawat Inap/Jalan

Nama		kunjungan_pasien
Deskripsi		View ini ditujukan untuk memudahkan dalam membuat report pelayanan terhadap pasien dalam setiap kedatangan/pendaftaran pasien rawat inap/jalan(menggunakan condition). Diambil dari entity Pasien, Riwayat Pendaftaran, Pembayaran.
No.	Nama Kolom	Tipe Data
1.	no_rekam_medis	varchar (8)
2.	nama_pasien	varchar (50)
3.	no_registrasi	varchar (8)
4.	diagnosa	varchar(50)
5.	total_pembayaran	number(10,2)

Tabel 25 View Data Dokter

Nama		data_dokter
Deskripsi		View ini ditujukan untuk melihat data dokter yang bekerja pada rumah sakit beserta poliklinik tempat ia melakukan praktek. Diambil dari entity Dokter dan Poliklinik.
No.	Nama Kolom	Tipe Data

1.	id_dokter	varchar(8)
2.	nama_dokter	varchar(50)
3.	spesialisasi	varchar(35)
4.	nama_poliklinik	varchar(30)

4.8 Package

Tabel 26 Rancangan Package pendaftaran_pasien

Nama Package	pendaftaran_pasien
Deskripsi	Menampung semua function dan procedure yang berkaitan dengan proses pendaftara pasien sampai dengan proses pembayaran.

5. IMPLEMENTASI PENGOLAHAN DATA

5.1 DDL

Tabel 27 DDL - Poliklinik

Nama Tabel	POLIKLINIK																												
Deskripsi	Menyimpan Data Poliklinik yang ada di Rumah Sakit																												
Script SQL																													
<pre>CREATE TABLE poliklinik (id_poliklinik VARCHAR2(8) NOT NULL, nama_poliklinik VARCHAR2(30) NOT NULL, jml_dokter NUMBER(3) NOT NULL, CONSTRAINT poliklinik_pk PRIMARY KEY (id_poliklinik),);</pre>																													
Screenshot Hasil																													
<table><tr><th></th><th>❖ COLUMN_NAME</th><th>❖ DATA_TYPE</th><th>❖ NULLABLE</th><th>DATA_DEFAULT</th><th>❖ COLUMN_ID</th><th>❖ COMMENTS</th></tr><tr><td>1</td><td>ID_POLIKLINIK</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>1 (null)</td><td></td></tr><tr><td>2</td><td>NAMA_POLIKLINIK</td><td>VARCHAR2(30 BYTE)</td><td>No</td><td>(null)</td><td>2 (null)</td><td></td></tr><tr><td>3</td><td>JML_DOKTER</td><td>NUMBER(3,0)</td><td>No</td><td>(null)</td><td>3 (null)</td><td></td></tr></table>			❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS	1	ID_POLIKLINIK	VARCHAR2(8 BYTE)	No	(null)	1 (null)		2	NAMA_POLIKLINIK	VARCHAR2(30 BYTE)	No	(null)	2 (null)		3	JML_DOKTER	NUMBER(3,0)	No	(null)	3 (null)	
	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS																							
1	ID_POLIKLINIK	VARCHAR2(8 BYTE)	No	(null)	1 (null)																								
2	NAMA_POLIKLINIK	VARCHAR2(30 BYTE)	No	(null)	2 (null)																								
3	JML_DOKTER	NUMBER(3,0)	No	(null)	3 (null)																								

Tabel 28 DDL - Dokter

Nama Tabel	DOKTER
Deskripsi	Data Dokter

Script SQL

```
CREATE TABLE dokter (
    id_dokter          VARCHAR2(8) NOT NULL,
    nama_dokter        VARCHAR2(50) NOT NULL,
    no_izin_praktek    VARCHAR2(8) NOT NULL,
    phone_number        VARCHAR2(13) NOT NULL,
    spesialisasi        VARCHAR2(35) NOT NULL,
    gaji               NUMBER(8, 2) NOT NULL,
    poliklinik_id_poliklinik VARCHAR2(8) ,
    CONSTRAINT dokter_pk PRIMARY KEY ( id_dokter ),
    CONSTRAINT dokter__un UNIQUE ( no_izin_praktek, phone_number ),
    CONSTRAINT poliklinik_dokter2_fk FOREIGN KEY (
poliklinik_id_poliklinik )
        REFERENCES poliklinik ( id_poliklinik )
    ON DELETE SET NULL,
    CONSTRAINT gaji_dokter_min
        CHECK (gaji > 0)
);
```

Screenshot Hasil

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_DOKTER	VARCHAR2(8 BYTE)	No	(null)	1 (null)	
2 NAMA_DOKTER	VARCHAR2(50 BYTE)	No	(null)	2 (null)	
3 NO_Izin_PRAKTEK	VARCHAR2(8 BYTE)	No	(null)	3 (null)	
4 PHONE_NUMBER	VARCHAR2(13 BYTE)	No	(null)	4 (null)	
5 SPESIALISASI	VARCHAR2(35 BYTE)	No	(null)	5 (null)	
6 GAJI	NUMBER(8,2)	No	(null)	6 (null)	
7 POLIKLINIK_ID_POLIKLINIK	VARCHAR2(8 BYTE)	No	(null)	7 (null)	

Tabel 29 DDL - Pasien

Nama Tabel	PASIEN																																			
Deskripsi	Menyimpan Data Pasien Rumah Sakit																																			
Script SQL																																				
<pre>CREATE TABLE pasien (no_rekam_medis VARCHAR2(8) NOT NULL, nama_pasien VARCHAR2(50) NOT NULL, gender VARCHAR2(20) NOT NULL, tgl_lahir DATE NOT NULL, CONSTRAINT pasien_pk PRIMARY KEY (no_rekam_medis), CONSTRAINT cek_valid_gender CHECK (gender = 'PEREMPUAN' OR gender = 'LAKI-LAKI'));</pre>																																				
Screenshot Hasil																																				
<table><tr><th></th><th>❖ COLUMN_NAME</th><th>❖ DATA_TYPE</th><th>❖ NULLABLE</th><th>DATA_DEFAULT</th><th>❖ COLUMN_ID</th><th>❖ COMMENTS</th></tr><tr><td>1</td><td>NO_REKAM_MEDIS</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>1</td><td>(null)</td></tr><tr><td>2</td><td>NAMA_PASIEN</td><td>VARCHAR2(50 BYTE)</td><td>No</td><td>(null)</td><td>2</td><td>(null)</td></tr><tr><td>3</td><td>GENDER</td><td>VARCHAR2(20 BYTE)</td><td>No</td><td>(null)</td><td>3</td><td>(null)</td></tr><tr><td>4</td><td>TGL_LAHIR</td><td>DATE</td><td>No</td><td>(null)</td><td>4</td><td>(null)</td></tr></table>			❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS	1	NO_REKAM_MEDIS	VARCHAR2(8 BYTE)	No	(null)	1	(null)	2	NAMA_PASIEN	VARCHAR2(50 BYTE)	No	(null)	2	(null)	3	GENDER	VARCHAR2(20 BYTE)	No	(null)	3	(null)	4	TGL_LAHIR	DATE	No	(null)	4	(null)
	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS																														
1	NO_REKAM_MEDIS	VARCHAR2(8 BYTE)	No	(null)	1	(null)																														
2	NAMA_PASIEN	VARCHAR2(50 BYTE)	No	(null)	2	(null)																														
3	GENDER	VARCHAR2(20 BYTE)	No	(null)	3	(null)																														
4	TGL_LAHIR	DATE	No	(null)	4	(null)																														

Tabel 30 DDL - Riwayat Pendaftaran

Nama Tabel	RIWAYAT PENDAFTARAN
Deskripsi	Menyimpan Data Pendaftaran Suatu Pasien
Script SQL	

```

CREATE TABLE riwayat_pendaftaran (

    no_registrasi          VARCHAR2(8) NOT NULL,

    symptom                VARCHAR2(200) NOT NULL,

    diagnosa               VARCHAR2(50) NOT NULL,

    tipe_rawat             VARCHAR2(15) NOT NULL,

    tgl_registrasi         DATE NOT NULL,

    usia_pasien            NUMBER(3) NOT NULL,

    pasien_no_rekam_medis  VARCHAR2(8) NOT NULL,

    poliklinik_id_poliklinik VARCHAR2(8) ,

    CONSTRAINT riwayat_pendaftaran_pk PRIMARY KEY ( no_registrasi ),

    CONSTRAINT riwayat_pendaftaran_pasien_fk FOREIGN KEY (
pasien_no_rekam_medis )

        REFERENCES pasien ( no_rekam_medis )

        ON DELETE SET NULL,

    CONSTRAINT rp_poliklinik_fk FOREIGN KEY ( poliklinik_id_poliklinik
)

        REFERENCES poliklinik ( id_poliklinik )

        ON DELETE SET NULL,

    CONSTRAINT cek_valid_tipe_rawat

        CHECK (tipe_rawat = 'INAP' OR tipe_rawat = 'JALAN')

);

```

Screenshot Hasil

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	NO_REGISTRASI		VARCHAR2(8 BYTE)	No		(null)		1	(null)	
2	SYMPTOM		VARCHAR2(200 BYTE)	No		(null)		2	(null)	
3	DIAGNOSA		VARCHAR2(50 BYTE)	No		(null)		3	(null)	
4	TIPE_RAWAT		VARCHAR2(15 BYTE)	No		(null)		4	(null)	
5	TGL_REGISTRASI		DATE	No		(null)		5	(null)	
6	USIA_PASIEN		NUMBER(3,0)	No		(null)		6	(null)	
7	PASIEN_NO_REKAM_MEDIS		VARCHAR2(8 BYTE)	No		(null)		7	(null)	
8	POLIKLINIK_ID_POLIKLINIK		VARCHAR2(8 BYTE)	No		(null)		8	(null)	

Tabel 31 DDL - Tindakan

Nama Tabel	TINDAKAN																																										
Deskripsi	Menyimpan Jenis jenis tindakan yang dapat dilakukan terhadap pasien																																										
Script SQL																																											
<pre>CREATE TABLE tindakan (id_tindakan VARCHAR2(8) NOT NULL, nama_tindakan VARCHAR2(50) NOT NULL, deskripsi VARCHAR2(500) NOT NULL, tarif NUMBER(10, 2) NOT NULL, rp_no_registrasi VARCHAR2(8) NOT NULL, CONSTRAINT tindakan_pk PRIMARY KEY (id_tindakan), CONSTRAINT min_tarif CHECK (tarif > 0), CONSTRAINT tindakan_rp_fk FOREIGN KEY (rp_no_registrasi) REFERENCES riwayat_pendaftaran (no_registrasi) ON DELETE CASCADE);</pre>																																											
Screenshot Hasil																																											
<table><tr><th></th><th>❖ COLUMN_NAME</th><th>❖ DATA_TYPE</th><th>❖ NULLABLE</th><th>DATA_DEFAULT</th><th>❖ COLUMN_ID</th><th>❖ COMMENTS</th></tr><tr><td>1</td><td>ID_TINDAKAN</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>1 (null)</td><td></td></tr><tr><td>2</td><td>NAMA_TINDAKAN</td><td>VARCHAR2(50 BYTE)</td><td>No</td><td>(null)</td><td>2 (null)</td><td></td></tr><tr><td>3</td><td>DESKRIPSI</td><td>VARCHAR2(500 BYTE)</td><td>No</td><td>(null)</td><td>3 (null)</td><td></td></tr><tr><td>4</td><td>TARIF</td><td>NUMBER(10,2)</td><td>No</td><td>(null)</td><td>4 (null)</td><td></td></tr><tr><td>5</td><td>RP_NO_REGISTRASI</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>5 (null)</td><td></td></tr></table>			❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS	1	ID_TINDAKAN	VARCHAR2(8 BYTE)	No	(null)	1 (null)		2	NAMA_TINDAKAN	VARCHAR2(50 BYTE)	No	(null)	2 (null)		3	DESKRIPSI	VARCHAR2(500 BYTE)	No	(null)	3 (null)		4	TARIF	NUMBER(10,2)	No	(null)	4 (null)		5	RP_NO_REGISTRASI	VARCHAR2(8 BYTE)	No	(null)	5 (null)	
	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS																																					
1	ID_TINDAKAN	VARCHAR2(8 BYTE)	No	(null)	1 (null)																																						
2	NAMA_TINDAKAN	VARCHAR2(50 BYTE)	No	(null)	2 (null)																																						
3	DESKRIPSI	VARCHAR2(500 BYTE)	No	(null)	3 (null)																																						
4	TARIF	NUMBER(10,2)	No	(null)	4 (null)																																						
5	RP_NO_REGISTRASI	VARCHAR2(8 BYTE)	No	(null)	5 (null)																																						

Tabel 32 DDL - Pembayaran

Nama Tabel	PEMBAYARAN																												
Deskripsi	Pembayaran dalam satu kali pendaftaran																												
Script SQL																													
<pre>CREATE TABLE pembayaran (id_pembayaran VARCHAR2(8) NOT NULL, total_pembayaran NUMBER(12, 2) NOT NULL, rp_no_registrasi VARCHAR2(8) NOT NULL, CONSTRAINT pembayaran_pk PRIMARY KEY (id_pembayaran,rp_no_registrasi), CONSTRAINT pembayaran_rp_fk FOREIGN KEY (rp_no_registrasi) REFERENCES riwayat_pendaftaran (no_registrasi);</pre>																													
Screenshot Hasil																													
<table><tr><th></th><th>COLUMN_NAME</th><th>DATA_TYPE</th><th>NULLABLE</th><th>DATA_DEFAULT</th><th>COLUMN_ID</th><th>COMMENTS</th></tr><tr><td>1</td><td>ID_PEMBAYARAN</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>1</td><td>(null)</td></tr><tr><td>2</td><td>TOTAL_PEMBAYARAN</td><td>NUMBER(12,2)</td><td>No</td><td>(null)</td><td>2</td><td>(null)</td></tr><tr><td>3</td><td>RP_NO_REGISTRASI</td><td>VARCHAR2(8 BYTE)</td><td>No</td><td>(null)</td><td>3</td><td>(null)</td></tr></table>			COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	1	ID_PEMBAYARAN	VARCHAR2(8 BYTE)	No	(null)	1	(null)	2	TOTAL_PEMBAYARAN	NUMBER(12,2)	No	(null)	2	(null)	3	RP_NO_REGISTRASI	VARCHAR2(8 BYTE)	No	(null)	3	(null)
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS																							
1	ID_PEMBAYARAN	VARCHAR2(8 BYTE)	No	(null)	1	(null)																							
2	TOTAL_PEMBAYARAN	NUMBER(12,2)	No	(null)	2	(null)																							
3	RP_NO_REGISTRASI	VARCHAR2(8 BYTE)	No	(null)	3	(null)																							

Tabel 33 DDL - View Kunjungan Pasien

Nama View	kunjungan_pasien
Deskripsi	View ini ditujukan untuk memudahkan dalam membuat report pelayanan terhadap pasien dalam setiap kedatangan/pendaftaran pasien rawat inap/jalan(menggunakan condition).

Script SQL

View Pasien Rawat Inap

```
CREATE OR REPLACE VIEW kunjungan_ps_inap AS

SELECT a.no_rekam_medis, a.nama_pasien, b.no_registrasi, b.diagnosa,
c.total_pembayaran

FROM PASIEN a, RIWAYAT_PENDAFTARAN b, PEMBAYARAN c

WHERE b.tipe_rawat = 'inap'
```

View Pasien Rawat Jalan

```
CREATE OR REPLACE VIEW kunjungan_ps_jalan AS

SELECT a.no_rekam_medis, a.nama_pasien, b.no_registrasi, b.diagnosa,
c.total_pembayaran

FROM PASIEN a, RIWAYAT_PENDAFTARAN b, PEMBAYARAN c

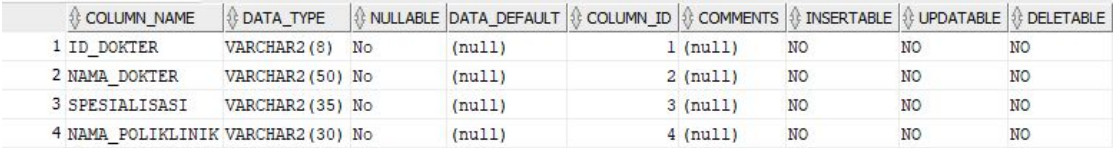
WHERE b.tipe_rawat = 'jalan'
```

Screenshot Hasil

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS	❖ INSERTABLE	❖ UPDATABLE	❖ DELETABLE
1 NO_REKAM_MEDIS	VARCHAR2(8)	No	(null)	1 (null)	NO	NO	NO	NO
2 NAMA_PASIEN	VARCHAR2(50)	No	(null)	2 (null)	NO	NO	NO	NO
3 NO_REGISTRASI	VARCHAR2(8)	No	(null)	3 (null)	NO	NO	NO	NO
4 DIAGNOSA	VARCHAR2(50)	No	(null)	4 (null)	NO	NO	NO	NO
5 TOTAL_PEMBAYARAN	NUMBER(12,2)	No	(null)	5 (null)	NO	NO	NO	NO

Tabel 34 DDL - View Data Dokter

Nama View	data_dokter
Deskripsi	<p>View ini ditujukan untuk melihat data dokter yang bekerja pada rumah sakit beserta poliklinik tempat ia melakukan praktek.</p> <p>Diambil dari entity Dokter dan Poliklinik.</p>

Script SQL																																																					
<pre>CREATE OR REPLACE VIEW data_dokter AS SELECT a.id_dokter, a.nama_dokter, a.spesialisasi, b.nama_poliklinik FROM DOKTER a, POLIKLINIK b</pre>																																																					
Screenshot Hasil																																																					
 <table border="1"> <thead> <tr> <th>COLUMN_NAME</th><th>DATA_TYPE</th><th>NULLABLE</th><th>DATA_DEFAULT</th><th>COLUMN_ID</th><th>COMMENTS</th><th>INSERTABLE</th><th>UPDATABLE</th><th>DELETABLE</th></tr> </thead> <tbody> <tr> <td>1 ID_DOKTER</td><td>VARCHAR2(8)</td><td>No</td><td>(null)</td><td>1 (null)</td><td>NO</td><td>NO</td><td>NO</td><td>NO</td></tr> <tr> <td>2 NAMA_DOKTER</td><td>VARCHAR2(50)</td><td>No</td><td>(null)</td><td>2 (null)</td><td>NO</td><td>NO</td><td>NO</td><td>NO</td></tr> <tr> <td>3 SPESIALISASI</td><td>VARCHAR2(35)</td><td>No</td><td>(null)</td><td>3 (null)</td><td>NO</td><td>NO</td><td>NO</td><td>NO</td></tr> <tr> <td>4 NAMA_POLIKLINIK</td><td>VARCHAR2(30)</td><td>No</td><td>(null)</td><td>4 (null)</td><td>NO</td><td>NO</td><td>NO</td><td>NO</td></tr> </tbody> </table>									COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE	1 ID_DOKTER	VARCHAR2(8)	No	(null)	1 (null)	NO	NO	NO	NO	2 NAMA_DOKTER	VARCHAR2(50)	No	(null)	2 (null)	NO	NO	NO	NO	3 SPESIALISASI	VARCHAR2(35)	No	(null)	3 (null)	NO	NO	NO	NO	4 NAMA_POLIKLINIK	VARCHAR2(30)	No	(null)	4 (null)	NO	NO	NO	NO
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE																																													
1 ID_DOKTER	VARCHAR2(8)	No	(null)	1 (null)	NO	NO	NO	NO																																													
2 NAMA_DOKTER	VARCHAR2(50)	No	(null)	2 (null)	NO	NO	NO	NO																																													
3 SPESIALISASI	VARCHAR2(35)	No	(null)	3 (null)	NO	NO	NO	NO																																													
4 NAMA_POLIKLINIK	VARCHAR2(30)	No	(null)	4 (null)	NO	NO	NO	NO																																													

5.2 DML

Tabel 35 DML - Registrasi Poliklinik

Nama Fitur	Registrasi Poliklinik
Deskripsi	Menambahkan data poliklinik. Data poliklinik terdiri dari id poliklinik, nama poliklinik, dan jumlah dokter. Saat insert pertama kali jml dokter di set 0 karena belum ada dokter yang terdaftar di poliklinik tersebut.
Script SQL	
<pre>INSERT ALL INTO poliklinik(id_poliklinik, nama_poliklinik, jml_dokter) VALUES ('PLKNK','POLI ANAK',0) INTO poliklinik(id_poliklinik, nama_poliklinik, jml_dokter) VALUES ('PLKNK','POLI BEDAH UMUM',0) INTO poliklinik(id_poliklinik, nama_poliklinik, jml_dokter)</pre>	

```
VALUES ('PLKNK', 'POLI BEDAH PLASTIK', 0)

INSERT INTO poliklinik(id_poliklinik, nama_poliklinik, jml_dokter)

VALUES ('PLKNK', 'POLI ORTHOPEDI', 0)

SELECT * FROM dual;
```

Screenshot Hasil

	ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER
1	PLKNK001	POLI ANAK	0
2	PLKNK002	POLI BEDAH UMUM	0
3	PLKNK003	POLI BEDAH PLASTIK	0
4	PLKNK004	POLI ORTHOPEDI	0

Tabel 36 DML - Registrasi Dokter

Nama Fitur	Registrasi Dokter
Deskripsi	Menambahkan data dokter. Data dokter terdiri id dokter, nama, nomor izin praktek, nomor handphone, spesialisasi, gaji, dan id dari poliklinik dokter tersebut. Poliklinik dimana dokter tersebut terdaftar, haruslah poliklinik yang memang terdaftar di rumah sakit tersebut.
Script SQL	
<pre>INSERT ALL INTO DOKTER (ID_DOKTER, NAMA_DOKTER, NO_Izin_PRAKTEK, PHONE_NUMBER, SPESIALISASI, GAJI, POLIKLINIK_ID_POLIKLINIK) VALUES ('DKTR','dr. Rizka Auliarahmi', '07071650', '089789996541', 'Spesialis Anak', 950000, 'PLKNK001'); INTO DOKTER (ID_DOKTER, NAMA_DOKTER, NO_Izin_PRAKTEK, PHONE_NUMBER, SPESIALISASI, GAJI, POLIKLINIK_ID_POLIKLINIK) VALUES ('DKTR', 'dr. Dzakira Fabillah', '11090633', '087807895697', 'Spesialis Bedah Umum', 300000, 'PLKNK002'); INTO DOKTER (ID_DOKTER, NAMA_DOKTER, NO_Izin_PRAKTEK, PHONE_NUMBER, SPESIALISASI, GAJI, POLIKLINIK_ID_POLIKLINIK)</pre>	

```
VALUES ('DKTR', 'dr. Ricat', '03122001', '085721032001', 'Spesialis
Bedah Plastik', 300000, 'PLKNK003');
```

```
INTO DOKTER (ID_DOKTER, NAMA_DOKTER, NO_IJIN_PRAKTEK, PHONE_NUMBER,
SPESIALISASI, GAJI, POLIKLINIK_ID_POLIKLINIK)
```

```
VALUES ('DKTR', 'dr. Putri Syalwa', '03122407', '085721032000',
'Spesialis Orthopedi', 900000, 'PLKNK004');
```

```
SELECT * FROM dual;
```

Screenshot Hasil

ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPESIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK
1 DKTR0001	dr. Rizka Auliahmami	07071650	089789996541	Spesialis Anak	950000	PLKNK001
2 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Bedah Umum	300000	PLKNK002
3 DKTR0004	dr. Ricat	03122001	085721032001	Spesialis Bedah Plastik	300000	PLKNK003
4 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Orthopedi	900000	PLKNK004

Tabel 37 DML - Delete Dokter

Nama Fitur	Delete Dokter																												
Deskripsi	Menghapus suatu dokter.																												
Script SQL																													
DELETE FROM DOKTER WHERE (id_dokter = 'DKTR0001');																													
Screenshot Hasil																													
<table><tr><th>ID_DOKTER</th><th>NAMA_DOKTER</th><th>NO_IJIN_PRAKTEK</th><th>PHONE_NUMBER</th><th>SPESIALISASI</th><th>GAJI</th><th>POLIKLINIK_ID_POLIKLINIK</th></tr><tr><td>1 DKTR0003</td><td>dr. Putri Syalwa</td><td>03122407</td><td>085721032000</td><td>Spesialis Jantung</td><td>900000</td><td>PLKNK003</td></tr><tr><td>2 DKTR0004</td><td>dr. RICAT</td><td>03122001</td><td>085721032001</td><td>Spesialis Jantung</td><td>300000</td><td>PLKNK003</td></tr><tr><td>3 DKTR0002</td><td>dr. Dzakira Fabillah</td><td>11090633</td><td>087807895697</td><td>Spesialis Paru</td><td>300000</td><td>PLKNK002</td></tr></table>		ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPESIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK	1 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKNK003	2 DKTR0004	dr. RICAT	03122001	085721032001	Spesialis Jantung	300000	PLKNK003	3 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK002
ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPESIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK																							
1 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKNK003																							
2 DKTR0004	dr. RICAT	03122001	085721032001	Spesialis Jantung	300000	PLKNK003																							
3 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK002																							

Tabel 38 DML - Update Data Dokter

Nama Fitur	Update Data Dokter
-------------------	--------------------

Deskripsi	Mengubah data dokter.																												
Script SQL																													
<pre>UPDATE dokter SET phone_number = '089765111234' WHERE (nama_dokter = 'dr. RICAT');</pre>																													
Screenshot Hasil																													
<table><tr><th>ID_DOKTER</th><th>NAMA_DOKTER</th><th>NO_IJIN_PRAKTEK</th><th>PHONE_NUMBER</th><th>SPELIALISASI</th><th>GAJI</th><th>POLIKLINIK_ID_POLIKLINIK</th></tr><tr><td>1 DKTR0003</td><td>dr. Putri Syalwa</td><td>03122407</td><td>085721032000</td><td>Spesialis Jantung</td><td>900000</td><td>PLKKNK003</td></tr><tr><td>2 DKTR0004</td><td>dr. RICAT</td><td>03122001</td><td>089765111234</td><td>Spesialis Jantung</td><td>300000</td><td>PLKKNK003</td></tr><tr><td>3 DKTR0002</td><td>dr. Dzakira Fabillah</td><td>11090633</td><td>087807895697</td><td>Spesialis Paru</td><td>300000</td><td>PLKKNK002</td></tr></table>		ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPELIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK	1 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKKNK003	2 DKTR0004	dr. RICAT	03122001	089765111234	Spesialis Jantung	300000	PLKKNK003	3 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKKNK002
ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPELIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK																							
1 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKKNK003																							
2 DKTR0004	dr. RICAT	03122001	089765111234	Spesialis Jantung	300000	PLKKNK003																							
3 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKKNK002																							

Tabel 39 DML - Registrasi Pasien

Nama Fitur	Registrasi Pasien
Deskripsi	Menambahkan data pasien. Data pasien terdiri dari nomor rekam medis, nama, usia, dan jenis kelamin. Data pasien di insert/didaftarkan hanya pada saat kunjungan pertama pasien.
Script SQL	
<pre>INSERT ALL INTO PASIEN (NO_REKAM_MEDIS, NAMA_PASIEN, GENDER, TGL_LAHIR) VALUES ('P','MARA', 'PEREMPUAN', '18-Jan-2000') INTO PASIEN (NO_REKAM_MEDIS, NAMA_PASIEN, GENDER, TGL_LAHIR) VALUES ('P', 'ZAHRA', 'PEREMPUAN', '14-Jan-1998') INTO PASIEN (NO_REKAM_MEDIS, NAMA_PASIEN, GENDER, TGL_LAHIR) VALUES ('P','ASEP', 'LAKI-LAKI', '22-Jan-1999') INTO PASIEN (NO_REKAM_MEDIS, NAMA_PASIEN, GENDER, TGL_LAHIR) VALUES ('P','CICI', 'PEREMPUAN', '22-Jan-1995') SELECT * FROM dual;</pre>	

Screenshot Hasil			
NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR
P0000001	MARA	PEREMPUAN	18-01-2000
P0000002	ZAHRA	PEREMPUAN	14-01-1998
P0000003	ASEP	LAKI-LAKI	22-01-1999
P0000004	CICI	PEREMPUAN	22-01-1995

Tabel 40 DML - Update Data Pasien

Nama Fitur	Update Pasien																
Deskripsi	Meng-update data pasien.																
Script SQL																	
<pre>UPDATE pasien set gender = 'PEREMPUAN' where (no_rekam_medis = '19000003')</pre>																	
Screenshot Hasil																	
<table><tr><th>NO_REKAM_MEDIS</th><th>NAMA_PASIEN</th><th>GENDER</th><th>TGL_LAHIR</th></tr><tr><td>P0000001</td><td>MARA</td><td>PEREMPUAN</td><td>18-01-2000</td></tr><tr><td>P0000002</td><td>ZAHRA</td><td>PEREMPUAN</td><td>14-01-1998</td></tr><tr><td>P0000003</td><td>ASEP</td><td>PEREMPUAN</td><td>22-01-1999</td></tr></table>		NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR	P0000001	MARA	PEREMPUAN	18-01-2000	P0000002	ZAHRA	PEREMPUAN	14-01-1998	P0000003	ASEP	PEREMPUAN	22-01-1999
NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR														
P0000001	MARA	PEREMPUAN	18-01-2000														
P0000002	ZAHRA	PEREMPUAN	14-01-1998														
P0000003	ASEP	PEREMPUAN	22-01-1999														

Tabel 41 DML - Insert Riwayat Pendaftaran

Nama Fitur	Insert Riwayat Pendaftaran
Deskripsi	Data pendaftaran terdiri dari symptom, diagnosa, tipe rawat, tanggal kedatangan, dan nomor registrasi. Insert data akan dilakukan jika ada orang yang ingin berobat atau berkonsultasi

	baik orang yang baru pertama kali atau sudah pernah. Value dari umur akan diisi secara otomatis menggunakan trigger berdasarkan data tgl_lahir pada data pasien.
--	--

Script SQL

```

INSERT ALL

    INTO RIWAYAT_PENDAFTARAN(NO_REGISTRASI, SYMPTOM, DIAGNOSA,
    TIPE_RAWAT, TGL_REGISTRASI, PASIEN_NO_REKAM_MEDIS,
    POLIKLINIK_ID_POLIKLINIK)

    VALUES( 'R', 'sakit perut', 'usus buntu', 'INAP', '15-Dec-2020',
    '19000001' , 'PLKNK002')

    INTO RIWAYAT_PENDAFTARAN(NO_REGISTRASI, SYMPTOM, DIAGNOSA,
    TIPE_RAWAT, TGL_REGISTRASI, PASIEN_NO_REKAM_MEDIS,
    POLIKLINIK_ID_POLIKLINIK)

    VALUES('R', 'tulang linu', 'infeksi tulang', 'INAP',
    '15-Dec-2020', '19000002' , 'PLKNK004')

    INTO RIWAYAT_PENDAFTARAN(NO_REGISTRASI, SYMPTOM, DIAGNOSA,
    TIPE_RAWAT, TGL_REGISTRASI, PASIEN_NO_REKAM_MEDIS,
    POLIKLINIK_ID_POLIKLINIK)

    VALUES( 'R', 'sakit punggung', 'skoliosis', 'INAP', '15-Dec-2020',
    '19000003' , 'PLKNK004')

    INTO RIWAYAT_PENDAFTARAN(NO_REGISTRASI, SYMPTOM, DIAGNOSA,
    TIPE_RAWAT, TGL_REGISTRASI, PASIEN_NO_REKAM_MEDIS,
    POLIKLINIK_ID_POLIKLINIK)

    VALUES('R', 'pendarahan', 'patah tulang hidung', 'INAP',
    '15-Dec-2020', '19000004' , 'PLKNK003')

SELECT * FROM dual;

```

Screenshot Hasil

NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE_RAWAT	TGL_REGISTRASI	USIA_PASIEN	PASIEN_NO_REKAM_MEDIS	POLIKLINIK_ID_POLIKLINIK
R0000001	sakit perut	usus buntu	INAP	15-12-2020	20	P0000001	PLKNK001
R0000002	tulang linu	infeksi tulang	INAP	15-12-2020	23	P0000002	PLKNK002
R0000003	sakit punggung	skoliosis	INAP	15-12-2020	21	P0000003	PLKNK004
R0000004	pendarahan	patah tulang hidung	INAP	15-12-2020	25	P0000004	PLKNK001

Tabel 42 DML - Insert Pembayaran

Nama Fitur	Insert Pembayaran
-------------------	-------------------

Deskripsi	Menambahkan data pembayaran. Data pembayaran yang terdiri dari id pembayaran, total pembayaran, dan nomor registrasi. Satu pendaftaran maka akan menghasilkan satu proses pembayaran. Saat insert total_pembayaran akan di set 0 dan nilainya secara otomatis akan bertambah ketika di insertkan tindakan yang memiliki no_registrasi yang sama.																				
Script SQL																					
<pre>INSERT ALL INTO pembayaran(id_pembayaran, total_pembayaran, rp_no_registrasi) VALUES ('B',0,'12200001') INTO pembayaran(id_pembayaran, total_pembayaran, rp_no_registrasi) VALUES ('B',0,'12200002') INTO pembayaran(id_pembayaran, total_pembayaran, rp_no_registrasi) VALUES ('B',0,'12200003') INTO pembayaran(id_pembayaran, total_pembayaran, rp_no_registrasi) VALUES ('B',0,'12200004') SELECT * FROM dual;</pre>																					
Screenshot Hasil																					
<table><tr><th></th><th>ID_PEMBAYARAN</th><th>TOTAL_PEMBAYARAN</th><th>RP_NO_REGISTRASI</th></tr><tr><td>1</td><td>00000001</td><td></td><td>0 12200001</td></tr><tr><td>2</td><td>00000002</td><td></td><td>0 12200002</td></tr><tr><td>3</td><td>00000003</td><td></td><td>0 12200003</td></tr><tr><td>4</td><td>00000004</td><td></td><td>0 12200004</td></tr></table>			ID_PEMBAYARAN	TOTAL_PEMBAYARAN	RP_NO_REGISTRASI	1	00000001		0 12200001	2	00000002		0 12200002	3	00000003		0 12200003	4	00000004		0 12200004
	ID_PEMBAYARAN	TOTAL_PEMBAYARAN	RP_NO_REGISTRASI																		
1	00000001		0 12200001																		
2	00000002		0 12200002																		
3	00000003		0 12200003																		
4	00000004		0 12200004																		

Tabel 43 DML - Insert Tindakan

Nama Fitur	Insert Tindakan
Deskripsi	Menambahkan tindakan pada suatu pendaftaran. Tindakan-tindakan ini akan di data oleh admin untuk mendapatkan jumlah biaya yang perlu dibayar pasien, berdasarkan tarif tarif setiap tindakan

Script SQL

```

INSERT ALL

    INTO tindakan(id_tindakan, nama_tindakan, deskripsi, tarif,
    RP_no_registrasi)

    VALUES ('T','Konnsultasi','Pemeriksaan fisik oleh dokter',170000 ,
    '12200001')

    INTO tindakan(id_tindakan, nama_tindakan, deskripsi, tarif,
    RP_no_registrasi)

    VALUES ('T','USG','USG untuk memastikan radang usus buntu', 523000
    , '12200001')

    INTO tindakan(id_tindakan, nama_tindakan, deskripsi, tarif,
    RP_no_registrasi)

    VALUES ('T','cek darah','Tes darah untuk memeriksa kesehatan secara
    keseluruhan',385000 , '12200001')

    INTO tindakan(id_tindakan, nama_tindakan, deskripsi, tarif,
    RP_no_registrasi)

    VALUES ('T','Apendektomi laproskopi','prosedur pembedahan dengan
    metode operasi terbuka',9000000 , '12200001')

SELECT * FROM dual;

```

Screenshot Hasil

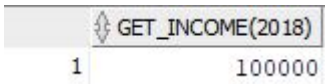
ID_TINDAKAN	NAMA_TINDAKAN	DESKRIPSI	TARIF	RP_NO_REGISTRASI
1 12200001	Konnsultasi	Pemeriksaan fisik oleh dokter	170000	12200001
2 12200002	USG	USG untuk memastikan radang usus buntu	523000	12200001
3 12200003	cek darah	Tes darah untuk memeriksa kesehatan secara keseluruhan	385000	12200001
4 12200004	Apendektomi laproskopi	prosedur pembedahan dengan metode operasi terbuka	9000000	12200001

5.3 PL/SQL

5.3.1 Function

Tabel 44 PL/SQL – Function get income

Nama Function	get_income
Parameter	in_year IN NUMBER
Deskripsi	Menghitung total pemasukan RS dalam satu tahun. Diambil dari entity Riwayat_Pendaftaran dan entity Pembayaran

Script SQL
<pre> CREATE OR REPLACE FUNCTION get_income(in_year PLS_INTEGER) RETURN NUMBER IS pemasukan NUMBER := 0; BEGIN SELECT SUM(total_pembayaran) INTO pemasukan FROM pembayaran INNER JOIN riwayat_pendaftaran ON no_registrasi = rp_no_registrasi GROUP BY EXTRACT(YEAR FROM tgl_registrasi) HAVING EXTRACT(YEAR FROM tgl_registrasi) = in_year; RETURN pemasukan; END; </pre>
Screenshot Hasil
 <p>The screenshot shows a SQL query execution result. The query is 'GET_INCOME(2018)'. The result is a single row with the value '100000'.</p>

5.3.2 Procedure

Tabel 45 PL/SQL – Procedure total_tarif

Nama Procedure	total_tarif
Parameter	no_regis in VARCHAR2

Deskripsi	procedure total_tarif menampilkan struk pembayaran dari suatu pendaftaran, menampilkan nomor registrasi, tindakan apa saja yang dilakukan dan total tarif nya
Script SQL	
<pre> create or replace PROCEDURE total_tarif (no_regis IN VARCHAR2) AS CURSOR cur_tdk IS SELECT nama_tindakan, tarif FROM tindakan; jml_tarif NUMBER; nama_tdk tindakan.nama_tindakan%TYPE; tarif_tdk tindakan.tarif%TYPE; BEGIN dbms_output.put_line('no registrasi : ' no_regis); dbms_output.put_line('tarif' ' ' 'tindakan'); SELECT SUM(tarif) INTO jml_tarif FROM tindakan WHERE tindakan.rp_no_registrasi=no_regis; OPEN cur_tdk; LOOP FETCH cur_tdk INTO nama_tdk, tarif_tdk; EXIT WHEN cur_tdk%NOTFOUND; dbms_output.put_line(tarif_tdk ' ' nama_tdk); END LOOP; CLOSE cur_tdk; dbms_output.put_line('total tarif : ' jml_tarif); </pre>	

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(SQLERRM);

END;

Screenshot Hasil

```
Connecting to the database ADM-RS.  
no registrasi :12200001  
tarif tindakan  
170000 Konnsultasi  
523000 USG  
385000 cek darah  
9000000 Apendektomi laproskopi  
total tarif :10078000  
Process exited.  
Disconnecting from the database ADM-RS.
```

Tabel 46 PL/SQL - Procedure jml_riwat_pendaftaran

Nama Procedure	jml_riwat_pendaftaran
Parameter	pol in VARCHAR2, bln in VARCHAR2, thn in VARCHAR2
Deskripsi	Menghitung jumlah pendaftaran berdasarkan nama poliklinik, bulan dan tahun yang dipilih.
Script SQL	
<pre>create or replace PROCEDURE jml_riwat_pendaftaran(pol in VARCHAR2, bln in number, thn in number) IS rp_count NUMBER; BEGIN -- get jumlah riwayat pendaftaran based on poliklinik id</pre>	

```

SELECT count(*) INTO rp_count

FROM RIWAYAT_PENDAFTARAN

WHERE poliklinik_id_poliklinik = (SELECT id_poliklinik FROM
POLIKLINIK WHERE nama_poliklinik = pol)

AND (SELECT

EXTRACT( MONTH FROM TO_DATE(
tgl_registrasi, 'DD-MM-YYYY' ) ) MONTH FROM DUAL) = bln

AND (SELECT

EXTRACT( YEAR FROM TO_DATE(
tgl_registrasi, 'DD-MM-YYYY' ) ) YEAR FROM DUAL) = thn;

-- print out information

dbms_output.put_line( 'Nama Poliklinik' || ' : ' || pol);

dbms_output.put_line( 'Bulan' || ' : ' || bln);

dbms_output.put_line( 'Tahun' || ' : ' || thn);

dbms_output.put_line( 'Jumlah Riwayat Pendaftaran' || ' : ' ||
rp_count);

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line( SQLERRM );

END;

```

Screenshot Hasil

Nama Poliklinik : POLI ORTHOPEDI
 Bulan : 12
 Tahun : 2020
 Jumlah Riwayat Pendaftaran : 2
 Process exited.
 Disconnecting from the database ADM-RS.

NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE...	TGL_REGISTRASI	USIA_PASIEN	PASIEN_...	POLIKLINI...
1 12200001	sakit perut	usus buntu	INAP	15-12-2020		20 19000001	PLKNK002
2 12200002	tulang linu	infeksi tulang	INAP	15-12-2020		23 19000002	PLKNK004
3 12200003	sakit punggung	skoliosis	INAP	15-12-2020		21 19000003	PLKNK004
4 12200004	pendarahan	patah tulang hidung	INAP	15-12-2020		25 19000004	PLKNK003

ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER
1 PLKNK004	POLI ORTHOPEDI	0
2 PLKNK001	POLI ANAK	3
3 PLKNK002	POLI BEDAH UMUM	1
4 PLKNK003	POLI BEDAH PLASTIK	2

Tabel 47 PL/SQL - Procedure Insert Poliklinik

Nama Procedure	POLIKLINIK_INSERT
Parameter	I_NAMA_POLIKLINIK
Deskripsi	Procedur untuk melakukan insert data poliklinik baru. ID poliklinik nilainya akan didapatkan secara otomatis karena adanya trigger, dan nilai jml_dokter akan di set sebagai 0.
Script SQL	
<pre> create or replace PROCEDURE POLIKLINIK_INSERT (I_NAMA_POLIKLINIK IN POLIKLINIK.NAMA_POLIKLINIK%TYPE) IS BEGIN INSERT INTO POLIKLINIK("ID_POLIKLINIK","NAMA_POLIKLINIK","JML_DOKTER") VALUES('PLKNK', I_NAMA_POLIKLINIK, 0); COMMIT; END;</pre>	

Screenshot Hasil			
	ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER
1	PLKNK004	POLI ORTHOPEDI	0
2	PLKNK001	POLI ANAK	3
3	PLKNK002	POLI BEDAH UMUM	2
4	PLKNK003	POLI BEDAH PLASTIK	2
5	PLKNK006	POLI GIGI	0
6	PLKNK005	POLI KANDUNGAN	0

Tabel 48 PL/SQL - Procedure Insert Dokter

Nama Procedure	DOKTER_INSERT
Parameter	I_NAMA_DOKTER,I_NO_IZIN_PRAKTEK,I_PHONE_NUMBER, I_SPESIALISASI,I_GAJI, I_POLIKLINIK_ID_POLIKLINIK I
Deskripsi	Procedur untuk melakukan insert data dokter baru. ID Dokter nilainya akan didapatkan secara otomatis karena adanya trigger
Script SQL	
<pre> CREATE OR REPLACE PROCEDURE DOKTER_INSERT (I_NAMA_DOKTER IN DOKTER.NAMA_DOKTER%TYPE, I_NO_IZIN_PRAKTEK IN DOKTER.NO_IZIN_PRAKTEK%TYPE, I_PHONE_NUMBER IN DOKTER.PHONE_NUMBER%TYPE, I_SPESIALISASI IN DOKTER.SPESIALISASI%TYPE, I_GAJI IN DOKTER.GAJI%TYPE, I_POLIKLINIK_ID_POLIKLINIK IN DOKTER. POLIKLINIK_ID_POLIKLINIK%TYPE) IS BEGIN INSERT INTO DOKTER("ID_DOKTER","NAMA_DOKTER","NO_IZIN_PRAKTEK","PHONE_NUMBER","SPES IALISASI","GAJI","POLIKLINIK_ID_POLIKLINIK") </pre>	

```
VALUES ('DKTR', I_NAMA_DOKTER, I_NO_IJIN_PRAKTEK, I_PHONE_NUMBER, I_SPESIALISASI, I_GAJI, I_POLIKLINIK_ID_POLIKLINIK );
```

```
COMMIT;
```

```
END;
```

Screenshot Hasil

	ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPESIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK
1	DKTR0005	dr. Anis	11090683	089502290209	Spesialisasi Paru	900000	PLKNK002
2	DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK002
3	DKTR0004	dr. RICAT	03122001	089765111234	Spesialis Jantung	300000	PLKNK003
4	DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKNK003

Tabel 49 PL/SQL - Procedure Insert Pasien

Nama Procedure	PASIEN_INSERT
Parameter	I_NAMA_PASIEN, I_GENDER, I_TGL_LAHIR
Deskripsi	Procedure untuk menambahkan pasien baru, no rekam medis akan didapatkan secara otomatis karena adanya trigger
Script SQL	
<pre>create or replace PROCEDURE PASIEN_INSERT (I_NAMA_PASIEN IN PASIEN.NAMA_PASIEN%TYPE, I_GENDER IN PASIEN.GENDER%TYPE, I_TGL_LAHIR IN PASIEN.TGL_LAHIR%TYPE) IS BEGIN INSERT INTO PASIEN("NO_REKAM_MEDIS", "NAMA_PASIEN", "GENDER", "TGL_LAHIR") VALUES ('P', I_NAMA_PASIEN, I_GENDER, I_TGL_LAHIR);</pre>	

COMMIT;
END;

Screenshot Hasil

	NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR
1	19000001	MARA	PEREMPUAN	18-01-2000
2	19000004	CICI	PEREMPUAN	22-01-1995
3	19000002	ZAHRA	PEREMPUAN	14-01-1998
4	19000003	ASEP	LAKI-LAKI	22-01-1999

Tabel 50 PL/SQL - Procedure Insert Riwayat Pendaftaran

Nama Procedure	RIWAYAT_PENDAFTARAN_INSERT
Parameter	I_SYMPTOM , I_DIAGNOSA, I_TIPE_RAWAT , I_TGL_REGISTRASI , I_PASIEN_NO_REKAM_MEDIS, I_POLIKLINIK_ID_POLIKLINIK
Deskripsi	Procedure untuk menambahkan riwayat pendaftaran baru. no registrasi dan usia pasien akan didapatkan secara otomatis karena adanya trigger.
Script SQL	
<pre> create or replace PROCEDURE RIWAYAT_PENDAFTARAN_INSERT (I_SYMPTOM IN RIWAYAT_PENDAFTARAN.SYMPTOM%TYPE, I_DIAGNOSA IN RIWAYAT_PENDAFTARAN.DIAGNOSA%TYPE, I_TIPE_RAWAT IN RIWAYAT_PENDAFTARAN.TIPE_RAWAT%TYPE, I_TGL_REGISTRASI IN RIWAYAT_PENDAFTARAN.TGL_REGISTRASI%TYPE, I_PASIEN_NO_REKAM_MEDIS IN RIWAYAT_PENDAFTARAN.PASIEN_NO_REKAM_MEDIS%TYPE, I_POLIKLINIK_ID_POLIKLINIK IN RIWAYAT_PENDAFTARAN.POLIKLINIK_ID_POLIKLINIK%TYPE) IS </pre>	


```
BEGIN
```

```
    INSERT INTO
    RIWAYAT_PENDAFTARAN("NO_REGISTRASI","SYMPTOM","DIAGNOSA","TIPE_RAWAT","
    TGL_REGISTRASI","PASIE_NO_REKAM_MEDIS","POLIKLINIK_ID_POLIKLINIK")
```

```
    VALUES('R',I_SYMPTOM, I_DIAGNOSA, I_TIPE_RAWAT, I_TGL_REGISTRASI,
    I_PASIE_NO_REKAM_MEDIS, I_POLIKLINIK_ID_POLIKLINIK);
```

```
    COMMIT;
```

```
END;
```

Screenshot Hasil

	NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE...	TGL_REGISTRASI	USIA_PASIE	PASIE_N...	POLIKLINI...
1	12200001	sakit perut	usus buntu	INAP	15-12-2020		20 19000001	PLKNK001
2	12200002	tulang linu	infeksi tulang	INAP	15-12-2020		23 19000002	PLKNK002
3	12200003	sakit punggung	skoliosis	INAP	15-12-2020		21 19000003	PLKNK004
4	12200004	pendarahan	patah tulang hidung	INAP	15-12-2020		25 19000004	PLKNK001

Tabel 51 PL/SQL - Procedure Insert Tindakan

Nama Procedure	TINDAKAN_INSERT
Parameter	I_NAMA_TINDAKAN, I_DESKRIPSI, I_TARIF, I_NO_REGIS
Deskripsi	Procedure untuk menambahkan pasien baru, id tindakan akan didapatkan secara otomatis karena adanya trigger
Script SQL	
<pre>CREATE OR REPLACE PROCEDURE TINDAKAN_INSERT (I_NAMA_TINDAKAN IN TINDAKAN.NAMA_TINDAKAN%TYPE, I_DESKRIPSI IN TINDAKAN.DESKRIPSI%TYPE, I_TARIF IN TINDAKAN.TARIF%TYPE, I_NO_REGIS IN TINDAKAN.RP_NO_REGISTRASI%TYPE) IS BEGIN</pre>	

```

INSERT INTO TINDAKAN("ID_TINDAKAN","NAMA_TINDAKAN","DESKRIPSI",
"TARIF","RP_NO_REGISTRASI")

VALUES ('T', I_NAMA_TINDAKAN, I_DESKRIPSI, I_TARIF, I_NO_REGIS);

COMMIT;

END;

```

Screenshot Hasil

	ID_TINDAKAN	NAMA_TINDAKAN	DESKRIPSI	TARIF	RP_NO_REGISTRASI
1	12200001	Konsultasi	Pemeriksaan fisik oleh dokter	170000	12200001
2	12200002	USG	USG untuk memastikan radang usus buntu	523000	12200001
3	12200003	cek darah	Tes darah untuk memeriksa kesehatan secara keseluruhan	385000	12200001
4	12200004	Apendektomi laproskopi	prosedur pembedahan dengan metode operasi terbuka	9000000	12200001

Tabel 52 PL/SQL - Procedure Insert Pembayaran

Nama Procedure	PEMBAYARAN_INSERT
Parameter	I_RP_NO_REGISTRASI
Deskripsi	Procedure akan menambahkan data pembayaran baru. ID pembayaran dan total pembayaran akan bertambah secara otomatis dengan adanya trigger
Script SQL	
<pre> create or replace PROCEDURE PEMBAYARAN_INSERT (I_RP_NO_REGISTRASI IN PEMBAYARAN.RP_NO_REGISTRASI%TYPE) IS BEGIN INSERT INTO PEMBAYARAN("ID_PEMBAYARAN","TOTAL_PEMBAYARAN","RP_NO_REGISTRASI") VALUES ('B',0, I_RP_NO_REGISTRASI); COMMIT; END; </pre>	


Screenshot Hasil		
	⚡ ID_PEMBAYARAN	⚡ TOTAL_PEMBAYARAN ⚡ RP_NO_REGISTRASI
1	00000001	0 12200001
2	00000002	0 12200002
3	00000003	0 12200003
4	00000004	0 12200004

Tabel 53 PL/SQL - Procedure Delete Dokter

Nama Procedure	DOKTER_DELETE																
Parameter	D_ID_DOKTER																
Deskripsi	Procedure akan menghapus data dokter sesuai dengan id dokter yang akan dihapus.																
Script SQL																	
<pre>create or replace PROCEDURE DOKTER_DELETE (D_ID_DOKTER IN DOKTER.ID_DOKTER%TYPE) IS BEGIN DELETE DOKTER WHERE ID_DOKTER = D_ID_DOKTER; COMMIT; END;</pre>																	
Screenshot Hasil																	
<table><tr><th></th><th>ID_POLIKLINIK</th><th>NAMA_POLIKLINIK</th><th>JML_DOKTER</th></tr><tr><td>1</td><td>PLKNK001</td><td>POLI ANAK</td><td>0</td></tr><tr><td>2</td><td>PLKNK002</td><td>POLI BEDAH UMUM</td><td>0</td></tr><tr><td>3</td><td>PLKNK003</td><td>POLI BEDAH PLASTIK</td><td>0</td></tr></table>			ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER	1	PLKNK001	POLI ANAK	0	2	PLKNK002	POLI BEDAH UMUM	0	3	PLKNK003	POLI BEDAH PLASTIK	0
	ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER														
1	PLKNK001	POLI ANAK	0														
2	PLKNK002	POLI BEDAH UMUM	0														
3	PLKNK003	POLI BEDAH PLASTIK	0														

5.3.3 Trigger

Tabel 54 PL/SQL - Trigger total_pembayaran

Nama Trigger	total_pembayaran
Aksi	INSERT OR UPDATE
Jenis	AFTER
Deskripsi	Set total pembayaran berdasarkan jumlah tarif
Script SQL	
<pre>create or replace TRIGGER Insert_Total_Pembayaran AFTER INSERT ON tindakan FOR EACH ROW WHEN (NEW.rp_no_registrasi IS NOT NULL) BEGIN UPDATE pembayaran bayar SET bayar.total_pembayaran = bayar.total_pembayaran + :NEW.tarif WHERE :NEW.rp_no_registrasi = bayar.rp_no_registrasi; END;</pre>	
Screenshot Hasil	
	

Tabel 55 PL/SQL - Trigger jumlah_dokter

Nama Trigger	jumlah_dokter
Aksi	INSERT OR DELETE
Jenis	AFTER
Deskripsi	Set jumlah dokter berdasarkan poliklinik
Script SQL	
<pre> create or replace TRIGGER jumlah_dokter AFTER INSERT OR DELETE ON DOKTER REFERENCING OLD AS OLD NEW AS NEW FOR EACH ROW DECLARE jml number; BEGIN IF INSERTING THEN UPDATE POLIKLINIK SET jml_dokter = (jml_dokter + 1) WHERE id_poliklinik = :NEW.poliklinik_id_poliklinik; ELSIF DELETING THEN UPDATE POLIKLINIK SET jml_dokter = (jml_dokter - 1) WHERE id_poliklinik = :OLD.poliklinik_id_poliklinik; END IF; END;</pre>	
Screenshot Hasil	

ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPELIALISASI	GAJI	POLIKLINIK_ID_POLIKLINIK
1 DKTR0001	dr. GUMMY	07071642	089789876543	Spesialis Anak	150000	PLKNK001
2 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK002
3 DKTR0003	dr. Putri Syalwa	03122407	085721032000	Spesialis Jantung	900000	PLKNK003
4 DKTR0004	dr. RICAT	03122001	085721032001	Spesialis Jantung	300000	PLKNK003
5 DKTR0005	dr. Rizka Auliarahmi	07071650	089789996541	Spesialis Anak	150000	PLKNK001

ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER
1 PLKNK001	Poli Anak	2
2 PLKNK002	Poli Paru	1
3 PLKNK003	Poli Jantung	2

Tabel 56 PL/SQL - Trigger PREVENT_DOKTER_DEL

Nama Trigger	PREVENT_DOKTER_DEL
Aksi	DELETE
Jenis	BEFORE
Deskripsi	Mencegah di deletenya suatu dokter apabila poliklinik tempat ia praktek tinggal memiliki 1 dokter (tidak ada dokter lain)
Script SQL	
<pre> create or replace TRIGGER PREVENT_DOKTER_DEL BEFORE DELETE ON DOKTER FOR EACH ROW DECLARE jml number; BEGIN SELECT jml_dokter INTO jml FROM POLIKLINIK WHERE id_poliklinik = :old.poliklinik_id_poliklinik; IF jml = 1 THEN </pre>	

```
RAISE_APPLICATION_ERROR (-20000, 'Dokter Pada Poliklinik Minimal 1, Tunggu Ada Dokter Baru');
```

```
ROLLBACK;
```

```
END IF;
```

```
END;
```

Screenshot Hasil

```
DELETE FROM DOKTER WHERE (id_dokter = 'DKTR0003');
```

ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPECIALISASI	GAJI	POLIKLINIK_ID_POLIK...
1 DKTR0003	dr. Ricat	03122001	085721032001	Spesialis Beda...	300000	PLKNK004
2 DKTR0005	dr. Anis	11090683	089502290209	Spesialisasi Paru	900000	PLKNK002
3 DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK001

```
ORA-20000: Dokter Pada Poliklinik Minimal 1, Tunggu Ada Dokter Baru
```

Keterangan : Dokter pada Poliklinik dengan ID PLKNK004 hanya ada 1 Dokter, sehingga delete tidak dapat dilakukan.

Tabel 57 PL/SQL - Trigger usia_pasien

Nama Trigger	usia_pasien
Aksi	INSERT
Jenis	BEFORE
Deskripsi	Set usia pasien saat mendaftar berdasarkan tanggal lahir
Script SQL	
<pre>create or replace TRIGGER usia_pasien BEFORE INSERT ON RIWAYAT_PENDAFTARAN REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW DECLARE</pre>	

```

birthdate date;

age number(3,0);

BEGIN

    SELECT tgl_lahir INTO birthdate FROM PASIEN WHERE
no_rekam_medis = :new.pasien_no_rekam_medis;

    SELECT TRUNC (TO_NUMBER (SYSDATE - TO_DATE (birthdate)) /
365.25) INTO AGE FROM DUAL;

    :new.usia_pasien := age;

END;
```

Screenshot Hasil

	NO_R...	NAMA_PA...	GENDER	TGL_LAHIR
1	19000004	CICI	PEREMPUAN	22-01-1995
2	19000003	ASEP	LAKI-LAKI	22-01-1999
3	19000002	ZAHRA	PEREMPUAN	14-01-1998
4	19000001	MARA	PEREMPUAN	18-01-2000

	NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE...	TGL_REGISTRASI	USIA_PASIEN	PASIEN_...	POLIKLINI...
1	12200001	sakit perut	usus buntu	INAP	15-12-2020	20	19000001	PLKNK002
2	12200002	tulang linu	infeksi tulang	INAP	15-12-2020	23	19000002	PLKNK004
3	12200003	sakit punggung	skoliosis	INAP	15-12-2020	21	19000003	PLKNK004
4	12200004	pendarahan	patah tulang hidung	INAP	15-12-2020	25	19000004	PLKNK003

Tabel 58 PL/SQL - Trigger POLI_ID

Nama Trigger	POLI_ID
Aksi	INSERT
Jenis	BEFORE
Deskripsi	Membuat id poliklinik secara otomatis dan sequence ketika insert data
Script SQL	


```

CREATE SEQUENCE POLIKLINIK_SEQ INCREMENT BY 1;

CREATE OR REPLACE TRIGGER Poli_ID
  BEFORE INSERT
  ON POLIKLINIK
  REFERENCING NEW AS NEW OLD AS OLD
  FOR EACH ROW
  BEGIN
    :NEW.ID_POLIKLINIK := :NEW.ID_POLIKLINIK ||
trim(to_char(POLIKLINIK_SEQ.nextval, '000'));
END Poli_ID;

```

Screenshot Hasil

ID_POLIKLINIK	NAMA_POLIKLINIK	JML_DOKTER
PLKNK001	POLI ORTHOPEDI	1
PLKNK002	POLI ANAK	1
PLKNK003	POLI BEDAH UMUM	1
PLKNK004	POLI GIGI	0
PLKNK005	POLI KANDUNGAN	0

Tabel 59 PL/SQL - Trigger DOKTER_ID

Nama Trigger	DOKTER_ID
Aksi	INSERT
Jenis	BEFORE
Deskripsi	Membuat id dokter secara otomatis dan sequence ketika insert data
Script SQL	
<pre> CREATE SEQUENCE DOKTER_SEQ INCREMENT BY 1; CREATE OR REPLACE TRIGGER Dokter_ID BEFORE INSERT ON DOKTER REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW BEGIN :NEW.ID_DOKTER := :NEW.ID_DOKTER trim(to_char(DOKTER_SEQ.nextval, '0000')); END Dokter_ID; </pre>	

Screenshot Hasil						
ID_DOKTER	NAMA_DOKTER	NO_IJIN_PRAKTEK	PHONE_NUMBER	SPEKIALISASI	GAJI	POLIKLINIK_ID_POLIK
DKTR0001	dr. Ricat	03122001	085721032001	Spesialis Bedah Plastik	300000	PLKNK004
DKTR0002	dr. Dzakira Fabillah	11090633	087807895697	Spesialis Paru	300000	PLKNK001
DKTR0003	dr. Anis	11090683	089502290209	Spesialisasi Paru	900000	PLKNK002

Tabel 60 PL/SQL - Trigger REK_MEDIS_ID

Nama Trigger	REK_MEDIS_ID																				
Aksi	INSERT																				
Jenis	BEFORE																				
Deskripsi	Membuat no rekaman medis secara otomatis dan sequence ketika insert data																				
Script SQL																					
<pre>CREATE SEQUENCE NOREK_MEDIS_SEQ INCREMENT BY 1; CREATE OR REPLACE TRIGGER REK_MEDIS_ID BEFORE INSERT ON PASIEN REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW BEGIN :NEW.no_rekam_medis := :NEW.no_rekam_medis trim(to_char(DOKTER_SEQ.nextval, '0000000')); END REK_MEDIS_ID;</pre>																					
Screenshot Hasil																					
<table><tr><th>NO_REKAM_MEDIS</th><th>NAMA_PASIEN</th><th>GENDER</th><th>TGL_LAHIR</th></tr><tr><td>P0000001</td><td>MARA</td><td>PEREMPUAN</td><td>18-01-2000</td></tr><tr><td>P0000002</td><td>ZAHRA</td><td>PEREMPUAN</td><td>14-01-1998</td></tr><tr><td>P0000003</td><td>ASEP</td><td>LAKI-LAKI</td><td>22-01-1999</td></tr><tr><td>P0000004</td><td>CICI</td><td>PEREMPUAN</td><td>22-01-1995</td></tr></table>		NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR	P0000001	MARA	PEREMPUAN	18-01-2000	P0000002	ZAHRA	PEREMPUAN	14-01-1998	P0000003	ASEP	LAKI-LAKI	22-01-1999	P0000004	CICI	PEREMPUAN	22-01-1995
NO_REKAM_MEDIS	NAMA_PASIEN	GENDER	TGL_LAHIR																		
P0000001	MARA	PEREMPUAN	18-01-2000																		
P0000002	ZAHRA	PEREMPUAN	14-01-1998																		
P0000003	ASEP	LAKI-LAKI	22-01-1999																		
P0000004	CICI	PEREMPUAN	22-01-1995																		

--

Tabel 61 PL/SQL - Trigger REGISTRASI_NO

Nama Trigger	REGISTRASI_NO																																								
Aksi	INSERT																																								
Jenis	BEFORE																																								
Deskripsi	Membuat no registrasi secara otomatis dan sequence ketika insert data																																								
Script SQL																																									
<pre>CREATE SEQUENCE NO_REGISTRASI_SEQ INCREMENT BY 1; CREATE OR REPLACE TRIGGER REGISTRASI_NO BEFORE INSERT ON RIWAYAT_PENDAFTARAN REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW BEGIN :NEW.no_registrasi := :NEW.no_registrasi trim(to_char(NO_REGISTRASI_SEQ.nextval, '0000000')); END REGISTRASI_NO;</pre>																																									
Screenshot Hasil																																									
<table><tr><th>NO_REGISTRASI</th><th>SYMPTOM</th><th>DIAGNOSA</th><th>TIPE_RAWAT</th><th>TGL_REGISTRASI</th><th>USIA_PASIE</th><th>PASIEN_NO_REKAM_MEDIS</th><th>POLIKLINIK_ID_POLIKLINIK</th></tr><tr><td>R0000001</td><td>sakit perut</td><td>usus buntu</td><td>INAP</td><td>15-12-2020</td><td>20</td><td>P0000001</td><td>PLKNK001</td></tr><tr><td>R0000002</td><td>tulang linu</td><td>infeksi tulang</td><td>INAP</td><td>15-12-2020</td><td>23</td><td>P0000002</td><td>PLKNK002</td></tr><tr><td>R0000003</td><td>sakit punggung</td><td>skoliosis</td><td>INAP</td><td>15-12-2020</td><td>21</td><td>P0000003</td><td>PLKNK004</td></tr><tr><td>R0000004</td><td>pendarahan</td><td>patah tulang hidung</td><td>INAP</td><td>15-12-2020</td><td>25</td><td>P0000004</td><td>PLKNK001</td></tr></table>		NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE_RAWAT	TGL_REGISTRASI	USIA_PASIE	PASIEN_NO_REKAM_MEDIS	POLIKLINIK_ID_POLIKLINIK	R0000001	sakit perut	usus buntu	INAP	15-12-2020	20	P0000001	PLKNK001	R0000002	tulang linu	infeksi tulang	INAP	15-12-2020	23	P0000002	PLKNK002	R0000003	sakit punggung	skoliosis	INAP	15-12-2020	21	P0000003	PLKNK004	R0000004	pendarahan	patah tulang hidung	INAP	15-12-2020	25	P0000004	PLKNK001
NO_REGISTRASI	SYMPTOM	DIAGNOSA	TIPE_RAWAT	TGL_REGISTRASI	USIA_PASIE	PASIEN_NO_REKAM_MEDIS	POLIKLINIK_ID_POLIKLINIK																																		
R0000001	sakit perut	usus buntu	INAP	15-12-2020	20	P0000001	PLKNK001																																		
R0000002	tulang linu	infeksi tulang	INAP	15-12-2020	23	P0000002	PLKNK002																																		
R0000003	sakit punggung	skoliosis	INAP	15-12-2020	21	P0000003	PLKNK004																																		
R0000004	pendarahan	patah tulang hidung	INAP	15-12-2020	25	P0000004	PLKNK001																																		

Tabel 62 PL/SQL - Trigger TINDAKAN_ID

Nama Trigger	TINDAKAN_ID
---------------------	-------------

Aksi	INSERT
Jenis	BEFORE
Deskripsi	Membuat id tindakan secara otomatis dan sequence ketika insert data
Script SQL	
<pre>CREATE SEQUENCE TINDAKAN_SEQ INCREMENT BY 1; CREATE OR REPLACE TRIGGER TINDAKAN_ID BEFORE INSERT ON TINDAKAN REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW BEGIN :NEW.id_tindakan:= :NEW.id_tindakan trim(to_char(TINDAKAN_SEQ.nextval, '0000000')); END TINDAKAN_ID;</pre>	
Screenshot Hasil	

Tabel 63 PL/SQL - Trigger PEMBAYARAN_ID

Nama Trigger	PEMBAYARAN_ID
Aksi	INSERT
Jenis	BEFORE
Deskripsi	Membuat id pembayaran secara otomatis dan sequence ketika insert data
Script SQL	

```

CREATE SEQUENCE PEMBAYARAN_SEQ INCREMENT BY 1;

CREATE OR REPLACE TRIGGER PEMBAYARAN_ID
  BEFORE INSERT
  ON PEMBAYARAN
  REFERENCING NEW AS NEW OLD AS OLD
  FOR EACH ROW
  BEGIN
    :NEW.id_pembayaran := :NEW.id_pembayaran ||
trim(to_char(PEMBAYARAN_SEQ.nextval, '0000000'));
END PEMBAYARAN_ID;

```

Screenshot Hasil

5.4 Package

Tabel 64 Package - Package pendaftaran_pasien

Nama Package	pendaftaran_pasien
Deskripsi	Menampung semua procedure yang berkaitan dengan proses pendaftaran pasien sampai dengan proses pembayaran.
Script SQL Package Specification	
<pre> create or replace package pendaftaran_pasien as procedure total_tarif (no_regis IN VARCHAR2); procedure jml_riwayat_pendaftaran(pol in VARCHAR2, bln in number, thn in number); PROCEDURE RIWAYAT_PENDAFTARAN_INSERT (I_SYMPTOM IN RIWAYAT_PENDAFTARAN.SYMPTOM%TYPE, I_DIAGNOSA IN RIWAYAT_PENDAFTARAN.DIAGNOSA%TYPE, </pre>	

```

        I_TIPE_RAWAT IN RIWAYAT_PENDAFTARAN.TIPE_RAWAT%TYPE,

        I_TGL_REGISTRASI IN RIWAYAT_PENDAFTARAN.TGL_REGISTRASI%TYPE,

        I_PASIEN_NO_REKAM_MEDIS IN
RIWAYAT_PENDAFTARAN.PASIEN_NO_REKAM_MEDIS%TYPE,

        I_POLIKLINIK_ID_POLIKLINIK IN
RIWAYAT_PENDAFTARAN.POLIKLINIK_ID_POLIKLINIK%TYPE

    );

PROCEDURE TINDAKAN_INSERT (

    I_NAMA_TINDAKAN IN TINDAKAN.NAMA_TINDAKAN%TYPE,

    I_DESKRIPSI IN TINDAKAN.DESKRIPSI%TYPE,

    I_TARIF IN TINDAKAN.TARIF%TYPE,

    I_NO_REGIS IN TINDAKAN.RP_NO_REGISTRASI%TYPE

);

PROCEDURE PEMBAYARAN_INSERT (

    I_RP_NO_REGISTRASI IN PEMBAYARAN.RP_NO_REGISTRASI%TYPE

);

end;
```

Script SQL Package Body

```

create or replace package body pendaftaran_pasien
as

    PROCEDURE total_tarif (no_regis IN VARCHAR2)

    AS

        CURSOR cur_tdk IS

            SELECT nama_tindakan, tarif FROM tindakan;

        jml_tarif NUMBER;

        nama_tdk tindakan.nama_tindakan%TYPE;
```

```

        tarif_tdk tindakan.tarif%TYPE;

BEGIN

    dbms_output.put_line('no registrasi :' || no_regis);
    dbms_output.put_line('tarif' || ' ' || 'tindakan');

    SELECT SUM(tarif)
    INTO jml_tarif
    FROM tindakan
    WHERE tindakan.rp_no_registrasi=no_regis;

    OPEN cur_tdk;

    LOOP

        FETCH cur_tdk INTO nama_tdk, tarif_tdk;

        EXIT WHEN cur_tdk%NOTFOUND;

        dbms_output.put_line(tarif_tdk || ' ' || nama_tdk);

    END LOOP;

    CLOSE cur_tdk;

    dbms_output.put_line('total tarif      :' || jml_tarif);

EXCEPTION

    WHEN OTHERS THEN

        dbms_output.put_line( SQLERRM );

END;

PROCEDURE jml_riwayat_pendaftaran(

    pol in VARCHAR2, bln in number, thn in number

)

IS

    rp_count NUMBER;

BEGIN

```

```

-- get jumlah riwayat pendaftaran based on poliklinik id

SELECT count(*) INTO rp_count

FROM RIWAYAT_PENDAFTARAN

WHERE poliklinik_id_poliklinik = (SELECT id_poliklinik FROM
POLIKLINIK WHERE nama_poliklinik = pol)

AND (SELECT

EXTRACT( MONTH FROM
TO_DATE( tgl_registrasi, 'DD-MM-YYYY' ) ) MONTH FROM DUAL) = bln

AND (SELECT

EXTRACT( YEAR FROM TO_DATE(
tgl_registrasi, 'DD-MM-YYYY' ) ) YEAR FROM DUAL) = thn;

-- print out information

dbms_output.put_line( 'Nama Poliklinik' || ' : ' || pol);

dbms_output.put_line( 'Bulan' || ' : ' || bln);

dbms_output.put_line( 'Tahun' || ' : ' || thn);

dbms_output.put_line( 'Jumlah Riwayat Pendaftaran' || ' : ' ||
rp_count);

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line( SQLERRM );

END;

PROCEDURE RIWAYAT_PENDAFTARAN_INSERT (

I_SYMPTOM IN RIWAYAT_PENDAFTARAN.SYMPTOM%TYPE,

I_DIAGNOSA IN RIWAYAT_PENDAFTARAN.DIAGNOSA%TYPE,

I_TIPE_RAWAT IN RIWAYAT_PENDAFTARAN.TIPE_RAWAT%TYPE,

I_TGL_REGISTRASI IN RIWAYAT_PENDAFTARAN.TGL_REGISTRASI%TYPE,

I_PASIEN_NO_REKAM_MEDIS IN
RIWAYAT_PENDAFTARAN.PASIEN_NO_REKAM_MEDIS%TYPE,

```



```

        I_POLIKLINIK_ID_POLIKLINIK IN
RIWAYAT_PENDAFTARAN.POLIKLINIK_ID_POLIKLINIK%TYPE)

IS

BEGIN

        INSERT INTO
RIWAYAT_PENDAFTARAN("SYMPTOM","DIAGNOSA","TIPE_RAWAT","TGL_REGISTRASI",
"PASIEN_NO_REKAM_MEDIS","POLIKLINIK_ID_POLIKLINIK")

        VALUES(I_SYMPTOM, I_DIAGNOSA, I_TIPE_RAWAT, I_TGL_REGISTRASI,
I_PASIEN_NO_REKAM_MEDIS, I_POLIKLINIK_ID_POLIKLINIK);

        COMMIT;

END;

PROCEDURE TINDAKAN_INSERT (

        I_NAMA_TINDAKAN IN TINDAKAN.NAMA_TINDAKAN%TYPE,

        I_DESKRIPSI IN TINDAKAN.DESKRIPSI%TYPE,

        I_TARIF IN TINDAKAN.TARIF%TYPE,

        I_NO_REGIS IN TINDAKAN.RP_NO_REGISTRASI%TYPE)

IS

BEGIN

        INSERT INTO TINDAKAN("NAMA_TINDAKAN","DESKRIPSI",
"TARIF","RP_NO_REGISTRASI")

        VALUES( I_NAMA_TINDAKAN, I_DESKRIPSI, I_TARIF, I_NO_REGIS);

        COMMIT;

END;

PROCEDURE PEMBAYARAN_INSERT (

        I_RP_NO_REGISTRASI IN PEMBAYARAN.RP_NO_REGISTRASI%TYPE)

IS

BEGIN

        INSERT INTO PEMBAYARAN("RP_NO_REGISTRASI")

        VALUES(I_RP_NO_REGISTRASI);

        COMMIT;

```

END;

end;

Screenshot Hasil

- Memanggil procedure

SQL :

```
execute pkg_func_proc.  
jml_rwayat_pendaftaran('Anak','November','2018')
```

Output :

```
Connecting to the database ADM-RS.  
Nama Poliklinik : Anak  
Bulan : November  
Tahun : 2018  
Jumlah Riwayat Pendaftaran : 2  
Process exited.  
Disconnecting from the database ADM-RS.
```

- Memanggil procedure

SQL :

```
execute pkg_func_proc. total_tarif('12200001')
```

Output :

```
Connecting to the database ADM-RS.  
no registrasi :12200001  
tarif tindakan  
170000 Konnsultasi  
523000 USG  
385000 cek darah  
9000000 Apendektomi laproskopi  
total tarif :10078000  
Process exited.  
Disconnecting from the database ADM-RS.
```

6. KESIMPULAN

Berdasarkan kebutuhan kebutuhan yang kami deskripsikan pada ruang lingkup, pengerjaan Tugas Besar Administrasi Rumah Sakit ini dapat dikatakan telah selesai dikerjakan, namun tim kami sendiri merasa perlunya revisi ketika sudah di review oleh ibu bapak dosen mata kuliah sistem basis data untuk mengetahui bagian mana saja yang perlu diperbaiki.

Kendala yang kami hadapi selama mengerjakan tugas besar Administrasi Rumah Sakit ini diantaranya yaitu saat menentukan entity apa saja yang diperlukan untuk sistem administrasi rumah sakit ini serta atribut dari setiap entitasnya, dimana kami beberapa kali menggantinya agar lebih tepat dan efisien. Selain itu terdapat kendala di bagian trigger tabel pasien, dimana sebelumnya rencana kami adalah nilai dari atribut umur diisi dan di update secara otomatis oleh trigger berdasarkan tanggal lahir pasien, namun ketika trigger telah dibuat ternyata insert tidak dapat dilakukan karena berdasarkan eksplorasi yang dilakukan melalui berbagai website, trigger tidak boleh melakukan operasi terhadap tabel yang sama dengan tabel yang menyebabkan trigger terpanggil. Namun akhirnya dapat terselesaikan dengan memindahkan attribute usia ke entity riwayat pendaftaran.

Selain itu untuk sekarang kelompok kami masih memiliki kendala di bagian trigger untuk sequence, dimana input kedua dari data, tidak sesuai dengan yang seharusnya, yaitu increament sebanyak 20. misal ID data 1 =DKTR0001, seharusnya data kedua adalah DKTR0020 namun output yang diperoleh ialah DKTR0020.

7. LESSON LEARNED

Lesson learned kelompok :

Dengan tugas besar mata kuliah sistem basis data ini, kelompok kami mendapatkan pelajaran mengenai bagaimana kerjasama tim yang baik terutama ketika masing-masing dari kami terpisahkan oleh jarak karena sedang berlangsungnya pandemi. Kami mempelajari bahwa kerja sama tim yang baik dipengaruhi oleh komunikasi yang baik sehingga tim kami memiliki pemahaman terhadap ide tugas besar yang selaras dan sebisa mungkin meminimalisir adanya salah tangkap ketika pengerjaan. Selain itu tentu saja dengan adanya tugas besar ini membuat masing-masing dari kami menjadi lebih memahami tentang hal-hal yang berkaitan dengan database. Selain itu, karena studi kasus yang kami angkat adalah administrasi rumah sakit, kami juga mendapatkan pengetahuan mengenai bagaimana gambaran sistem administrasi rumah sakit serta kebutuhan-kebutuhannya.

Lesson learned masing-masing anggota

Dzakira Fabillah : Mendapatkan banyak pelajaran mengenai database, karena sebelum merancang dan mengimplementasikan tugas ini, saya yang sebelumnya tidak paham menjadi lebih paham karena mengeksplorasi lebih dalam untuk memenuhi kebutuhan sistem yang akan dibuat, misalnya ketika menentukan relationship antar entity, menentukan tipe data, membuat function/procedure/trigger/package, dan masih banyak lagi . Saya juga merasa sangat terbantu dengan penjelasan dari ibu bapa dosen ketika me review tugas besar kelompok kami maupun kelompok lain ketika jam mata kuliah. Ketika ada kendala dalam pengerjaan, saya merasa sangat terbantu dengan berdiskusi bersama anggota kelompok atau dengan bertanya kepada kelompok lain.

Putri Syalwa Abdullah : Dari pengerjaan tugas ini, saya menjadi lebih mengetahui terkait database. Seperti menentukan entity, atribut, beserta relationship nya, juga membuat procedure, trigger, DML, dan lain-lain. Untuk kendala yang saya hadapi saat pengerjaan, saya bertanya kepada teman-teman, mengeksplor dari internet, serta penjelasan dari bapak dan ibu yang sangat membantu. Komunikasi yang berjalan dengan baik walau di masa daring seperti ini membantu saya dan teman sekelompok dalam menyelesaikan masalah dan menyelesaikan tugas ini.

Rizka Auliarahmi : Saya menjadi lebih mengerti tentang database dan implementasinya dari pengerjaan tugas besar ini. penjelasan materi dari bapak dan ibu dosen juga bantuan dari rekan sekelompok membuat saya dapat mengatasi masalah yang terjadi pada saat pengerjaan seperti pembuatan DML, pembuatan function, procedure, trigger, dan package. Pemahaman saya khususnya terhadap relationship antar entity menjadi lebih baik saat langsung mengimplementasikannya.

Dokumentasi Masalah dan Solusi

Anda dianjurkan untuk membuat log book sebagai keterangan tambahan mengenai solusi dari masalah yang Anda hadapi saat proses pengerjaan Tugas Besar untuk diambil hikmah ketika menghadapi masalah yang sama di waktu yang akan datang.