

**PENGEMBANGAN APLIKASI CUSTOMER
RELATIONSHIP MANAGEMENT STUDI KASUS
*MEDICAL REPRESENTATIVE***

*Application Development Of Customer Relationship Management Case Study
Medical Representative*

Laporan ini disusun untuk memenuhi salah satu syarat
menyelesaikan Pendidikan
PROGRAM STUDI DIPLOMA III TEKNIK INFORMATIKA
Di Jurusan Teknik Komputer dan Informatika

Oleh
KoTA 104
Arif Khairuddin 141511004
Helmi Fauzi 141511011
Inda Nabila Maulida 141511012



**POLITEKNIK NEGERI BANDUNG
2017**



Nama : Arif Khairuddin

NIM : 141511004

- Tempat, Tanggal Lahir : Bandung, 09 Maret 1995.
- SD Lulus Tahun : 2007 – SD Negeri Babakan Priangan 3 Bandung
- SLTP Lulus Tahun : 2010 – SMP Negeri 3 Bandung
- SLTA Lulus Tahun : 2014 – SMK Negeri 4 Bandung
- Prestasi yang pernah dicapai :-



Nama : Helmi Fauzi

NIM : 141511011

Tempat, Tanggal Lahir : Bandung, 19 Juli 1996.

SD Lulus Tahun : 2008 – SD Negeri Majalaya 06

SLTP Lulus Tahun : 2011 – SMP Negeri 01 Majalaya

SLTA Lulus Tahun : 2014 – SMA Negeri 01 Majalaya

Prestasi yang pernah dicapai :-



Nama : Inda Nabila Maulida

NIM : 141511012

Tempat, Tanggal Lahir : Bandung, 08 Agustus 1996.

SD Lulus Tahun : 2008 – SD Negeri Buah Batu 05 Bandung

SLTP Lulus Tahun : 2011 – SMP Negeri 28 Bandung

SLTA Lulus Tahun : 2014 – SMA BPI 1 Bandung

Prestasi yang pernah dicapai :-

ABSTRAK

Pengembangan aplikasi *customer relationship management* (CRM) studi kasus *medical representative* merupakan aplikasi yang dibuat untuk membantu pekerjaan *medical representative* (*medical rep*) dan supervisor *medical rep*. *Medical rep* adalah orang yang bekerja di bidang pemasaran pada industri farmasi, untuk mempromosikan produk perusahaan kepada dokter sebagai penulis resep atau yang mewakili konsumen akhir (pasien) dalam menentukan obat yang akan dikonsumsinya. Pekerjaan *medical rep* yang dibantu oleh aplikasi ini meliputi pencatatan data dokter, mengklasifikasi besar potensi peresepan dokter, mengelola rencana kunjungan, pencatatan kunjungan dan membuat laporan. Aplikasi ini juga membantu pekerjaan supervisor *medical rep* yang merupakan atasan *medical rep* dalam hal mengelola tugas dan target *medical rep*, memonitor laporan serta mengevaluasi pencapaian target *medical rep*. Pembuatan aplikasi dilakukan karena saat ini aplikasi yang menerapkan CRM untuk industri farmasi belum dikembangkan secara luas di Indonesia. Aplikasi ini dikembangkan untuk *medical rep* pada perangkat *mobile* berbasis Android dengan minimal versi sistem operasi 5.0 (Lollipop), dan *website* untuk supervisor *medical rep*. Penggunaan Android untuk *medical rep* ini dilakukan karena menyesuaikan dengan pekerjaan *medical rep* yang banyak dilakukan di lapangan atau di luar kantor, sehingga mudah digunakan dan mendukung mobilitas *medical rep*, sedangkan *website* untuk supervisor *medical rep* karena pekerjaannya dominan di kantor dan menggunakan komputer. Aplikasi ini juga mengintegrasikan *medical rep* dengan supervisor *medical rep*, sehingga membutuhkan *internet* pada penggunaannya. Dengan adanya aplikasi ini, pengguna aplikasi yakni *medical rep* dan supervisor *medical rep* dapat terbantu dalam mengerjakan tugas pemasarannya.

Kata Kunci: *customer relationship management, medical representative, pemasaran obat, industri farmasi*

ABSTRACT

Application development of customer relationship management (CRM) case study medical representative is an application created to assist the work of medical representative (medical rep) and supervisor of medical rep. Medical rep is a person who works in marketing in the pharmaceutical industry, to promote the company's product to physicians as a prescription writer or who represent the final consumer (patient) in determining the medicine to be consumed. Medical rep jobs assisted by this application include physician's data recording, classifying potential of physician's prescribing, managing visit plan, recording visits and reporting. This application also assists the work of supervisor of medical rep who is medical rep's superior in terms of managing task and target of medical rep, monitoring reports and evaluating the target achievement of medical rep. The making of the application is done because the current application that implements CRM for the pharmaceutical industry has not been developed widely in Indonesia. This application is developed for medical rep on Android-based mobile devices with a minimum version 5.0 operating system (Lollipop), and a website for supervisor of medical rep. The use of Android for medical rep is done because it adapts to the medical rep work mostly done in the field or outside the office, so it is easy to use and support medical rep mobility, while the website for supervisor of medical rep because the work is dominant in the office and using a computer. This application also integrates medical rep with the supervisor of medical rep, thus requiring the internet on its use. With this application, users of the application such as medical rep and supervisor of medical rep can be helpful in doing their marketing tasks.

Keywords: *customer relationship management, medical representative, marketing of the medicine, pharmaceutical industry*

KATA PENGANTAR

Alhamdulillahirabbil'alamiiin, puji dan syukur tim penulis panjatkan kehadiran Allah SWT atas segala karunia dan rahmat-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan topik “Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical Representative*” sesuai dengan rencana.

Penulis sangat menyadari bahwa dalam penyusunan laporan Tugas Akhir ini tidak lepas dari bantuan berbagai pihak. Untuk itu kami mengucapkan terima kasih kepada:

1. Orang tua yang telah memberikan dukungan moril maupun materil yang luar biasa banyak kepada kami hingga penyelesaian Tugas Akhir;
2. Bapak Bambang Wisnuadhi, S.Si, M.T., selaku Dosen Pembimbing 1 dan Ibu Idah Suhartini, S.Kom., MMSI., selaku Dosen Pembimbing 2 yang telah banyak memberikan bimbingan, dukungan dan masukan sejak awal sehingga tim penulis dapat menyelesaikan Tugas Akhir;
3. Ibu Nurjannah Syakrani, Dra., M.T., Dr., selaku Pengaji 1 dan Ibu Ade Hodijah selaku Pengaji 2 yang telah mengevaluasi, memberi kritik dan saran yang membangun kepada tim penulis yang dijadikan sebagai acuan dalam menyelesaikan Tugas Akhir;
4. Ibu Santi Sundari, S.Si., M.T., Ibu Nurjannah Syakrani, Dra., M.T., Dr., Ibu Transmissia Semiawan, BSCS., M.IT., Bapak Urip Teguh Setijohatmo, BSCS., M.Kom., Bapak Suprihanto, BSEE., M.Sc., Bapak Jonner Hutahaean, BSET., M.Info.Sys., selaku koordinator Tugas Akhir tahun 2017 yang telah memberikan arahan-arahan dan mengatur jadwal demi kelancaran pelaksanaan Tugas Akhir;
5. Seluruh Dosen Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung, yang telah membantu dan memberikan ilmu kepada tim penulis, sehingga kami dapat menyusun Laporan Tugas Akhir ini;

6. Bapak Yuda, Bapak Edo, Bapak Bambang Pamungkas, dan Mbak Astrid selaku narasumber dalam wawancara mengenai Tugas Akhir tim penulis, yang tanpa bantuan mereka Tugas Akhir kami tidak dapat berjalan sesuai rencana;
7. Kakak, Adik, dan Saudara tim penulis yang senantiasa menyemangati dan memberikan dukungannya dalam penggerjaan Tugas Akhir;
8. Teman-teman seperjuangan, khususnya JTK 2014 yang memberikan dukungan dan bantuan dalam penggerjaan Tugas Akhir;
9. Semua pihak yang telah membantu yang tidak dapat kami sebut satu persatu, terima kasih atas doa yang tulus dan dukungannya.

Semoga Allah SWT berkenan membalas segala kebaikan pihak-pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini. Penulis berharap laporan ini dapat bermanfaat bagi semua pihak. Akhir kata, semoga Allah SWT selalu memberikan rahmat dan hidayah-Nya kepada kita semua.

Bandung, Juli 2017

KoTA 104

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	xii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Ruang Lingkup.....	4
1.5 Batasan Masalah.....	5
1.6 Metodologi Penyelesaian Tugas Akhir	6
1.6.1 Pendefinisian Masalah.....	6
1.6.2 Studi Literatur.....	6
1.6.3 Pengembangan Aplikasi	6
1.6.4 Penarikan Kesimpulan.....	10
BAB II.....	11
TINJAUAN PUSTAKA	11
2.1 Teori Pendukung	11
2.1.1 Customer Relationship Management	11
2.1.2 Pelaku Pemasaran Perusahaan Farmasi	15

2.1.3 Penerapan CRM	17
2.2 Teknologi dan Tools Pendukung	29
2.2.1 Teknologi dan Tools Pendukung dalam Pengembangan Aplikasi	29
BAB III	33
ANALISIS	33
3.1 Analisis Sistem yang Sedang Berjalan (As-Is) di Perusahaan Farmasi	33
3.1.1 Analisis Stakeholder dan Perannya	33
3.1.2 Analisis Bisnis Proses	35
3.1.3 Business Rules.....	50
3.1.4 Domain Model.....	51
3.2 Analisis Requirement.....	53
3.3 Analisis Aplikasi Sejenis	63
3.3.1 Insightly CRM	63
3.3.2 Base CRM	74
3.3.3 MediVisits	84
3.4 Evaluasi Hasil Analisis Aplikasi Sejenis	91
3.5 Usulan Solusi Penyelesaian Masalah	94
3.5.1 Stakeholder dan Perannya	96
3.5.2 Bisnis Proses.....	96
3.5.3 Business Rules.....	98
3.5.4 Domain Model.....	99
3.5.5 System Sequence diagram (SSD) dan Operation Contract (OC)	99
BAB IV	129
PERANCANGAN	129
4.1 Perancangan Mobile Application.....	129
4.1.1 Perancangan Arsitektur	129
4.1.2 Perancangan Class	130
4.1.3 Perancangan Interaksi Antarobjek.....	136
4.1.4 Perancangan User Interface	201
4.2 Perancangan Web Frontend	220
4.2.1 Perancangan Arsitektur	220
4.2.2 Perancangan Interaksi Antarobjek.....	221

4.2.3 Perancangan User Interface	273
4.3 Perancangan Backend	293
4.3.1 Perancangan Arsitektur	293
4.3.2 Perancangan Class	295
4.4 Perancangan Database.....	312
BAB V.....	315
IMPLEMENTASI.....	315
5.1 Model Deployment.....	315
5.2 Rancangan yang Diimplementasi.....	316
5.3 Hasil Implementasi.....	319
BAB VI	339
PENUTUP.....	339
6.1 Kesimpulan	339
6.2 Saran.....	340
DAFTAR PUSTAKA	341
LAMPIRAN A.....	345

DAFTAR GAMBAR

Gambar 1 Bisnis Proses Pengelolaan Rencana Kunjungan	36
Gambar 2 Bisnis Proses Mencatat Data Dokter Baru	37
Gambar 3 Bisnis Proses Pencatatan Potensi Peresepan Dokter	39
Gambar 4 Bisnis Proses Detailing dan Follow Up Dokter	42
Gambar 5 Bisnis Proses Rekap Hasil Kunjungan	45
Gambar 6 Bisnis Proses Rekap Hasil Transaksi	47
Gambar 7 Bisnis Proses Monitoring Pencapaian Target Medical Representative	49
Gambar 8 Domain Model Sistem yang Berjalan	53
Gambar 9 Strategic Dependency System As-Is	56
Gambar 10 Strategic Performance Goals Of Stakeholder.....	57
Gambar 11 Strategic Dependency System to-be.....	58
Gambar 12 Initial Specification for CRM Software	59
Gambar 13 Tampilan Fitur Task pada Aplikasi Insightly CRM.....	64
Gambar 14 Tampilan Fitur Contact pada Aplikasi Insightly CRM	66
Gambar 15 Tampilan Fitur Organization pada Aplikasi Insightly CRM.....	67
Gambar 16 Tampilan Fitur Opportunities pada Aplikasi Insightly CRM	69
Gambar 17 Tampilan Fitur Emails pada Aplikasi Insightly CRM	70
Gambar 18 Tampilan Fitur Calendar pada Aplikasi Insightly CRM	71
Gambar 19 Tampilan Fitur Leads pada Aplikasi Base CRM	75
Gambar 20 Tampilan Fitur Contact pada Aplikasi Base CRM	77
Gambar 21 Tampilan Fitur Deals pada Aplikasi Base CRM.....	78
Gambar 22 Tampilan Fitur Tasks pada Aplikasi Base CRM.....	79
Gambar 23 Tampilan Fitur Calendar pada Aplikasi Base CRM	80
Gambar 24 Tampilan Fitur Communication pada Aplikasi Base CRM	81
Gambar 25 Tampilan Fitur Report pada Aplikasi Base CRM	81
Gambar 26 Tampilan Fitur Dashboard pada Aplikasi Base CRM.....	82
Gambar 27 Tampilan Fitur Salesvisit pada Aplikasi MediVisits	85
Gambar 28 Tampilan Fitur Doctor pada Aplikasi MediVisits.....	86
Gambar 29 Tampilan Fitur Product pada Aplikasi MediVisits	87

Gambar 30 Tampilan Fitur Account pada Aplikasi MediVisits	88
Gambar 31 Tampilan Hasil Export Laporan Kunjungan Aplikasi MediVisits.....	89
Gambar 32 Use Case Diagram CRM Application	95
Gambar 33 Domain Model dari Sistem yang Diusulkan	99
Gambar 34 SSD Manage Task And Target Medical Representative.....	100
Gambar 35 SSD Show Report.....	102
Gambar 36 SSD Show Medical rep Target Achievement	104
Gambar 37 SSD Manage Visit Plan.....	106
Gambar 38 SSD Manage Doctor Data	109
Gambar 39 SSD Check Potential Doctor	111
Gambar 40 - SSD Classify Doctor	113
Gambar 41 SSD Add Daily Visit	114
Gambar 42 SSD Record Transaction	116
Gambar 43 SSD Show Product	117
Gambar 44 SSD Generate Report	118
Gambar 45 SSD Manage Product	120
Gambar 46 SSD Manage Medical Representative Account	123
Gambar 47 SSD Manage Supervisor Account.....	126
Gambar 48 Arsitektur CRM Mobile Application/Android	129
Gambar 49 Class Diagram Frontend Android	130
Gambar 50 SD Manage Visit Plan	137
Gambar 51 Referensi Visit Plan By Medrep.....	139
Gambar 52 SD Referensi Add Visit Plan.....	141
Gambar 53 SD Referensi Get Doctor By Medrep And Practice Day	143
Gambar 54 SD Referensi Save Visit Plan.....	145
Gambar 55 SD Referensi Update Visit Plan	147
Gambar 56 Referensi Update Visit Plan Data	149
Gambar 57 SD Manage Doctor Data	150
Gambar 58 SD Referensi Manage Lead Doctor	151
Gambar 59 SD Referensi Get Doctor By Medrep and Status.....	153
Gambar 60 SD Referensi Add Lead Doctor Data.....	155
Gambar 61 SD Referensi Save New Lead Doctor	157

Gambar 62 SD Referensi Update Doctor Data	160
Gambar 63 SD Referensi Update Doctor.....	162
Gambar 64 Referensi Delete Doctor Data	164
Gambar 65 Referensi Show Doctor's Detail	166
Gambar 66 Referensi Manage Lead Qualified Doctor	168
Gambar 67 SD Classify Doctor	170
Gambar 68 Referensi Get Classify Doctor.....	171
Gambar 69 Referensi Check Potential Doctor	174
Gambar 70 SD Referensi Get Product Task Medrep	176
Gambar 71 SD Referensi Save Potential Prescription	178
Gambar 72 SD Add Daily Visit	180
Gambar 73 SD Referensi Input Visit Result.....	181
Gambar 74 SD Referensi Get Doctor For Visit By Medrep	182
Gambar 75 SD Referensi Save Daily Visit	184
Gambar 76 SD Referensi Record Transaction	185
Gambar 77 SD Referensi Save Transaction.....	187
Gambar 78 SD Referensi Get Product Task Medrep	189
Gambar 79 SD Generate Report	191
Gambar 80 SD Referensi Save Generate Report	193
Gambar 81 SD Show Product	197
Gambar 82 SD Referensi Show Detail Product	199
Gambar 83 SD Referensi Get Product Task Medrep	200
Gambar 84 Arsitektur Frontend Website	220
Gambar 85 SD Manage Medical Representative's Task and Target.....	222
Gambar 86 SD Referensi Get List Medrep For Task and Target	225
Gambar 87 SD Referensi Add Task and Target.....	227
Gambar 88 SD Referensi Send Notification	229
Gambar 89 SD Referensi Get Detail Medrep Task and Target	230
Gambar 90 SD Show Report.....	232
Gambar 91 SD Referensi Get List Medrep For Show Report	234
Gambar 92 SD Referensi Get All Report By Type.....	235
Gambar 93 SD Referensi Get Report By Id.....	238

Gambar 94 SD Show Medical Representative's Target Achievement.....	241
Gambar 95 SD Referensi Calculate Target Achievement.....	242
Gambar 96 SD Manage Product	245
Gambar 97 SD Referensi Get List Product	248
Gambar 98 SD Referensi Add Product Data.....	249
Gambar 99 SD Referensi Update Product	251
Gambar 100 SD Referensi Delete Product.....	252
Gambar 101 SD Manage Medical Representative Account	254
Gambar 102 SD Referensi Get List Medical Representative Account	257
Gambar 103 SD Referensi Add Medical Representative Account Data	259
Gambar 104 SD Referensi Update Medical Representative Account	260
Gambar 105 SD Referensi Delete Medical Representative Account.....	262
Gambar 106 SD Manage Supervisor Account	264
Gambar 107 SD Referensi Get List Supervisor Account.....	267
Gambar 108 SD Referensi Add Supervisor Account Data	268
Gambar 109 SD Referensi Update Supervisor Account	270
Gambar 110 SD Referensi Delete Supervisor Account	271
Gambar 111 Arsitektur Backend.....	294
Gambar 112 Class Diagram Backend	295
Gambar 113 Conceptual Data Model Pengembangan Aplikasi CRM.....	313
Gambar 114 Physical Data Model Pengembangan Apikasi CRM.....	314
Gambar 115 Model Deployment Pengembangan Aplikasi CRM.....	315

DAFTAR TABEL

Tabel 1 Penerapan CRM Secara Umum	20
Tabel 2 Penerapan CRM Studi Kasus Medical Representative	21
Tabel 3 Nama-Nama Kelas, Persentase Akumulasi dan Persentase Jumlah Item Keseluruhan.....	26
Tabel 4 Nama Dokter Beserta Besar Potensi Persentasenya	26
Tabel 5 Nama Dokter Beserta Potensi Peresepan yang Sudah Terurut	27
Tabel 6 Nama Dokter Beserta Pengklasifikasiannya	27
Tabel 7 Stakeholder dan Perannya.....	34
Tabel 8 Penjelasan Bisnis Proses Pengelolaan Rencana Kunjungan	36
Tabel 9 Penjelasan Bisnis Proses Mencatat Data Dokter Baru.....	38
Tabel 10 Penjelasan Bisnis Proses Pencatatan Potensi Peresepan Dokter.....	40
Tabel 11 Penjelasan Bisnis Proses Detailing dan Follow Up Dokter	43
Tabel 12 Penjelasan Bisnis Proses Rekap Hasil Kunjungan.....	45
Tabel 13 Penjelasan Bisnis Proses Rekap Hasil Transaksi	47
Tabel 14 Penjelasan Bisnis Proses Monitoring Pencapaian Target Medical Representative	49
Tabel 15 Business Rules Sistem As-Is.....	50
Tabel 16 Daftar Goals untuk Setiap Stakeholder.....	54
Tabel 17 Strategic performance goals of the stakeholder	55
Tabel 18 Strategic dependency system To-Be	55
Tabel 19 Functional Requirements Aplikasi CRM.....	60
Tabel 20 Hubungan Functional Requirement, Spesification, dan Penerapan CRM	61
Tabel 21 Tabel Hasil Analisis Aplikasi Insightly CRM	72
Tabel 22 Kekurangan dan Kelebihan Aplikasi Base CRM.....	82
Tabel 23 Kekurangan dan Kelebihan Aplikasi MediVisits.....	89
Tabel 24 Evaluasi Requirement dan Aplikasi Sejenis	91
Tabel 25 Stakeholder dan Perannya pada Aplikasi CRM.....	96

Tabel 26 Keterhubungan Bisnis Proses, Use Case dan Functional Requirement .	97
Tabel 27 Business Rules pada Aplikasi CRM	98
Tabel 28 Contract CO1 requestManageTaskAndTargetMedicalRep	101
Tabel 29 Contract CO2 chooseMedicalRep.....	101
Tabel 30 Contract CO3 enterTaskAndTargetMedicalRep.....	101
Tabel 31 Contract CO1 requestShowReport.....	103
Tabel 32 Contract CO2 chooseReportByType	103
Tabel 33 Contract CO3 chooseReportById	103
Tabel 34 Contract CO4 chooseReportById	104
Tabel 35 Contract CO1 requestShowMedicalRepresentativeTargetAchievement	105
Tabel 36 Contract CO2 choosePeriod.....	105
Tabel 37 Contract CO1 requestManageVisitPlan	107
Tabel 38 Contract CO2 chooseDate.....	107
Tabel 39 Contract CO3 enterVisitPlanData.....	107
Tabel 40 Contract CO4 updateVisitPlanData	108
Tabel 41 Contract CO1 requestManageDoctor.....	110
Tabel 42 Contract CO2 requestFormAddLeadDoctor	110
Tabel 43 Contract CO3 enterDataLeadDoctor.....	110
Tabel 44 Contract CO4 chooseDoctor	110
Tabel 45 Contract CO5 updateDoctorData	111
Tabel 46 Contract CO1 requestCheckPotentialDoctor	112
Tabel 47 Contract CO3 enterDataPotential.....	112
Tabel 48 Contract CO1 requestClassificationDoctor.....	113
Tabel 49 Contract CO2 classifyDoctor	113
Tabel 50 Contract CO1 requestAddDailyVisitResult	115
Tabel 51 Contract CO2 chooseDoctor	115
Tabel 52 Contract CO3 enterDailyVisitResult	115
Tabel 53 Contract CO1 requestRecordTransaction	116
Tabel 54 Contract CO2 enterTransactionData	117
Tabel 55 Contract CO1 requestListProduct	118
Tabel 56 Contract CO2 chooseProduct.....	118

Tabel 57 Contract CO1 requestGenerateReport	119
Tabel 58 Contract CO2 enterGenerateReport	119
Tabel 59 Contract CO3 collectResultVisitAndTransactionData	119
Tabel 60 Contract CO1 requestManageProduct	121
Tabel 61 Contract CO2 requestFormAddProduct.....	121
Tabel 62 Contract CO3 enterProductData	121
Tabel 63 Contract CO4 chooseProduct.....	122
Tabel 64 Contract CO5 updateProductData.....	122
Tabel 65 Contract CO1 requestManageMedicalRepresentativeAccount	124
Tabel 66 Contract CO2 requestFormAddMedicalRepresentativeAccount.....	124
Tabel 67 Contract CO3 enterMedicalRepresentativeAccontData	124
Tabel 68 Contract CO4 chooseMedicalRepresentativeAccount.....	124
Tabel 69 Contract CO5 updateMedicalRepresentativeAccountData.....	125
Tabel 70 Contract CO1 requestManageSupervisorAccount	127
Tabel 71 Contract CO2 requestFormAddSupervisorAccount	127
Tabel 72 Contract CO3 enterSupervisorAcountData.....	127
Tabel 73 Contract CO4 chooseSupervisorAccount	127
Tabel 74 Contract CO5 updateSupervisorAccountData	128
Tabel 75 Penjelasan Arsitektur CRM Mobile Application/Android	130
Tabel 76 Keterangan SD_ManageVisitPlan	138
Tabel 77 Keterangan SD_refGetVisitPlanByMedrep	139
Tabel 78 Keterangan SD_refAddVisitPlan	142
Tabel 79 Keterangan SD_refGetDoctorByMedrepAndPracticeDay	143
Tabel 80 Keterangan SD_refSaveVisitPlan	145
Tabel 81 Keterangan SD_refUpdateVisitPlan	148
Tabel 82 Keterangan SD_refUpdateVisitPlanData.....	149
Tabel 83 Keterangan SD_refManageLeadDoctor.....	152
Tabel 84 Keterangan SD_refGetDoctorByMedrepAndStatus	153
Tabel 85 Keterangan SD_refLeadDoctorAdd.....	156
Tabel 86 Keterangan SD_refSaveNewLeadDoctor	157
Tabel 87 Keterangan SD_refUpdateDoctorData	161
Tabel 88 Keterangan SD_refUpdateDoctor	162

Tabel 89 Keterangan SD_refDeleteDoctorData.....	165
Tabel 90 Keterangan SD_refShowDoctorDetail.....	167
Tabel 91 Keterangan SD_refManageLeadQualifiedDoctor.....	169
Tabel 92 Keterangan SD_ClassifyDoctor	170
Tabel 93 Keterangan SD_refGetClassifyDoctor.....	172
Tabel 94 Keterangan SD_refCheckPotentialDoctor	175
Tabel 95 Keterangan SD_refGetProductTaskMedrep	177
Tabel 96 Keterangan SD_refSavePotentialPrescription	178
Tabel 97 Keterangan SD_AddDailyVisit.....	180
Tabel 98 Keterangan SD_refInputVisitResult	182
Tabel 99 Keterangan SD_refGetDoctorForVisitByMedrep	183
Tabel 100 Keterangan SD_refSaveDailyVisit	184
Tabel 101 Keterangan SD_refRecordTransaction	186
Tabel 102 Keterangan SD_refSaveTransaction	188
Tabel 103 Keterangan SD_refGetProductTaskMedrep	189
Tabel 104 Keterangan SD_GenerateReport.....	192
Tabel 105 Keterangan SD_refSaveGenerateReport.....	194
Tabel 106 Keterangan SD_ShowProduct.....	198
Tabel 107 Keterangan SD_refShowDetailProduct	199
Tabel 108 Keterangan SD_refGetProductTaskMedrep	201
Tabel 109 Tampilan Daftar Visit Plan	202
Tabel 110 Tampilan Add Visit Plan.....	203
Tabel 111 Tampilan Update Visit Plan.....	204
Tabel 112 Tampilan Daftar Data Dokter Lead dan Dokter Lead Qualified	206
Tabel 113 Tampilan Form add New Lead Doctor	208
Tabel 114 Tampilan Form Update Doctor Data	209
Tabel 115 Tampilan Detail Doctor Data.....	210
Tabel 116 Tampilan Form Check Potential Doctor	211
Tabel 117 Tampilan Classify Doctor	212
Tabel 118 Tampilan Pilihan Dokter yang Akan Dikunjungi	214
Tabel 119 Tampilan Add Daily Visit.....	215
Tabel 120 Tampilan Record Transaction	216

Tabel 121 Tampilan Generate Report	217
Tabel 122 Tampilan List Product.....	218
Tabel 123 Tampilan Detail Product	219
Tabel 124 Penjelasan Arsitektur Frontend Website.....	220
Tabel 125 Keterangan SD_ManageTaskAndTargetMedrep.....	223
Tabel 126 Keterangan SD_refGetListMedrepForTaskAndTarget.....	226
Tabel 127 Keterangan SD_refAddTaskAndTarget.....	227
Tabel 128 Keterangan SD_refGetDetailMedrepTaskAndTarget.....	230
Tabel 129 Keterangan SD_ShowReport	232
Tabel 130 Keterangan SD_refGetListMedrepForShowReport.....	234
Tabel 131 Keterangan SD_refGetAllReportByType	236
Tabel 132 Keterangan SD_refGetReportById	239
Tabel 133 Keterangan SD_ShowMedicalRepresentativeTargetAchievement ...	241
Tabel 134 Keterangan SD_refCalculateTargetAchievement.....	243
Tabel 135 Keterangan SD_ManageProduct.....	246
Tabel 136 Keterangan SD_refGetListProduct	248
Tabel 137 Keterangan SD_refAddProductData	249
Tabel 138 Keterangan SD_refUpdateProduct.....	251
Tabel 139 Keterangan SD_refDeleteProduct.....	253
Tabel 140 Keterangan SD_ManageMedicalRepresentativeAccount.....	255
Tabel 141 Keterangan SD_refGetListMedicalRepresentativeAccount	258
Tabel 142 Keterangan SD_refAddMedicalRepresentativeAccountData.....	259
Tabel 143 Keterangan SD_refUpdateMedicalRepresentativeAccount.....	260
Tabel 144 Keterangan SD_refDeleteMedicalRepresentativeAccount.....	262
Tabel 145 Keterangan SD_ManageSupervisorAccount	265
Tabel 146 Keterangan SD_refGetListSupervisorAccount.....	267
Tabel 147 Keterangan SD_refAddSupervisorAccountData	268
Tabel 148 Keterangan SD_refUpdateSupervisorAccount	270
Tabel 149 Keterangan SD_refDeleteSupervisorAccount	272
Tabel 150 Tampilan Halaman Awal Task and Target Medical Rep.....	273
Tabel 151 Tampilan Form Add Medical Representative's Task and Target.....	274
Tabel 152 Tampilan Detail Medical Representative's Task and Target.....	276

Tabel 153 Tampilan Halaman Awal Show Report	277
Tabel 154 Tampilan list Laporan Kunjungan (Reports Of Visit)	278
Tabel 155 Tampilan Detail Reports Of Visit	278
Tabel 156 Tampilan List Laporan Transaksi (Reports Of Transaction)	280
Tabel 157 Tampilan Detail Reports Of Transaction	281
Tabel 158 Tampilan Daftar Hasil Perhitungan Pencapaian Target Medical Rep	282
Tabel 159 Tampilan Halaman Awal Manage Product.....	283
Tabel 160 Tampilan Halaman Add New Product.....	284
Tabel 161 Tampilan Halaman Edit Product Data	285
Tabel 162 Tampilan Halaman Delete Product Data	286
Tabel 163 Tampilan Daftar Data Akun Medical Representative.....	287
Tabel 164 Tampilan Form Add Medical Representative Account	287
Tabel 165 Tampilan Form Edit Medical Representative Account.....	288
Tabel 166 Tampilan Supervisor Dashboard.....	289
Tabel 167 Tampilan Daftar Data Akun Supervisor	290
Tabel 168 Tampilan Form Add Supervisor Account.....	291
Tabel 169 Tampilan Form Edit Supervisor Account	292
Tabel 170 Penjelasan Arsitektur Backend	294
Tabel 171 Penjelasan Node.....	316
Tabel 172 Penjelasan Komponen.....	316
Tabel 173 Rancangan Frontend yang Diimplementasi	317

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perusahaan adalah suatu bentuk organisasi yang didirikan untuk memproduksi barang atau jasa, serta memiliki tujuan tertentu yang ingin dicapai [1]. Tujuan-tujuan tersebut diantaranya ingin memperoleh keuntungan yang sebesar-besarnya, mempertahankan keberlangsungan usahanya, serta meningkatkan kesejahteraan karyawannya. Salah satu perusahaan di bidang produksi barang yaitu perusahaan farmasi yang memproduksi obat-obatan [2]. Ketatnya persaingan antarperusahaan dalam era ekonomi global memberikan tantangan utama bagi perusahaan dalam menjaga dan mengelola hubungan dengan pelanggan agar mampu memenangkan persaingan, tak terkecuali bagi perusahaan farmasi [3]. Salah satu strategi yang dapat dilakukan oleh perusahaan adalah dengan menerapkan strategi manajemen hubungan dengan pelanggan atau *customer relationship management* (CRM).

CRM adalah strategi inti dalam bisnis yang mengintegrasikan proses-proses dan fungsi-fungsi internal dengan semua jaringan eksternal untuk menciptakan serta mewujudkan nilai bagi para pelanggan secara menguntungkan. CRM didukung dengan data pelangan yang berkualitas dan teknologi informasi [4]. Penerapan strategi CRM meliputi mendapatkan pelanggan baru, menjaga hubungan dengan pelanggan yang ada, dan mengembangkan nilai-nilai pelanggan, sehingga dapat meningkatkan pertumbuhan dan keuntungan jangka panjang bagi perusahaan, termasuk perusahaan farmasi.

Pada perusahaan farmasi terdapat tenaga penjualan atau *sales force*, salah satunya yaitu *medical representative* (*medical rep*) yang menangani sejumlah kunjungan ke dokter dan tenaga medis lainnya yang berwenang menangani pasien guna mempromosikan produk perusahaannya yang dilakukan berdasarkan wilayah kerjanya [7].

Produk yang dipromosikan berupa obat-obatan *ethical*, yaitu obat yang diresepkan dokter dan tidak dijual bebas [8]. Selain kunjungan ke dokter, *medical rep* juga bertugas untuk melakukan survei ke apotek untuk mendapatkan informasi obat yang ditulis oleh dokter yang akan dikunjungi tersebut. *Medical rep* juga bertanggung jawab untuk menyerahkan laporan hasil kunjungan dan transaksi yang dilakukan kepada atasannya.

Proses pencatatan kunjungan harian *medical rep* ke dokter dimulai dari pencatatan data dokter beserta jadwal praktiknya, data resep dokter setelah berkunjung ke apotek, perencanaan kunjungan, kartu kunjungan yang harus ditandatangani dokter sebagai bukti kunjungan, serta pencatatan transaksi yang dilakukan antara *medical rep* dengan dokter. Seluruh pencatatan dilakukan secara manual dan *paper-based*, yang kemudian direkap ulang dan dicatat kembali ke komputer menggunakan Ms. Excel untuk di-*submit* ke perusahaan dengan berbentuk laporan pada keesokan harinya, bahkan bisa sampai satu minggu kemudian, sehingga membutuhkan waktu tambahan untuk membuat laporan tersebut. Hal ini menyebabkan penyerahan laporan kepada perusahaan yang tidak tepat waktu. *Medical rep* juga dituntut untuk menjaga ketelitiannya saat membuat laporan agar menghindari terjadinya duplikasi data maupun kesalahan dalam memasukkan data ke Excel. Disisi lain, dokter-dokter yang dikunjungi oleh *medical rep* harus diklasifikasi terlebih dahulu dengan menghitung potensi berdasarkan peresepan yang dilakukan dokter, namun *medical rep* terkadang tidak melakukan penentuan prioritas berdasarkan klasifikasi besar peresepan dokter yang akan dikunjungi, sehingga kemungkinan besar kunjungan tidak akan memberikan pencapaian target yang signifikan bahkan tidak terjadi kerja sama antara *medical rep* dan dokter.

Sedangkan untuk atasan *medical rep*, yaitu supervisor *medical rep* (supervisor) bertugas untuk memberikan tugas dan target penjualan yang harus dicapai kepada setiap *medical rep* yang ada di timnya. Oleh sebab itu, supervisor harus mampu memonitor pekerjaan dari *medical rep* meliputi kunjungan dan target penjualan yang dicapainya, selain itu supervisor juga menerima laporan kunjungan dan transaksi *medical rep* dengan dokter yang bekerja sama untuk mengevaluasi capaian target dari *medical rep*. Akan tetapi, karena *medical rep* perlu tambahan

waktu untuk membuat laporan, sehingga akan berpengaruh terhadap supervisor dalam mengevaluasi capaian target dari *medical rep*, yaitu menyebabkan keterlambatan dalam evaluasi dan penentuan target penjualan selanjutnya.

Berdasarkan permasalahan-permasalahan tersebut, dibuatlah suatu aplikasi yang menerapkan CRM strategis yaitu CRM yang berfokus untuk mengidentifikasi prospek pelanggan, pesaing, dan target perusahaan, CRM operasional yang berfokus kepada proyek-proyek otomatisasi seperti otomatisasi pemasaran dan penjualan, CRM analitis untuk mendukung kegiatan pemasaran dan penjualan, serta evaluasi pencapaian penjualan dari *sales representative* [4]. Aplikasi ini dapat membantu proses kerja dari *medical rep* agar lebih mudah dan cepat, serta membantu supervisor dalam memonitor pekerjaan yang dilakukan oleh *medical rep*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, terdapat rumusan masalah sebagai berikut:

1. Proses perekapan data kunjungan maupun transaksi harus dilakukan dari catatan *medical rep* ke komputer, karena tidak terdapat suatu model data yang terpusat. Akibatnya, saat perekapan ulang rentan terjadi kesalahan data atau *input* data yang berulang, dan data menjadi tidak akurat, sehingga laporan yang diserahkan kepada atasan *medical rep* tidak sesuai dengan hasil kerja *medical rep* serta rentan terjadinya kesalahan ketika proses analisis maupun evaluasi. Untuk menangani masalah ini diperlukan aplikasi yang menyediakan suatu model data yang terpusat.
2. Pekerjaan *medical rep* lebih banyak di lapangan sedangkan pembuatan laporan bersifat *office centric* (dilakukan di kantor). Karena tidak adanya sistem yang terintegrasi, maka pembuatan laporan dilakukan keesokan harinya bahkan seminggu kemudian, sehingga menyebabkan terlambatnya proses evaluasi oleh supervisor mengenai pencapaian target *medical rep*. Untuk dapat menyelesaikan masalah ini diperlukan suatu aplikasi yang dapat mengintegrasikan penyerahan laporan dari *medical rep* ke supervisor.

3. *Medical rep* tidak memperhatikan prioritas dokter yang harus dilakukan kunjungan berdasarkan besar potensi peresepan dokter apabila terjadi kesepakatan dengan dokter, sehingga sulit dalam mencapai target penjualan. Untuk itu diperlukan suatu aplikasi yang dapat mengklasifikasikan dokter berdasarkan besar potensi peresepan dokter agar *medical rep* dapat memprioritaskan kunjungannya.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dijelaskan, tujuan dari pengembangan aplikasi adalah:

1. Meminimalkan kesalahan data atau *input* data berulang yang dapat terjadi saat proses pembuatan laporan, karena data yang dimasukkan oleh *medical rep* langsung tercatat ke sistem dengan adanya model data terpusat dan tidak perlu direkap ulang, hal tersebut dapat meningkatkan ketelitian dari *medical rep*, karena proses *input* menjadi lebih singkat.
2. Mempercepat penyerahan laporan yang dilakukan oleh *medical rep* kepada supervisor, karena sistem terintegrasi maka data yang dimasukkan adalah data yang terjadi dilapangan dan pembuatan laporan tidak bersifat *office centric*, sehingga proses evaluasi pencapaian target *medical rep* dapat dilakukan dengan tepat waktu.
3. Membantu *medical rep* dalam mengklasifikasikan dokter yang akan dikunjungi berdasarkan besar potensi peresepan dokter terhadap produk sejenis dengan produk yang dipromosikan *medical rep*, sehingga memberikan informasi klasifikasi dokter kepada *medical rep* agar dapat memprioritaskan kunjungannya.

1.4 Ruang Lingkup

Ruang lingkup dalam pembangunan aplikasi ini adalah:

1. Aplikasi dibangun berbasis Android dan *website*.
2. Aplikasi berbasis Android menggunakan versi 5.0 (Lollipop), karena versi ini menyediakan fitur *all screen*, yaitu cocok digunakan di layar apapun.
3. Aplikasi harus terhubung dengan internet, karena memerlukan integrasi data dari *server* serta untuk mendukung sistem notifikasi.

4. Aplikasi membantu pekerjaan supervisor dalam mengelola target dan tugas *medical rep* (pencatatan tugas dan target *medical rep*), memonitor pencapaian target *medical rep*, serta perhitungan pencapaian target *medical rep*.
5. Aplikasi membantu pekerjaan yang dilakukan *medical rep* meliputi pencatatan data dokter, perhitungan potensi peresepan dokter, dan pembuatan laporan (laporan kunjungan dan laporan transaksi).
6. Aplikasi menampilkan informasi mengenai produk perusahaan yang harus dipromosikan oleh *medical rep* dan menyediakan pengelolaan produk yang dapat dilakukan oleh supervisor.

1.5 Batasan Masalah

Batasan masalah dalam Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical Representative* yaitu, sebagai berikut:

1. Aplikasi yang dibangun termasuk ke dalam jenis CRM strategis, operasional dan analitis. Aplikasi tidak termasuk ke dalam jenis CRM kolaboratif, karena aplikasi tidak mendukung hubungan perusahaan farmasi dengan perusahaan yang menjadi partner (hubungan antarperusahaan).
2. Aplikasi menangani proses pengelolaan data dokter, rencana kunjungan, serta pengelolaan tugas dan target *medical rep*, perhitungan besar potensi peresepan dokter, penentuan klasifikasi dokter, pencatatan hasil kunjungan dan transaksi, perekapan laporan kunjungan dan transaksi, perhitungan pencapaian target dari *medical rep*, pengelolaan produk, serta pengelolaan akun *medical rep* dan supervisor.
3. Proses perhitungan atau penentuan besar target untuk setiap *medical rep* yang dilakukan oleh supervisor tidak dilakukan oleh aplikasi yang dibangun.
4. Pada pengelolaan tugas dan target *medical rep*, aplikasi tidak menangani pembagian area kerja, sehingga penerapan studi kasus pada pengembangan aplikasi CRM berdasarkan satu wilayah kerja *medical rep*.
5. Hasil dari proses pengklasifikasian dokter yang dilakukan oleh sistem, terbatas pada pemberian informasi pendukung untuk *medical rep* dalam menentukan rencana kunjungan.

1.6 Metodologi Penyelesaian Tugas Akhir

Pada sub bab ini dijelaskan mengenai tahapan dan metodologi dalam menyelesaikan Tugas Akhir Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical Representative*.

1.6.1 Pendefinisan Masalah

Tahap paling awal dilakukan adalah mencari topik tugas akhir yang diambil. Kemudian diskusi dengan anggota kelompok dan komunikasi dengan dosen pembimbing dilakukan untuk menentukan kelayakan permasalahan yang diambil. Untuk memahami lebih dalam mengenai permasalahan yang terjadi, dilakukan wawancara ke *medical rep* pada suatu perusahaan farmasi, dan melakukan studi literatur.

1.6.2 Studi Literatur

Studi literatur ini berhubungan dengan tahap pendefinisan masalah, dilakukan dengan mencari referensi berupa buku bacaan, *thesis*, disertasi maupun jurnal yang terkait dengan topik tugas akhir, kemudian mempelajari aplikasi sejenis dengan memahami karakteristik dari aplikasi tersebut.

Tahapan pendefinisan masalah dan studi literatur menghasilkan latar belakang, rumusan masalah, batasan dan ruang lingkup, serta tujuan yang akan dicapai apabila masalah dapat terselesaikan.

1.6.3 Pengembangan Aplikasi

Dalam melakukan pengembangan aplikasi CRM mengikuti metode *rational unified process* (RUP) [9], yang terdiri atas tahapan sebagai berikut:

1. *Inception*

Tujuan dari tahap ini adalah mendapatkan gambaran besar mengenai aplikasi yang akan dibuat. Agar tujuan dari fase ini dapat tercapai, maka diperlukan analisis bisnis proses yang berjalan saat ini, aktor-aktor yang terlibat, *functional requirement*, rancangan anggaran biaya dan jadwal pengembangan aplikasi. Pada fase *inception* ini hanya dilakukan satu kali iterasi.

Kegiatan yang dilakukan pada fase ini meliputi

- a. pembuatan *business modeling* dengan *tools activity diagram*,
- b. pembuatan domain model,
- c. pembuatan rencana anggaran biaya,
- d. pembuatan rencana pengembangan aplikasi,
- e. pembuatan use case *model* inti, dan
- f. pembuatan use case *description* inti.

Hasil dari fase ini yaitu

- a. domain model,
- b. ruang lingkup aplikasi yang akan dibuat sudah terdefinisi,
- c. bisnis proses pekerjaan *medical rep* dan supervisor,
- d. rencana pengembangan aplikasi,
- e. rancangan anggaran biaya, dan
- f. use case *model* fitur utama.

2. *Elaboration*

Pada fase *elaboration* ini, lebih banyak ditekankan pada proses analisis dan desain aplikasi yang akan dibuat. Pada fase ini dilakukan analisis domain masalah dan arsitektur dari sistem secara keseluruhan. Fase ini juga memastikan bahwa *requirement* dan arsitektur sudah cukup stabil untuk diteruskan ke fase *construction*.

Pada fase elaboration ini terbagi menjadi tiga iterasi yang mana kegiatan setiap fasenya terbagi menjadi berikut:

Iterasi 1

- a. Analisis *requirement* menggunakan *goals oriented requirement engineering* (GORE).
- b. Analisis sistem sejenis.
- c. Membuat domain model sistem *to-be*.
- d. Pembuatan use case diagram berdasarkan hasil analisis *requirement*.
- e. Pembuatan use case *description*.
- f. Pembuatan system sequence diagram.

- g. Pembuatan operation contract.
- h. Pembuatan arsitektur sistem.
- i. Pembuatan sequence diagram untuk fitur inti sistem.
- j. Pembuatan mockup fitur inti.
- k. Pembuatan class diagram versi 1.
- l. Pembuatan entity relationship diagram versi 1.
- m. Pembuatan *prototype* berdasarkan *design* pada iterasi ini.

Iterasi 2

- a. Evaluasi dan *review* hasil analisis yang sudah dilakukan.
- b. Pembuatan sequence diagram fitur pendukung dari fitur inti sistem.
- c. Pembuatan mockup fitur pendukung dari fitur inti sistem.
- d. Pembuatan class diagram versi 2.
- e. Pembuatan entity relationship diagram versi 2.
- f. Pembuatan *prototype* berdasarkan *design* pada iterasi ini.

Iterasi 3

- a. Evaluasi dan *review* hasil analisis dan *design* yang sudah dilakukan.
- b. Pembuatan sequence diagram fitur lainnya dari sistem.
- c. Pembuatan mockup fitur lainnya dari sistem.
- d. Pembuatan class diagram versi 3.
- e. Pembuatan entity relationship diagram versi 3.
- f. Pembuatan *prototype* berdasarkan *design* pada iterasi ini.

Hasil yang harus dicapai pada fase elaboration adalah

- a. use case *model* dari setiap *functional requirement* minimal 80% terpenuhi,
- b. *non functional requirement* dan *requirement* lain dapat terdefinisi,
- c. deskripsi arsitektur sistem telah dibuat, dan
- d. *prototype* telah terimplementasi dan teruji.

3. *Construction*

Pada fase ini merupakan fase integrasi semua komponen dan fitur yang telah terimplementasi dalam bentuk *prototype* dan diuji kembali agar fungsi dari fitur-fitur sudah berjalan baik dan terintegrasi satu sama lain.

Pada fase ini dibagi menjadi tiga iterasi dengan masing-masing kegiatan iterasinya adalah sebagai berikut:

Iterasi 1

Integrasi implementasi dari fase *elaboration* iterasi 1 dan iterasi 2.

Iterasi 2

Integrasi implementasi dari fase *construction* iterasi 1 dan fase *elaboration* iterasi 3.

Iterasi 3

Review keseluruhan sistem yang telah dibuat.

Hasil yang harus dicapai pada fase ini, yaitu

- a. sistem yang dibuat telah terintegrasi secara keseluruhan dan telah teruji, serta,
- b. deskripsi sistem telah dibuat.

1.6.4 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan untuk mengetahui capaian yang didapat dari hasil pelaksanaan tugas akhir. Pada tahap ini menghasilkan kesimpulan dari tugas akhir yang dikerjakan.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar-dasar teori pendukung, teknologi dan *tools* yang digunakan untuk menyelesaikan masalah dan mencapai tujuan.

2.1 Teori Pendukung

Pada sub bab ini dijelaskan mengenai landasan teori yang digunakan pada Tugas Akhir Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical representative*.

2.1.1 *Customer Relationship Management*

1. Definisi *Customer Relationship Management*

Menurut kamus *online American Marketing Association* (AMA), *customer relationship management* (CRM) merupakan suatu disiplin dalam menggabungkan *database* pemasaran dan teknologi informasi dengan layanan pelanggan (*customer service*) dan fungsi komunikasi pemasaran. CRM berusaha untuk menciptakan komunikasi satu ke satu yang lebih bermakna dengan pelanggan dengan menerapkan data pelanggan (demografi, industri, riwayat pembelian, dan lain-lain.) ke setiap sarana informasi [10].

Menurut Kotler dan Keller (2012) CRM berfokus pada pengembangan program perusahaan untuk menarik dan mempertahankan pelanggan yang tepat dan memenuhi kebutuhan individu dari para pelanggan bernilai tersebut [11].

CRM adalah istilah yang mengacu pada penerapan, strategi dan teknologi yang digunakan perusahaan untuk mengelola dan menganalisis interaksi dan data pelanggan di seluruh siklus hidup pelanggan, dengan tujuan untuk meningkatkan hubungan bisnis dengan pelanggan, membantu dalam mempertahankan pelanggan dan mendorong pertumbuhan dari penjualan [12].

Menurut Buttle (2009) CRM adalah strategi inti dalam bisnis yang mengintegrasikan proses-proses dan fungsi-fungsi internal dengan semua jaringan eksternal untuk menciptakan serta mewujudkan nilai bagi para pelanggan secara menguntungkan. CRM didukung dengan data pelanggan yang berkualitas dan teknologi informasi [4].

Dari keempat pengertian di atas, dapat disimpulkan bahwa CRM adalah penerapan strategi pada perusahaan dalam menciptakan nilai pelanggan dan meningkatkan hubungan dengan pelanggan yang bertujuan untuk meningkatkan pendapatan atau keuntungan perusahaan, dan unggul dalam bersaing dengan perusahaan lain. Selain itu CRM juga merupakan usaha untuk menarik pelanggan baru, mempertahankan pelanggan yang bernilai bagi perusahaan dan hubungan jangka panjang. CRM dapat mengkombinasikan proses bisnis yang meliputi proses pemasaran, penjualan, serta layanan pelanggan dengan teknologi informasi.

2. Tipe-Tipe *Customer Relationship Management*

CRM dapat dibagi menjadi empat jenis berdasarkan perbedaan penerapan pada beberapa perusahaan menurut Buttle (2009 : 4), yaitu:

- a. CRM strategis : CRM sebagai strategi bisnis paling penting yang mengutamakan konsumen dan bertujuan memikat dan mempertahankan pelanggan yang menguntungkan.
- b. CRM operasional : CRM yang berfokus pada proses-proses otomatisasi, seperti otomatisasi penjualan, pemasaran dan otomatisasi layanan pelanggan (*customer service*).
- c. CRM analitis : CRM yang berfokus pada kegiatan penggalian data pelanggan atau *intelligent mining* untuk tujuan-tujuan strategis dan taktis perusahaan.
- d. CRM kolaboratif : kolaborasi CRM dengan teknologi di semua aspek organisasi untuk mengoptimalkan hubungan perusahaan, *partner* perusahaan dan pelanggan.

Sesuai dengan tujuan dari aplikasi yang telah dijelaskan pada bab sebelumnya, maka CRM yang akan dibangun pada aplikasi merupakan jenis CRM yang meliputi, strategis, operasional dan analitis.

Pada perusahaan, manajemen strategis yaitu proses yang dapat membantu sebuah perusahaan untuk menjaga kesesuaian perusahaan dengan kondisi lingkungan bisnis. Misalnya, jika lingkungan bisnis menuju persaingan yang lebih ketat, maka perusahaan mungkin memerlukan pengurangan biaya dan mengembangkan lebih banyak produk dan layanan agar mampu menghadapi kompetisi dengan pesaing. Kemudian, jika sebuah perusahaan menghadapi lingkungan bisnis global, manajer perusahaan mungkin perlu untuk memasuki pasar baru, mengembangkan kemitraan internasional, dan sebagainya [5]. Strategis berkaitan dengan proses identifikasi tujuan jangka panjang atau strategi perusahaan untuk mencapai tujuan-tujuannya [6]. Definisi strategi sendiri yaitu rencana perusahaan untuk menerapkan keputusan yang dibuat dalam mencapai tujuan perusahaan [5].

Operasional pada perusahaan berkaitan dengan fungsi rutin dan aktivitas suatu perusahaan [6]. Dapat dikatakan bahwa operasional adalah pelaksanaan dari rencana yang telah dikembangkan oleh perusahaan. Selanjutnya, analitis berkaitan dengan proses analisis yang ada di perusahaan, digunakan untuk berbagai tujuan, termasuk merancang dan melaksanakan target dari pemasaran yang dilakukan perusahaan, meningkatkan perolehan pelanggan, mendukung dalam pengambilan keputusan yang berkaitan dengan pengembangan produk atau layanan, peramalan keuangan, dan analisis profitabilitas pelanggan [business essen].

Selain itu menurut Buttle (2009 : 369) CRM dibagi menjadi tiga kelompok, sebagai berikut:

- a. *Enterprise CRM suites* : solusi CRM untuk perusahaan yang pendapatannya lebih dari satu miliar US dollar dan memiliki lebih dari seribu karyawan. CRM ini menyediakan fungsionalitas yang sangat luas meliputi seluruh bisnis proses yang ada di perusahaan dan melibatkan seluruh karyawan perusahaan yang berhubungan dengan klien, bukan

hanya *staff* penjualan dan pemasaran saja, tetapi personil akuntansi, *help desk*, perwakilan *call center* dan lain-lain.

- b. *Midmarket CRM* : solusi CRM untuk perusahaan yang pendapatannya kurang dari satu milyar US dollar dan memiliki kurang dari seribu karyawan. CRM ini juga menyediakan fungsionalitas yang luas, namun seringkali memiliki kemampuan yang lebih terbatas pada area tertentu dan lebih mudah untuk digunakan dibandingkan solusi enterprise CRM.
- c. *CRM speciality tools* : CRM yang fungsionalitasnya lebih kecil dari *enterprise CRM suites* dan *midmarket CRM*, tetapi kemampuannya lebih dalam atau mengerucut. Misalnya otomatisasi pemasaran (*marketing automation*), otomatisasi penjualan (*sales-force automation*), *customer service*, *partner channel management and collaboration*, *customer analytics*, dan manajemen data pelanggan.

Pada aplikasi yang dibuat, solusi CRM yang akan diterapkan termasuk kedalam kelompok *speciality tools*.

3. Tujuan dan Manfaat Customer Relationship Management

Tujuan utama dari CRM yaitu untuk membantu perusahaan dalam memahami pelanggan dengan lebih baik, membangun hubungan dengan pelanggan dan mempertahankan pelanggan yang menguntungkan [12]. Selain itu, CRM juga memiliki tujuan sebagai berikut:

- a. Mengidentifikasi pelanggan perusahaan
- b. Penciptaan nilai pelanggan
- c. Mengatur hubungan dengan pelanggan
- d. Penyesuaian dari pelanggan perusahaan dan strategi komunikasi perusahaan terhadap pelanggan yang berbeda
- e. Pengolahan informasi pelanggan

Manfaat penerapan CRM, khususnya untuk perusahaan farmasi, yaitu:

- a. Perusahaan dapat menganalisis pendapatan pelanggan, yaitu dokter, dan data biaya untuk mengidentifikasi nilai pelanggan saat ini dan masa depan
- b. CRM dapat diterapkan sebagai usaha merencanakan pemasaran perusahaan

- c. Menggambarkan produk yang banyak digunakan pelanggan, sehingga dapat dijadikan perencanaan untuk memproduksi produk baru atau mengetahui kebutuhan pelanggan
- d. Mengembangkan penetapan harga baru
- e. Mengetahui perilaku pelanggan dan cara mempertahankannya
- f. Mengetahui tingkat kepuasan pelanggan

2.1.2 Pelaku Pemasaran Perusahaan Farmasi

1. *Medical Representative*

Medical representative (medical rep) adalah orang yang bekerja di bidang *sales* (penjualan) dan *marketing* (pemasaran) pada industri farmasi. *Medical rep* ini mewakili perusahaan untuk mempromosikan produknya kepada dokter sebagai penulis resep atau yang mewakili konsumen akhir (pasien) dalam menentukan obat yang akan dikonsumsinya dan mungkin juga ke institusi lain seperti klinik, rumah sakit, serta rumah bersalin [14].

Medical rep mempromosikan produk atau obat-obatan *ethical*, yaitu obat yang diresepkan dokter dan tidak dijual bebas, oleh karena itu perusahaan membutuhkan tim promosi atau *medical rep*, agar mempermudah dokter dan tenaga medis yang berwenang menangani pasien mendapatkan informasi lengkap mengenai nama produk, indikasi, dosis, kontraindikasi, efek samping, dan semua yang berhubungan dengan obat yang dipromosikannya.

Pekerjaan *medical rep* ini berkaitan dengan pemasaran pada perusahaan farmasi, karena *medical rep* bertugas untuk *detailing* atau mempromosikan produk perusahaannya kepada dokter sasaran. *Medical rep* tidak bertugas untuk menjual produk perusahaan secara langsung, artinya tidak membawa serta produk yang dijual, namun mereka mendapatkan target penjualan yang biasanya berdasarkan wilayah kerjanya [7].

Medical rep tidak membawa obat-obatan atau produk perusahaannya secara langsung seperti *salesman*, karena terkait peraturan negara, yaitu perusahaan atau pabrikan harus menunjuk distributor untuk menjual ke pengecer. Sedangkan penjualan akan terjadi apabila *medical rep* melakukan kunjungan

dan mempromosikan produk perusahaannya, sehingga dokter bekerja sama untuk menuliskan resep dan resep ditebus pasien di apotek dan apotek membeli ulang ke distributor yang menjual. Oleh karena itu penjualan dihasilkan oleh *medical rep* secara tidak langsung.

2. Supervisor

Menurut Kamus Besar Bahasa Indonesia (KBBI) *online* yang dikeluarkan oleh Kementerian Pendidikan dan Kebudayaan (Kemendikbud) supervisor berarti pengawas utama, pengontrol utama, penyelia [1]. Supervisor *medical rep* ini merupakan atasan dari *medical rep* yang bertugas untuk mengelola tugas dan target *medical rep*, serta memonitor pencapaian target *medical rep* atau bertanggung jawab atas pelaksanaan tugas para *medical rep* yang dipimpinnya secara tepat dan efisien sesuai dengan yang telah ditentukan oleh perusahaan.

3. Area Sales Manager

Area sales manager (ASM) adalah manajer yang membawahi suatu wilayah kerja tertentu [14]. Wilayah kerja tersebut dapat berdasarkan kota atau provinsi tergantung ketetapan perusahaan farmasi. ASM dan supervisor sebenarnya termasuk dalam tingkatan manajer penjualan atau *sales manager*, namun tingkatan ASM berada di atas supervisor atau dengan kata lain ASM merupakan atasan dari supervisor.

Sales manager sendiri adalah yang bertanggung jawab untuk merencanakan, mengatur, mengarahkan, dan mengendalikan fungsi penjualan. Biasanya ada beberapa tingkatan dari *sales manager* ini, mulai dari *general sales manager* hingga supervisor yang mengatur langsung *salesperson* (*medical rep*) [10].

ASM ini berupa sebutan yang diberikan kepada manajer penjualan di tingkat regional, distrik, cabang, atau unit di perusahaan dengan jumlah tenaga penjualan yang besar. Sebutan ASM berbeda di tiap perusahaan, ada yang menyebut ASM sebagai *district sales manager* (DSM), *field sales manager* (FSM), dan sebagainya. Sesuai dengan pengertiannya, peran ASM yaitu berhubungan dengan pembagian wilayah kerja supervisor, merencanakan kuota/jumlah tenaga penjualan (supervisor maupun *medical rep*) dan biaya

yang dibutuhkan dalam wilayah kerjanya, serta mengontrol pekerjaan dari supervisor yang berada dalam timnya.

2.1.3 Penerapan CRM

Aplikasi *enterprise CRM suites* dapat digunakan oleh perusahaan farmasi yang memiliki lebih dari seribu tenaga penjualan dan perusahaan telah menjadi perusahaan global, lingkup dari *enterprise CRM suites* sangat luas, yaitu meliputi seluruh kegiatan bisnis yang ada di perusahaan bahkan hubungan antarperusahaan dapat didukung dengan *enterprise CRM suites*, selain itu, *enterprise CRM suites* melibatkan jumlah pengguna yang banyak. Fungsionalitas *enterprise CRM suites* dapat meliputi *business intelligence, purchasing, inventory, shipping/logistics*, dan *accounting*.

Midmarket CRM dapat juga menjadi alternatif pilihan untuk diterapkan di perusahaan farmasi karena menawarkan fungsionalitas yang hampir sama dengan *enterprise CRM suites*, yaitu dapat meliputi seluruh kegiatan bisnis yang ada di perusahaan, tetapi terdapat keterbatasan dalam mendukung kegiatan bisnis tertentu. *Midmarket CRM* juga tidak dapat digunakan untuk skala global karena tidak mendukung kegiatan antarperusahaan, sedangkan untuk jumlah tenaga penjualan yang dapat menggunakan *midmarket CRM* adalah perusahaan dengan jumlah tenaga penjualan kurang dari seribu. Dapat dikatakan *midmarket CRM* adalah versi ringan dari *enterprise CRM suites* karena penggunaan aplikasinya lebih sederhana.

Untuk CRM *speciality tools*, CRM ini digunakan untuk mendukung kegiatan bisnis tertentu. CRM *speciality tools* memiliki fungsionalitas yang sempit tetapi mengerucut atau kemampuan khusus yang lebih dalam pada kegiatan bisnis tertentu, misalnya *sales* atau *marketing*. CRM *speciality tools* ini dapat digunakan oleh perusahaan yang tergolong ke dalam *enterprise* atau *midmarket*, karena CRM *speciality tools* dapat diterapkan menjadi suatu fungsi pada *enterprise* maupun *midmarket CRM*, pengguna yang terlibat pun lebih sedikit dibandingkan *enterprise* dan *midmarket CRM*.

Dilihat dari fungsionalitas dan jumlah pengguna yang terlibat, maka penerapan CRM pada aplikasi yang dibangun termasuk kedalam CRM *speciality tools*, karena

fungsionalitasnya lebih spesifik pada kegiatan bisnis tertentu di perusahaan farmasi. Kegiatan bisnis yang didukung oleh aplikasi yang dibangun, yaitu penjualan dan pemasaran dengan menerapkan konsep dari *sales force automation* (SFA) dan *marketing automation* (MA).

Sales force automation (SFA) yaitu aplikasi yang digunakan oleh bagian *sales* atau *sales management* untuk membantu melakukan otomatisasi fungsi-fungsi *sales force*. *Sales force* yaitu tenaga penjualan pada suatu perusahaan, dimana pada perusahaan farmasi salah satu bagian dari *sales force* dikenal dengan sebutan *medical rep*. Dengan adanya *medical rep* dapat meningkatkan kinerja dan penjualan agar lebih efisien, efektif dan terkontrol, artinya *medical rep* dan supervisor mampu menjalankan tugas dengan tepat, cermat, hemat biaya, waktu, tenaga, dan mencapai hasil akhir yang diinginkan.

Fungsi SFA, yaitu memanfaatkan teknologi untuk membantu tugas *sales force* dan supervisor. Dilihat dari sisi *medical rep*, fungsinya yaitu sebagai berikut:

1. Mencatat daftar dokter.
2. Mengelola *visit plan* dari tugas yang diberikan.
3. Memberikan perhitungan besar potensi peresepan dan klasifikasi berdasarkan hasil perhitungan besar potensi peresepan dari dokter yang akan dikunjungi.
4. Pencatatan kesepakatan peresepan atau transaksi antara *medical rep* dengan dokter.
5. Mencatat barang atau obat yang akan dipromosikan *medical rep*.
6. Mencatat kompetitor atau pesaing perusahaan dari hasil survei peresepan dokter.
7. Mencatat laporan kunjungan harian maupun laporan transaksi.

Sedangkan dari sisi supervisor, berfungsi untuk:

1. Membantu dalam mencatat tugas dan target yang diberikan kepada *medical rep*.
2. Memonitor pencapaian target penjualan yang dilakukan oleh *medical rep*.
3. Mengoptimalkan penyampaian informasi maupun laporan dari *medical rep* dengan menyediakan fitur *generate report*. Fitur *generate report* dapat menghasilkan informasi besar transaksi dan kunjungan yang telah dilakukan oleh *medical rep*, produk yang banyak terjual dan besar jumlah pencapaian penjualan perbulan dari seluruh *medical rep*. Informasi yang diperoleh tersebut,

dapat membantu proses analisis untuk membuat langkah strategis maupun keputusan terkait penjualan kedepannya.

4. Mendapatkan informasi mengenai nama perusahaan atau nama produk pesaing yang dapat digunakan untuk membantu menentukan langkah strategis dalam penjualan produk perusahaan di wilayah kerja *medical rep* agar mampu bersaing dengan kompetitor.
5. Mengevaluasi pencapaian target dari *medical rep*, sehingga dapat digunakan untuk merencanakan strategi penjualan berikutnya.

Selain menerapkan konsep SFA, aplikasi juga menerapkan konsep *marketing automation* (MA). MA atau otomatisasi pemasaran adalah penerapan teknologi komputerisasi untuk mendukung pemasar dan manajemen pemasaran dalam pencapaian tujuan yang berhubungan dengan pekerjaan mereka [4].

MA menerapkan teknologi untuk proses pemasaran. Terdapat modul *campaign management* yang memungkinkan para pemasar untuk menggunakan data pelanggan dalam rangka mengembangkan, melaksanakan dan mengevaluasi komunikasi dan penawaran yang ditargetkan [4].

Pada aplikasi ini MA berkaitan dengan tugas supervisor dalam menghitung pencapaian target timnya, menghitung jumlah produk yang banyak digunakan pelanggan. Selain itu dengan adanya data prospek dokter juga dapat termasuk ke dalam MA, karena konsep MA sendiri memungkinkan para pemasar untuk menggunakan data dokter tersebut. Sebenarnya tugas utama *medical rep* yaitu untuk menciptakan *demand* (permintaan) produk yang dipromosikan atau dipasarkannya agar dokter mau meresepkan obat perusahaannya, sehingga hal ini berkaitan dengan pemasaran.

Penerapan fungsi dari aplikasi yang berkaitan dengan jenis-jenis CRM, dapat ditunjukkan pada Tabel 2.

Tabel 1 Penerapan CRM Secara Umum menurut Buttle (2009)

Jenis CRM	Strategic	Operational	Analytical
Lingkup	<p><i>Strategic</i> CRM berfokus pada pengembangan bisnis yang berorientasi pada pelanggan atau <i>customer-centric</i> dengan tujuan untuk mendapatkan pelanggan dan mempertahankan pelanggan dengan menciptakan dan memberikan nilai yang lebih baik dibandingkan dengan kompetitor.</p>	<p>Pada <i>operational</i> CRM berfokus pada proyek-proyek otomatisasi. CRM ini berfokus pada cara-cara perusahaan berhubungan dengan pelanggan. Dengan kata lain adalah cara perusahaan berhubungan dengan pelanggan melalui teknologi-teknologi yang ada.</p>	<p><i>Analytical</i> CRM fokus pada pengumpulan, penyimpanan, pengolahan serta penggunaan data yang berhubungan dengan pelanggan untuk meningkatkan nilai pelanggan dan perusahaan.</p>

Tabel 2 Penerapan CRM Studi Kasus *Medical Representative*

Jenis CRM	Strategic	Operational	Analytical
Lingkup Pada Studi Kasus <i>Medical Representative</i>	<p><i>Strategic</i> CRM digunakan untuk mengetahui data tentang pasar dari produk yang akan dipromosikan oleh <i>medical rep</i>, data dokter yang meliputi kinerja atau potensi peresepan dokter, dan data pesaing.</p> <p>Pencatatan data dokter beserta potensi peresepan dokter dilakukan dengan survei ke apotek, pencatatan potensi peresepan dokter ini digunakan untuk dapat mengidentifikasi target dokter atau prospek dokter, dan produk apa yang harus ditawarkan kepada dokter, sehingga menciptakan nilai bagi dokter, serta mencatat data pesaing dari produk yang dipromosikan <i>medical rep</i>.</p>	<p><i>Operational</i> CRM pada studi kasus <i>medical representative</i> digunakan untuk memanfaatkan teknologi-teknologi yang dapat membantu kegiatan bisnis sehari-hari yang dilakukan oleh <i>medical rep</i> dan supervisor, misalnya kegiatan kunjungan ke dokter, pencatatan transaksi, pengelolaan data produk, pelaporan, dan sebagainya.</p> <p>Pencatatan data hasil kunjungan, transaksi, dan pelaporan yang harus dibuat oleh <i>medical rep</i> terbantu dengan menggunakan <i>operational</i> CRM. Pengelolaan produk yang dilakukan supervisor pun akan terbantu dengan sistem yang dapat menyimpan data produk ke <i>database</i>.</p>	<p><i>Analytical</i> CRM menggunakan data dokter berdasarkan informasi yang diperoleh dari survei maupun dari proses bisnis sehari-hari untuk mendukung dalam meningkatkan nilai dokter maupun perusahaan, serta mengelola data pencapaian target dari <i>medical rep</i> sehingga supervisor dapat memonitor pekerjaan <i>medical rep</i>.</p> <p>Data dokter tersimpan pada <i>database</i> sehingga dapat dilakukan <i>monitoring</i> hubungan antara perusahaan dengan dokter dalam melakukan kerja sama dalam peresepan obat, sehingga dengan <i>analytical</i> CRM dapat membantu dalam meningkatkan nilai bagi dokter maupun perusahaan dan mempertahankan dokter.</p> <p>Adapun data dari proses bisnis yang dilakukan <i>medical rep</i>, dapat diolah dan memberikan informasi mengenai <i>medical rep</i> dalam melakukan tugas dan pencapaian targetnya sehingga dapat digunakan untuk evaluasi oleh supervisor dalam</p>

			mendukung rencana strategis dan operasional kedepannya.	
No	Penerapan Fungsi-Fungsi pada CRM Studi Kasus <i>Medical Rep</i>			
1.	Pengelolaan Tugas dan Target <i>Medical Rep</i> (<i>Medical Rep's Task and Target Management</i>)	<ul style="list-style-type: none"> - Penentuan target yang berbeda-beda kepada <i>medical rep</i> oleh supervisor sesuai dengan data dokter dan produk yang berkaitan dengan wilayah kerja dari <i>medical rep</i>. - Pemberian tugas dan target kepada setiap <i>medical rep</i> yang dilakukan oleh supervisor. 		
2.	Pengelolaan Data Dokter (<i>Doctor's Data Management</i>)	<ul style="list-style-type: none"> - Pencatatan data dokter yang akan menjadi target kunjungan <i>medical rep</i> (data prospek dokter). - Pencatatan rata-rata pasien yang ditangani dokter per hari. - Perhitungan besar potensi peresepan yang dilakukan dokter terhadap produk yang akan di-<i>detailing</i> oleh <i>medical rep</i>. 	<ul style="list-style-type: none"> - Menampilkan informasi mengenai dokter yang harus dikunjungi <i>medical rep</i> baik itu <i>detailing</i> maupun <i>follow up</i>. 	<ul style="list-style-type: none"> - Menampilkan klasifikasi dokter berdasarkan hasil perhitungan besar potensi peresepan dokter untuk membantu <i>medical rep</i> dalam memprioritaskan kunjungannya ke dokter. - Menyediakan informasi untuk mengembangkan prospek dokter dalam mencapai target peresepan.
3.	Pengelolaan Kunjungan Harian <i>Medical Rep</i> (<i>Daily Visit Management</i>)	<ul style="list-style-type: none"> - Mencatat perencanaan kunjungan yang akan dilakukan oleh <i>medical rep</i>. - Pencatatan data kompetitor atau pesaing perusahaan yang bekerja sama dengan dokter berupa nama perusahaan atau produk yang digunakan dokter. 	<ul style="list-style-type: none"> - Melakukan pencatatan hasil kunjungan yang telah dilakukan oleh <i>medical rep</i>. - Melakukan pencatatan transaksi atau kerja sama antara <i>medical rep</i> dan dokter. 	<ul style="list-style-type: none"> - Menyediakan informasi mengenai pesaing dan produknya untuk digunakan sebagai langkah strategis perusahaan di masa depan.

4.	Pengelolaan Data Produk (<i>Product's Data Management</i>)	<ul style="list-style-type: none"> - Pemberian daftar produk yang ditugaskan kepada <i>medical rep</i> untuk di-<i>detailing</i> kepada dokter. - Menampilkan daftar produk beserta penjelasan produk yang ditugaskan untuk <i>medical rep</i>. - Pengelolaan data produk yang dilakukan oleh supervisor. 	<ul style="list-style-type: none"> - Menyediakan informasi produk yang banyak terjual. - Memberikan gambaran wilayah atau target pasar dengan produk yang banyak terjual.
5.	Pelaporan (<i>Reporting</i>)	<ul style="list-style-type: none"> - <i>Generate</i> laporan kunjungan - <i>Generate</i> laporan transaksi 	
6.	Evaluasi Pencapaian Target <i>Medical Rep</i> (<i>Evaluation Medical Rep's Target Achievement</i>)		<ul style="list-style-type: none"> - Perhitungan pencapaian target dari <i>medical rep</i>.

Dengan menerapkan CRM, terdapat beberapa perhitungan dalam mendukung pekerjaan *medical rep* maupun supervisor, diantaranya:

1. Klasifikasi Dokter

Menurut hasil wawancara, klasifikasi dokter dilakukan untuk memberikan informasi kepada *medical rep* dalam menentukan prioritas kunjungan kepada masing-masing dokter sesuai kelasnya. Ada beberapa tahap dalam mengklasifikasikan dokter yaitu:

a. Menentukan potensi dokter.

Dalam melakukan tugasnya, *medical rep* harus mengetahui dokter tersebut berpotensi meresepkan obat yang akan ditawarkan atau tidak. Tidak semua dokter yang ada dalam suatu daerah disebut berpotensi. Terdapat persyaratan dalam menentukan dokter tersebut berpotensi. Menurut hasil wawancara dengan beberapa narasumber, dokter yang berpotensi meresepkan, dapat dilihat dari jenis obat apa yang sering diresepkan oleh dokter tersebut. Bila ada resep dokter yang meresepkan obat yang akan ditawarkan oleh *medical rep*, maka dokter tersebut disebut berpotensi

dengan mengesampingkan besar peresepan dokter tersebut dalam meresepkan obat sejenis tersebut. Berikut tahapan-tahapan dalam menentukan dokter berpotensi (dengan asumsi untuk dokter yang dicek merupakan dokter yang datanya belum tersimpan di perusahaan *medical rep* bekerja).

1. *Medical rep* melihat spesialis dokter tersebut, apakah spesialisnya sesuai dengan jenis obat yang akan ditawarkan *medical rep*.
 2. Jika benar, maka *medical rep* mencatat data dokter tersebut.
 3. *Medical rep* kemudian mengunjungi apotek terdekat untuk melihat rekap resep dokter tersebut dalam satu bulan terakhir.
 4. Jika dokter tersebut dalam waktu satu bulan terakhir ada resep obat sejenis dengan yang akan dipromosikan *medical rep*, maka dokter tersebut bisa disebut dokter potensial.
- b. Menghitung besar potensi peresepan dokter.

Setelah mengetahui dokter tersebut berpotensi, maka selanjutnya adalah menghitung besar peresepan dokter tersebut. Tujuannya adalah agar *medical rep* dapat mengetahui sejauh mana dokter tersebut dapat memenuhi target *medical rep*, kemudian dapat mengklasifikasikan dokter tersebut sesuai besar peresepannya. Menurut hasil wawancara dengan narasumber, menghitung besar peresepan dokter dapat dirumuskan sebagai berikut:

Besar Potensi Peresepan =

$$\text{JHPD} \times \text{JM} \times \text{XPH} \times \sum_{i=1}^n (\text{XOS}[i] \times \text{HOS}[i])$$

Keterangan :

JHPD = Jumlah hari praktek dokter dalam satu minggu

JM = Jumlah minggu dalam satu bulan tersebut

XPH = rata-rata pasien per hari

XOS = rata-rata obat sejenis diresepkannya kepada pasien

HOS = harga obat sejenis yang akan ditawarkan

n = banyak produk

Contoh :

Diketahui target *medical rep* untuk satu bulan adalah harus menghasilkan Rp10.000.000,00.

Dokter A praktek setiap hari Senin, Rabu dan Jumat setiap minggunya. Rata-rata pasien per hari ke dokter tersebut adalah 10 pasien. Kemudian banyak peresepan vitamin C (obat sejenis yang akan diresepkan oleh *medical rep*) rata-rata adalah 30 tablet per pasien. Harga vitamin C per tablet nya adalah Rp1000,00 (harga tersebut merupakan harga dari perusahaan *medical rep*)

Setelah mendapatkan informasi yang dibutuhkan, kemudian kalkulasi dengan menggunakan rumus di atas sebagai berikut:

Target *medical rep* dalam 1 bulan = Rp10.000.000,00

JHPD = 3

JM = 4

XPH = 10

XOS = 30

Harga obat yang akan ditawarkan = Rp1.000,00

Besar potensi peresepan = $((3*4*10* 30* 1000)) = 3600000$

Hasil perhitungan menghasilkan bahwa besar potensi peresepan dokter tersebut adalah sebesar Rp3.600.000,00. Hasil tersebut akan digunakan *medical rep* sebagai informasi dalam menentukan klasifikasi dokter.

c. Menentukan klasifikasi dokter.

Setelah mengetahui besar potensi peresepan dokter yang menjadi tugasnya *medical rep*, langkah berikutnya yaitu menentukan klasifikasi dokter menggunakan metode pareto ABC atau hukum pareto ABC. Metode pareto ABC adalah sebuah konsep dengan mengkategorikan menjadi tiga kategori, yaitu kategori A, B dan C. Tabel 3 dijelaskan mengenai pembagian persentase untuk menentukan kelas menurut heizer dan render[13].

Tabel 3 Nama-Nama Kelas, Persentase Akumulasi dan Persentase Jumlah Item Keseluruhan

Nama Kelas	Persentase Akumulasi	Persentase Jumlah Item Keseluruhan
A	60-70%	10-20%
B	15-25%	20-40%
C	5-10%	60-70%

Metode ini bertujuan agar supervisor ataupun *medical rep* dapat lebih memperhatikan dokter yang berada pada kelas A.

Tahapan proses pengklasifikasian dokter, ditunjukkan pada tabel 4 sampai tabel 6. Tabel 4 dijelaskan data dokter beserta besar persepannya, setelah itu data dokter diurutkan berdasarkan besar persentase potensi persepnan dari yang terbesar ke yang terkecil ditunjukkan pada tabel 5. Kemudian, hitung besar persentase akumulasi dan tentukan kelas nya sesuai aturan yang sudah dijelaskan pada tabel 3. Perhitungan besar persentase akumulasi dan penentuan kelas ditunjukkan pada tabel 6.

Tabel 4 Nama Dokter Beserta Besar Potensi Persentasenya

Nama Dokter	Besar Potensi Persepnan	Besar Persentase (Besar Potensi Persepnan / Jumlah * 100%)
Dr. Azhari	Rp. 1.500.000	13.79%
Dr. Abu	Rp. 480.000	4.41%
Dr. Rizal	Rp. 675.000	6.21%
Dr. James	Rp. 600.000	5.52%
Dr. Febri	Rp. 200.000	1.84%
Dr. Ahmad	Rp. 2.500.000	22.99%
Dr. Zaelani	Rp. 4.500.000	41.38%
Dr. Lilis	Rp. 120.000	1.10%
Dr. Ayum	Rp. 175.000	1.61%
Dr. Sari	Rp. 125.000	1.51%
Jumlah	Rp. 10.875.000	

Tabel 5 Nama Dokter Beserta Potensi Peresepean yang Sudah Terurut

Nama Dokter	Besar Potensi Peresepean	Besar Persentase (Besar Potensi Peresepean / Jumlah * 100%)
Dr. Zaelani	Rp. 4.500.000	41.38%
Dr. Ahmad	Rp. 2.500.000	22.99%
Dr. Azhari	Rp. 1.500.000	13.79%
Dr. Rizal	Rp. 675.000	6.21%
Dr. James	Rp. 600.000	5.52%
Dr. Abu	Rp. 480.000	4.41%
Dr. Febri	Rp. 200.000	1.84%
Dr. Ayum	Rp. 175.000	1.61%
Dr. Sari	Rp. 125.000	1.51%
Dr. Lilis	Rp. 120.000	1.10%
Jumlah	Rp. 10.875.000	

Tabel 6 Nama Dokter Beserta Pengklasifikasianya

Nama Dokter	Besar Potensi Peresepean	Besar Persentase (Besar Potensi Peresepean / Jumlah * 100%)	Persentase Akumulatif	Kategori
Dr. Zaelani	Rp. 4.500.000	41.38%	41.38%	A
Dr. Ahmad	Rp. 2.500.000	22.99%	64.4%	A
Dr. Azhari	Rp. 1.500.000	13.79%	13.79%	B
Dr. Rizal	Rp. 675.000	6.21%	20%	B
Dr. James	Rp. 600.000	5.52%	25.52%	B
Dr. Abu	Rp. 480.000	4.41%	4.41%	C
Dr. Febri	Rp. 200.000	1.84%	6.25%	C
Dr. Ayum	Rp. 175.000	1.61%	7.86%	C
Dr. Sari	Rp. 125.000	1.51%	9.37%	C
Dr. Lilis	Rp. 120.000	1.10%	10.47%	C
Jumlah	Rp. 10.875.000			

Hasil tersebut menjadi informasi untuk *medical rep* dalam menentukan tindakan terhadap dokter sesuai kategorinya. Tindakan tersebut dapat berupa sikap *detailing* yang harus dilakukan, ataupun banyak rencana kunjungan yang harus dikunjungi untuk setiap masing-masing kelas/kategori.

2. Perhitungan Pencapaian Target dari *Medical Representative*

Menurut David Jobber dalam bukunya yang berjudul “*Selling and Sales Management 8th Edition*”, perhitungan kinerja *salesforce* adalah perbandingan antara target *salesforce* dengan hasil yang diperoleh [15]. Proses evaluasi adalah sebagai berikut:

a. Penetapan Indikator Penilaian

Menurut C. Wardana dalam bukunya yang berjudul “*Kiat Sukses Pengawasan Repetisi Medis*”, penilaian kinerja *medical rep* dapat dilihat dari dua indikator, yaitu pencapaian target tiap *medical rep* dari seluruh transaksi dan total dokter yang bekerjasama [14]. Pada aplikasi yang akan dikembangkan ruang lingkup yang menjadi indikator penilaian yaitu pencapaian target dari tiap *medical rep*.

b. Menentukan Standar Kinerja dari Tiap Indikator

Menurut Dedi Rianto dalam bukunya yang berjudul “*Manajemen Kinerja Sumber Daya Manusia*”, mendefinisikan standar kinerja (*performance standards*) sebagai tingkat yang diharapkan dari kinerja, dan merupakan “pembanding kinerja”, atau “tujuan”, atau “target” tergantung pada pendekatan yang diambil [16].

Standar kinerja atau pembanding dari pencapaian target *medical rep* pada aplikasi yang dikembangkan adalah target yang diberikan oleh supervisor kepada *medical rep*.

c. Membandingkan Hasil Aktual dengan Standar yang Telah Dibuat

Langkah terakhir adalah membandingkan hasil yang telah diperoleh dengan standar yang telah dibuat. Hasil yang telah diperoleh atau hasil aktual adalah pencapaian target dari *medical rep*, sedangkan standar yang telah dibuat yaitu target yang diberikan oleh supervisor kepada *medical rep*.

Perbandingan hasil aktual dan standar yang telah dibuat ini menjadi dasar dalam melakukan perhitungan pencapaian target dari *medical rep* yang berupa rumus perbandingan atau rasio. Rumus pencapaian target menurut Wardana (2015) adalah sebagai berikut:

$$\text{Pencapaian target} = \frac{\text{Total kesepakatan penjualan obat dokter}}{\text{Target penjualan dari } medical \ rep}$$

Keterangan :

Total kesepakatan penjualan obat dokter : jumlah keseluruhan transaksi yang berhasil dilakukan oleh *medical rep* (dalam rupiah)

Target penjualan dari *medical rep* : besar target yang diberikan supervisor kepada *medical rep* (dalam rupiah)

Hasil perhitungan di atas akan menghasilkan nilai pencapaian target *medical rep* yang menyatakan berapa kali lipat pencapaian target *medical rep* dari target penjualan yang diberikan supervisor.

2.2 Teknologi dan *Tools* Pendukung

Pada sub bab ini dijelaskan mengenai teknologi dan *tools* pendukung yang digunakan untuk mengembangkan aplikasi CRM studi kasus *medical representative* dan teknologi yang digunakan oleh pengguna aplikasi.

2.2.1 Teknologi dan *Tools* Pendukung dalam Pengembangan Aplikasi

1. CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi berbasis PHP dan mengimplementasi konsep MVC (*model, view, controller*) [17]. Pada pengembangan aplikasi CRM studi kasus *medical rep*, CodeIgniter digunakan untuk membuat *web backend* aplikasi, karena CodeIgniter menyediakan struktur yang baik dalam pembuatan *project* dan dapat terintegrasi dengan mudah dengan *library* REST sehingga dapat mempercepat pembuatan *web backend* aplikasi.

2. Android

Pemrograman Android adalah pemrograman berbasis Java untuk membuat aplikasi pada *device smartphone*, *tablet* maupun *device* lainnya yang menggunakan sistem operasi berbasis Android [18]. Pada pengembangan aplikasi CRM studi kasus *medical rep*, teknologi Android cocok digunakan untuk aplikasi *medical rep*, karena pekerjaannya banyak dilakukan diluar kantor atau lapangan sehingga mudah untuk dibawa kemana-mana.

3. AngularJS

AngularJS adalah suatu *framework javascript* yang pertama kali dikembangkan oleh Misko Hevery dan Adam Abrons pada tahun 2009. Saat ini dikembangkan oleh Google. AngularJS ini *open source* yang artinya bisa dikembangkan oleh siapa saja [19]. Pada pengembangan aplikasi CRM studi kasus *medical rep*, AngularJS digunakan untuk membuat *web frontend* aplikasi yang akan digunakan supervisor, karena AngularJS menyediakan kemudahan kepada *developer* untuk membuat *web app* dengan lebih baik dan dapat mengakses *web backend* dengan lebih mudah sehingga dapat mempercepat pembuatan *web frontend* aplikasi yang akan digunakan oleh supervisor.

4. Firebase Cloud Messaging

Firebase cloud messaging (FCM), sebelumnya dikenal sebagai Google *cloud messaging* (GCM), merupakan suatu layanan gratis yang bangun oleh Google sebagai solusi *cross-platform* untuk pesan dan pemberitahuan [20]. Pada pengembangan aplikasi CRM studi kasus *medical rep*, FCM akan digunakan untuk membantu penyampaian notifikasi ke aplikasi Android *medical rep*, karena FCM merupakan layanan yang dapat diintegrasikan dengan Android sehingga *developer* dapat terbantu dalam pembuatan notifikasi ke aplikasi Android *medical rep*.

5. REST

REST adalah suatu arsitektur metode komunikasi yang sering diterapkan dalam pengembangan berbasis *web*. Arsitektur REST, yang umumnya dijalankan via HTTP (*hypertext transfer protocol*), melibatkan proses pembacaan laman *web* tertentu yang memuat sebuah *file XML* atau *JSON* [21]. Pada pengembangan

aplikasi CRM studi kasus *medical rep*, REST digunakan sebagai jembatan komunikasi untuk mengakses *web backend* aplikasi.

6. MySQL

MySQL merupakan perangkat lunak sistem manajemen *database* dari SQL (*structured query language*). Pada pengembangan aplikasi CRM studi kasus *medical rep*, MySQL akan digunakan untuk mengelola *database* aplikasi, karena MySQL mampu menampung data yang cukup besar dan memiliki kecepatan yang stabil serta *syntax* yang digunakan untuk proses pengambilan atau memasukkan data (*query*) lebih mudah dipahami dan tidak terlalu rumit sehingga akan membantu *developer* dalam membuat proses memasukkan ataupun mengambil data dari MySQL.

7. GORE

Goals oriented requirements engineering (GORE) merupakan salah satu pendekatan dalam melakukan rekayasa kebutuhan (*requirements engineering*), dengan menggunakan metode rekayasa kebutuhan berorientasi pada tujuan. GORE merekayasa kebutuhan yang merasionalkan berbagai kebutuhan yang diperlukan oleh sebuah sistem yang akan dibuat berdasarkan tujuan-tujuan yang dirumuskan sehingga diharapkan kebutuhan yang didapatkan bukan hanya berdasarkan data dan proses bisnis manual [22].

Ada tiga kategori utama dari konsep GORE, yaitu unsur-unsur yang sengaja dibuat, relasi, dan aktor atau *stakeholder*. Unsur-unsur yang sengaja dibuat meliputi tujuan utama (*goal*), tugas (*task*), tujuan pendukung (*softgoal*), dan sumber daya (*resource*). Hal-hal tersebut sengaja dibuat karena digunakan untuk memungkinkan menjawab pertanyaan-pertanyaan mengenai perilaku tertentu, mengenai informasi yang dipilih untuk dimasukkan dalam kebutuhan sistem, alternatif apa yang dipertimbangkan, kriteria apa yang digunakan untuk menentukan pilihan alternatif, dan alasan apa yang mendasari pemilihan alternatif pada analisis sistem yang akan dibuat [23].

Pada pembuatan GORE terdapat beberapa tahapan yang harus dilakukan yaitu, sebagai berikut:

a. *Strategic dependency system As-Is*

Pada tahap awal penyusunan GORE, terlebih dahulu mengidentifikasi tujuan-tujuan *stakeholder* berdasarkan bisnis proses sistem yang sedang berjalan, yang digambarkan dengan *strategic dependency system as-is*.

b. *Strategic performance goals of the stakeholder*

Setelah mengetahui tujuan-tujuan *stakeholder* pada sistem yang berjalan, tahap berikutnya adalah mengelompokan lebih *detail* tujuan-tujuan dari masing-masing *stakeholder* yang digambarkan dengan *strategic performance goals of stakeholder*.

c. *Strategic dependency system To-Be*

Tahap berikutnya menggambarkan tujuan yang dapat dicapai dengan adanya pengembangan aplikasi CRM, yang digambarkan dengan *strategic dependency system To-Be*.

d. *Initial Specification*

Pada tahap ini dijelaskan mengenai *requirements* pada aplikasi yang akan dibangun berdasarkan tujuan-tujuan *stakeholder* yang telah dipaparkan sebelumnya.

BAB III

ANALISIS

Pada bab ini dijelaskan mengenai sistem yang sedang berjalan di perusahaan farmasi dan aplikasi yang sejenis. Tujuan dari analisis sistem yang sedang berjalan adalah untuk mengetahui proses-proses yang sedang berjalan. Lingkup analisis sistem yang dilakukan pada sistem yang sedang berjalan yaitu: analisis mengenai bisnis proses, pengguna dan peran pengguna, *business rules*, dan *domain model*. Selain itu terdapat tiga aplikasi sejenis yang dianalisis yaitu: Insightly CRM, Base CRM dan MediVisits. Tujuan dari analisis sistem yang sejenis adalah untuk mengetahui cara kerja dari aplikasi melalui fitur-fitur yang ada, sehingga dapat menyimpulkan kelebihan dan kekurangan dari masing-masing aplikasi. Hasil dari tahap ini dapat digunakan untuk membuat usulan solusi serta menentukan kebutuhan-kebutuhan atau *requirements* untuk perancangan Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical representative*.

3.1 Analisis Sistem yang Sedang Berjalan (*As-Is*) di Perusahaan Farmasi

Analisis sistem berjalan yang dilakukan merupakan hasil wawancara terhadap beberapa *medical rep* di beberapa perusahaan farmasi dan terhadap salah satu supervisor.

3.1.1 Analisis *Stakeholder* dan Perannya

Untuk menganalisis proses-proses yang terdapat pada sistem yang berjalan, maka perlu mengetahui *stakeholder* dan perannya terlebih dahulu. Tabel 7 menjelaskan *stakeholder* beserta peran dan tugasnya.

Tabel 7 Stakeholder dan Perannya

No	Stakeholder	Role, Task, Rule
1	Supervisor	<p><i>Role :</i></p> <p>Supervisor merupakan atasan dari <i>Medical rep.</i></p> <p><i>Tasks :</i></p> <ul style="list-style-type: none"> a. Merencanakan target penjualan, sumber daya, waktu dan biaya yang diperlukan. b. Memberikan arahan mengenai pekerjaan yang harus dikerjakan <i>medical rep.</i> c. Mengelola tugas dan target untuk <i>medical rep.</i> d. Memonitor capaian target dari <i>medical rep.</i> e. Mengevaluasi capaian target dari <i>medical rep.</i> <p><i>Rules :</i></p> <ul style="list-style-type: none"> -
2	<i>Medical representative</i>	<p><i>Role :</i></p> <p><i>Medical rep</i> mewakili perusahaan untuk <i>detailing</i> atau mempromosikan produknya kepada dokter.</p> <p><i>Tasks :</i></p> <ul style="list-style-type: none"> a. Membuat perencanaan kunjungan ke dokter b. Mencatat data dokter. c. Melakukan survei apotek untuk mengetahui besar potensi peresepan dokter. d. Mencatat kompetitor atau produk pesaing. e. Melakukan kunjungan ke dokter. f. Mencatat hasil kunjungan. g. Mencatat hasil transaksi. h. Membuat laporan kunjungan dan hasil transaksi <p><i>Rules :</i></p> <ul style="list-style-type: none"> a. <i>Medical rep</i> tidak boleh menjual obat secara langsung kepada tenaga medis. b. <i>Medical rep</i> dan perusahaan tidak boleh memaksa dan mengarahkan dokter untuk meresepkan obat tertentu kepada pasiennya
3	Dokter	<p><i>Role :</i></p> <p>Dokter merupakan pelanggan atau sasaran utama dari perusahaan farmasi, sebagai orang yang mewakili konsumen akhir yaitu pasien.</p> <p><i>Tasks :</i></p> <ul style="list-style-type: none"> a. Meresepkan obat perusahaan sesuai dengan yang di-<i>detailing</i> oleh <i>medical rep.</i>

	<p>b. Merekomendasikan obat perusahaan sesuai dengan yang di-detailing oleh <i>medical rep</i>.</p> <p><i>Rules :</i></p> <p>Dokter berhak mendapatkan <i>reward</i> dari perusahaan farmasi, sesuai dengan Permenkes dan Kode Etik <i>International Pharmaceutical Manufactures Group</i> (IPMG)[9].</p>
--	---

3.1.2 Analisis Bisnis Proses

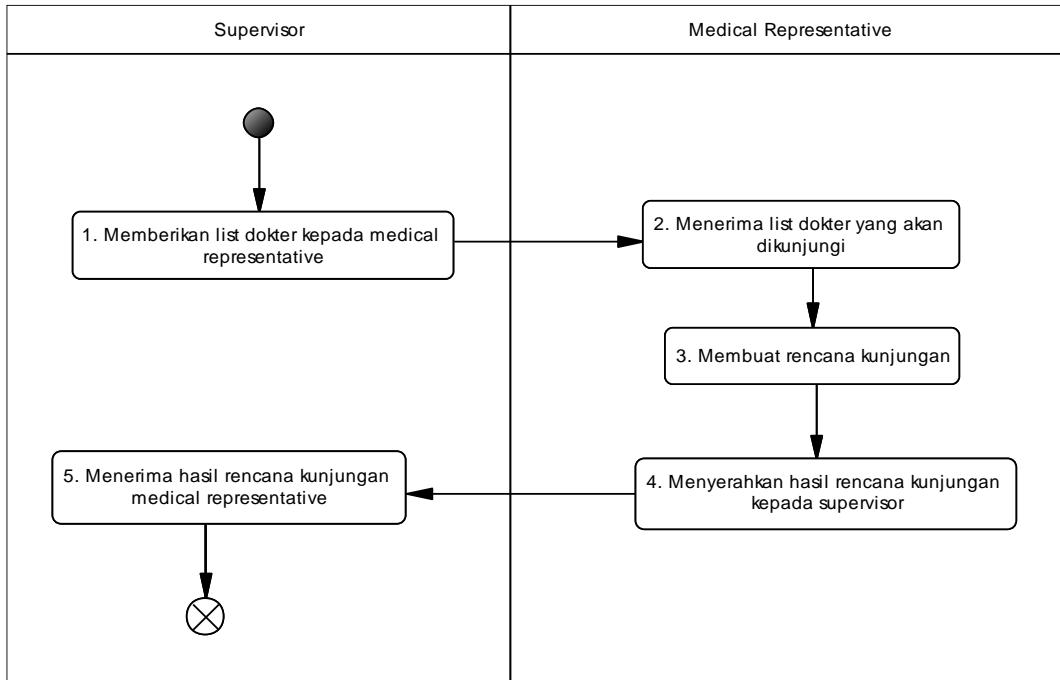
Dari hasil wawancara, secara garis besar terdapat dua aktor yang terlibat dalam sistem berjalan, yaitu *medical rep* dan supervisor. Peran dan tanggung jawab aktor-aktor tersebut sudah dijelaskan pada tabel 7, aktor pendukung atau *stakeholder* yang terlibat diantaranya yaitu, dokter dan apotek.

Proses pekerjaan atau kegiatan yang terdapat pada sistem yang sedang berjalan saat ini sebagai berikut:

Untuk *medical representative* proses yang dilakukan, yaitu:

1. Mengelola Rencana Kunjungan

Proses pengelolaan rencana kunjungan dilakukan oleh *medical rep*, untuk dapat mengelola waktu kunjungan berdasarkan target dari kunjungan dengan besar peresepan dari dokter dan jadwal praktik dokter. Bisnis proses pembuatan rencana kunjungan digambarkan pada Gambar 1 serta penjelasan proses pada Tabel 8.



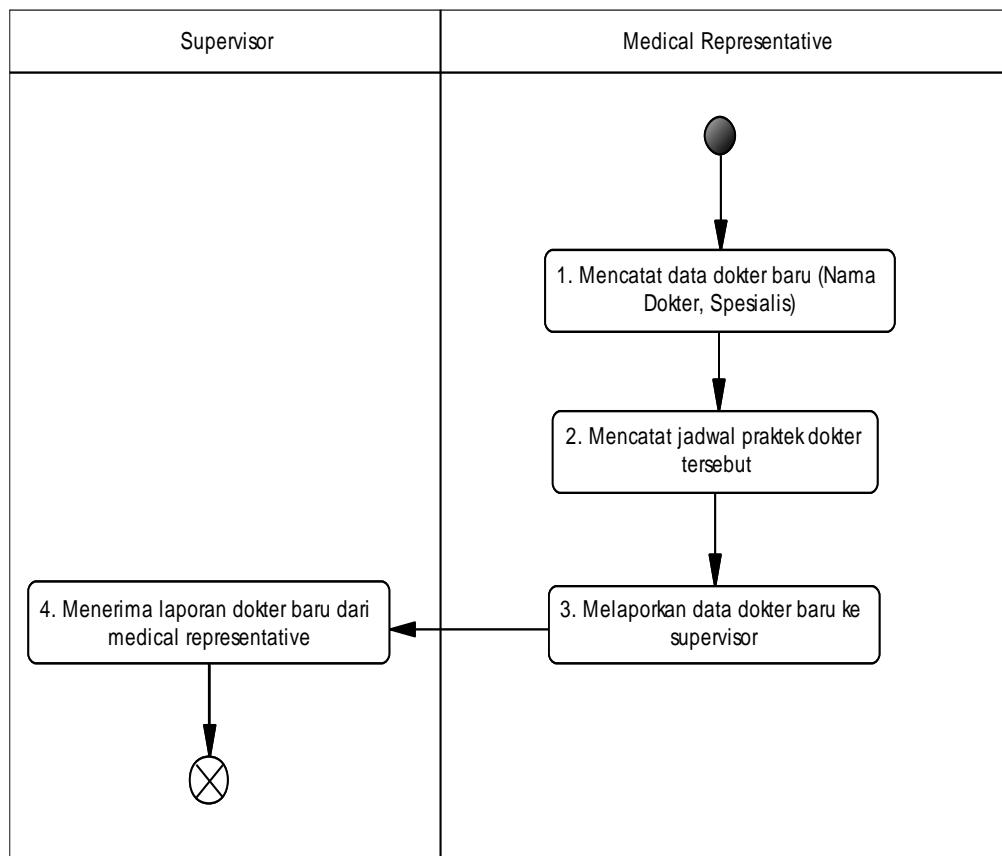
Gambar 1 Bisnis Proses Pengelolaan Rencana Kunjungan

Tabel 8 Penjelasan Bisnis Proses Pengelolaan Rencana Kunjungan

Nama Proses	Pengelolaan Rencana Kunjungan
No Proses	BP-AI-01
Business Process	<p>1. Memberikan <i>list</i> dokter kepada <i>medical rep</i></p> <p>Supervisor memberikan tugas kepada <i>medical rep</i> berupa <i>list</i> dokter yang harus dikunjungi. Didalam <i>list</i> dokter sudah terdapat informasi besar peresepan masing-masing dokter.</p> <p>2. Menerima <i>list</i> dokter yang akan dikunjungi</p> <p>3. Membuat rencana kunjungan</p> <p><i>Medical rep</i> membuat rencana kunjungan per harinya dalam satu bulan. Rencana kunjungan berisi dokter mana saja yang akan dikunjungi pada hari tersebut melihat berdasarkan besar potensi peresepan dokter dan jadwal praktik dokter tersebut.</p> <p>4. Menyerahkan hasil rencana kunjungan kepada supervisor</p> <p>Setelah selesai membuat rencana kunjungan, <i>medical rep</i> menyerahkan hasilnya kepada supervisor.</p> <p>5. Menerima hasil rencana kunjungan <i>medical rep</i></p>
Business Rules	-

2. Mencatat Data Dokter Baru

Proses pencatatan dokter baru dapat dilakukan ketika *medical rep* melakukan survei ke apotek atau melakukan kunjungan kepada dokter dan menemukan data dokter spesialis baru yang sesuai dengan produk yang di-*detailing* *medical rep*. Alur proses mencatat dokter baru digambarkan pada Gambar 2 serta penjelasan proses terdapat pada Tabel 9.



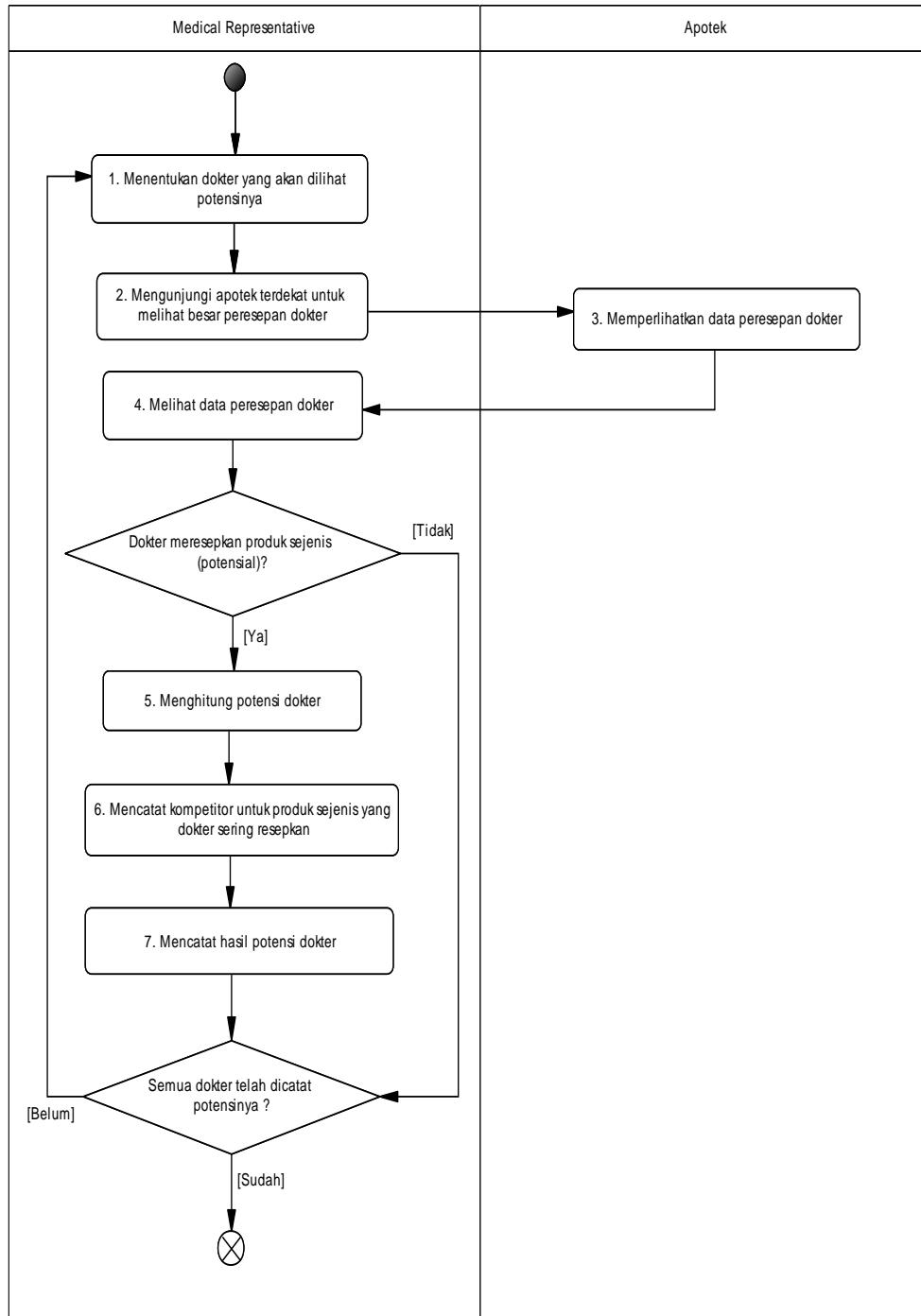
Gambar 2 Bisnis Proses Mencatat Data Dokter Baru

Tabel 9 Penjelasan Bisnis Proses Mencatat Data Dokter Baru

Nama Proses	Mencatat Data Dokter Baru
No Proses	BP-AI-02
<i>Business Process</i>	<p>1. Mencatat data dokter baru (nama dokter, spesialis)</p> <p><i>Medical rep</i> mencatat data dokter baru yang terdiri dari nama dokter beserta spesialis dokter tersebut.</p> <p>2. Mencatat jadwal praktek dokter</p> <p>Mencatat jadwal dokter agar <i>medical rep</i> mengetahui waktu praktek dokter sehingga <i>medical rep</i> mengetahui waktu yang tepat untuk mengunjungi dokter. Jadwal praktek yang dicatat adalah: nama dokter, hari praktek, dan jam praktek.</p> <p>3. Melaporkan data dokter baru ke supervisor</p> <p>4. Menerima laporan dokter baru dari <i>medical rep</i></p>
<i>Business Rules</i>	-

3. Mencatat Potensi Peresepan Dokter

Proses mencatat potensi dokter harus dilakukan oleh *medical rep* untuk mengetahui dokter yang berpotensi untuk meresepkan produk sejenis yang akan di-detailing oleh *medical rep*. Untuk mengetahui potensi peresepan tersebut *medical rep* harus melakukan survei ke apotek dan menghitung potensi peresepan dokter dari catatan peresepan dokter di apotek tersebut. Selain itu *medical rep* juga melakukan pencatatan kompetitor atau pesaing dari perusahaan berupa nama perusahaan atau nama produk pesaing. Lebih jelasnya alur proses ini digambarkan pada Gambar 3 dengan penjelasan pada Tabel 10.



Gambar 3 Bisnis Proses Pencatatan Potensi Peresepan Dokter

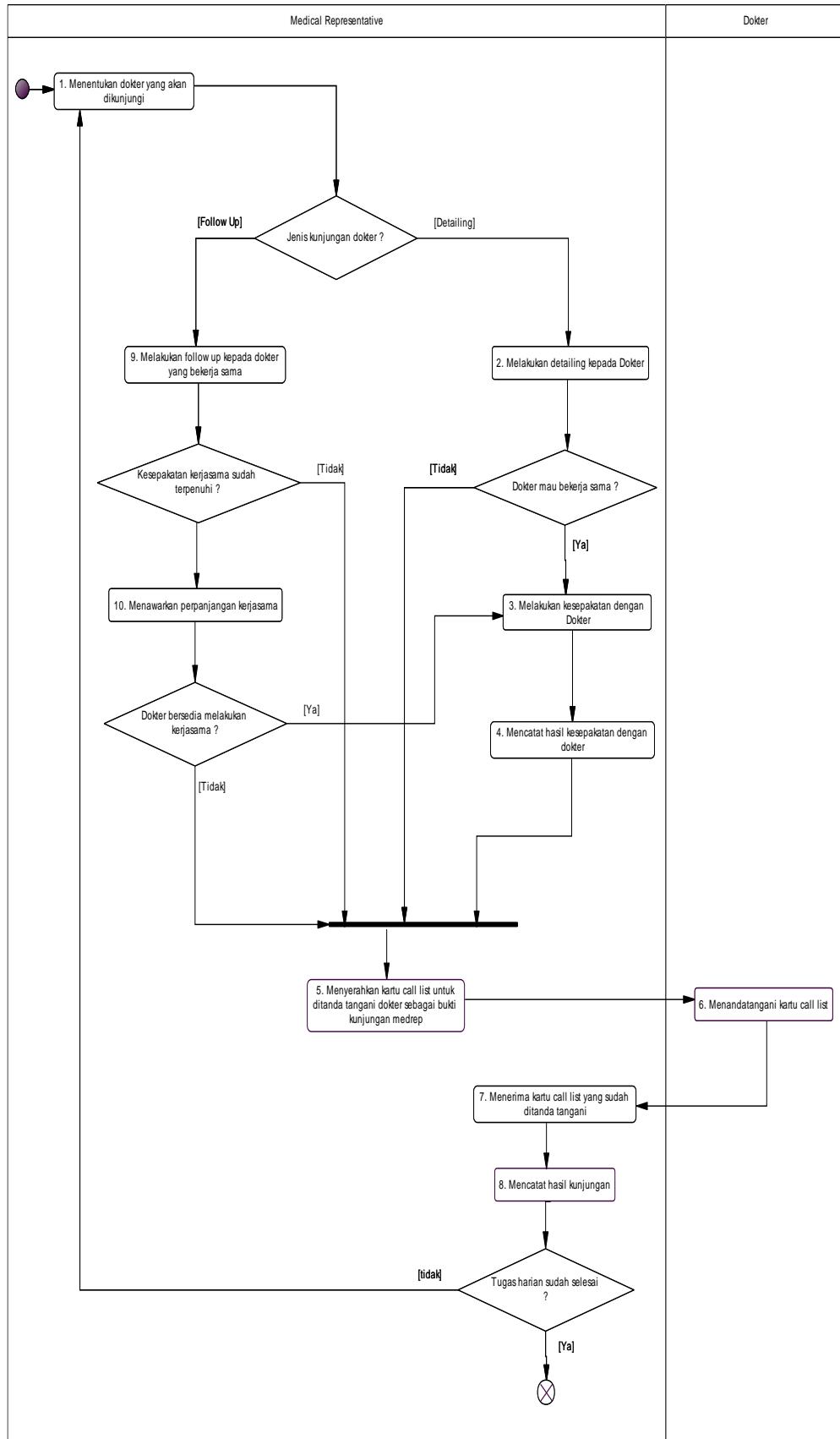
Tabel 10 Penjelasan Bisnis Proses Pencatatan Potensi Peresepan Dokter

Nama Proses	Proses Pencatatan Potensi Peresepan Dokter
No Proses	BP-AI-03
<i>Business Process</i>	<p>1. Menentukan dokter yang akan dilihat potensi peresepannya <i>Medical rep</i> memilih dokter dari <i>list</i> dokter yang ada untuk mengkualifikasi potensi peresepan dokter tersebut.</p> <p>2. Mengunjungi apotek untuk melihat besar peresepan dokter Setelah memilih dokter, <i>medical rep</i> melakukan kunjungan ke apotek yang dekat atau yang berada sesuai dengan alamat dokter tersebut bekerja, misalnya apotek rumah sakit dimana dokter bekerja. Hal ini dilakukan untuk mengetahui data peresepan dokter.</p> <p>3. Memperlihatkan data peresepan dokter Apotek kemudian memberikan akses kepada <i>medical rep</i> untuk melihat rekap peresepan dokter terkait.</p> <p>4. Melihat data peresepan dokter <i>Medical rep</i> melihat data peresepan tersebut, apabila dokter terkait pernah meresepkan produk sejenis maka dokter tersebut dikatakan berpotensi untuk meresepkan produk perusahaan dan dilakukan kunjungan ke dokter, sehingga bisa berlanjut ke proses selanjutnya (Proses No.5), sebaliknya apabila dokter tersebut tidak pernah meresepkan produk sejenis maka dokter tersebut dikatakan tidak berpotensi untuk meresepkan produk, sehingga tidak dilakukan kunjungan dan <i>medical rep</i> tidak perlu melakukan proses selanjutnya.</p> <p>5. Menghitung potensi peresepan dokter <i>Medical rep</i> melakukan perhitungan potensi peresepan dari data peresepan dokter yaitu produk sejenis dan jumlah produk yang diresepkan, kemudian data jumlah pasien perharinya, data jumlah hari dari jadwal praktek dokter selama seminggu.</p> <p>6. Mencatat kompetitor untuk produk sejenis yang dokter sering resepkan Pada proses ini <i>medical rep</i> melakukan pencatatan nama kompetitor beserta nama produknya.</p> <p>7. Mencatat hasil perhitungan potensi peresepan dokter Data dari proses 5 dan proses 6 kemudian dicatat sebagai hasil potensi peresepan dokter tersebut.</p>

<p><i>Business Rules</i></p>	<ol style="list-style-type: none"> 1. Data yang digunakan untuk mengetahui potensi peresepan dokter berupa data peresepan satu bulan terakhir. 2. Perhitungan potensi dokter digunakan selama satu bulan periode dimulai dari tanggal 1 hingga tanggal terakhir pada bulan tersebut. 3. Perhitungan potensi dokter menggunakan rumus sebagai berikut $\text{Besar Potensi Peresepan} = JHPD \times JM \times XPH \times \sum_{i=1}^n (XOS[i] \times HOS[i]) \text{ (hal. 19)}$
------------------------------	--

4. *Detailing* dan *Follow Up* Dokter

Proses *detailing* merupakan salah satu kegiatan promosi pada perusahaan farmasi, dan yang bertanggung jawab melakukan *detailing* yaitu *medical rep*. Tujuannya memperkenalkan produk kepada dokter dengan mengunjungi dokter tersebut baik ke rumah sakit tempatnya bekerja, rumah bersalin, tempat praktik dokter, dan sebagainya. Dengan *detailing* juga, *medical rep* berusaha agar dokter mau meresepkan produk perusahaan kepada pasien. *Follow up* dilakukan untuk memantau peresepan obat oleh dokter dan salah satu cara mempertahankan dokter sebagai pelanggan perusahaan farmasi. Alur proses digambarkan pada Gambar 4 dan penjelasan proses pada Tabel 11.

Gambar 4 Bisnis Proses *Detailing* dan *Follow Up* Dokter

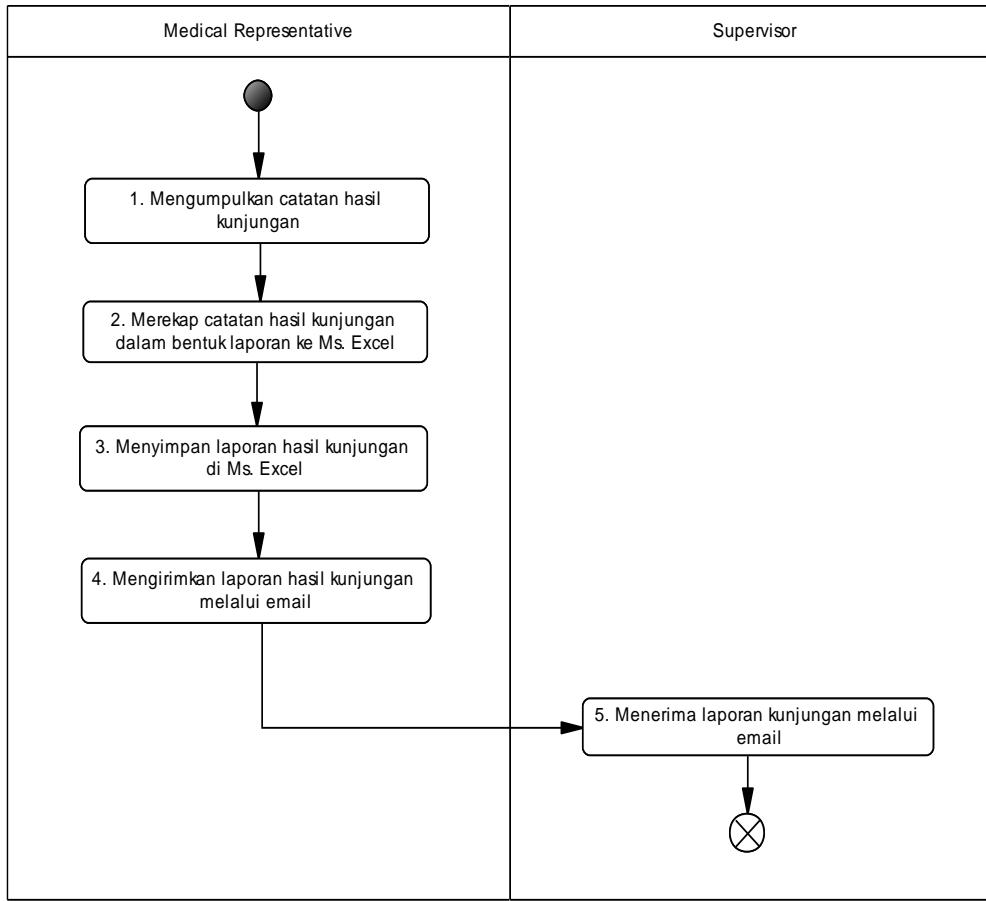
Tabel 11 Penjelasan Bisnis *Proses Detailing* dan *Follow Up* Dokter

Nama Proses	<i>Detailing</i> dan <i>Follow Up</i> Dokter
No Proses	BP-AI-04
Business Process	<p>1. Menentukan dokter yang akan dikunjungi</p> <p><i>Medical rep</i> menentukan terlebih dahulu dokter yang akan dikunjunginya.</p> <p>2. Melakukan <i>detailing</i> kepada dokter</p> <p>Jika dokter yang ditentukan tersebut jenis kunjungannya <i>detailing</i>, maka <i>medical rep</i> harus melakukan <i>detailing</i> ke dokter tersebut.</p> <p>3. Melakukan kesepakatan dengan dokter</p> <p>Dari <i>detailing</i> tadi, jika dokter tersebut tertarik untuk meresepkan obat yang ditawarkan <i>medical rep</i>, maka keduanya melakukan kesepakatan untuk jumlah resep yang harus dokter tersebut penuhi.</p> <p>4. Mencatat hasil kesepakatan dengan dokter</p> <p>Setelah melakukan kesepakatan, <i>medical rep</i> mencatat hasil kesepakatannya untuk dilaporkan kepada supervisor.</p> <p>5. Menyerahkan kartu <i>visit list</i> untuk ditanda tangani dokter sebagai bukti kunjungan <i>medical rep</i></p> <p>Setelah proses <i>detailing</i> selesai, maka <i>medical rep</i> menyerahkan kartu <i>visit list</i> untuk ditanda tangani dokter sebagai bukti kunjungan <i>medical rep</i> kepada dokter tersebut.</p> <p>6. Menandatangani kartu <i>visit list</i></p> <p>Proses ini dilakukan oleh dokter, dengan menandatangani pada kartu <i>visit list</i>, medrep mendapatkan bukti atas kunjungannya kepada dokter yang dikunjungi.</p> <p>7. Menerima kartu <i>visit list</i> yang sudah ditanda tangani</p> <p>8. Mencatat hasil kunjungan</p> <p><i>Medical rep</i> wajib menuliskan hasil kunjungannya untuk nanti dilaporkan kepada supervisor.</p> <p>9. Melakukan <i>follow up</i> kepada dokter yang bekerja sama</p> <p><i>Medical rep</i> melakukan <i>follow up</i>, jika dokter yang akan dikunjungi nya tersebut sudah mau bekerja sama untuk</p>

	<p>meresepkan produk <i>medical rep</i>. Melakukan <i>follow up</i> ini diantaranya menanyakan apakah kerja sama berjalan baik atau tidak, atau menanyakan apakah masih kurang mengerti tentang produknya atau sudah paham.</p> <p>10. Menawarkan perpanjangan kerja sama</p> <p>Jika dokter sudah memenuhi kesepakatan kerja sama, maka <i>medical rep</i> menawarkan kembali kepada dokter tersebut apakah akan meneruskan kerja samanya atau tidak</p>
<i>Business Rules</i>	Dokter yang ditentukan oleh <i>medical rep</i> untuk kunjungan merupakan dokter yang terdapat pada <i>list</i> tugas <i>medical rep</i> . Apabila terdapat dokter baru selain pada <i>list</i> , maka menjalankan bisnis proses Mencatat Dokter Baru (BP-AI-02).

5. Merekap Hasil Kunjungan

Proses perekapan hasil kunjungan dilakukan oleh *medical rep* berdasarkan catatan *medical rep* selama dilapangan yang direkap ulang ke Ms.Excel, hasil perekapan kunjungan adalah laporan kunjungan, yang berisi data-data dokter yang sudah dikunjungi dan hasil kunjungan dari setiap dokter. Proses rekap hasil kunjungan digambarkan pada Gambar 5 dan penjelasan proses pada Tabel 12.



Gambar 5 Bisnis Proses Rekap Hasil Kunjungan

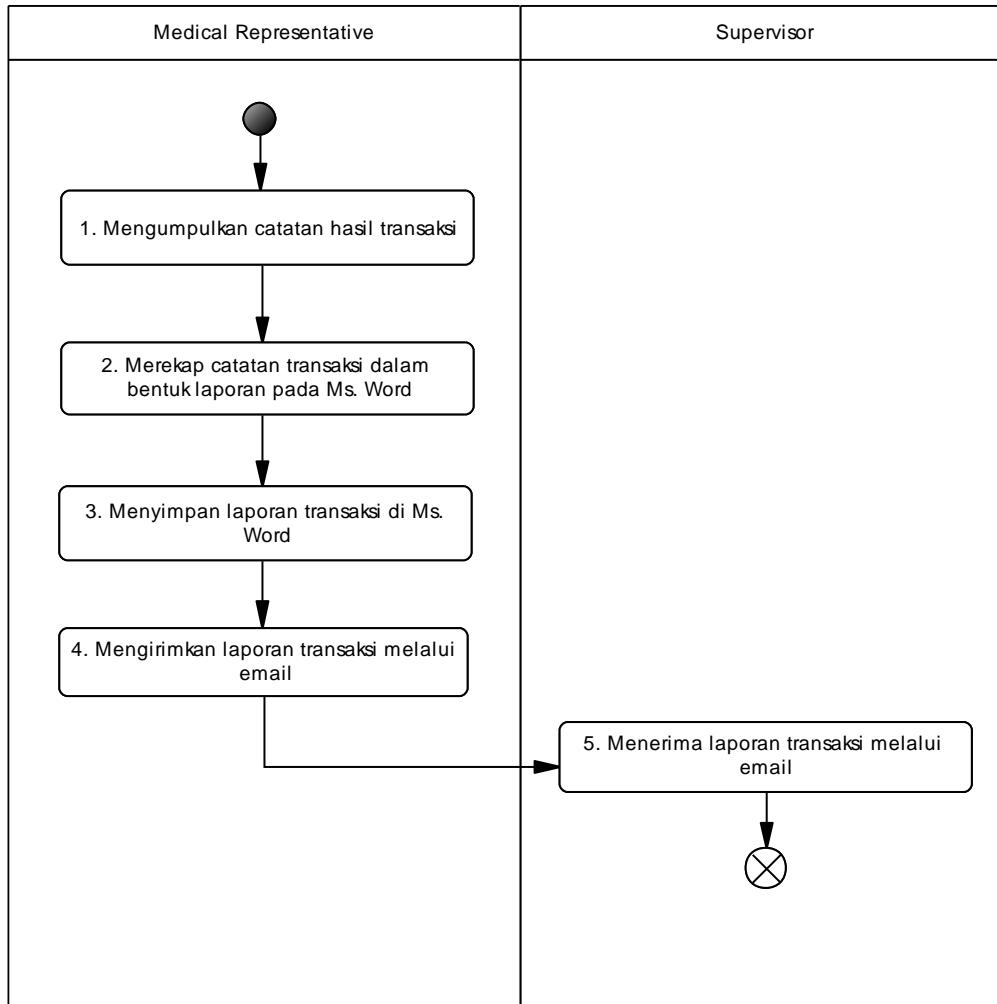
Tabel 12 Penjelasan Bisnis Proses Rekap Hasil Kunjungan

Nama Proses	Rekap Hasil Kunjungan
No Proses	BP-AI-05
Business Process	<p>1. Mengumpulkan catatan hasil kunjungan</p> <p><i>Medical rep</i> mengumpulkan catatan hasil kunjungan dari setiap dokter yang sudah dikunjunginya. Catatan hasil kunjungan berisi informasi bukti kunjungan yang berupa tanda tangan dokter yang dikumpulkan melalui kartu <i>visit list</i> dan <i>comment</i>, yaitu catatan mengenai kejadian selama kunjungan, penilaian <i>medical rep</i> terhadap sikap, karakter dan respon dokter yang dikunjungi, potensi peresepehan dokter, serta produk pesaing yang digunakan oleh dokter.</p> <p>2. Merekap catatan hasil kunjungan dalam bentuk laporan ke Ms. Excel</p> <p>Setelah mengumpulkan catatan hasil kunjungan, <i>medical rep</i> melakukan proses rekap hasil kunjungan dari catatan <i>medical</i></p>

	<p><i>rep</i> ke Ms.Excel, proses rekap ini menghasilkan laporan hasil kunjungan yang formatnya ditentukan oleh perusahaan.</p> <ol style="list-style-type: none"> 3. Menyimpan laporan hasil kunjungan di Ms. Excel <p>Proses ini dilakukan setelah <i>medical rep</i> selesai merekap ulang seluruh hasil kunjungan ke Ms. Excel.</p> <ol style="list-style-type: none"> 4. Mengirimkan laporan hasil kunjungan melalui <i>email</i> <p><i>Medical rep</i> kemudian mengirimkan dokumen laporan hasil kunjungan kepada supervisor melalui <i>email</i>.</p> <ol style="list-style-type: none"> 5. Menerima laporan kunjungan melalui <i>email</i> <p>Setelah dokumen laporan hasil kunjungan terkirim melalui <i>email</i>, maka supervisor menerima dokumen tersebut dari <i>email</i>.</p>
<i>Business Rules</i>	<ol style="list-style-type: none"> 1. Laporan hasil kunjungan dibuat per bulan. 2. Catatan hasil kunjungan yang dikumpulkan adalah catatan yang belum pernah dilaporkan oleh <i>medical rep</i> kepada supervisor, atau laporan kunjungan yang dibuat yaitu kunjungan yang sesuai dengan tugas dan target pada bulan atau periode tertentu. 3. <i>Medical rep</i> mengirimkan <i>email</i> kepada supervisor sesuai dengan jadwal yang ditentukan oleh supervisor.

6. Merekap Hasil Transaksi

Proses perekapan hasil transaksi dilakukan oleh *medical rep* berdasarkan transaksi-transaksi yang terjadi ketika melakukan kunjungan dengan dokter dan direkap ulang ke Ms. Word, hasil perekapan transaksi ini menghasilkan laporan hasil transaksi. Alur proses rekap hasil transaksi digambarkan pada Gambar 6 dan penjelasan proses pada Tabel 13.



Gambar 6 Bisnis Proses Rekap Hasil Transaksi

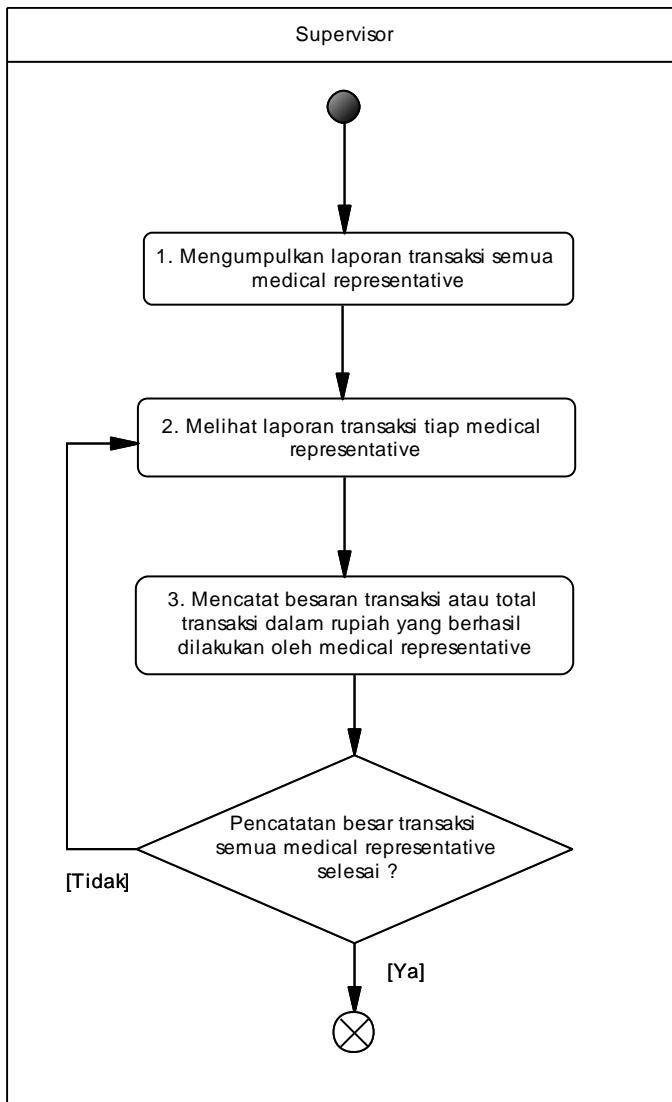
Tabel 13 Penjelasan Bisnis Proses Rekap Hasil Transaksi

Nama Proses	Rekap Hasil Transaksi
No Proses	BP-AI-06
Business Process	<p>1. Mengumpulkan catatan hasil transaksi</p> <p><i>Medical rep</i> mengumpulkan catatan hasil transaksi yang terjadi saat kunjungan dengan dokter. Catatan hasil transaksi berisi informasi kesepakatan kerja sama berupa besar target peresepan, yang dilakukan antara dokter dengan <i>medical rep</i> untuk menggunakan produk perusahaan yang telah di-<i>detailing</i>.</p> <p>2. Merekap catatan transaksi dalam bentuk laporan ke Ms. Word</p> <p>Setelah mengumpulkan seluruh catatan transaksi, <i>medical rep</i> melakukan proses rekap transaksi yang terjadi dari catatan</p>

	<p><i>medical rep</i> ke Ms. Word, proses rekap ini menghasilkan laporan transaksi yang formatnya ditentukan oleh perusahaan.</p> <ol style="list-style-type: none"> 3. Menyimpan laporan transaksi di Ms. Word <p>Proses ini dilakukan setelah <i>medical rep</i> selesai merekap ulang seluruh transaksi ke Ms. Word.</p> <ol style="list-style-type: none"> 4. Mengirimkan laporan transaksi melalui <i>email</i> <p><i>Medical rep</i> kemudian mengirimkan dokumen laporan transaksi kepada supervisor melalui <i>email</i>.</p> <ol style="list-style-type: none"> 5. Menerima laporan transaksi melalui <i>email</i> <p>Setelah dokumen laporan transaksi terkirim melalui <i>email</i>, maka supervisor menerima dokumen tersebut dari <i>email</i>.</p>
<i>Business Rules</i>	<ol style="list-style-type: none"> 1. Laporan transaksi dibuat per bulan. 2. Catatan transaksi yang dikumpulkan adalah catatan yang belum pernah dilaporkan oleh <i>medical rep</i> kepada supervisor, atau laporan transaksi yang dibuat yaitu transaksi yang sesuai dengan tugas dan target pada bulan atau periode tertentu. 3. <i>Medical rep</i> mengirimkan <i>email</i> kepada supervisor sesuai dengan jadwal yang ditentukan oleh supervisor.

Sedangkan untuk supervisor *medical rep* bertanggung jawab untuk melakukan *monitoring* pencapaian target dari *medical rep*.

Monitoring pencapaian target dilakukan oleh supervisor untuk melihat performa kerja dari setiap *medical rep*, dimana informasi ini dapat menjadi bahan evaluasi supervisor untuk tiap *medical rep* dalam timnya. Alur proses *monitoring* pencapaian target *medical rep* ini digambarkan pada Gambar 7 serta untuk penjelasan proses tersebut terdapat pada Tabel 14.



Gambar 7 Bisnis Proses *Monitoring Pencapaian Target Medical Representative*

Tabel 14 Penjelasan Bisnis Proses *Monitoring Pencapaian Target Medical Representative*

Nama Proses	Proses <i>Monitoring Pencapaian Target Medical Representative</i>
No Proses	BP-AI-07
Business Process	<p>1. Mengumpulkan laporan transaksi semua <i>medical rep</i></p> <p>Supervisor mengumpulkan semua laporan transaksi terakhir yang diberikan oleh <i>medical rep</i>.</p> <p>2. Melihat laporan transaksi tiap <i>medical rep</i></p>

	<p>Setelah melakukan pengumpulan laporan, supervisor kemudian melihat setiap laporan transaksi <i>medical rep</i>.</p> <p>3. Mencatat besaran transaksi atau total transaksi dalam rupiah yang berhasil dilakukan oleh <i>medical representative</i></p> <p>Supervisor melakukan pencatatan seluruh transaksi yang berhasil dilakukan oleh masing-masing <i>medical rep</i>, kemudian dihitung total transaksi yang didapatkan oleh setiap <i>medical rep</i>. Proses ini termasuk kedalam <i>monitoring</i> pencapaian target <i>medical rep</i> yang dapat digunakan untuk evaluasi penjualan dari tim yang dipegang oleh supervisor.</p>
Bussiness Rules	-

3.1.3 Business Rules

Dari bisnis proses yang sedang berjalan, diperoleh *business rules* yang dijelaskan pada Tabel 15.

Tabel 15 *Business Rules* Sistem As-Is

No	<i>Business Rules</i>	Proses Terkait
1	<p>1. Data yang digunakan untuk mengetahui potensi peresepan dokter berupa data peresepan satu bulan terakhir yang didapatkan dari apotek yang disurvei.</p> <p>2. Perhitungan potensi dokter digunakan selama satu bulan periode dimulai dari tanggal 1 hingga tanggal terakhir pada bulan tersebut.</p> <p>3. Perhitungan potensi dokter menggunakan rumus sebagai berikut:</p> $\text{Besar Potensi Peresepan} = JHPD \times JM \times XPH \times \sum_{i=1}^n (XOS[i] \times HOS[i]) \text{ (hal. 19)}$ <p>4. Pencatatan kompetitor berupa pencatatan nama perusahaan atau nama produk pesaing yang diresepkan oleh dokter.</p>	[BP-AI-03] Mencatat Potensi Peresepan Dokter
2	Urutan dokter yang akan dikunjungi ditentukan oleh <i>medical rep</i> sesuai <i>list</i> dokter yang diberikan oleh supervisor.	[BP-AI-04] Bisnis Proses <i>Detailing</i> dan <i>Follow Up</i> Dokter.
3	<p>1. Laporan hasil kunjungan dilakukan per bulan.</p> <p>2. Catatan hasil kunjungan yang dikumpulkan adalah catatan yang belum pernah dilaporkan oleh <i>medical rep</i> kepada Supervisor.</p>	[BP-AI-05] Bisnis Proses Rekap Hasil Kunjungan

4	1. Laporan hasil kunjungan dilakukan per bulan. 2. Catatan hasil kunjungan yang dikumpulkan adalah catatan yang belum pernah dilaporkan oleh <i>medical rep</i> kepada Supervisor.	[BP-AI-06] Bisnis Proses Rekap Hasil Transaksi
---	---	--

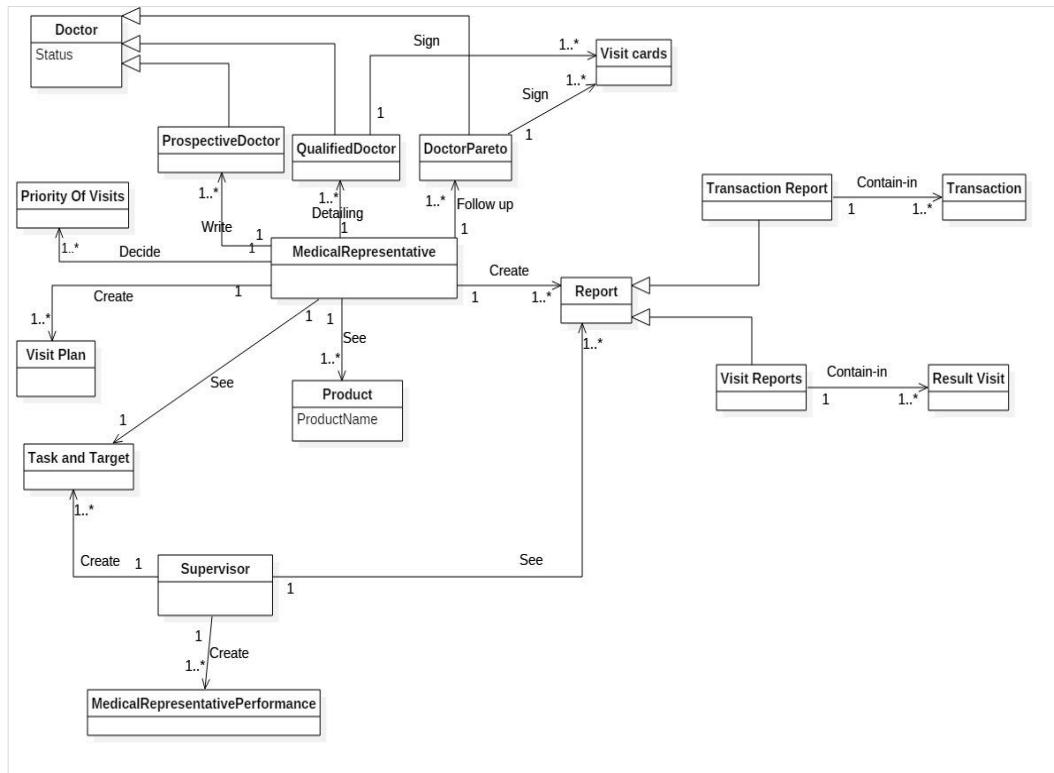
3.1.4 Domain Model

Dari hasil analisis, diperoleh beberapa data yang dibutuhkan pada bisnis proses yang sedang berjalan, antara lain:

1. Data *medical rep*
 - a. Nama *medical rep*
 - b. Nomor kontak *medical rep*
 - c. *Email medical rep*
 - d. Alamat *medical rep*
2. Data supervisor
 - a. Nama supervisor
 - b. Nomor kontak supervisor
 - c. *Email supervisor*
 - d. Alamat supervisor
3. Data dokter
 - a. Nama dokter
 - b. Spesialisasi dokter
 - c. Nomor kontak dokter
 - d. *Email dokter*
 - e. Jadwal praktek
4. Data jadwal praktek dokter
 - a. Hari praktek
 - b. Jam praktek
 - c. Alamat dokter
 - d. dokter
5. Data produk
 - a. Nama produk
 - b. Jenis obat: tablet, kapsul, botol, dan sebagainya.

- c. Cara kerja obat
 - d. Indikasi
 - e. Aturan pakai
 - f. Komposisi
 - g. Peringatan obat
 - h. Kontraindikasi
 - i. Efek samping
- 6. Data kunjungan
 - a. Dokter
 - b. Hasil kunjungan
 - c. Tanggal kunjungan
 - 7. Data transaksi
 - a. Dokter
 - b. Tanggal awal transaksi
 - c. Tanggal akhir transaksi
 - d. Besar transaksi

Berdasarkan bisnis proses dan *business rules* pada sistem yang sedang berjalan, diperoleh *domain model* pada Gambar 8.



Gambar 8 Domain Model Sistem yang Berjalan

3.2 Analisis *Requirement*

Setelah melakukan analisis pada sistem yang sedang berjalan, selanjutnya adalah menganalisis *requirement*. Analisis *requirement* pada sistem yang akan dikembangkan menggunakan metode *goals oriented requirements engineering* (GORE).

Goals Oriented Requirements Engineering (GORE)

Pada tahap *goals oriented requirements engineering*, analisis yang dilakukan adalah menentukan tujuan dan tugas-tugas dari setiap *stakeholder* yang ada pada sistem yang sedang berjalan, sehingga dapat menghasilkan *requirements* untuk mengembangkan aplikasi CRM berdasarkan tujuan-tujuan setiap *stakeholder* tersebut.

a. *Strategic dependency system As-Is*

Pada tahap awal penyusunan GORE, terlebih dahulu mengidentifikasi tujuan-tujuan *stakeholder* berdasarkan bisnis proses sistem yang sedang berjalan, yang digambarkan dengan *strategic dependency system as-is* pada Gambar 9.

Pada Gambar 9, terlihat bahwa ada tujuan yang ingin dicapai oleh *medical rep*, supervisor maupun tujuan keduanya. Selain tujuan, terdapat *softgoal* bersama yang diharapkan. Agar lebih jelas, tabel 16 menunjukkan tujuan-tujuan dari masing-masing *stakeholder* yang terlibat.

Tabel 16 Daftar *Goals* untuk Setiap *Stakeholder*

<i>Goal Medical Rep</i>	<i>Softgoal</i>	<i>Goal Supervisor</i>
<ol style="list-style-type: none"> 1. Kunjungan terprioritaskan dengan baik. 2. Kunjungan terencana dengan baik 3. Data dokter terkelola dengan baik 4. Hasil kunjungan tercatat dengan baik 5. Hasil transaksi tercatat dengan baik. 	<ol style="list-style-type: none"> 1. Pengklasifikasian dokter dengan cepat. 2. Laporan yang cepat dan tepat 	<ol style="list-style-type: none"> 1. Tugas dan target <i>medical rep</i> terkelola dengan baik. 2. Pencapaian target <i>medical rep</i> tercatat dengan baik 3. Data <i>medical rep</i> terkelola dengan baik

b. *Strategic performance goals of the stakeholder*

Setelah mengetahui tujuan-tujuan *stakeholder* pada sistem yang berjalan, tahap berikutnya adalah mengelompokan lebih detail tujuan-tujuan dari masing-masing *stakeholder* yang digambarkan dengan *strategic performance goals of stakeholder* pada Gambar 10. Tabel 17 menunjukkan tujuan-tujuan yang sudah dikelompokkan berdasarkan *stakeholder*.

Tabel 17 *Strategic performance goals of the stakeholder*

<i>Goal Medical Rep</i>	<i>Goal Supervisor</i>
<p><i>Hardgoal :</i></p> <ul style="list-style-type: none"> 1. Kunjungan terprioritaskan dengan baik. 2. Data dokter terkelola dengan baik. 3. Hasil kunjungan tercatat dengan baik. 4. Hasil transaksi tercatat dengan baik. 5. Membuat laporan dengan mudah. 6. Dapat melihat tugas dan target 7. Kunjungan terencana dengan baik. <p><i>Softgoal :</i> Pengklasifikasian dokter dengan cepat</p>	<p>1. Pencapaian target <i>medical rep</i> termonitor dengan baik.</p> <p>2. Laporan kunjungan termonitor dengan baik.</p> <p>3. Laporan transaksi termonitor dengan baik.</p> <p>4. Data <i>medical rep</i> terkelola dengan baik.</p> <p>5. Tugas dan target <i>medical rep</i> terkelola dengan baik.</p> <p><i>Softgoal :</i> Laporan yang cepat dan tepat</p>

c. *Strategic dependency system To-Be*

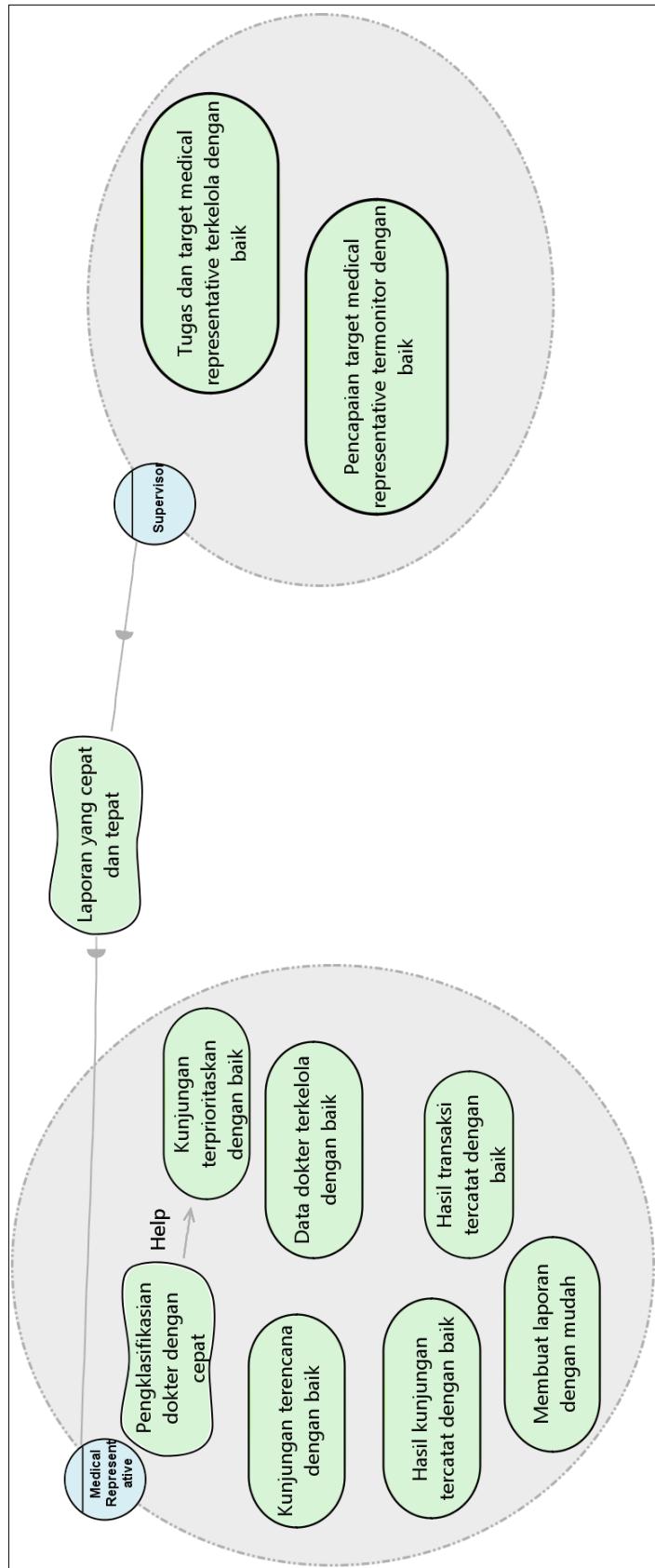
Tahap berikutnya menggambarkan tujuan yang dapat dicapai dengan adanya pengembangan aplikasi CRM, yang digambarkan dengan *strategic dependency system To-Be* pada Gambar 11. Pengembangan aplikasi CRM diharapkan menjadi solusi dari permasalahan-permasalahan yang timbul dari proses bisnis yang sedang berjalan. Untuk menunjang tujuan-tujuan agar tercapai dengan baik, terdapat *goal* dan aktor tambahan yang dijelaskan pada Tabel 18 dan 19.

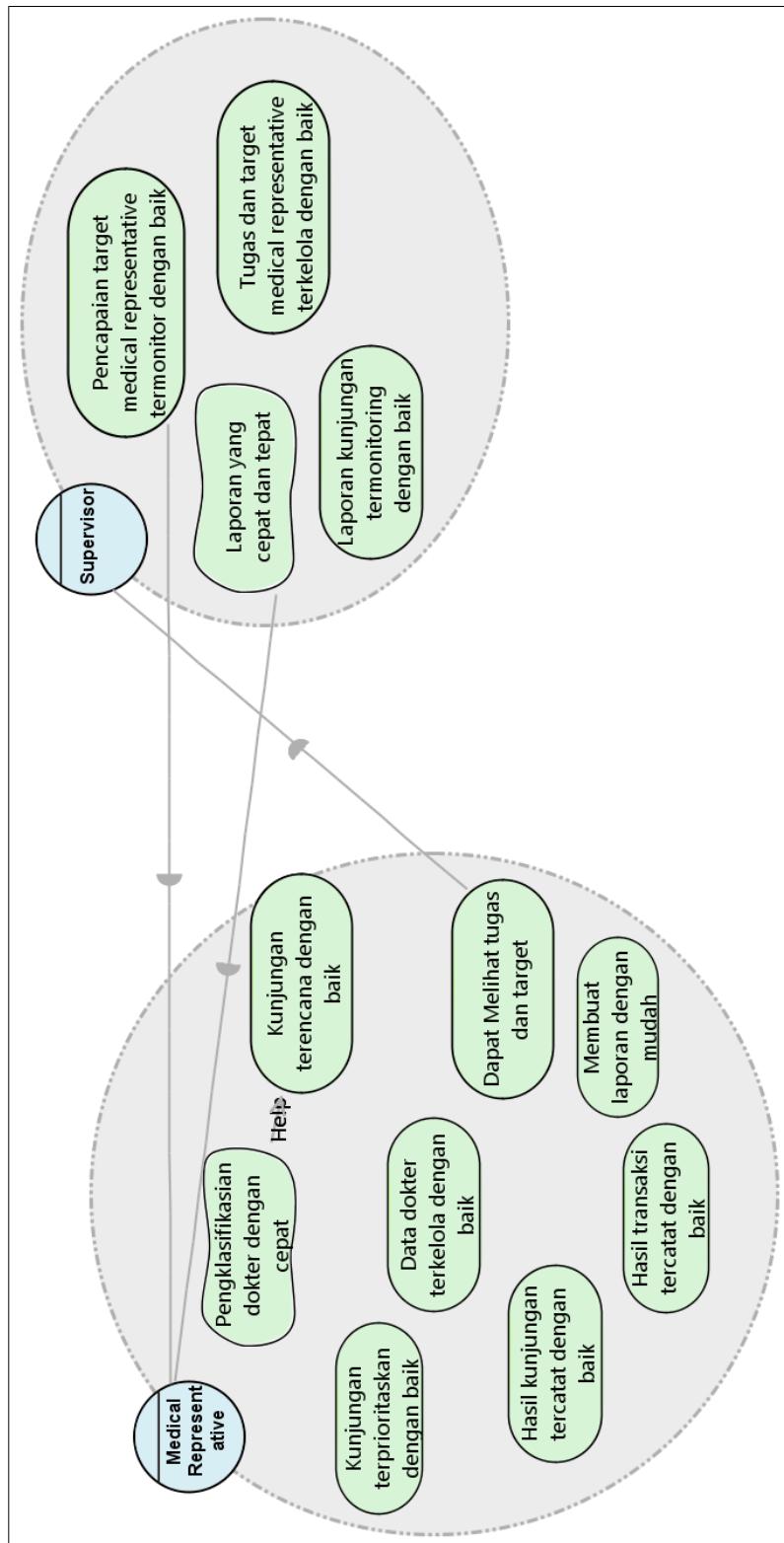
Tabel 18 *Strategic dependency system To-Be*

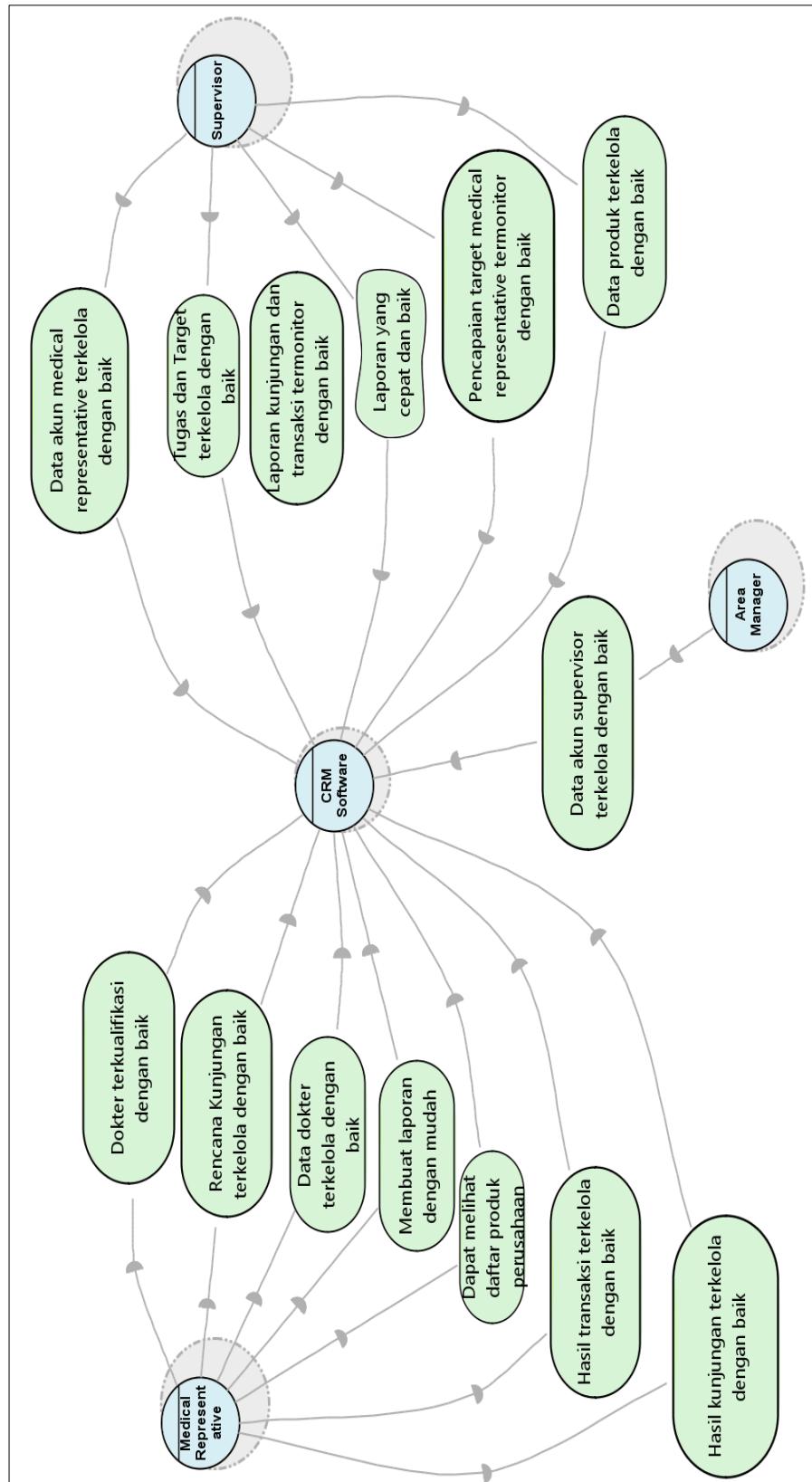
<i>Goal tambahan</i>	Keterangan
1. Data produk terkelola dengan baik	Untuk mengelola data produk yang ditugaskan kepada supervisor.
2. Data akun supervisor terkelola dengan baik	Untuk mengelola data akun supervisor

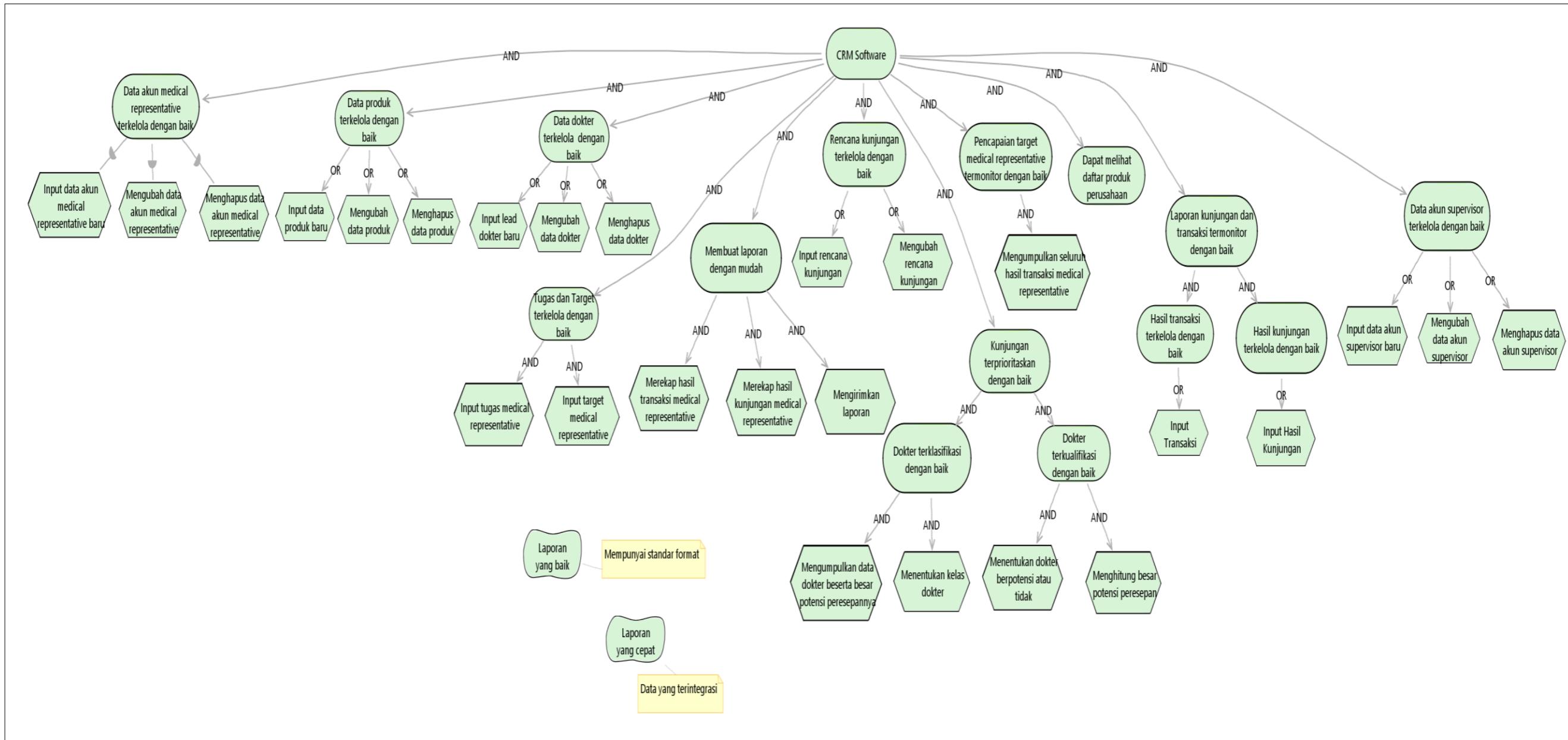
d. *Initial specification*

Pada tahap ini dijelaskan mengenai *requirements* pada aplikasi yang dikembangkan berdasarkan tujuan-tujuan *stakeholder* yang telah dijelaskan sebelumnya, digambarkan pada Gambar 12.

Gambar 9 *Strategic Dependency System As-Is*

Gambar 10 *Strategic Performance Goals Of Stakeholder*

Gambar 11 *Strategic Dependency System to-be*



Gambar 12 Initial Specification for CRM Software

Keterangan :

- = *Goals (functional requirement)*
- = *Specification*
- = *Softgoals (non functional requirement)*

Setelah digambarkan *initial specification*, berikut daftar *requirements* dan spesifikasi yang berhasil diperoleh berdasarkan analisis *requirements*, termasuk *initial specification* dijelaskan pada Tabel 19.

Tabel 19 *Functional Requirements* Aplikasi CRM

<i>Id</i>	<i>Functional Requirement</i>	<i>Specification</i>
[REQ-F-01]	Aplikasi menyediakan fitur untuk mengelola tugas dan target <i>medical representative</i>	<ol style="list-style-type: none"> 1. <i>Input</i> tugas <i>medical representative</i> 2. <i>Input</i> target <i>medical representative</i>
[REQ-F-02]	Aplikasi menyediakan fitur untuk menampilkan informasi pencapaian target <i>medical representative</i>	<ol style="list-style-type: none"> 1. Mengumpulkan seluruh hasil transaksi <i>medical representative</i>
[REQ-F-03]	Aplikasi menyediakan fitur untuk menampilkan laporan yang dikirimkan oleh <i>medical representative</i>	<ol style="list-style-type: none"> 1. Menerima laporan transaksi <i>medical representative</i> 2. Menerima laporan kunjungan <i>medical representative</i>
[REQ-F-04]	Aplikasi menyediakan fitur untuk mengklasifikasikan dokter	<ol style="list-style-type: none"> 1. Mengumpulkan data dokter beserta besar potensi peresepannya 2. Menentukan kelas dokter
[REQ-F-05]	Aplikasi menyediakan fitur untuk mengelola data dokter	<ol style="list-style-type: none"> 1. <i>Input</i> data <i>lead</i> dokter baru 2. Ubah data dokter 3. Hapus data dokter
[REQ-F-06]	Aplikasi menyediakan fitur untuk mengkualifikasi dokter	<ol style="list-style-type: none"> 1. Menentukan dokter berpotensi atau tidak 2. Menghitung besar potensi peresepan
[REQ-F-07]	Aplikasi menyediakan fitur untuk mengelola rencana kunjungan <i>medical representative</i>	<ol style="list-style-type: none"> 1. <i>Input</i> rencana kunjungan 2. Mengubah rencana kunjungan
[REQ-F-08]	Aplikasi menyediakan fitur untuk memasukkan informasi hasil kunjungan harian dengan dokter	<ol style="list-style-type: none"> 1. <i>Input</i> hasil kunjungan
[REQ-F-09]	Aplikasi menyediakan fitur untuk memasukkan informasi hasil transaksi (kesepakatan) dengan dokter	<ol style="list-style-type: none"> 1. <i>Input</i> hasil transaksi

[REQ-F-10]	Aplikasi menyediakan fitur untuk membuat laporan (<i>generate report</i>)	<ol style="list-style-type: none"> 1. Merekap hasil transaksi <i>medical representative</i>. 2. Merekap hasil kunjungan <i>medical representative</i>.
[REQ-F-11]	Aplikasi menyediakan fitur untuk menampilkan informasi <i>product</i> perusahaan.	
[REQ-F-12]	Aplikasi menyediakan fitur untuk mengelola produk perusahaan	<ol style="list-style-type: none"> 1. <i>Input</i> data produk baru 2. Ubah data produk 3. Hapus data produk
[REQ-F-13]	Aplikasi menyediakan fitur untuk mengelola akun <i>medical representative</i>	<ol style="list-style-type: none"> 1. <i>Input</i> data akun <i>medical representative</i> baru 2. Ubah data akun <i>medical representative</i> 3. Hapus data akun <i>medical representative</i>
[REQ-F-14]	Aplikasi menyediakan fitur untuk mengelola akun supervisor	<ol style="list-style-type: none"> 1. <i>Input</i> data akun supervisor baru 2. Ubah data akun supervisor 3. Hapus data akun supervisor

Berdasarkan batasan masalah poin 2, 3, 4 dan 5 serta ruang lingkup poin 4, 5 dan 6, diperoleh penerapan CRM yang dikaitkan dengan *functional requirement* dan *specification* dari hasil analisis *requirement* untuk aplikasi yang dibangun akan dijelaskan pada Tabel 20.

Tabel 20 Hubungan *Functional Requirement*, *Spesification*, dan Penerapan CRM

<i>Id Functional Requirement</i>	<i>Spesification</i>	Penerapan CRM
[REQ-F-01]	1. <i>Input</i> tugas <i>medical representative</i>	Operasional
	2. <i>Input</i> target <i>medical representative</i>	Operasional
[REQ-F-02]	1. Mengumpulkan seluruh hasil transaksi <i>medical representative</i>	Analitis
[REQ-F-03]	1. Menerima laporan transaksi <i>medical representative</i>	Operasional, Analitis

	2. Menerima laporan kunjungan <i>medical representative</i>	Operasional, Analitis
[REQ-F-04]	1. Mengumpulkan data dokter beserta besar potensi peresepannya	Strategis
	2. Menentukan kelas dokter	Analitis
[REQ-F-05]	1. <i>Input</i> data <i>lead</i> dokter baru	Strategis
	2. <i>Update</i> data dokter	Operasional
	3. Hapus data dokter	Operasional
[REQ-F-06]	1. Menentukan dokter berpotensi atau tidak	Strategis
	2. Menghitung besar potensi peresepan	Strategis
[REQ-F-07]	1. <i>Input</i> rencana kunjungan	Strategis
	2. Mengubah rencana kunjungan	Strategis
[REQ-F-08]	1. <i>Input</i> hasil kunjungan	Operasional
[REQ-F-09]	1. <i>Input</i> hasil transaksi	Operasional
[REQ-F-10]	1. Merekap hasil transaksi <i>medical representative</i> .	Operasional
	2. Merekap hasil kunjungan <i>medical representative</i> .	Operasional
[REQ-F-11]	1. Menampilkan informasi <i>product</i> perusahaan.	Operasional
[REQ-F-12]	1. <i>Input</i> data produk baru	Operasional
	2. Ubah data produk	Operasional
	3. Hapus data produk	Operasional

[REQ-F-13]	1. <i>Input</i> data akun <i>medical representative</i> baru	Operasional
	2. Ubah data akun <i>medical representative</i>	Operasional
	3. Hapus data akun <i>medical representative</i>	Operasional
[REQ-F-14]	1. <i>Input</i> data akun supervisor baru	Operasional
	2. Ubah data akun supervisor	Operasional
	3. Hapus data akun supervisor	Operasional

3.3 Analisis Aplikasi Sejenis

3.3.1 Insightly CRM

Insightly CRM merupakan suatu aplikasi CRM buatan perusahaan Insightly Inc. yang berbasis *mobile* maupun *website*. Namun, pada aplikasi *mobile* ataupun *website* tetap digunakan oleh pelaku bisnis yang sama. Aplikasi di akses melalui : <https://play.google.com/store/apps/details?id=com.insightly.droid&hl=in>. Berikut ini fitur-fitur yang terdapat pada Insightly CRM.

1. *Tasks*

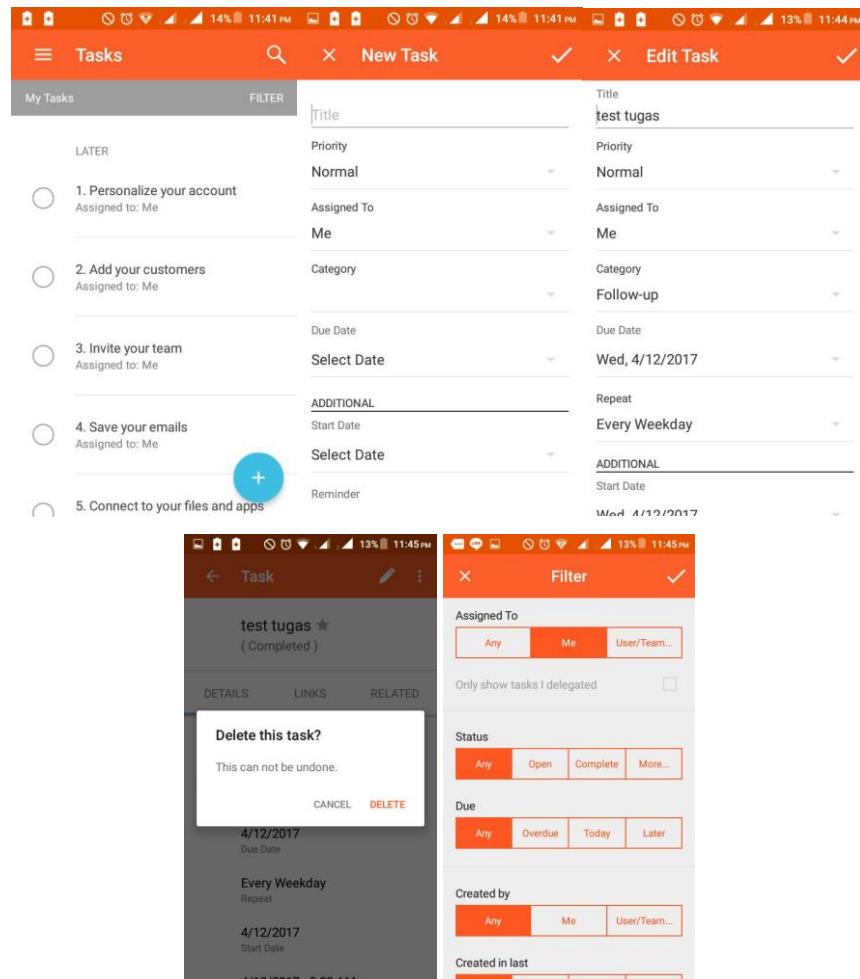
Fitur ini merupakan fitur pengelolaan tugas *sales* atau *medical rep* (jika dikaitkan dengan studi kasus). Pada fitur ini dapat menampilkan tugas yang telah ditambahkan, menambahkan tugas, menandai tugas yang telah diselesaikan, menghapus tugas, mem-filter tugas dan *reminder* tugas.

Filter tugas ini, yaitu:

- a. *Filter* berdasarkan *assign* (*any, me, user/team*).
- b. *Filter* berdasarkan status (*any, open, complete, not started, in progress, waiting, deferred*).
- c. *Filter* berdasarkan tanggal berakhirnya tugas (*any, overdue, today, later*).
- d. *Filter* berdasarkan penulis tugas (*any, me, user/team*).
- e. *Filter* berdasarkan waktu pembuatan tugas terakhir (*any, 24 hours, 7 days, 30 days*).

Data yang dicatat pada fitur ini:

- a. Judul tugas
- b. Prioritas tugas (*low, normal, high*)
- c. Orang yang akan mengerjakan tugas tersebut
- d. Kategori tugas (*Email, follow-up, get started, meeting, phone visit, to-do*)
- e. Batas terakhir tugas.
- f. *Reminder.*
- g. *Progress*
- h. Status (*not started, in progress, waiting, completed, deferred*).
- i. *Permissions (public, private)*.



Gambar 13 Tampilan Fitur Task pada Aplikasi Insightsly CRM

2. *Contacts*

Fitur ini merupakan fitur untuk mengelola *customer* yang sudah berhasil bekerja sama. Jika dikaitkan dengan studi kasus, fitur ini mengelola data dokter yang sudah berhasil bekerja sama dengan *medical rep*. Pada fitur ini dapat menampilkan kontak yang telah ditambahkan, menambah kontak, *edit* kontak dan *filter* kontak.

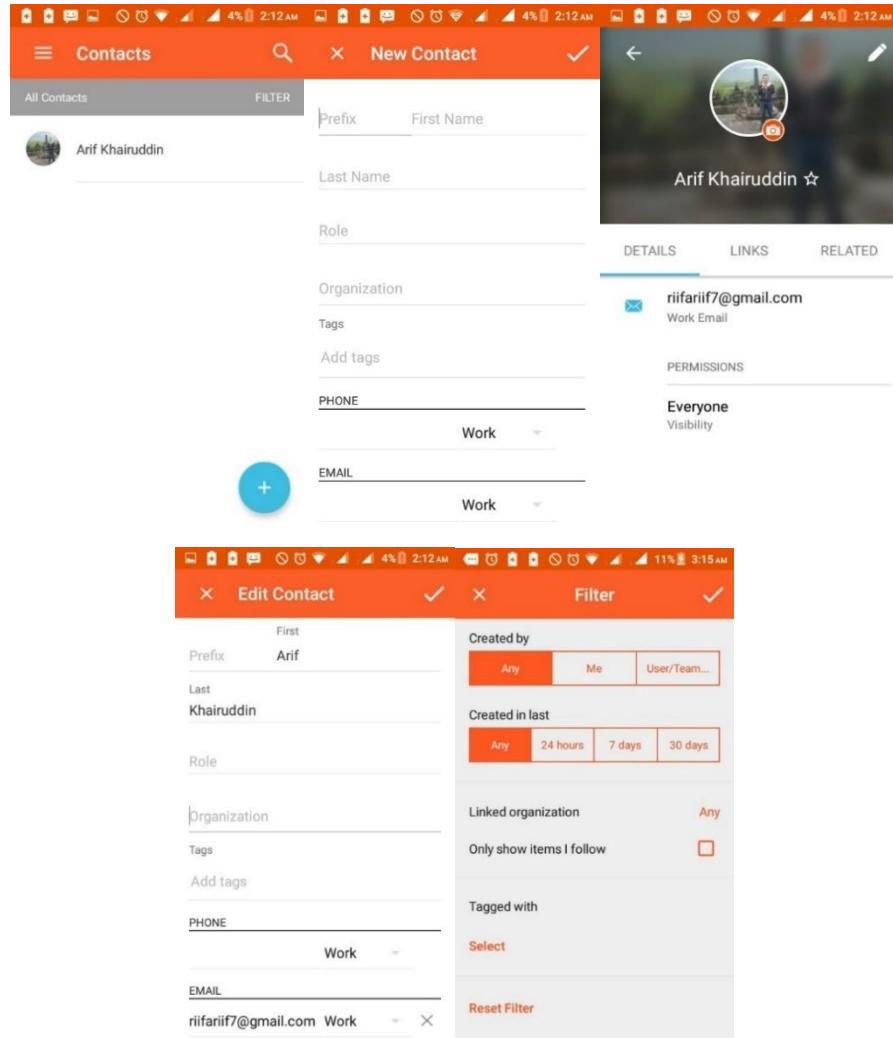
Filter kontak ini, yaitu:

- a. *Filter* berdasarkan penulis kontak (*any, me, user/team*).
- b. *Filter* berdasarkan waktu pembuatan kontak terakhir (*any, 24 hours, 7 days, 30 days*).

Data yang dicatat untuk keperluan fitur ini adalah:

- a. *Firstname*
- b. *Lastname*
- c. *Role*
- d. *Organization*
- e. *Tags*
- f. Nomor telepon
- g. *Email*.
- h. Alamat *website*
- i. Alamat perusahaan

Tampilan fitur *Contacts* pada aplikasi insightly CRM yang digambarkan pada Gambar 14.



Gambar 14 Tampilan Fitur *Contact* pada Aplikasi Insightsly CRM

3. *Organizations*

Pada fitur ini merupakan fitur untuk mengelola data perusahaan yang berhasil bekerja sama dengan *sales*. Jika dikaitkan dengan studi kasus, fitur ini mengelola data rumah sakit yang berhasil ditemukan oleh *medical rep*. Pada fitur ini dapat menampilkan *organization* yang sudah ditambahkan, menambah *organization*, edit *organization*, hapus *organization* dan filter *organization*.

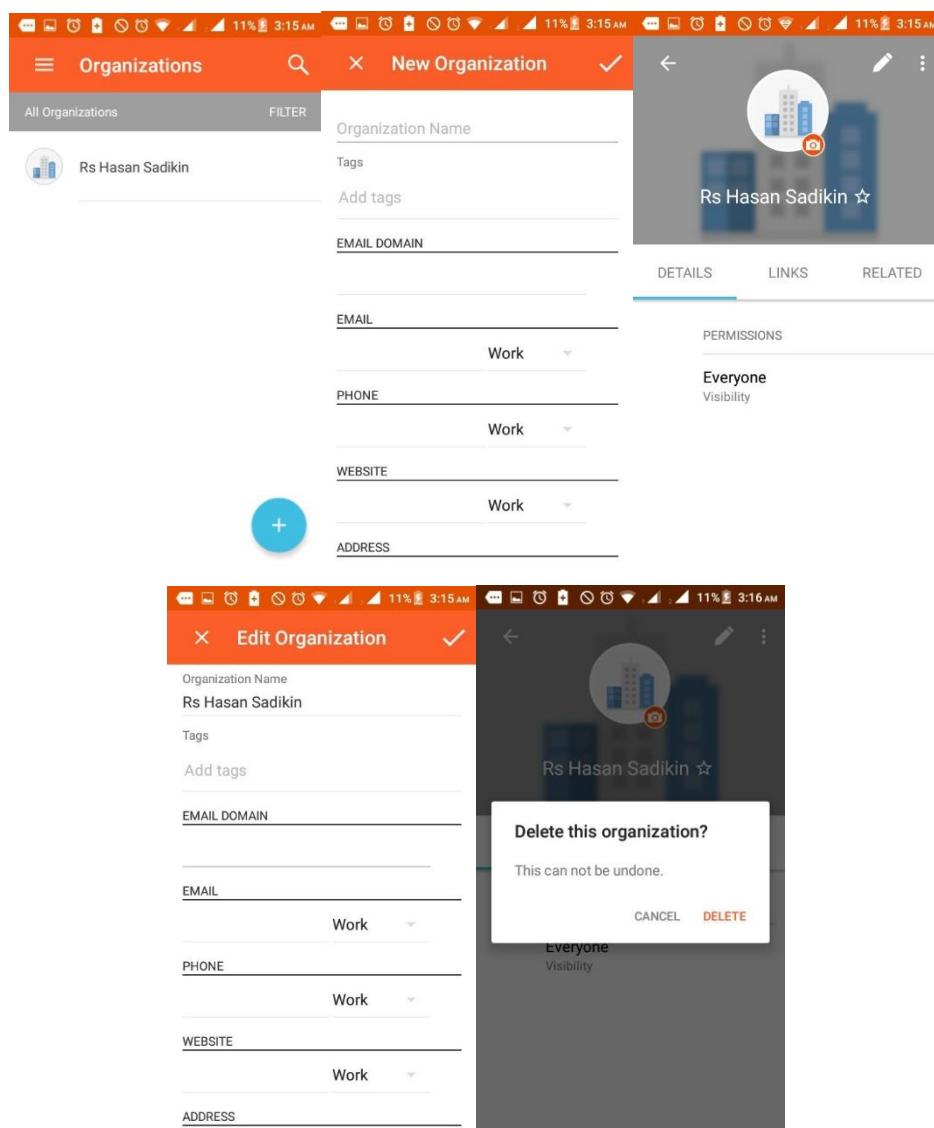
Filter organizations ini, yaitu :

- Filter* berdasarkan penulis kontak (*any, me, user/team*).
- Filter* berdasarkan waktu pembuatan kontak terakhir (*any, 24 hours, 7 days, 30 days*).

Data yang dicatat untuk keperluan fitur ini adalah :

- a. Nama *organization*
- b. *Tags*
- c. *Email organization*
- d. Telepon *organization*
- e. Alamat *website organization*
- f. Alamat *organization*

Berikut tampilan fitur *Organizations* pada aplikasi insightly CRM



Gambar 15 Tampilan Fitur *Organization* pada Aplikasi Insightly CRM

4. *Opportunities*

Pada fitur ini merupakan fitur untuk mengelola *customer* yang belum bekerja sama tetapi berpotensi untuk bekerja sama dengan *sales*. Jika dikaitkan dengan studi kasus, fitur ini mengelola data dokter yang belum bekerja sama dengan *medical rep*, namun memiliki potensi untuk kerja sama. Pada fitur ini dapat menampilkan *opportunities* yang telah ditambahkan, menambahkan *opportunities*, *edit opportunities*, *delete opportunities* dan *filter opportunities*.

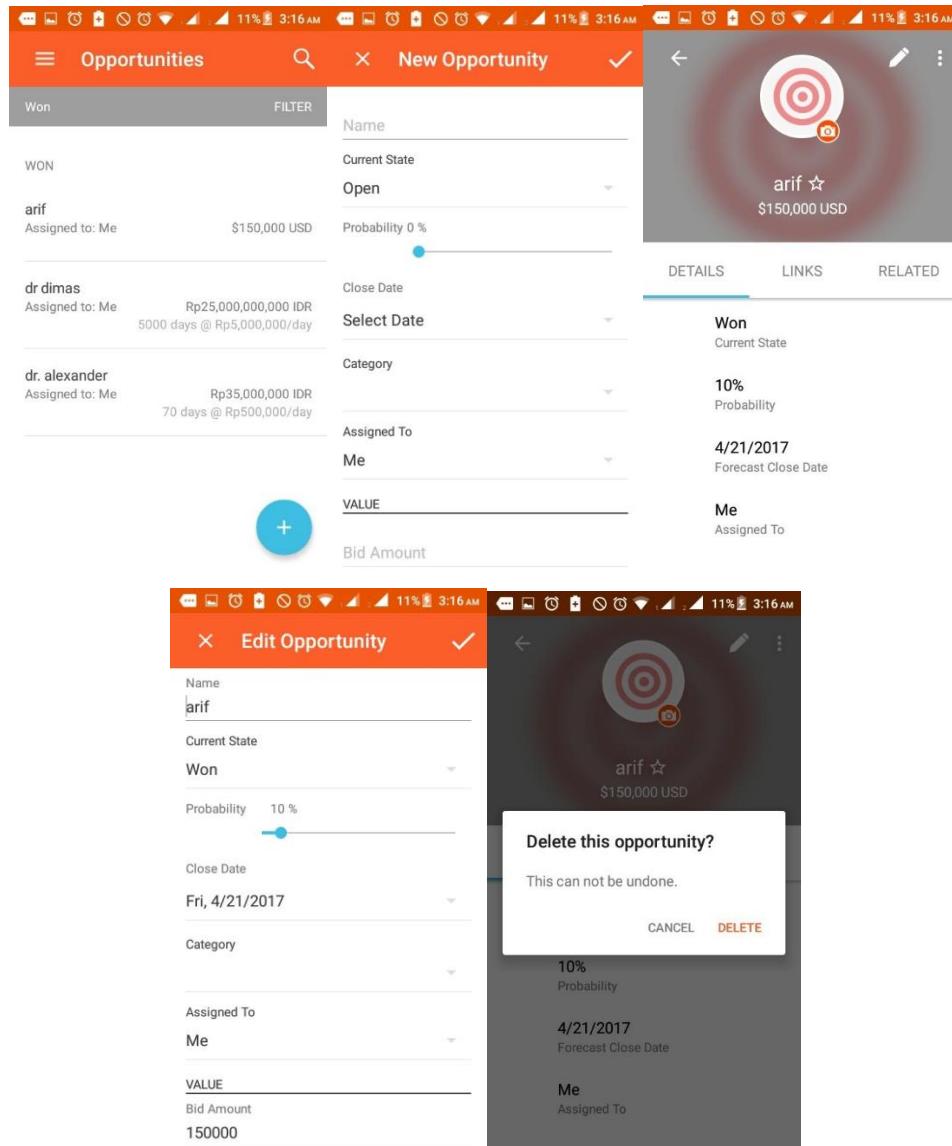
Filter opportunities ini, yaitu:

- a. *Filter* berdasarkan *assign* (*any, me, user/team*).
- b. *Filter* berdasarkan status (*any, open, complete, not started, in progress, waiting, deferred*).
- c. *Filter* berdasarkan penulis tugas (*any, me, user/team*).
- d. *Filter* berdasarkan waktu pembuatan tugas terakhir (*any, 24 hours, 7 days, 30 days*).

Data yang dibutuhkan pada fitur ini:

- a. Nama *opportunities*
- b. Status terakhir (*Open, Suspended, Lost, Won, Abandoned*)
- c. *Probability*
- d. *Close Date*
- e. *Category*
- f. Orang yang bertanggung jawab terhadap *opportunities* tersebut.
- g. Besar nilai transaksi jika berhasil

Berikut tampilan fitur *Opportunities* pada aplikasi insightly CRM:

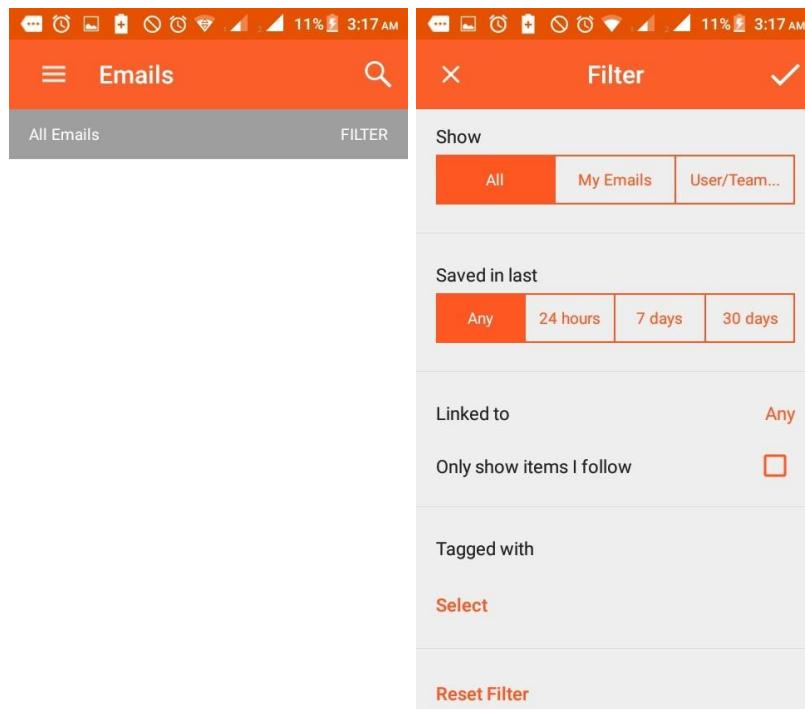


Gambar 16 Tampilan Fitur *Opportunities* pada Aplikasi Insightly CRM

5. *Emails*

Pada fitur ini merupakan fitur untuk mengelola *email* yang tersinkronisasi dengan *email* yang didaftarkan ke aplikasi. Fitur ini menampilkan *email* yang diterima dan *filter email*. Sejauh ini hanya dua yang dapat diidentifikasi, karena setelah melakukan uji coba, tidak diketahui cara untuk menambahkan *email* ataupun mensinkronisasikan *email* dengan *email* yang didaftarkan.

Berikut tampilan fitur *Emails* pada aplikasi insightly CRM



Gambar 17 Tampilan Fitur *Emails* pada Aplikasi Insightly CRM

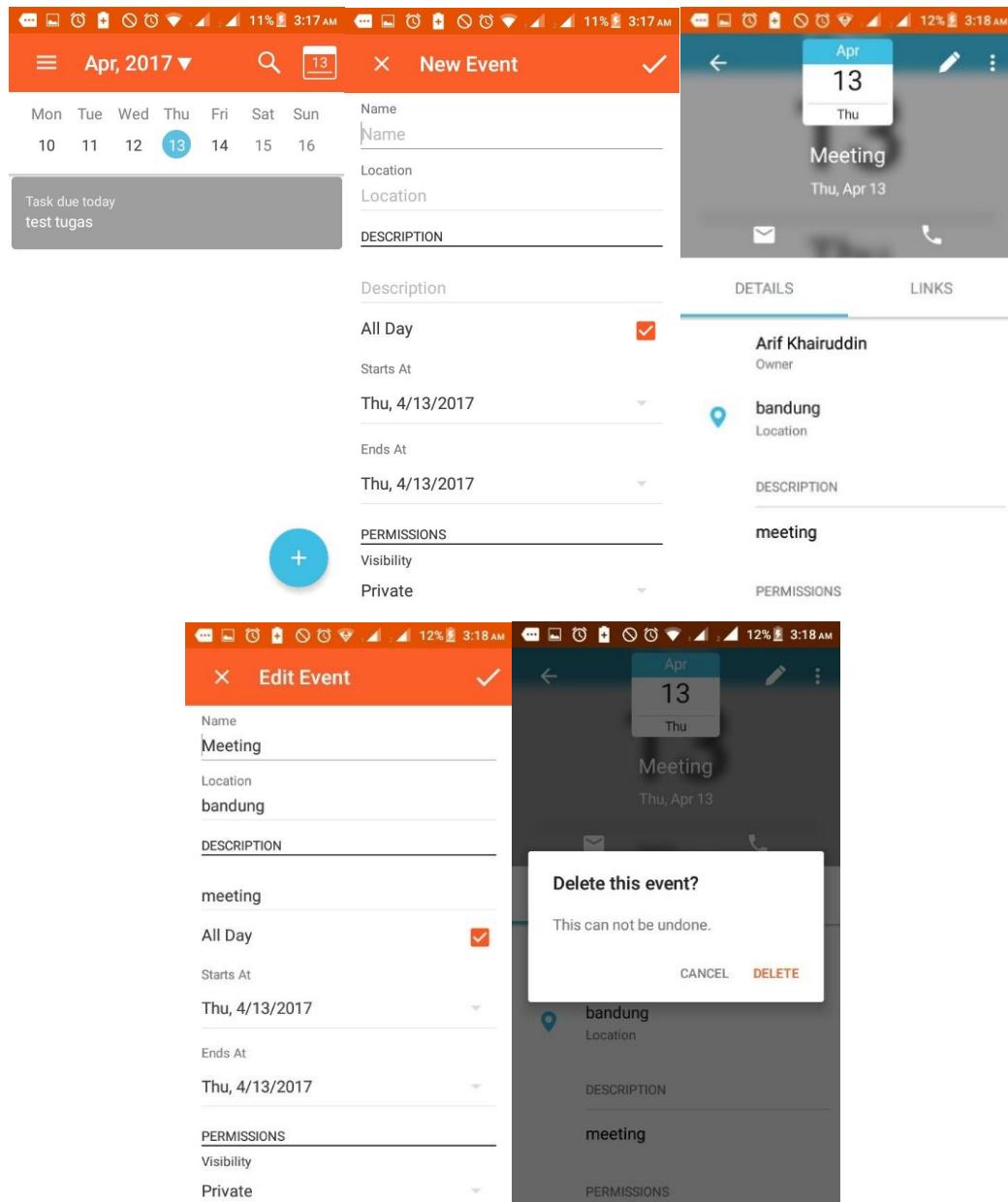
6. *Calendar*

Pada fitur ini merupakan fitur untuk mengelola agenda/jadwal. Tugas yang dibuat pada fitur *tasks* sudah terintegrasi dengan fitur *calendar*. Fitur ini dapat menampilkan *event* atau tugas yang sudah dibuat sebelumnya berdasarkan tanggal, menambahkan *event* atau tugas, *edit event* atau tugas, dan hapus *event* atau tugas.

Data yang dibutuhkan untuk fitur ini adalah (khususnya untuk *event*):

- a. Nama *event*
- b. Lokasi *event*
- c. Deskripsi *event*
- d. Waktu mulai *event*
- e. Waktu akhir *event*

Berikut tampilan fitur *Calendar* pada aplikasi insightly CRM



Gambar 18 Tampilan Fitur *Calendar* pada Aplikasi Insightly CRM

Setelah menganalisis dan mencoba fitur dari aplikasi Insightly CRM *mobile*, fitur-fitur yang terdapat didalamnya mempunyai kelebihan dan kekurangan masing-masing, yang dijelaskan pada Tabel 21.

Tabel 21 Tabel Hasil Analisis Aplikasi Insightly CRM

No	Menu Aplikasi	Specification	Jenis CRM	Kelebihan	Kekurangan
1	<i>Tasks</i>	1. Menyimpan data tugas	Operasional	Dapat menyimpan data tugas dan terintegrasi dengan fitur <i>calendar</i>	1. Tidak bisa menandai <i>opportunities</i> pada penambahan tugas. Karena jika terkait dengan studi kasus, tugas harus berkaitan dengan data dokter agar <i>medical rep</i> mengetahui tugas hariannya, dokter mana saja yang harus dikunjunginya.
2	<i>Contacts</i>	1. Menyimpan data <i>contacts</i>	Operasional	Dapat menyimpan data <i>contacts</i>	1. Jika ada data <i>opportunities</i> yang berhasil <i>deal</i> , data <i>contacts</i> harus dituliskan ulang pada fitur <i>contact</i> ini.
3	<i>Organizations</i>	1. Menyimpan data <i>organization/ perusahaan</i>	Operasional	Dapat menyimpan data <i>organization/ perusahaan</i>	
4	<i>Opportunities</i>	1. Menyimpan data <i>opportunities</i>	Operasional	Dapat menyimpan data <i>opportunities</i> .	1. Data <i>opportunities</i> yang terdapat pada aplikasi, tidak sesuai dengan studi kasus yang diangkat. 2. Data <i>opportunities</i> tidak menyimpan data <i>history</i> hasil

					kunjungan sebelumnya. 3. Jika sudah berhasil bekerjasama, data <i>opportunities</i> tidak otomatis berpindah ke <i>contacts</i> .
5	<i>Emails</i>	1. Mengirim email kepada <i>customer</i>	Operasional	-	1. Kurang informasi mengenai penambahan <i>email</i> , sehingga <i>user</i> kurang bisa menggunakan fitur ini.
6	<i>Calendar</i>	1. Menyimpan data <i>event</i> dan <i>task</i> .	Strategis	1. Dapat menyimpan data <i>event</i> dan <i>task</i> dengan cara pemilihan tanggal yang <i>simple</i> , sehingga membuat <i>user</i> menjadi lebih mudah untuk menggunakananya. 2. Sudah terintegrasi dengan pemilihan tanggal untuk fitur <i>tasks</i> .	

3.3.2 Base CRM

Base CRM merupakan suatu aplikasi CRM dibuat oleh perusahaan Base CRM (Future Simple) berbasis Android yang ditujukan untuk membantu pengelolaan penjualan berbagai jenis perusahaan, artinya aplikasi bisa digunakan oleh siapapun yang berperan sebagai tenaga penjualan. Aplikasi dapat diakses melalui : <https://play.google.com/store/apps/details?id=com.futuresimple.base&hl=in>.

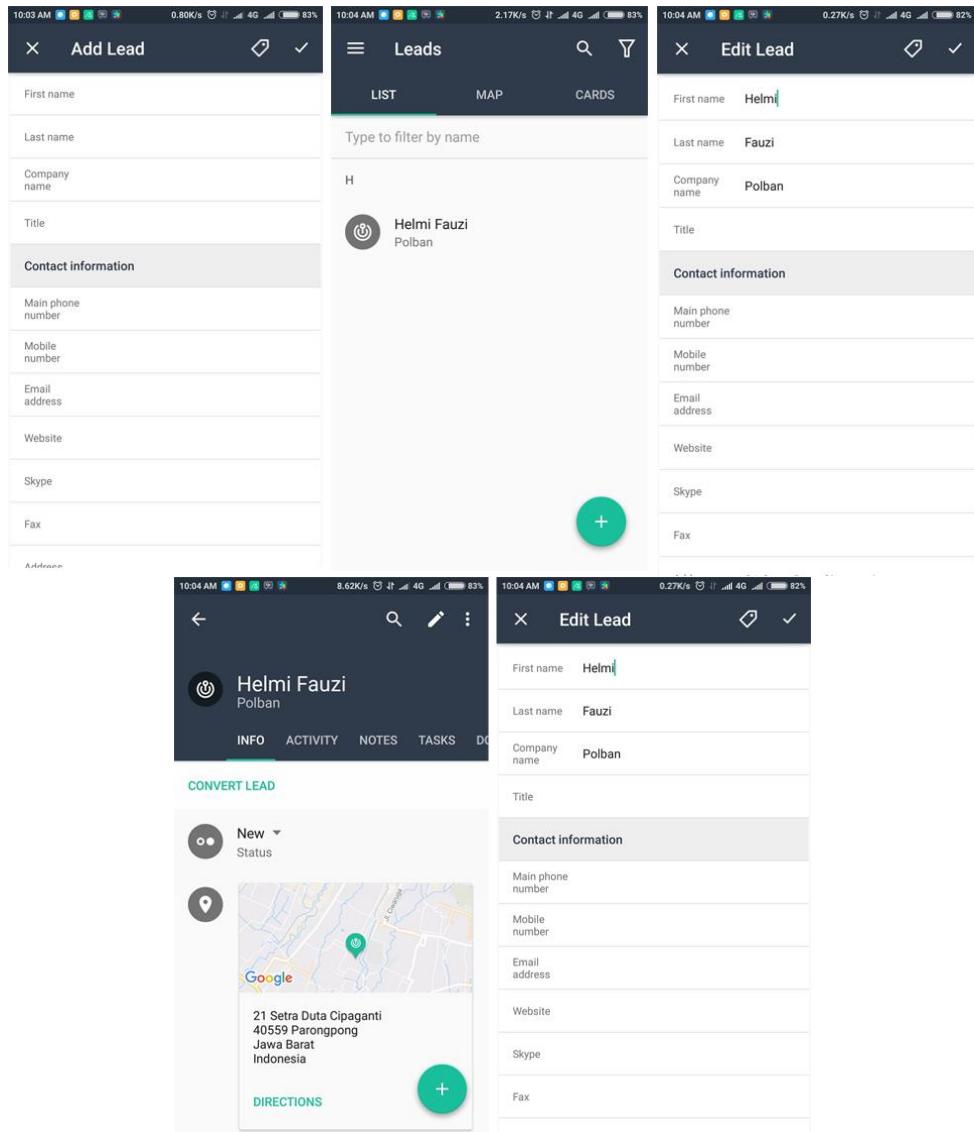
Fitur-fitur yang terdapat pada Base CRM diantaranya:

1. *Leads*

Fitur ini merupakan fitur untuk mengelola data calon *customer*. Jika dikaitkan dengan studi kasus tentunya fitur ini sangat *relevant* dengan pengelolaan data dokter. Fitur ini dapat menampilkan *leads* yang telah ditambahkan, menambahkan *leads* baru, mengedit data *leads*, menghapus *leads*.

Data yang terdapat pada fitur ini adalah:

- a. Nama *leads* terdiri dari nama awal dan akhir.
- b. Nama perusahaan tempat *leads* bekerja
- c. Gelar
- d. Informasi kontak seperti nomor telepon, nomor seluler, *email*, *website*, *Skype*, *fax*, dan alamat *leads*.
- e. Status *leads* (*new*, *working*, *unqualified*).
- f. Pemilik data *leads* atau orang yang bertanggung jawab terhadap *leads* tersebut.
- g. Sosial media *leads*



Gambar 19 Tampilan Fitur *Leads* pada Aplikasi Base CRM

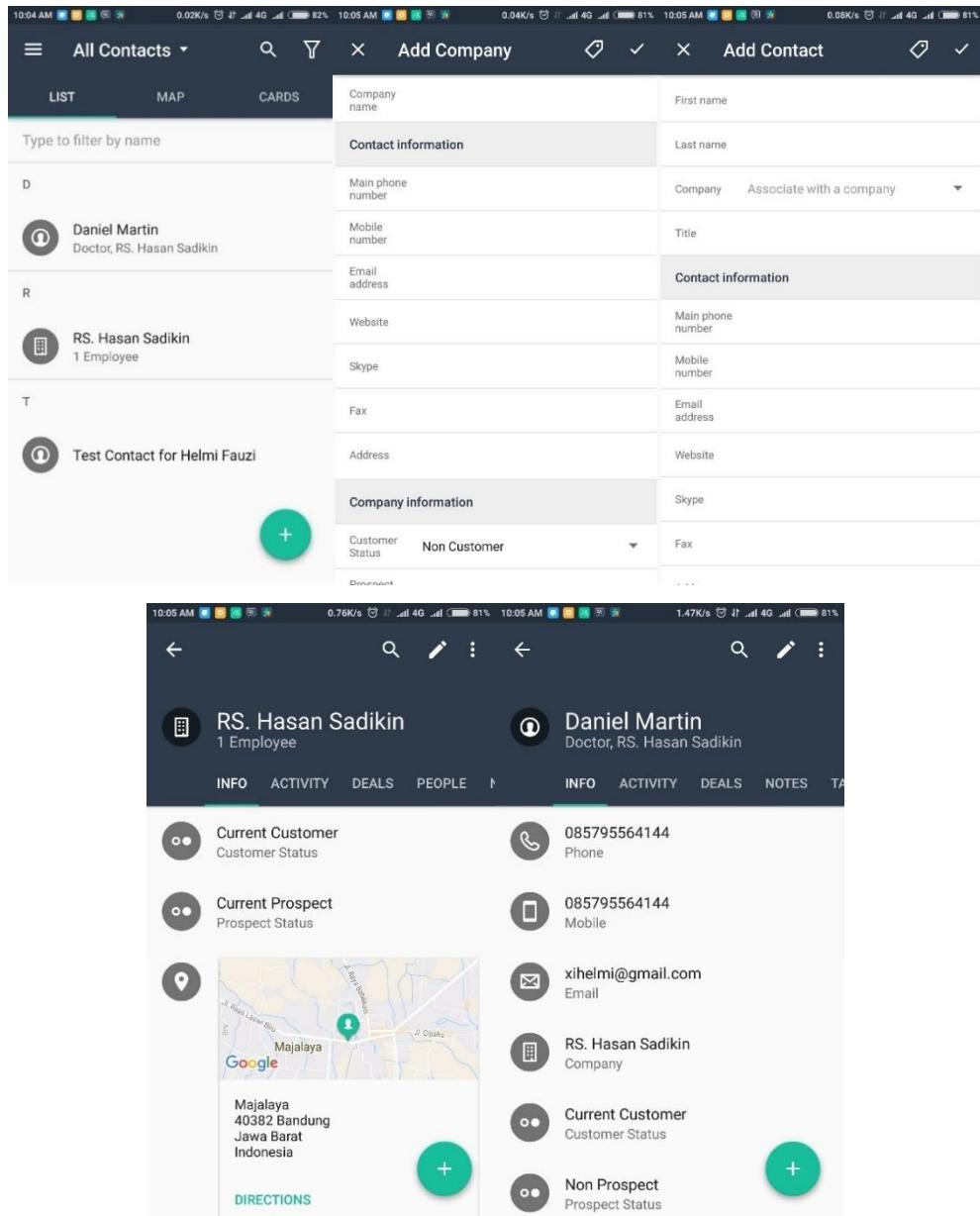
2. *Contact*

Fitur ini merupakan fitur untuk mengelola data *customer*. Jika dikaitkan dengan studi kasus, fitur ini mengelola data dokter yang sudah berhasil bekerja sama dengan *medical rep*. Fitur ini membagi kontak ke dalam dua jenis, yaitu kontak perusahaan dan kontak perseorangan. Fitur ini dapat menampilkan kontak yang telah ditambahkan, menambah kontak, edit kontak, dan hapus kontak.

Data untuk kontak perusahaan yang terdapat pada fitur ini adalah:

- a. Nama perusahaan
- b. Informasi kontak seperti nomor telepon, nomor seluler, *email*, *website*, *Skype*, *fax*, dan alamat perusahaan.
- c. Status *customer* (*past customer*, *current customer*, *non customer*).
- d. Status prospek (*current prospect*, *lost prospect*, *non prospect*).
- e. Pemilik data *leads* atau orang yang bertanggung jawab terhadap kontak tersebut.
- f. Sosial media perusahaan.

Untuk kontak pribadi, hampir sama dengan data pada kontak perusahaan, perbedaannya nama *customer* dan terdapat tambahan keterkaitan dengan kontak perusahaan yang telah dibuat. Gambar 20 berikut merupakan tampilan fitur *contact*.



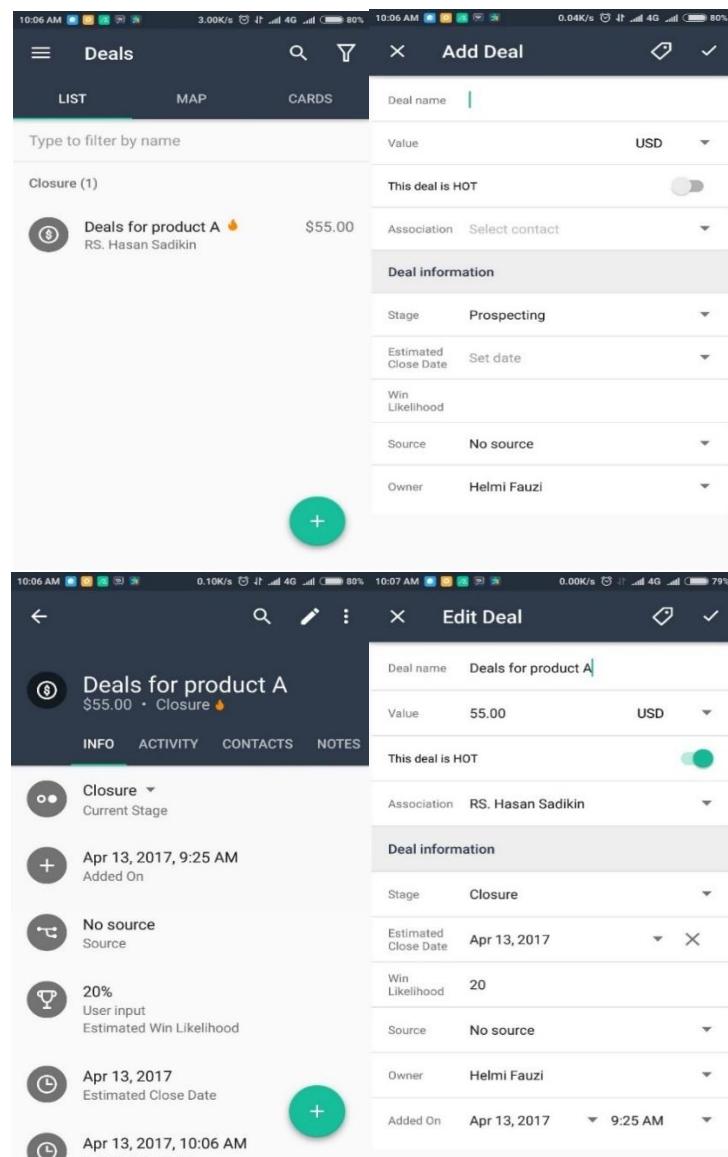
Gambar 20 Tampilan Fitur *Contact* pada Aplikasi Base CRM

3. Deals

Fitur ini merupakan fitur untuk mengelola *progress* penjualan perusahaan terhadap *customer* atau bisa dikatakan sebagai *sales pipeline*. Jika dikaitkan dengan studi kasus fitur ini dapat mengelola proses *detailing* produk perusahaan terhadap dokter, artinya tidak hanya mengelola transaksi yang telah terjadi, tetapi mengelola ‘proses’ *detailing medical representative* dengan dokter dari awal sampai terjadinya transaksi.

Data yang terdapat pada fitur ini adalah :

- a. Nama *deal*
- b. Jumlah *deal*
- c. *Contact* yang terkait
- d. Tahapan *deal* (*Prospecting, Qualified, Closure, Won, Unqualified, Lost*)
- e. Perkiraan tanggal *close deal*
- f. Kemungkinan berhasil *deal*
- g. Orang yang bertanggung jawab terhadap *deal*



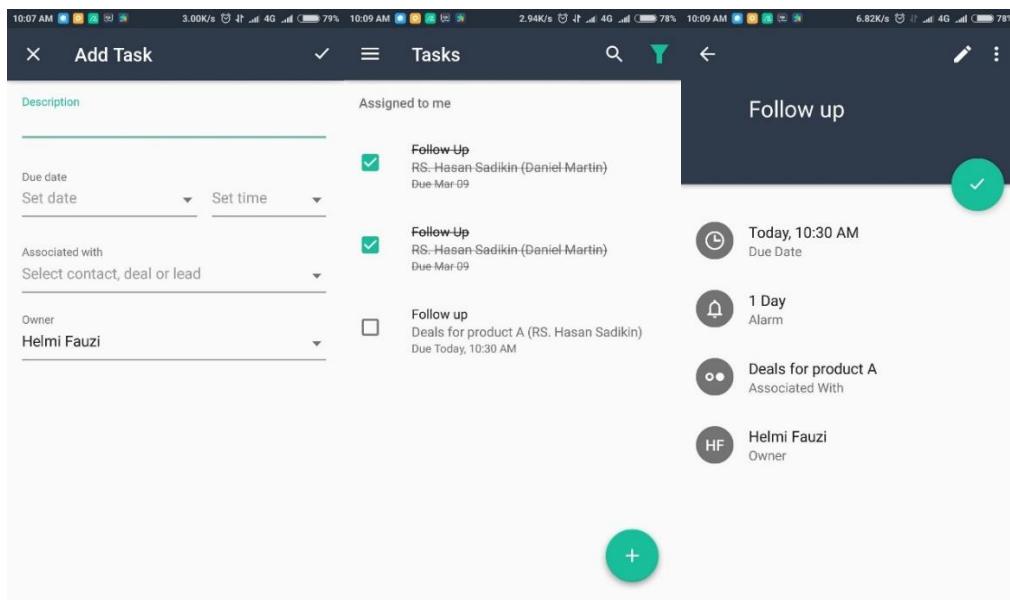
Gambar 21 Tampilan Fitur *Deals* pada Aplikasi Base CRM

4. Tasks

Fitur ini merupakan fitur pengelolaan tugas *sales* atau *medical rep* (jika dikaitkan dengan studi kasus). Pada fitur ini dapat menampilkan tugas yang telah ditambahkan, menambahkan tugas, menandai tugas yang telah diselesaikan, menghapus tugas, dan *reminder* tugas.

Data yang terdapat pada fitur ini adalah:

- a. Deskripsi *task*
- b. *Due date/ tenggat waktu*
- c. *Reminder*
- d. Data yang terkait (*lead, deal, atau contact*)
- e. Orang yang bertanggung jawab terhadap *task*



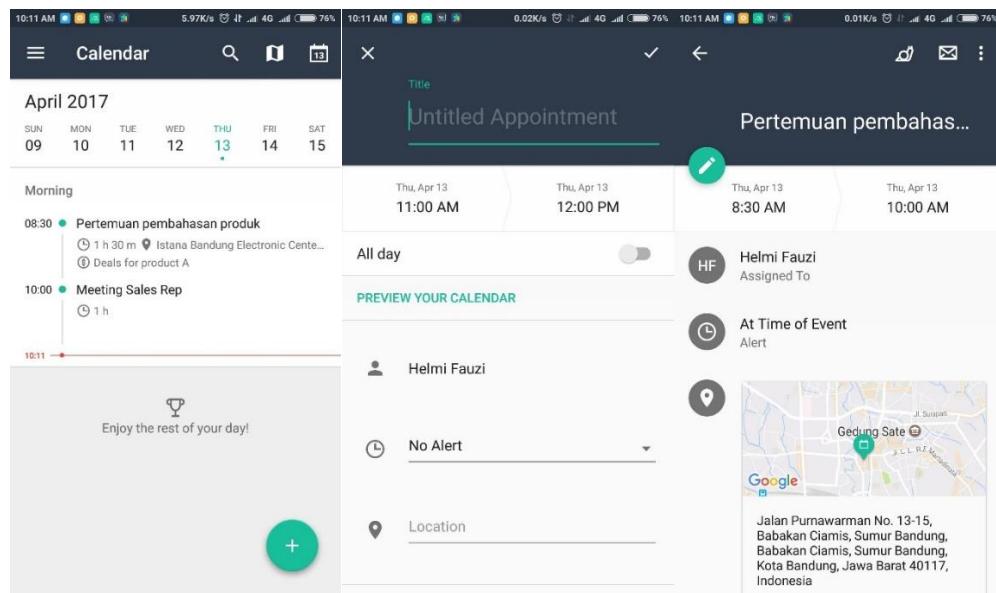
Gambar 22 Tampilan Fitur *Tasks* pada Aplikasi Base CRM

5. Calendar

Pada fitur ini merupakan fitur untuk mengelola agenda pertemuan yang akan dilakukan *sales*. Pada fitur ini dapat menampilkan agenda yang telah dibuat berdasarkan tanggal, menambahkan agenda baru, *edit* agenda, hapus agenda, *reminder* agenda.

Data yang terdapat pada fitur ini adalah:

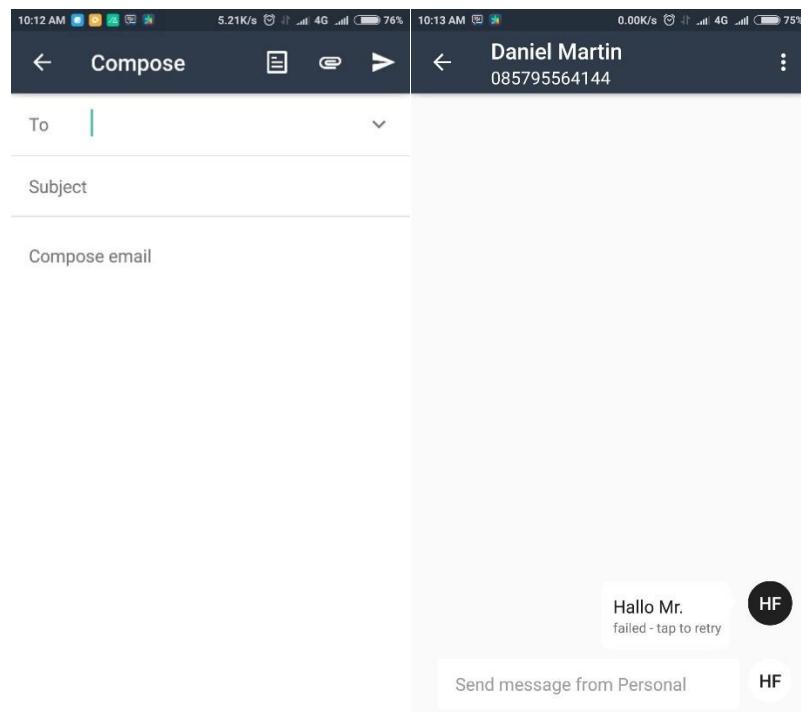
- a. Judul agenda
- b. Waktu mulai dan selesai
- c. *Reminder*
- d. Lokasi pertemuan
- e. Peserta
- f. *Deals*
- g. Deskripsi agenda



Gambar 23 Tampilan Fitur *Calendar* pada Aplikasi Base CRM

6. *Communication*

Fitur ini merupakan fitur yang menyediakan sarana komunikasi untuk *sales* dengan kontak yang ada seperti *email* dan sms, sehingga tidak perlu membuka aplikasi lain.



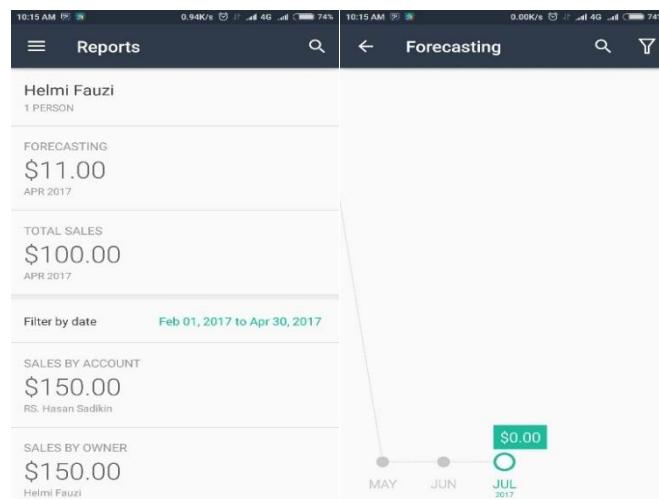
Gambar 24 Tampilan Fitur *Communication* pada Aplikasi Base CRM

7. Report

Fitur ini merupakan fitur untuk memonitor laporan penjualan yang telah terjadi.

Fitur ini juga menyediakan *forecast* untuk peramalan penjualan tiap bulan.

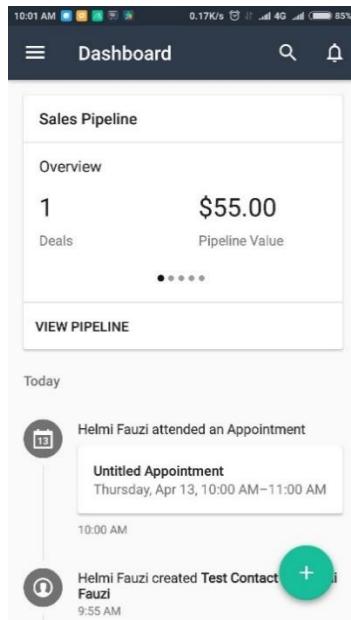
Fitur ini ditampilkan pada Gambar 25.



Gambar 25 Tampilan Fitur *Report* pada Aplikasi Base CRM

8. Dashboard

Pada fitur *dashboard* ini dapat memonitor *sales pipeline* atau proses penjualan kepada *customer* dan juga terdapat *timeline history* yang telah dilakukan selama menggunakan aplikasi.



Gambar 26 Tampilan Fitur *Dashboard* pada Aplikasi Base CRM

Kelebihan dan kekurangan aplikasi :

Setelah menganalisis dan mencoba fitur dari aplikasi Base CRM, fitur-fitur yang terdapat didalamnya mempunyai kelebihan dan kekurangan masing-masing yang dijelaskan pada Tabel 22.

Tabel 22 Kekurangan dan Kelebihan Aplikasi Base CRM

No	Menu Aplikasi	Specification	Jenis CRM	Kelebihan	Kekurangan
1	<i>Leads</i>	1. Menyimpan data <i>leads</i> .	Operasional	Dapat menyimpan data <i>leads</i> . Jika dikaitkan dengan studi kasus, fitur ini dapat digunakan untuk mencatat data dokter, dan dapat membedakan kualifikasi dokter tersebut.	
		2. Membedakan kualifikasi dokter	Strategis		

2	<i>Contact</i>	1. Menyimpan data <i>contact</i> perusahaan	Operasional	Dapat menyimpan data <i>contact</i> perusahaan ataupun perorangan. Jika dikaitkan dengan studi kasus fitur ini dapat digunakan ketika dokter sepakat untuk meresepkan produk, data dokter bisa dibedakan sebagai pelanggan dan disimpan sebagai kontak.	
3	<i>Deals</i>	1. Mengelola data penjualan	Operasional	Dapat mengelola proses penjualan <i>sales</i> dengan <i>customer</i> . Jika dikaitkan dengan studi kasus, fitur ini dapat digunakan untuk mencatat transaksi/kesepakatan yang terjadi antara <i>medical rep</i> dengan dokter, yang mana nantinya data digunakan untuk pelaporan.	
4	<i>Tasks</i>	1. Mengelola tugas yang akan dilakukan	Strategis	Dapat mengelola tugas yang akan dilakukan dan terdapat <i>reminder</i> tugas tersebut. Jika dikaitkan dengan studi kasus, fitur ini dapat digunakan untuk mencatat rencana kunjungan (<i>visit plan</i>)	
5	<i>Calendar</i>	1. Mengelola agenda pertemuan	Strategis	Dapat mengelola agenda pertemuan yang akan dilakukan <i>sales</i> .	
6	<i>Communication</i>	1. Mengirim email kepada <i>customer</i>	Operasional	Dengan adanya fitur ini kita tidak perlu mengirim <i>email</i> dengan aplikasi lain, dan dapat mengirim	Dengan mengizinkan akses pihak lain, tentunya keamanan

				pesan dengan mudah melalui aplikasi	privasi berkurang
7	<i>Report</i>	1. Memonitor angka penjualan.	Analitis	Dapat memonitor angka penjualan, dan dapat mengetahui ramalan penjualan kedepannya sehingga <i>sales</i> dapat merencanakan strateginya.	
		2. Menampilkan ramalan penjualan	Analitis		
8	<i>Dashboard</i>	1. Memonitor <i>sales pipeline</i>	Analitis	Dapat memonitor <i>sales pipeline</i> , dan mengetahui apa saja yang telah dilakukan ketika menggunakan aplikasi	

3.3.3 MediVisits

MediVisits merupakan salah satu aplikasi Android buatan zTek Technologies yang dapat membantu pekerjaan *medical rep* profesional. Pekerjaan yang dibantu meliputi *planning* kunjungan, mengelola dokter, mengklasifikasikan dokter, dan juga informasi produk perusahaan. Aplikasi dapat dilihat melalui : <https://play.google.com/store/apps/details?id=ca.ztek.medivisits.free&hl=in>. Fitur-fitur yang terdapat pada MediVisits diantaranya:

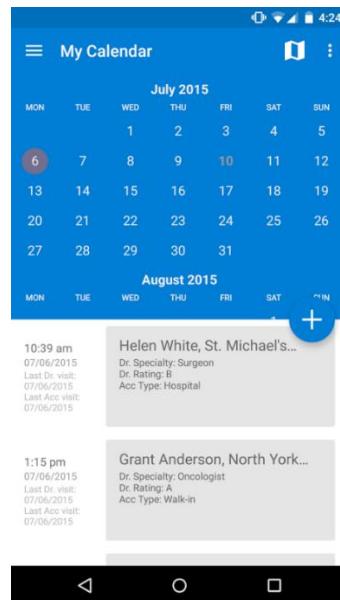
1. *Salesvisits*

Fitur ini merupakan fitur yang digunakan untuk pencatatan ataupun perencanaan kunjungan ke dokter. Hal ini sangat *relevant* dengan studi kasus, dalam hal pembuatan rencana kunjungan (*visit plan*) dan pencatatan kartu *visit list*. Fitur ini disertai dengan *reminder*.

Data yang terdapat pada fitur ini adalah:

- Nama dokter
- Nama perusahaan

- c. Waktu pengingat/ *reminder*
- d. Jenis kontak (*visit, phone, email/SMS, other*)
- e. Waktu kunjungan
- f. Produk yang akan dipromosikan
- g. Catatan



Gambar 27 Tampilan Fitur *Salesvisit* pada Aplikasi MediVisits

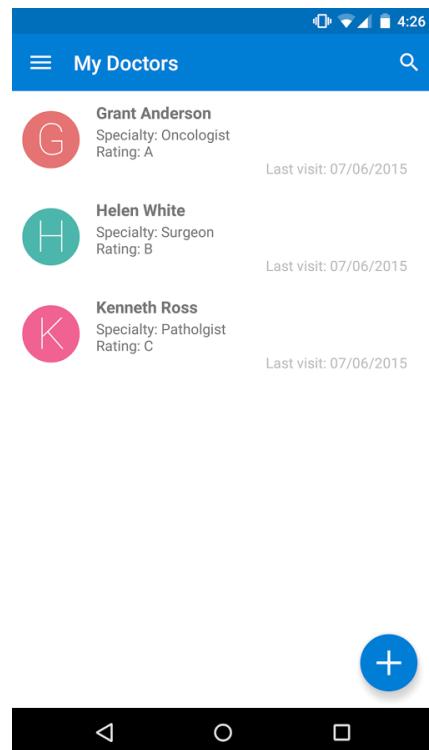
2. *Doctors*

Fitur ini merupakan fitur yang dapat membantu *medical sales rep* dalam mengelola dokter.

Data yang terdapat pada fitur ini adalah:

- a. Gambar dokter
- b. Nama dokter
- c. Nomor ID
- d. Spesialis
- e. *Rating* (hal ini digunakan untuk mengklasifikasikan dokter) beserta alasannya
- f. Perusahaan terkait
- g. *Email*

- h. Nomor telepon
- i. Alamat
- j. Ulang tahun
- k. Gender



Gambar 28 Tampilan Fitur *Doctor* pada Aplikasi MediVisits

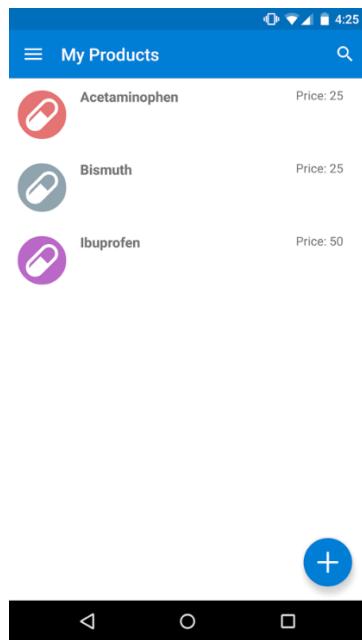
3. *Product*

Fitur ini merupakan fitur yang dapat membantu *medical rep* dalam mengelola informasi produk.

Data yang terdapat pada fitur ini adalah:

- a. Gambar produk
- b. Nama produk
- c. Nama umum
- d. Nama brand
- e. *Manufacturer*/ pabrikan
- f. Kode obat
- g. *Product family*

- h. Kekuatan
- i. Paket
- j. Bahan aktif
- k. Deskripsi produk
- l. Harga



Gambar 29 Tampilan Fitur *Product* pada Aplikasi MediVisits

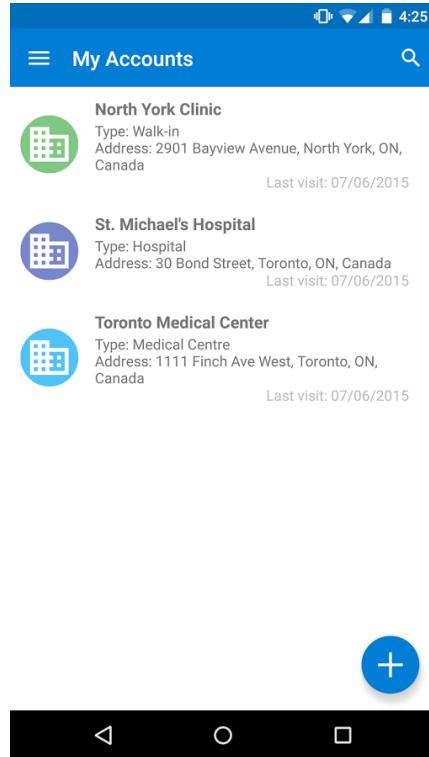
4. *Account*

Fitur ini merupakan fitur yang dapat membantu *medical rep* dalam mengelola data perusahaan, dalam hal ini rumah sakit atau apotek.

Data yang terdapat pada fitur ini adalah:

- a. Gambar perusahaan
- b. Nama perusahaan
- c. Nomor ID
- d. Tipe perusahaan
- e. Alamat
- f. *Rating*, beserta alasannya
- g. *Website*
- h. *Email*

- i. Nomor telepon
- j. Catatan



Gambar 30 Tampilan Fitur *Account* pada Aplikasi MediVisits

5. *Export Salesvisit*

Fitur ini digunakan untuk meng-*export* data kunjungan ke Excel sebagai laporan kunjungan *medical rep*.

Data yang dibutukan pada fitur ini adalah:

- a. *Start date*
- b. *End date*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Doctor	Account	Type	Scheduled	De-scheduled	Time	Products	Notes	Status	Name	Message	Feedback	Reason	Sample
2	Helmi Fauz	RS. Hasan	Visit	04/15/2017	5:10		Amoxicillin,		Planned		Amoxicillin			
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														

Gambar 31 Tampilan Hasil *Export* Laporan Kunjungan Aplikasi MediVisits

Kelebihan dan kekurangan aplikasi :

Setelah menganalisis dan mencoba fitur dari aplikasi MediVisit, fitur-fitur yang terdapat didalamnya mempunyai kelebihan dan kekurangan masing-masing yang dijelaskan pada Tabel 23.

Tabel 23 Kekurangan dan Kelebihan Aplikasi MediVisits

No	Menu Aplikasi	Specification	Jenis CRM	Kelebihan	Kekurangan
1	<i>Salesvisit</i>	1. Menyimpan agenda kunjungan	Strategis	Dapat menyimpan agenda kunjungan <i>medical rep</i> , sehingga kunjungan dapat lebih terencana dan tercatat. Selain itu juga dengan adanya <i>reminder</i> , bisa menjadi pengingat bagi <i>medical rep</i> ketika akan melakukan kunjungan	Tidak meng-handle waktu <i>salesvisit</i> yang besamaan, artinya aplikasi dapat membuat dua atau lebih agenda yang berbeda dalam waktu yang sama. Tidak dibedakannya antara agenda kunjungan dan pencatatan kunjungan yang telah dilakukan, sehingga saat dilakukan <i>export</i> data <i>salesvisit</i> ini menjadi laporan, akan ada informasi

					kunjungan yang belum tentu telah selesai dilakukan.
2	<i>Doctor</i>	1. Menyimpan data dokter.	Operasional	Dapat menyimpan data dokter, beserta klasifikasi dokter tersebut, selain itu juga terdapat status <i>last visit</i> sehingga <i>medical rep</i> dapat mengetahui kapan ia mengunjungi dokter tersebut	Aplikasi tidak membedakan dokter mana yang terkualifikasi, ataupun sebaliknya.
		2. Menyimpan data klasifikasi dokter	Strategis		
3	<i>Product</i>	1. Menyimpan informasi <i>product</i>	Operasional	Dapat menyimpan informasi lengkap produk, sehingga dapat membantu <i>medical rep</i> ketika akan mempromosikan produknya	
4	<i>Account</i>	1. Menyiiman informasi perusahaan	Operasional	Dapat menyimpan informasi perusahaan seperti rumah sakit, klinik, dan sebagainya.	
5	<i>Export Salesvisit</i>	1. Dapat merekap seluruh data kunjungan dalam bentuk Excel.	Operasional	Dapat melakukan <i>export</i> data kunjungan menjadi rekapan seluruh data kunjungan dalam bentuk Excel, sehingga dapat membantu <i>medical rep</i> dalam pelaporan	Bentuk rekapan masih berdasarkan data kunjungan yang di-input sehingga <i>medical rep</i> masih harus membuat laporan secara manual yang sesuai dengan format seharusnya.

3.4 Evaluasi Hasil Analisis Aplikasi Sejenis

Berdasarkan analisis yang telah dilakukan pada sub bab sebelumnya, diperoleh fitur-fitur aplikasi sejenis yang dapat dijadikan sebagai acuan dalam menyelesaikan hasil analisis *requirement*. Tabel 24 menjelaskan fitur-fitur aplikasi sejenis yang dapat menyelesaikan *requirement*.

Tabel 24 Evaluasi *Requirement* dan Aplikasi Sejenis

No	Fitur	Menu yang Menunjang Fitur			Aplikasi yang Menjadi Patokan	Alasan
		Insightly	BaseCRM	MediVisits		
<i>Medical representative</i>						
1	[REQ-F-05] Aplikasi dapat mengelola data dokter	<i>Opportunity & Contact</i>	<i>Leads & Contact</i>	<i>Doctor</i>	BaseCRM	Fitur pada BaseCRM pada <i>requirement</i> ini, melengkapi spesifikasi dibandingkan dengan fitur lain. Selain itu, tampilannya lebih menarik dan informasi yang ditampilkan cukup lengkap.

2	[REQ-F-07] Aplikasi dapat mengelola rencana kunjungan	<i>Calendar</i>	<i>Calendar & Task</i>	<i>SalesVisit</i>	MediVisits	Secara data yang dibutuhkan, pada aplikasi MediVisits sesuai dengan studi kasus, tetapi secara konsep masih ambigu, karena rencana kunjungan tersebut dapat dikatakan sebagai laporan kunjungan, padahal belum tentu rencana kunjungan tersebut benar dilakukan.
3	[REQ-F-11] Aplikasi dapat menampilkan daftar produk perusahaan	-	-	<i>Product</i>	MediVisits	Satu-satunya aplikasi sejenis yang dianalisis yang mempunyai fitur menampilkan produk, hanya MediVisit
4	[REQ-F-06] Aplikasi dapat mengkualifikasi dokter	-	-	-	-	-
5	[REQ-F-10] Aplikasi dapat membuat laporan sesuai format	-	-	-	-	-
6	[REQ-F-08] Aplikasi dapat mengelola hasil kunjungan	-	<i>Deals</i>	<i>SalesVisit</i>	BaseCRM	Secara data yang dibutuhkan, BaseCRM hampir sesuai dengan kebutuhan studi kasus

7	[REQ-F-04] Aplikasi dapat mengklasifikasi sikan dokter	-	-	-	-	-
8	[REQ-F-09] Aplikasi dapat mengelola hasil transaksi	-	-	-	-	-
Supervisor						
9	[REQ-F-01] Aplikasi dapat mengelola tugas dan target <i>medical representative</i>	-	-	-	-	-
10	[REQ-F-02] Aplikasi dapat menghitung pencapaian target <i>medical representative</i> secara berkala	-	-	-	-	-
11	[REQ-F-03] Aplikasi dapat menampilkan laporan <i>medical representative</i>	-	-	-	-	-
12	[REQ-F-12] Aplikasi dapat mengelola produk perusahaan	-	-	-	-	-
13	[REQ-F-13] Aplikasi dapat mengelola akun <i>medical representative</i>	-	-	-	-	-
Area Sales Manager						
14	[REQ-F-14] Aplikasi dapat mengelola akun supervisor	-	-	-	-	-

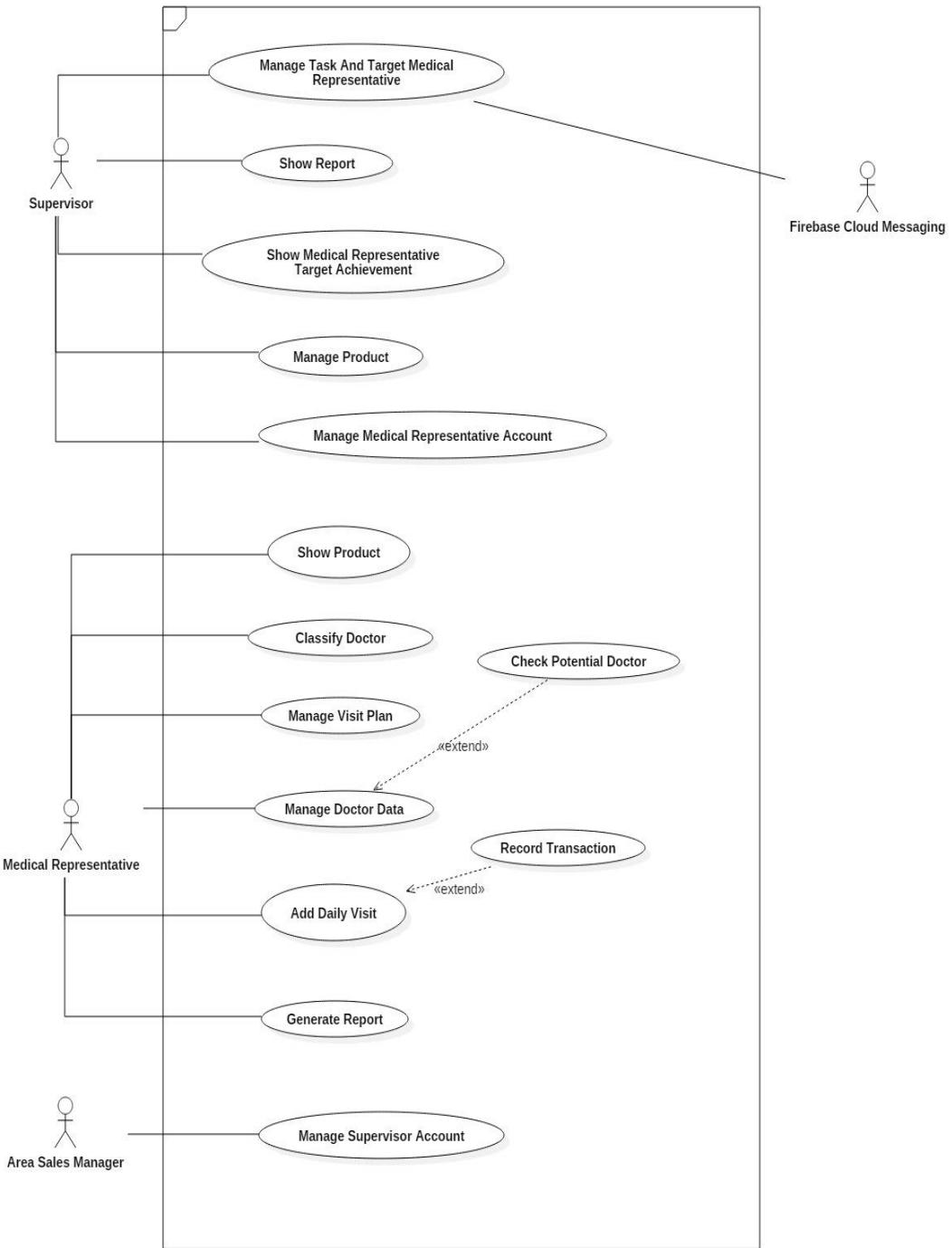
Dari Tabel 24 dapat disimpulkan bahwa aplikasi-aplikasi yang ada belum menjawab seluruh kebutuhan *stakeholder* pada studi kasus *medical representative*. Beberapa fitur aplikasi yang sesuai dengan kebutuhan *stakeholder*, akan dijadikan sebagai acuan baik dari segi tampilan, data, ataupun alur proses dalam pembuatan aplikasi. Adapun kebutuhan yang tidak terselesaikan oleh fitur aplikasi sejenis akan menjadi pengembangan, sehingga dapat diselesaikan oleh aplikasi yang dibangun.

3.5 Usulan Solusi Penyelesaian Masalah

Dari hasil analisis dan evaluasi sistem yang sedang berjalan maupun aplikasi sejenis, aplikasi yang dibangun harus memenuhi persyaratan sebagai berikut:

1. Aplikasi dapat mengelola data dokter.
2. Aplikasi dapat mengelola tugas dan target *medical rep*.
3. Aplikasi dapat mengelola rencana kunjungan.
4. Aplikasi dapat menghitung pencapaian target *medical rep* secara berkala.
5. Aplikasi dapat menampilkan daftar produk perusahaan.
6. Aplikasi dapat mengkualifikasi dokter.
7. Aplikasi dapat mengklasifikasikan dokter.
8. Aplikasi dapat membuat laporan sesuai format (*generate report*).
9. Aplikasi dapat mengelola hasil kunjungan.
10. Aplikasi dapat mengelola hasil transaksi.
11. Aplikasi dapat menampilkan laporan *medical rep*.
12. Aplikasi dapat mengelola akun *medical rep* dan supervisor.
13. Aplikasi dapat mengelola produk yang harus dipromosikan *medical rep*.

Setiap fitur yang telah disebutkan secara umum dapat digambarkan dengan *use case diagram* pada Gambar 32.



Gambar 32 Use Case Diagram CRM Application

Untuk penjelasan lebih detail mengenai interaksi antara pengguna (aktor) dengan sistem dapat dilihat pada lampiran dokumen *software requirements specification* (SRS).

3.5.1 Stakeholder dan Perannya

Berdasarkan hasil analisis yang menghasilkan kebutuhan serta fitur-fitur yang diperlukan oleh aplikasi, selanjutnya dijelaskan tentang *stakeholder* dan perannya dalam aplikasi CRM yang akan dikembangkan.

Stakeholder yang terlibat pada aplikasi yaitu *area sales manager*, supervisor dan *medical rep*. Dengan adanya aplikasi yang dikembangkan, maka peran setiap *stakeholder* tersebut akan berbeda dengan peran sebelumnya. *Stakeholder* dan perannya dijelaskan pada Tabel 25.

Tabel 25 *Stakeholder* dan Perannya pada Aplikasi CRM

No	Stakeholder	Peran
1.	<i>Area Sales Manager</i>	<i>Area Sales Manager</i> adalah atasan dari supervisor, pihak yang menggunakan perangkat <i>website</i> untuk melakukan pengelolaan akun dari supervisor.
2.	Supervisor	Supervisor adalah pihak yang mengelola seluruh tugas dan target harian <i>medical representative</i> , kemudian supervisor juga memiliki kepentingan untuk memonitor laporan yang buat oleh <i>medical representative</i> , serta mengetahui capaian target dari <i>medical representative</i> . Supervisor merupakan pengguna perangkat <i>website</i> serta dapat menggunakannya.
3.	<i>Medical representative</i>	<i>Medical representative</i> adalah pihak yang menggunakan perangkat <i>mobile</i> Android serta dapat menggunakannya. <i>Medical representative</i> memiliki kepentingan untuk mengelola data dokter, mengelola rencana kunjungan, menyeleksi data dokter yang berpotensial, mencatat kompetitor atau produk pesaing, mencatat transaksi, mengakses <i>daily visit</i> , mengakses produk, dan membuat laporan.

3.5.2 Bisnis Proses

Pada bagian ini dijelaskan keterhubungan antara bisnis proses *as-is* (sistem berjalan) dengan use case dan fungsional *requirements*. Keterhubungan dijelaskan untuk mengetahui bisnis proses yang akan dikomputasi pada aplikasi yang dibangun.

Tabel 26 Keterhubungan Bisnis Proses, Use Case dan *Functional Requirement*

No	Bisnis Proses As-Is	Use Case	<i>Functional Requirement</i>
[K-1]	Mencatat hasil kesepakatan dengan dokter	<i>Case Record Transaction</i>	[REQ-F-09]
[K-2]	Mencatat hasil kunjungan	<i>Case Manage Daily Visit</i>	[REQ-F-08]
[K-3]	Mencatat data dokter baru	<i>Case Manage Doctor Data</i>	[REQ-F-05]
[K-4]	Mencatat jadwal praktik dokter	<i>Case Manage Doctor Data</i>	[REQ-F-05]
[K-5]	Menghitung potensi peresepan dokter	<i>Case Check Potential Doctor</i>	[REQ-F-06]
[K-6]	Mencatat kompetitor untuk produk sejenis yang dokter sering resepkan	<i>Case Check Potential Doctor</i>	[REQ-F-06]
[K-7]	Mencatat hasil potensi peresepan dokter	<i>Case Check Potential Doctor</i>	[REQ-F-06]
[K-8]	Mengumpulkan laporan semua <i>medical representative</i>	<i>Case Show Medical Representative Target Achievement</i>	[REQ-F-02]
[K-9]	Menghitung pencapaian target berdasarkan data laporan	<i>Case Show Medical Representative Target Achievement</i>	[REQ-F-02]
[K-10]	Mencatat hasil penilaian kinerja <i>medical representative</i>	<i>Case Show Medical Representative Target Achievement</i>	[REQ-F-02]
[K-11]	Merekap hasil kunjungan	<i>Case Generate Report</i>	[REQ-F-10]
[K-12]	Mengirim laporan kunjungan ke supervisor	<i>Case Generate Report, Case Show Report</i>	[REQ-F-10][REQ-F-03]
[K-13]	Merekap hasil transaksi	<i>Case Generate Report</i>	[REQ-F-10]
[K-14]	Mengirim laporan kunjungan ke supervisor	<i>Case Generate Report, Case Show Report</i>	[REQ-F-10][REQ-F-03]
[K-15]	Membuat rencana kunjungan	<i>Case Manage Visit Plan</i>	[REQ-F-07]

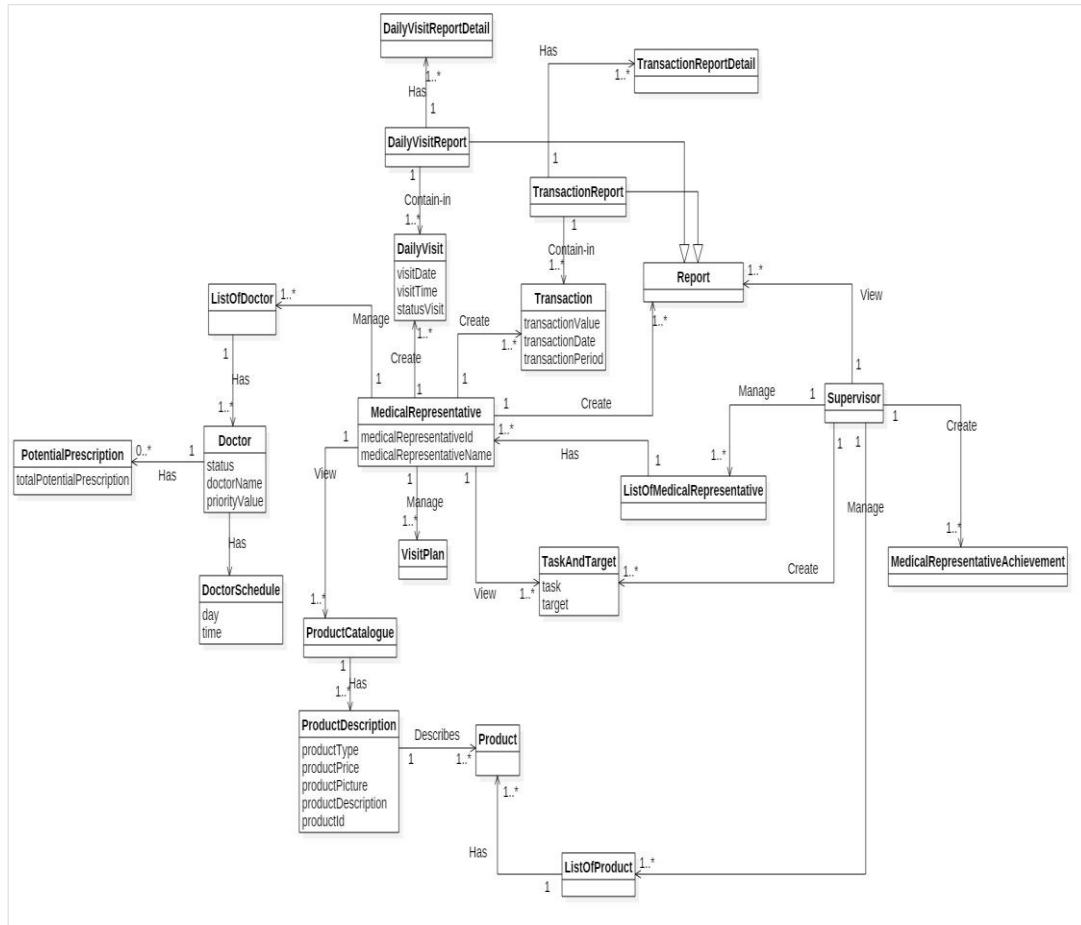
3.5.3 Business Rules

Tabel 27 *Business Rules* pada Aplikasi CRM

No	<i>Business Rule</i>	Proses Terkait
1	<i>Medical rep</i> atau Supervisor harus memiliki akun untuk menggunakan aplikasi	
2	<ol style="list-style-type: none"> 1. Data yang digunakan untuk mengetahui potensi peresepan dokter berupa data peresepan satu bulan terakhir yang didapatkan dari apotek yang disurvei. 2. Perhitungan potensi dokter digunakan selama satu bulan periode dimulai dari tanggal 1 hingga tanggal terakhir pada bulan tersebut. 3. Perhitungan potensi dokter menggunakan rumus sebagai berikut : $\text{Besar Potensi Peresepan} = JHPD \times JM \times XPH \times \sum_{i=1}^n (XOS[i] \times HOS[i])$ (hal. 19) 4. Pencatatan kompetitor berupa pencatatan nama perusahaan atau nama produk pesaing yang diresepkan oleh dokter. 	[K-5] Mencatat Potensi Peresepan Dokter
3	Dokter yang ditentukan oleh <i>medical rep</i> untuk kunjungan merupakan dokter yang terdapat pada <i>list</i> tugas <i>medical rep</i> . Apabila memiliki data dokter baru harus ditambahkan terlebih dahulu ke <i>list</i> dokter.	[K-2] <i>Detailing</i> dan <i>Follow Up</i> Dokter
4	Perhitungan pencapaian target dari <i>medical rep</i> dilakukan berdasarkan data transaksi yang dicatat oleh <i>medical rep</i> , dengan rumus sebagai berikut : $\text{Pencapaian Target} = \frac{\text{Total kesepakatan penjualan obat dokter}}{\text{Target penjualan dari medical rep}}$ (hal. 23)	[K-9] Menghitung pencapaian target berdasarkan data laporan
5	Laporan hasil kunjungan dibuat per bulan oleh <i>medical rep</i> .	[K-11] Rekap Hasil Kunjungan
6	Laporan hasil transaksi dibuat per bulan oleh <i>medical rep</i> .	[K-13] Rekap Hasil Transaksi

3.5.4 Domain Model

Berikut adalah *domain model* dari sistem yang diusulkan

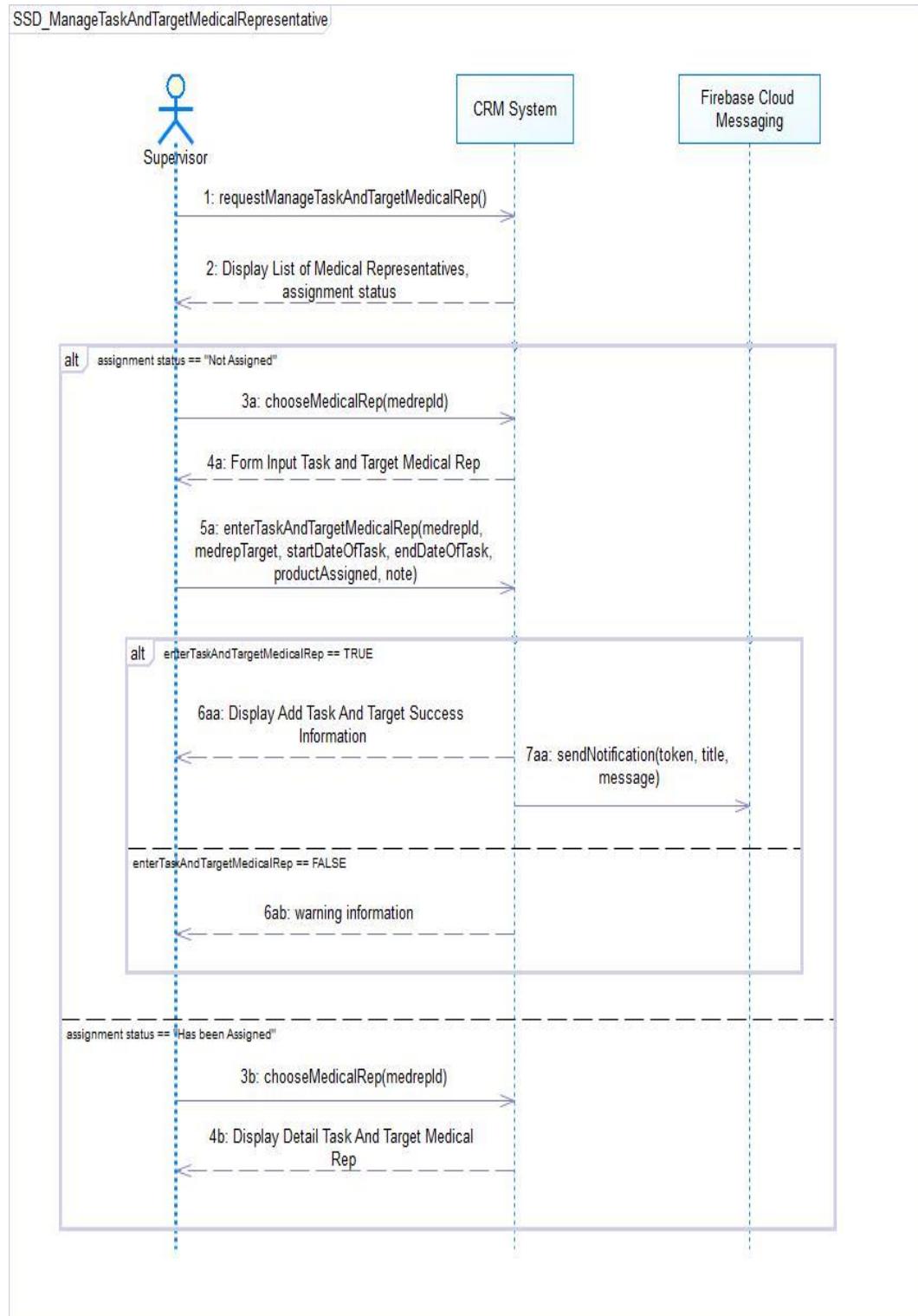


Gambar 33 *Domain Model* dari Sistem yang Diusulkan

3.5.5 System Sequence diagram (SSD) dan Operation Contract (OC)

Berdasarkan hasil analisis yang telah dilakukan dan telah digambarkan dengan use case diagram beserta use case textnya, didapatkan interaksi-interaksi antara aktor dengan aplikasi yang digambarkan dengan *system sequence diagram* (SSD). Proses interaksi aktor dengan sistem pada SSD dijelaskan lebih detail pada laporan SRS.

1. SSD dan OC Manage Task and Target Medical Representative (SSD-01)



Gambar 34 SSD Manage Task And Target Medical Representative

Operation contract dari *system sequence diagram Manage Task And Target Medical Representative* dijelaskan pada Tabel 28, Tabel 29, dan Tabel 30.

Tabel 28 *Contract CO1 requestManageTaskAndTargetMedicalRep*

<i>Operation</i>	requestManageTaskAndTargetMedicalRep()
<i>Cross References</i>	Use Case <i>Manage Task And Target Medical Representative</i>
<i>Preconditions</i>	Supervisor belum memasukkan tugas dan target untuk <i>Medical Representative</i>
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi taskAndTarget dari TaskAndTarget terbentuk 2. Instansiasi listOfMedrep dari ListOfMedicalRep terbentuk 3. Atribut listOfMedrep.listOfMedrep diisi data list medical rep

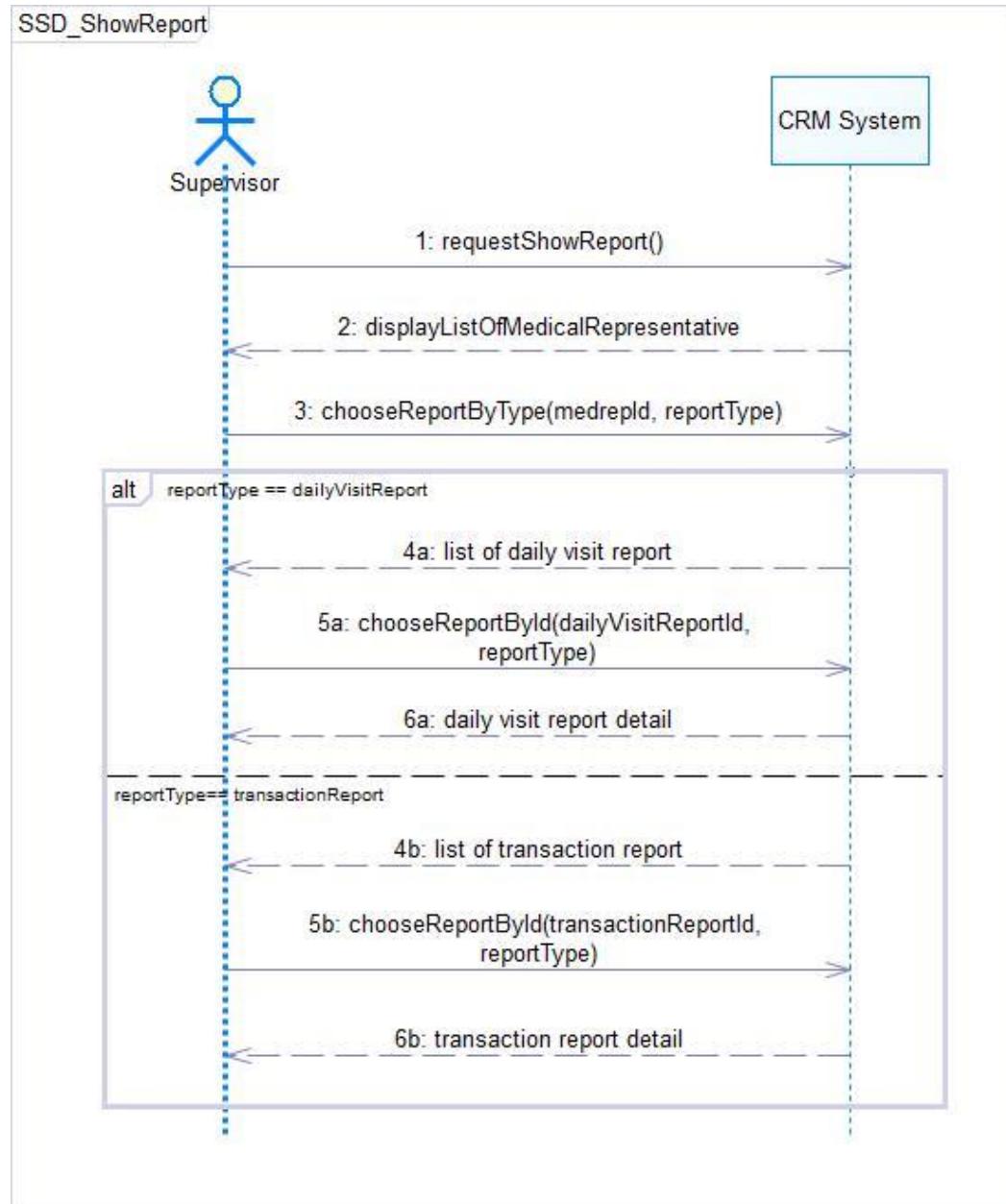
Tabel 29 *Contract CO2 chooseMedicalRep*

<i>Operation</i>	chooseMedicalRep(medrepId)
<i>Cross References</i>	Use Case <i>Manage Task And Target Medical Representative</i>
<i>Preconditions</i>	Supervisor belum memasukkan tugas dan target untuk <i>Medical Representative</i>
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi medicalRep dari MedicalRepresentative terbentuk 2. Atribut medicalRep.idMedrep diisi data idMedrep

Tabel 30 *Contract CO3 enterTaskAndTargetMedicalRep*

<i>Operation</i>	enterTaskAndTargetMedicalRep(medrepId, medrepTarget, startDateOfTask, endDateOfTask, productAssigned, note)
<i>Cross References</i>	Use Case <i>Manage Task And Target Medical Representative</i>
<i>Preconditions</i>	Supervisor belum memasukkan tugas dan target untuk <i>Medical Representative</i>
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Atribut taskAndTarget.medrepId diisi data medrepId 2. Atribut taskAndTarget.medrepTarget diisi data medrepTarget 3. Atribut taskAndTarget.startDateOfTask diisi data startDateOfTask 4. Atribut taskAndTarget.endDateOfTask diisi data endDateOfTask 5. Atribut taskAndTarget.productAssigned diisi data productAssigned 6. Atribut taskAndTarget.note diisi data note 7. TaskAndTarget berasosiasi dengan MedicalRepresentative sesuai dengan medrepId yang sama

2. SSD dan OC Show Report (SSD-02)



Gambar 35 SSD Show Report

Operation contract dari *system sequence diagram Show Report* dijelaskan pada Tabel 31, Tabel 32, Tabel 33 dan Tabel 34.

Tabel 31 *Contract CO1 requestShowReport*

<i>Operation</i>	requestShowReport()
<i>Cross Reference</i>	Use Case <i>Show Report</i>
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Supervisor telah memiliki data <i>Medical Representative</i> 2. <i>Medical Representative</i> telah men-generate laporan
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi listOfMedicalRepresentative dari ListOfMedicalRepresentative terbentuk 2. Atribut listOfMedrep diisi data list medical rep

Tabel 32 *Contract CO2 chooseReportByType*

<i>Operation</i>	chooseReportByType(medrepId, reportType)
<i>Cross Reference</i>	Use Case <i>Show Report</i>
<i>Precondition</i>	<ol style="list-style-type: none"> 1. Supervisor telah memiliki data <i>Medical Representative</i> 2. <i>Medical Representative</i> telah meng-generate laporan
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi medicalRepresentative dari MedicalRepresentative terbentuk 2. Instansiasi dailyVisitReport dari dailyVisitReport terbentuk 3. Instansiasi transactionReport dari transactionReport terbentuk. 4. Atribut dailyVisitReport.medrepId diisi medrepId. 5. Atribut transactionReport.medrepId diisi medrepId 6. Instansiasi listDailyVisitReport dari List DailyVisitReport terbentuk. 7. Instansiasi listTransactionReport dari ListTransactionReport terbentuk 8. DailyVisitReport berasosiasi dengan MedicalRepresentative sesuai medrepId yang sama. 9. TransactionReport berasosiasi dengan MedicalRepresentative sesuai medrepId yang sama

Tabel 33 *Contract CO3 chooseReportById*

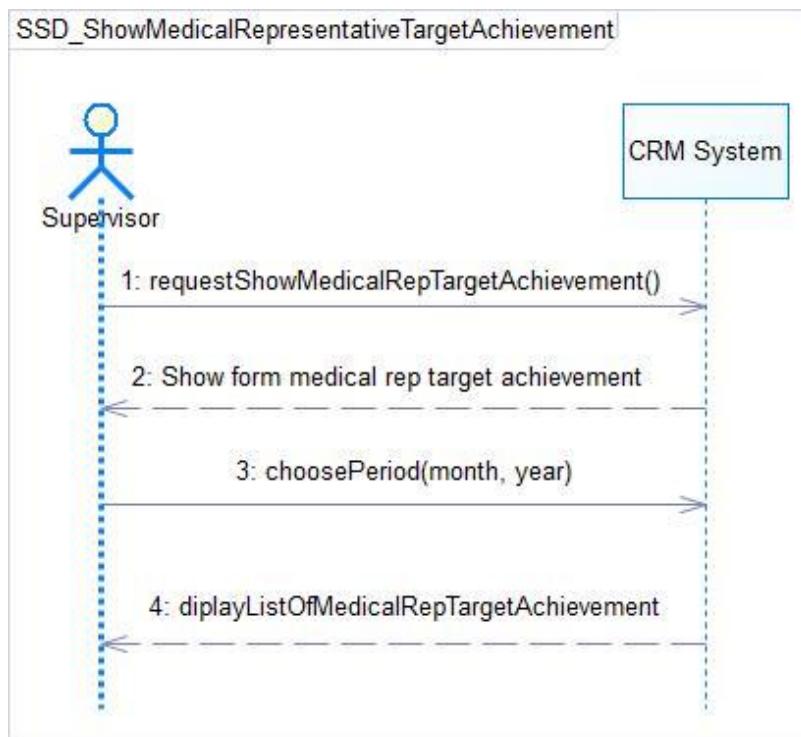
<i>Operation</i>	chooseReportById(dailyVisitReportId, reportType)
<i>Cross Reference</i>	Use Case <i>Show Report</i>
<i>Precondition</i>	Sistem telah menampilkan daftar laporan kunjungan

<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi dailyVisitReport dari DailyVisitReport terbentuk 2. Atribut dailyVisitReport.dailyVisitReportId diisi dailyVisitReportId 3. Instansiasi dailyVisitReportDetail dari DailyVisitReportDetail 4. dailyVisitReportDetail diisi data detail laporan kunjungan. 5. DailyVisitReport berasosiasi dengan DailyVisitReportDetail sesuai dailyVisitReportId yang sama
----------------------	--

Tabel 34 *Contract CO4 chooseReportById*

<i>Operation</i>	chooseReportById(transactionReportId, reportType)
<i>Cross Reference</i>	Use Case <i>Show Report</i>
<i>Precondition</i>	Sistem telah menampilkan daftar laporan transaksi
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi transactionReport dari TransactionReport terbentuk 2. Atribut transaction.transactionReportId diisi transactionReportId 3. Instansiasi transactionReportDetail dari TransactionReportDetail 4. transactionReportDetail diisi data detail laporan transaksi. 5. TransactionReport berasosiasi dengan TransactionReportDetail sesuai transactionReportId yang sama

3. SSD dan OC Show Medical Representative Target Achievement (SSD-03)

Gambar 36 SSD *Show Medical rep Target Achievement*

Operation contract dari *system sequence diagram Show Medical Representative Target Achievement* dijelaskan pada Tabel 35 dan Tabel 36.

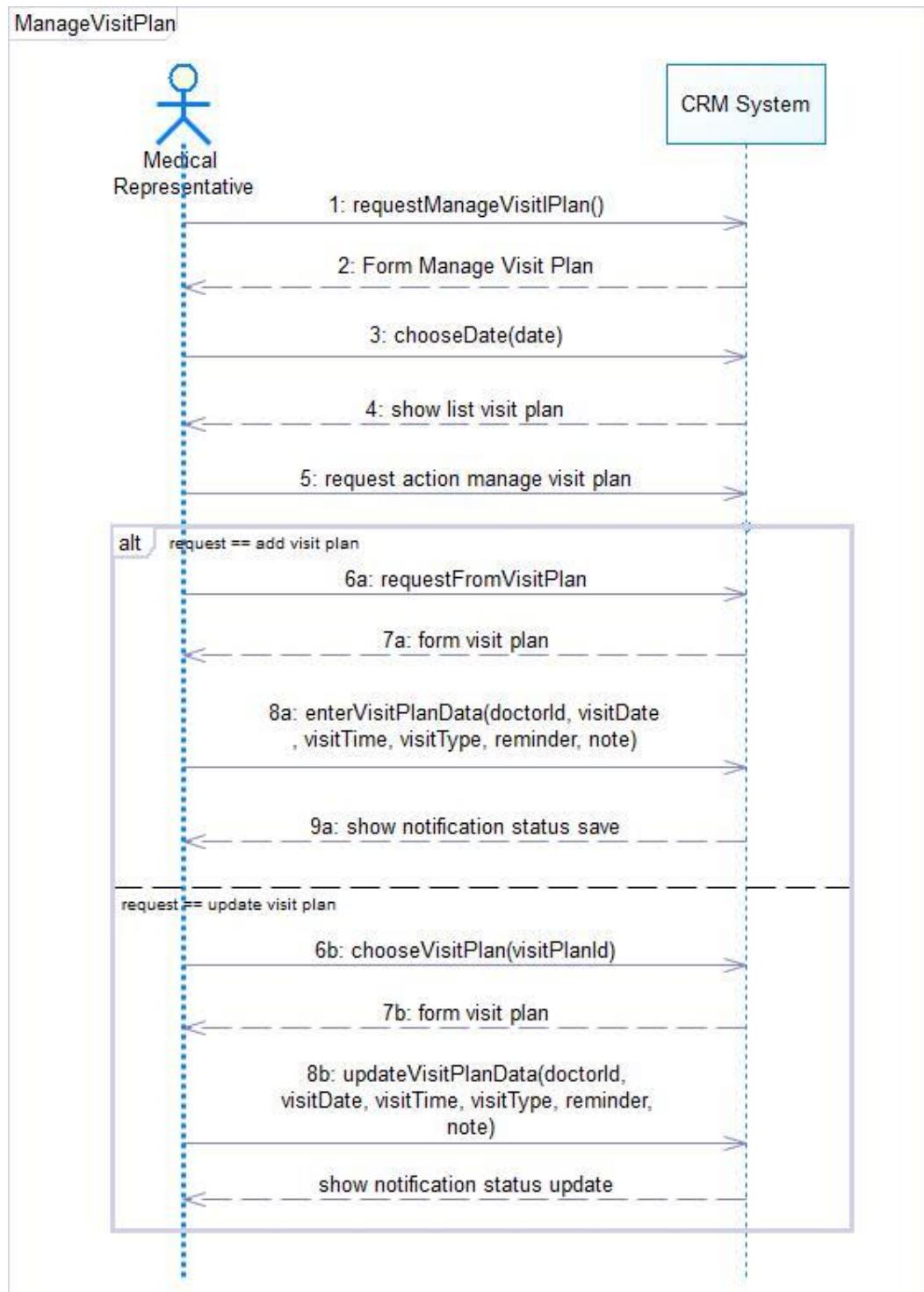
Tabel 35 *Contract CO1 requestShowMedicalRepresentativeTargetAchievement*

<i>Operation</i>	requestShowMedicalRepresentativeTargetAchievement()
<i>Cross Reference</i>	Use Case <i>Show Medical Representative Target Achievement</i>
<i>Precondition</i>	<ul style="list-style-type: none"> 1. Supervisor memiliki data <i>Medical Representative</i> 2. Supervisor telah menentukan target tiap <i>Medical Representative</i> 3. <i>Medical Representative</i> telah memasukkan hasil kunjungan dan transaksi
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi <i>medicalRepresentativeAchievement</i> dari <i>MedicalRepresentativeAchievement</i> terbentuk 2. Supervisor beraosiasi dengan <i>MedicalRepresentativeAchievement</i> sesuai supervisorId yang sama

Tabel 36 *Contract CO2 choosePeriod*

<i>Operation</i>	requestShowMedicalRepresentativeTargetAchievement()
<i>Cross Reference</i>	Use Case <i>Show Medical Representative Target Achievement</i>
<i>Precondition</i>	<ul style="list-style-type: none"> 1. Supervisor memiliki data <i>Medical Representative</i> 2. Supervisor telah menentukan target tiap <i>Medical Representative</i> 3. <i>Medical Representative</i> telah memasukkan hasil kunjungan dan transaksi
<i>Postcondition</i>	<ul style="list-style-type: none"> 3. Instansiasi <i>medicalRepresentativeAchievement</i> dari <i>MedicalRepresentativeAchievement</i> terbentuk 4. Supervisor beraosiasi dengan <i>MedicalRepresentativeAchievement</i> sesuai supervisorId yang sama

4. SSD dan OC Manage Visit Plan (SSD-04)



Gambar 37 SSD Manage Visit Plan

Operation contract dari *system sequence diagram Manage Visit Plan* dijelaskan pada Tabel 37, Tabel 38, Tabel 39, dan Tabel 40.

Tabel 37 *Contract CO1 requestManageVisitPlan*

<i>Operation</i>	requestManageVisitPlan
<i>Cross References</i>	Use Case <i>Manage Visit Plan</i>
<i>Preconditions</i>	<ul style="list-style-type: none"> 1. <i>Medical Representative</i> belum membuat rencana kunjungan 2. <i>Medical Representative</i> sudah mendapatkan tugas dan target dari Supervisor
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi visitPlan dari VisitPlan terbentuk

Tabel 38 *Contract CO2 chooseDate*

<i>Operation</i>	chooseDate(date)
<i>Cross References</i>	Use Case <i>Manage Visit Plan</i>
<i>Preconditions</i>	<ul style="list-style-type: none"> 1. <i>Medical Representative</i> belum membuat rencana kunjungan 2. <i>Medical Representative</i> sudah mendapatkan tugas dan target dari Supervisor
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Atribut visitPlan.date diisi data date

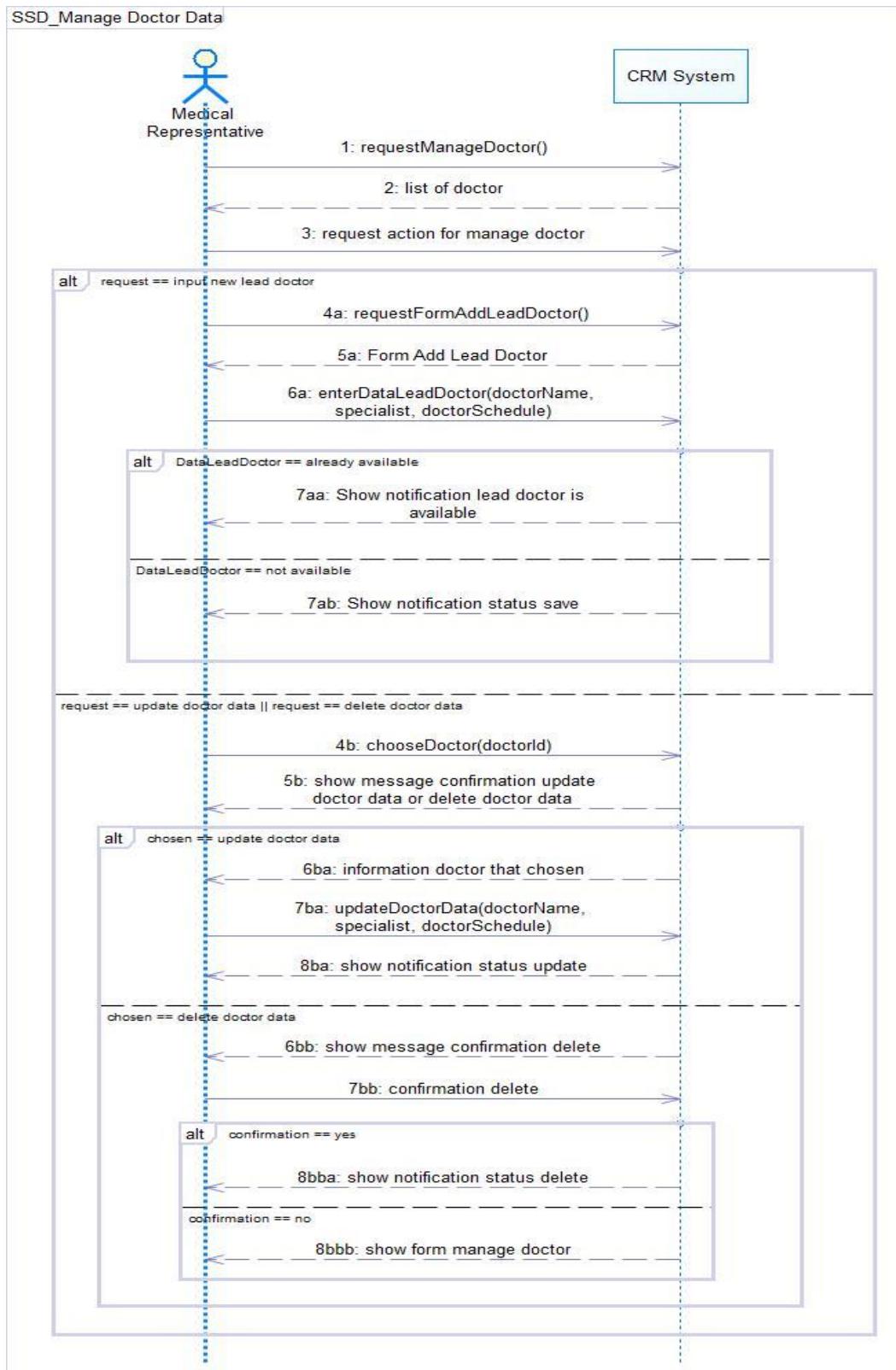
Tabel 39 *Contract CO3 enterVisitPlanData*

<i>Operation</i>	enterVisitPlanData(doctorId, visitDate, visitTime, visitType, reminder, note)
<i>Cross References</i>	Use Case <i>Manage Visit Plan</i>
<i>Preconditions</i>	<ul style="list-style-type: none"> 1. <i>Medical Representative</i> belum membuat rencana kunjungan 2. <i>Medical Representative</i> sudah mendapatkan tugas dan target dari Supervisor
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi visitPlan dari VisitPlan terbentuk. 2. Atribut visitPlan.doctorId diisi data doctorId. 3. Atribut visitPlan.visitDate diisi data visitDate. 4. Atribut visitPlan.visitTime diisi data visitTime. 5. Atribut visitPlan.reminder diisi data reminder. 6. Atribut visitPlan.note diisi note.

Tabel 40 *Contract CO4 updateVisitPlanData*

<i>Operation</i>	updateVisitPlanData(doctorId, visitDate, visitTime, visitType, reminder, note)
<i>Cross References</i>	Use Case <i>Manage Visit Plan</i>
<i>Preconditions</i>	<ol style="list-style-type: none">1. <i>Medical Representative</i> belum membuat rencana kunjungan2. <i>Medical Representative</i> sudah mendapatkan tugas dan target dari <i>Supervisor</i>
<i>Postcondition</i>	<ol style="list-style-type: none">1. Instansiasi visitPlan dari VisitPlan terbentuk.2. Atribut visitPlan.doctorId diisi data doctorId.3. Atribut visitPlan.visitDate diisi data visitDate.4. Atribut visitPlan.visitTime diisi data visitTime.5. Atribut visitPlan.reminder diisi data reminder.6. Atribut visitPlan.note diisi note.

5. SSD dan OC *Manage Doctor Data* (SSD-05)



Gambar 38 SSD *Manage Doctor Data*

Operation contract dari *system sequence diagram Manage Doctor Data* dijelaskan pada Tabel 41, Tabel 42, Tabel 43, Tabel 44, dan Tabel 45.

Tabel 41 *Contract CO1 requestManageDoctor*

<i>Operation</i>	requestManageDoctor()
<i>Cross References</i>	Use Case <i>Manage Doctor Data</i>
<i>Preconditions</i>	<i>None</i>
<i>Postcondition</i>	1. Instansiasi listOfDoctor dari ListOfDoctor terbentuk

Tabel 42 *Contract CO2 requestFormAddLeadDoctor*

<i>Operation</i>	requestFormAddLeadDoctor()
<i>Cross References</i>	Use Case <i>Check Manage Doctor Data</i>
<i>Preconditions</i>	<i>None</i>
<i>Postcondition</i>	1. Instansiasi doctor dari Doctor terbentuk

Tabel 43 *Contract CO3 enterDataLeadDoctor*

<i>Operation</i>	enterDataLeadDoctor(doctorName, specialist, doctorSchedule)
<i>Cross References</i>	Use Case <i>Manage Doctor Data</i>
<i>Preconditions</i>	<i>None</i>
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Atribut doctor.doctorName diisi data doctorName 2. Atribut doctor.specialist diisi data specialist 3. Instansiasi doctorSchedule dari DoctorSchedule terbentuk 4. Atribut doctorSchedule.practiceDay diisi data doctorSchedule.practiceDay 5. Atribut doctorSchedule.practiceStartTime diisi data doctorSchedule.practiceStartTime 6. Atribut doctorSchedule.practiceEndTime diisi data doctorSchedule.practiceEndTime 7. Atribut doctorSchedule.practiceLocation diisi data doctorSchedule.practiceLocation 8. <i>Class Doctor</i> berasosiasi dengan DoctorSchedule

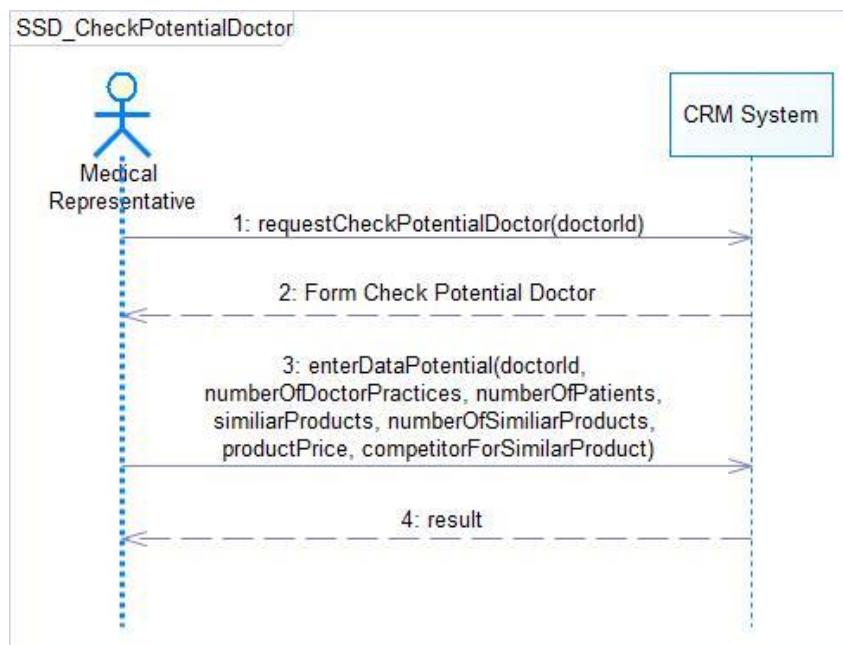
Tabel 44 *Contract CO4 chooseDoctor*

<i>Operation</i>	chooseDoctor(doctorId)
<i>Cross References</i>	Use Case <i>Manage Doctor Data</i>
<i>Preconditions</i>	Data dokter sudah ada
<i>Postcondition</i>	1. Atribut doctor.idDoctor diisi idDoctor

Tabel 45 *Contract CO5 updateDoctorData*

<i>Operation</i>	updateDoctorData(doctorName, specialist, doctorSchedule)
<i>Cross References</i>	Use Case <i>Check Manage Doctor Data</i>
<i>Preconditions</i>	Data produk sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Atribut doctor.doctorName diisi data doctorName 2. Atribut doctor.specialist diisi data specialist 3. Instansiasi doctorSchedule dari DoctorSchedule terbentuk 4. Atribut doctorSchedule.practiceDay diisi data doctorSchedule.practiceDay 5. Atribut doctorSchedule.practiceStartTime diisi data doctorSchedule.practiceStartTime 6. Atribut doctorSchedule.practiceEndTime diisi data doctorSchedule.practiceEndTime 7. Atribut doctorSchedule.practiceLocation diisi data doctorSchedule.practiceLocation 8. Class Doctor berasosiasi dengan DoctorSchedule

6. SSD dan OC Check Potential Doctor (SSD-06)

Gambar 39 SSD *Check Potential Doctor*

Operation contract dari *system sequence diagram* *Check Potential Doctor* dijelaskan pada Tabel 46, dan Tabel 47.

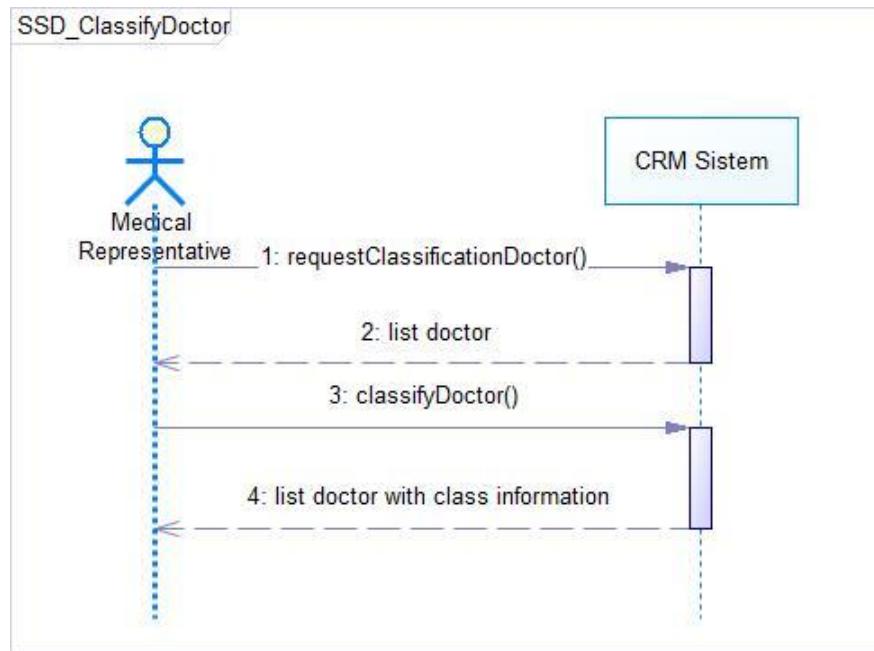
Tabel 46 *Contract CO1 requestCheckPotentialDoctor*

<i>Operation</i>	requestCheckPotentialDoctor(doctorId)
<i>Cross References</i>	Use Case <i>Check Potential Doctor</i>
<i>Preconditions</i>	Data dokter sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi doctor dari Doctor terbentuk 2. Atribut doctor.doctor diisi dengan data doctorId

Tabel 47 *Contract CO3 enterDataPotential*

<i>Operation</i>	enterDataPotential(doctorId, numberOfDoctorPractices,numberOfPatients, similiarProducts, numberOfSimiliarProducts, productPrice, competitorForSimiliarProduct)
<i>Cross References</i>	Use Case <i>Check Potential Doctor</i>
<i>Preconditions</i>	Data dokter, produk sejenis, dan peresepan dokter sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi potentialPrescription dari PotentialPrescription 2. Atribut potentialPrescription.doctorId diisi doctorId 3. Atribut potentialPrescription.numberOfDoctorPractices diisi numberOfDoctorPractices 4. Atribut potentialPrescription.numberOfPatients diisi numberOfPatients 5. Atribut potentialPrescription.similiarProducts diisi similiarProducts 6. Atribut potentialPrescription.numberOfSimilarProducts diisi data numberOfSimilarProducts 7. Atribut potentialPrescription.productPrice diisi productPrice 8. Atribut potentialPrescription.competitorForSimilarProduct diisi data competitorForSimilarProduct 9. Doctor berasosiasi dengan PotentialPrescription

7. SSD dan OC Classify Doctor (SSD-07)



Gambar 40 - SSD Classify Doctor

Operation contract dari *system sequence diagram Classify Doctor* dijelaskan pada Tabel 48, dan Tabel 49.

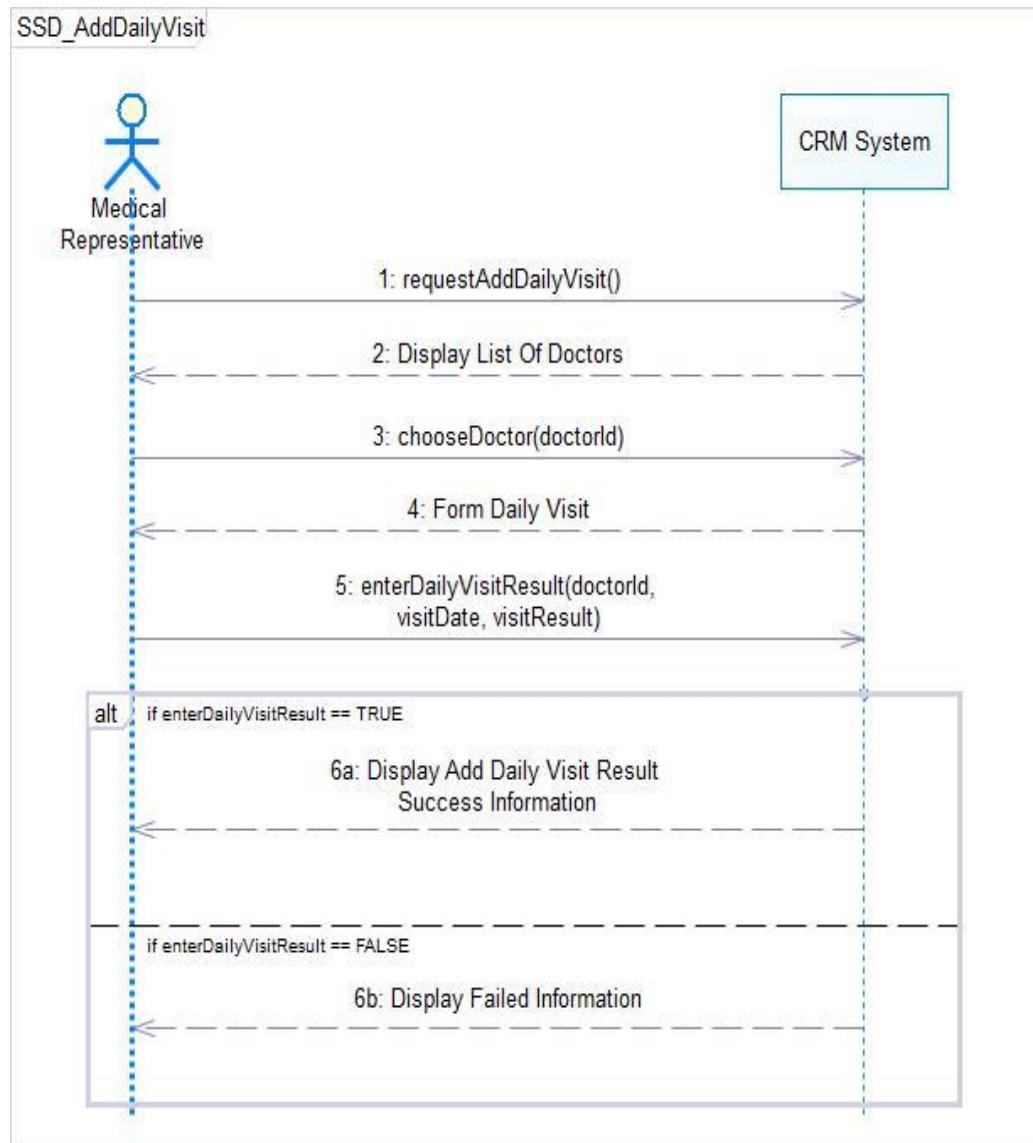
Tabel 48 Contract CO1 *requestClassificationDoctor*

<i>Operation</i>	<i>requestClassificationDoctor()</i>
<i>Cross References</i>	Use Case <i>Classify Doctor</i>
<i>Preconditions</i>	<i>Medical Representative</i> telah memiliki daftar dokter beserta besar potensi peresepannya.
<i>Postcondition</i>	<ol style="list-style-type: none"> Instansiasi <i>listDoctor</i> dari <i>ListDoctor</i> terbentuk. <i>listDoctor</i> diisi daftar dokter beserta besar potensi peresepannya

Tabel 49 Contract CO2 *classifyDoctor*

<i>Operation</i>	<i>requestClassificationDoctor()</i>
<i>Cross References</i>	Use Case <i>Classify Doctor</i>
<i>Preconditions</i>	<i>Medical Representative</i> telah memiliki daftar dokter beserta besar potensi peresepannya.
<i>Postcondition</i>	<ol style="list-style-type: none"> Instansiasi <i>listDoctor</i> dari <i>ListDoctor</i> terbentuk. <i>listDoctor</i> diisi daftar dokter beserta kelas dokter tersebut

8. SSD dan OC Add Daily Visit (SSD-08)



Gambar 41 SSD Add Daily Visit

Operation contract dari *system sequence diagram Add Daily Visit* dijelaskan pada Tabel 50, Tabel 51, dan Tabel 52.

Tabel 50 *Contract CO1 requestAddDailyVisitResult*

<i>Operation</i>	requestAddDailyVisit()
<i>Cross References</i>	Use Case <i>Add Daily Visit</i>
<i>Preconditions</i>	<i>Medical Representative</i> belum memasukkan hasil kunjungan harian
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi dailyVisit dari DailyVisit terbentuk 2. Instansiasi listOfDoctor dari ListOfDoctor terbentuk 3. Atribut listOfDoctor.listOfDoctor diisi data list dokter

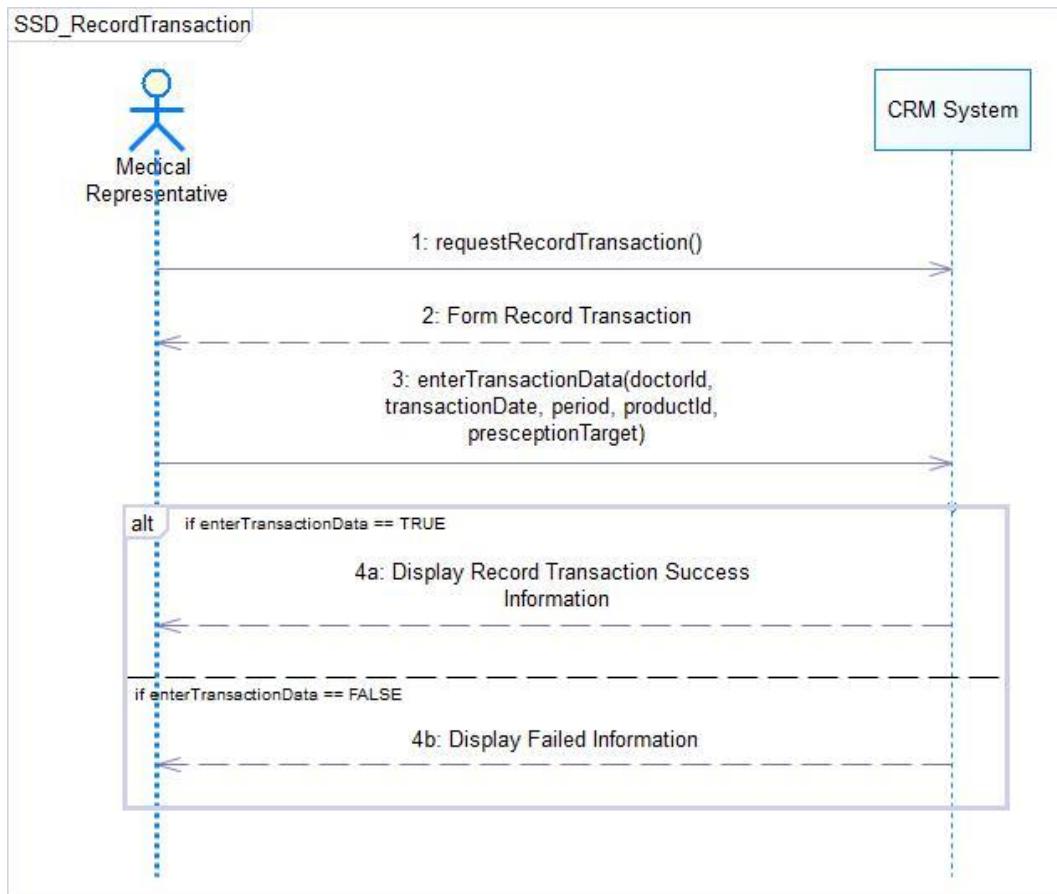
Tabel 51 *Contract CO2 chooseDoctor*

<i>Operation</i>	chooseDoctor(doctorId)
<i>Cross References</i>	Use Case <i>Add Daily Visit</i>
<i>Preconditions</i>	<i>Medical Representative</i> belum memasukkan hasil kunjungan harian
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi doctor dari Doctor terbentuk 2. Atribut doctor.doctorId diisi data doctorId

Tabel 52 *Contract CO3 enterDailyVisitResult*

<i>Operation</i>	enterDailyVisitResult(doctorId, visitDate, visitResult)
<i>Cross References</i>	Use Case <i>Add Daily Visit</i>
<i>Preconditions</i>	<i>Medical Representative</i> belum memasukkan hasil kunjungan harian
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. dailyVisit.doctorId diisi data doctorId 2. dailyVisit.visitResult diisi data visitResult 3. dailyVisit.visitDate diisi data visitDate 4. dailyVisit berasosiasi dengan MedicalRepresentative

9. SSD dan OC Record Transaction (SSD-09)



Gambar 42 SSD Record Transaction

Operation contract dari *system sequence diagram Record Transaction* dijelaskan pada Tabel 53, dan Tabel 54.

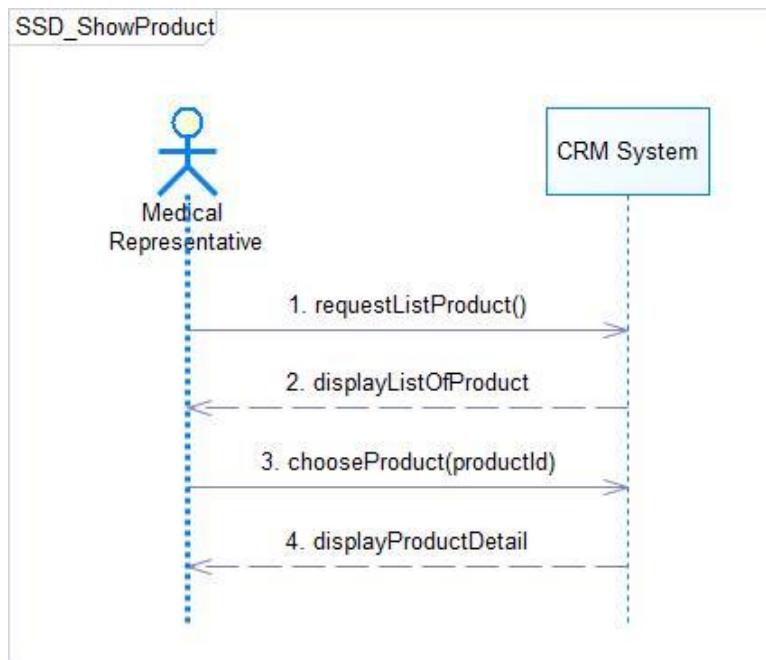
Tabel 53 Contract CO1 requestRecordTransaction

<i>Operation</i>	requestRecordTransaction()
<i>Cross References</i>	Use Case Record Transaction
<i>Preconditions</i>	Medical Representative belum memasukkan transaksi
<i>Postcondition</i>	<ol style="list-style-type: none"> Instansiasi transaction dari Transaction terbentuk Instansiasi listOfDoctor dari ListOfDoctor terbentuk Atribut listOfDoctor diisi data list dokter

Tabel 54 Contract CO2 enterTransactionData

<i>Operation</i>	enterTransactionData(doctorId, transactionDate, period, productId, prescriptionTarget)
<i>Cross References</i>	Use Case <i>Record Transaction</i>
<i>Preconditions</i>	<i>Medical Representative</i> belum memasukkan transaksi
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Atribut transaction.doctorId diisi data doctorId 2. Atribut transaction.transactionDate diisi data transactionDate 3. Atribut transaction.period diisi data period 4. Atribut transaction.productId diisi productId 5. Atribut transaction.prescriptionTarget diisi prescriptionTarget 6. transaction berasosiasi dengan MedicalRepresentative

10. SSD dan OC Show Product (SSD-10)



Gambar 43 SSD Show Product

Operation contract dari *system sequence diagram Show Product* dijelaskan pada Tabel 55, dan Tabel 56.

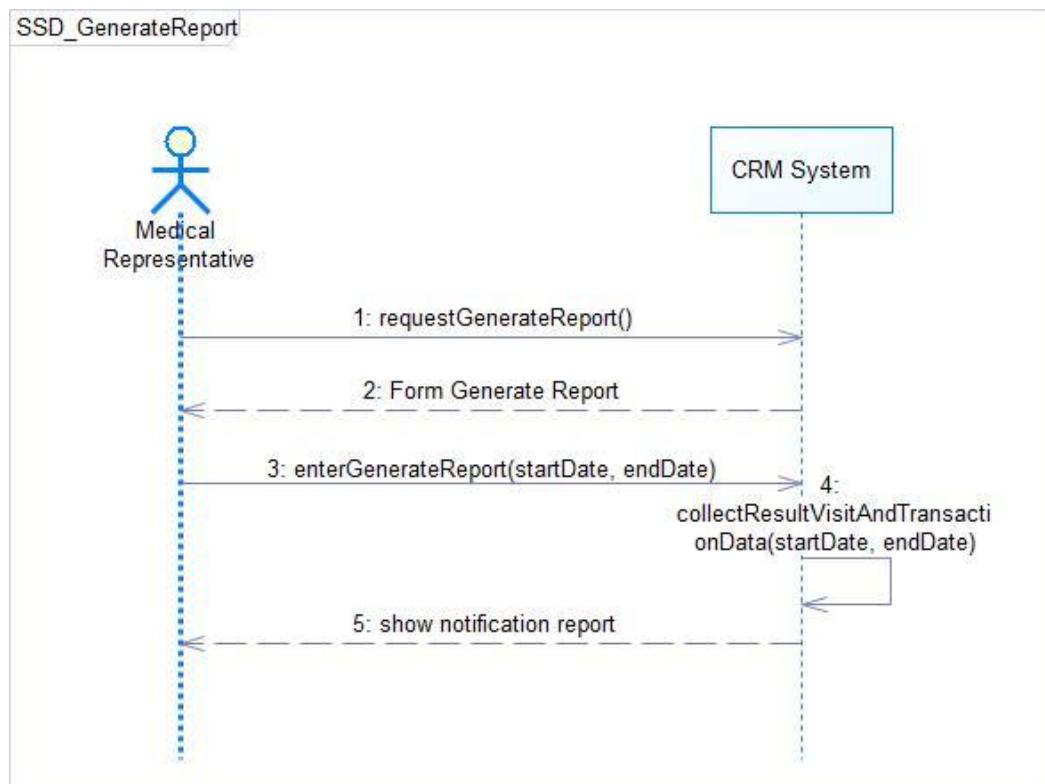
Tabel 55 *Contract CO1 requestListProduct*

<i>Operation</i>	requestListProduct()
<i>Cross Reference</i>	Use Case <i>Show Product</i>
<i>Precondition</i>	<i>None</i>
<i>Postcondition</i>	1. Instansiasi productCatalogue dari ProductCatalogue terbentuk

Tabel 56 *Contract CO2 chooseProduct*

<i>Operation</i>	chooseProduct(productId)
<i>Cross Reference</i>	Use Case <i>Show Product</i>
<i>Precondition</i>	<i>None</i>
<i>Postcondition</i>	1. Instansiasi productDesc dari ProductDescription terbentuk 2. Atribut productDesc.productId diisi productId 3. ProductCatalogue berasosiasi dengan ProductDescription sesuai productId yang sama

11. SSD dan OC *Generate Report* (SSD-11)

Gambar 44 SSD *Generate Report*

Operation contract dari *system sequence diagram Generate Report* dijelaskan pada Tabel 57, Tabel 58, dan Tabel 59.

Tabel 57 *Contract CO1 requestGenerateReport*

<i>Operation</i>	requestGenerateReport()
<i>Cross Reference</i>	Use Case <i>Generate Report</i>
<i>Precondition</i>	Sistem telah memiliki data kunjungan dan transaksi
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Instansiasi dailyVisitReport dari DailyVisitReport terbentuk. 2. Instansiasi transactionReport dari TransactionReport terbentuk.

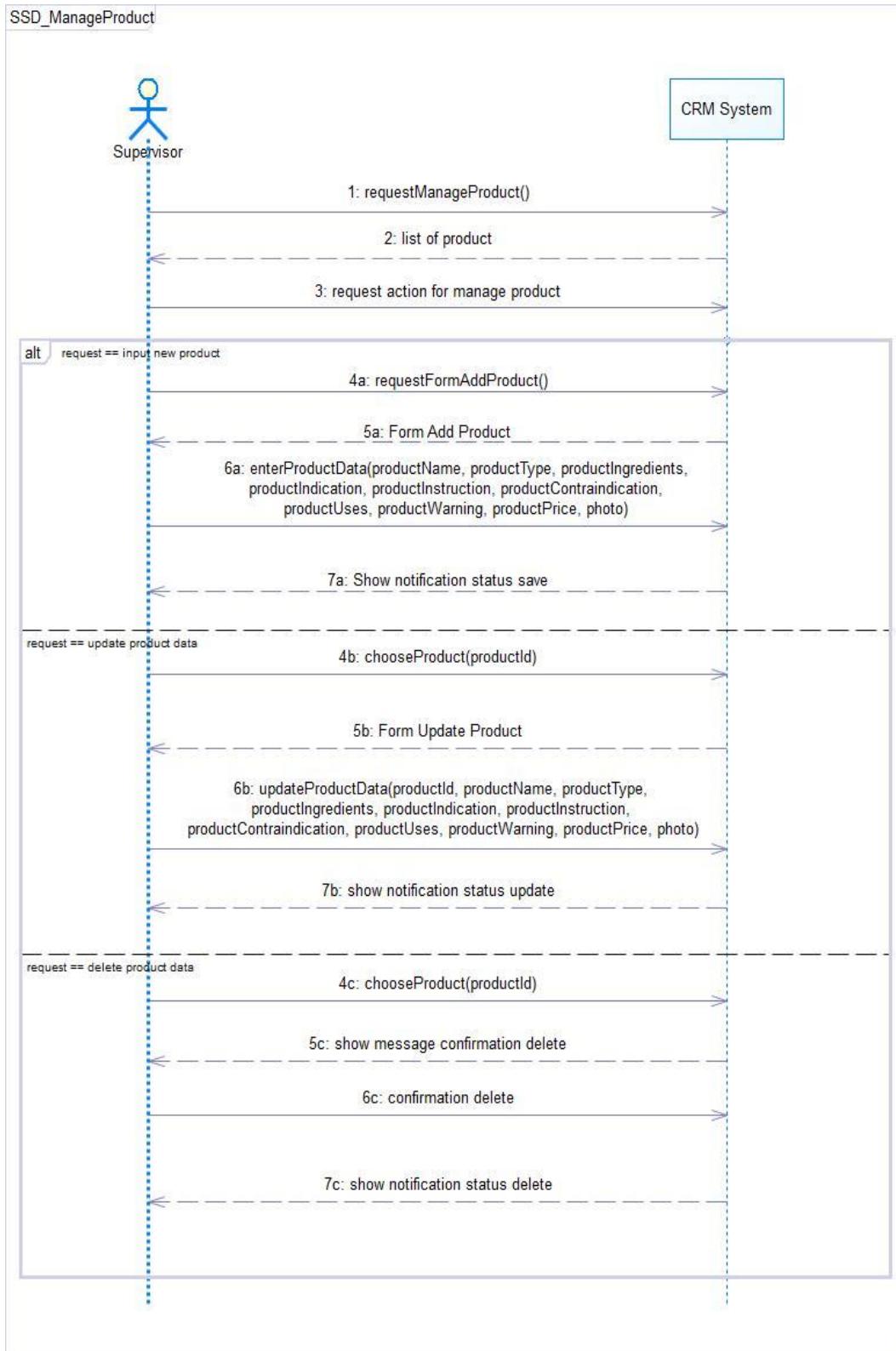
Tabel 58 *Contract CO2 enterGenerateReport*

<i>Operation</i>	enterGenerateReport(startDate, endDate)
<i>Cross Reference</i>	Use Case <i>Generate Report</i>
<i>Precondition</i>	Sudah ada data kunjungan <i>Medical Representative</i>
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Atribut dailyVisitReport.startDatePeriod diisi startDate 2. Atribut dailyVisitReport.endDatePeriod diisi endDate. 3. Atribut transactionReport.startDatePeriod diisi startDate. 4. Atribut transactionReport.endDatePeriod diisi endDate.

Tabel 59 *Contract CO3 collectResultVisitAndTransactionData*

<i>Operation</i>	collectResultVisitAndTransactionData(startDate, endDate)
<i>Cross Reference</i>	Use Case <i>Generate Report</i>
<i>Precondition</i>	<ul style="list-style-type: none"> 1. Atribut dailyVisitReport.startDatePeriod telah terisi. 2. Atribut dailyVisitReport.endDatePeriod telah terisi. 3. Atribut transactionReport.startDatePeriod telah terisi. 4. Atribut transactionReport.endDatePeriod telah terisi.
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. dailyVisitReport berasosiasi dengan dailyVisit sesuai dengan period 2. transactionReport berasosiasi dengan Transaction sesuai dengan period

12. SSD dan OC *Manage Product* (SSD-12)



Gambar 45 SSD *Manage Product*

Operation contract dari *system sequence diagram Manage Product* dijelaskan pada Tabel 60, Tabel 61, Tabel 62, Tabel 63 dan Tabel 64.

Tabel 60 *Contract CO1 requestManageProduct*

<i>Operation</i>	requestManageProduct()
<i>Cross References</i>	Use Case <i>Manage Product</i>
<i>Preconditions</i>	1. Supervisor terautentifikasi oleh sistem 2. Supervisor memiliki data Produk
<i>Postcondition</i>	1. Instansiasi listOfProduct dari ListOfProduct terbentuk

Tabel 61 *Contract CO2 requestFormAddProduct*

<i>Operation</i>	requestFormAddProduct()
<i>Cross References</i>	Use Case <i>Manage Product</i>
<i>Preconditions</i>	-
<i>Postcondition</i>	1. Instansiasi product dari Product terbentuk

Tabel 62 *Contract CO3 enterProductData*

<i>Operation</i>	enterProductData(productName, productType, productIngredients, productIndication, productInstruction, productContraindication, productUses, productWarning, productPrice, photo)
<i>Cross References</i>	Use Case <i>Manage Product</i>
<i>Preconditions</i>	1. Supervisor terautentifikasi oleh sistem. 2. Supervisor memiliki data Produk
<i>Postcondition</i>	1. Atribut product.productName diisi data productName. 2. Atribut product.productType diisi data productType. 3. Atribut product.productIngredients diisi data productIngredients. 4. Atribut product.productInstruction diisi data productInstruction. 5. Atribut product.productContraindication diisi data productContraindication. 6. Atribut product.productUses diisi data productUses. 7. Atribut product.productWarning diisi data productWarning. 8. Atribut product.productPrice diisi data productPrice. 9. Atribut product.productPicture diisi data photo. 10. Class Product berasosiasi dengan ProductDescription

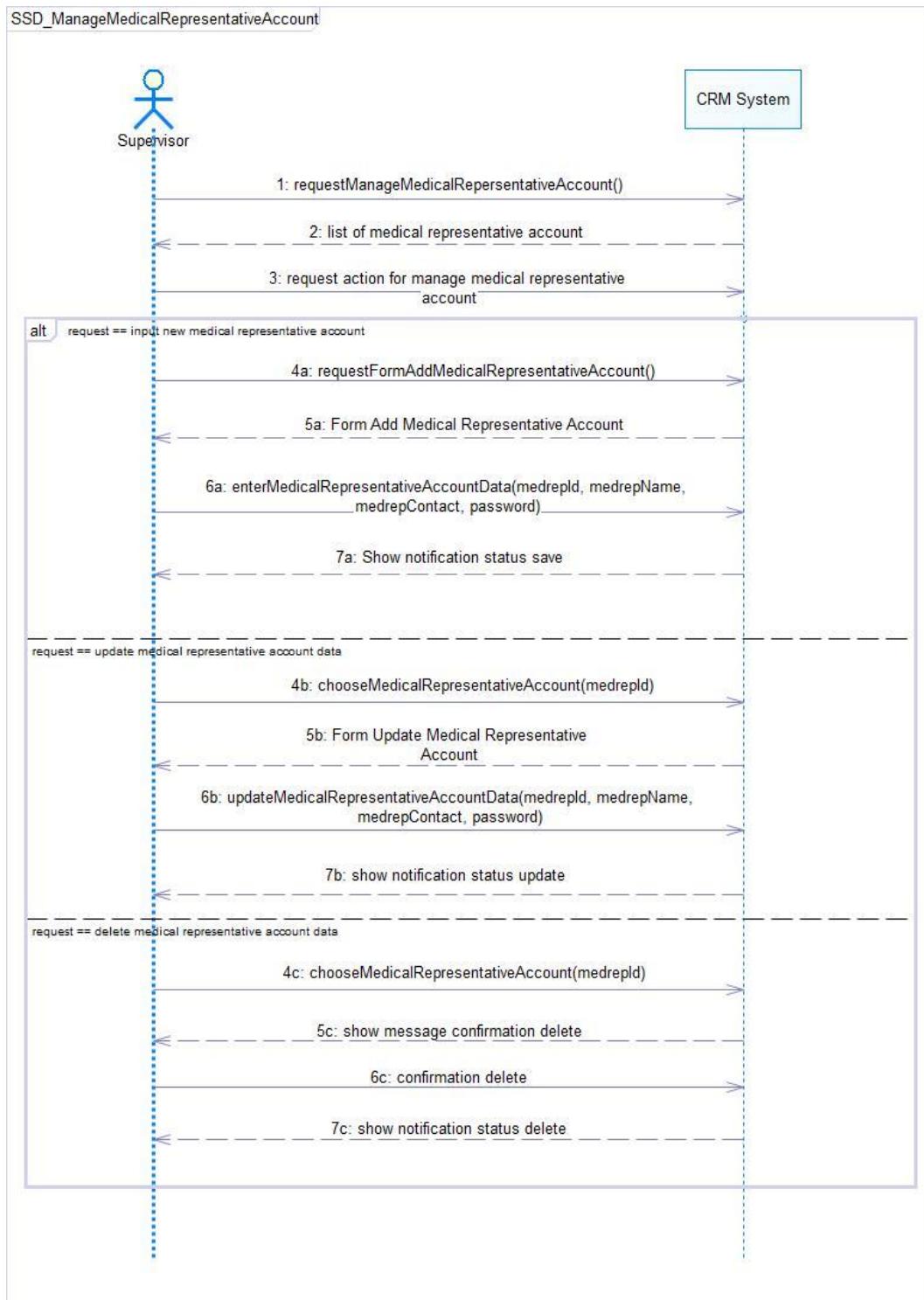
Tabel 63 *Contract CO4 chooseProduct*

<i>Operation</i>	chooseProduct(productId)
<i>Cross References</i>	Use Case <i>Manage Product</i>
<i>Preconditions</i>	Data produk sudah ada
<i>Postcondition</i>	1. Atribut product.productId diisi productId

Tabel 64 *Contract CO5 updateProductData*

<i>Operation</i>	updateProductData(productName, productType, productIngredients, productIndication, productInstruction, productContraindication, productUses, productWarning, productPrice, photo)
<i>Cross References</i>	Use Case <i>Manage Product</i>
<i>Preconditions</i>	Data produk sudah ada
<i>Postcondition</i>	<ul style="list-style-type: none"> 1. Atribut product.productName diisi data productName. 2. Atribut product.productType diisi data productType. 3. Atribut product.productIngredients diisi data productIngredients. 4. Atribut product.productInstruction diisi data productInstruction. 5. Atribut product.productContraindication diisi data productContraindication. 6. Atribut product.productUses diisi data productUses. 7. Atribut product.productWarning diisi data productWarning. 8. Atribut product.productPrice diisi data productPrice. 9. Atribut product.productPicture diisi data photo. 10. Class Product berasosiasi dengan ProductDescription

13. SSD dan OC *Manage Medical Representative Account* (SSD-13)



Gambar 46 SSD Manage Medical Representative Account

Operation contract dari *system sequence diagram* *Manage Medical Representative Account* dijelaskan pada Tabel 65, Tabel 66, Tabel 67, Tabel 68 dan Tabel 69.

Tabel 65 *Contract CO1 requestManageMedicalRepresentativeAccount*

<i>Operation</i>	requestManageMedicalRepresentativeAccount()
<i>Cross References</i>	Use Case <i>Manage Medical Representative Account</i>
<i>Preconditions</i>	<ol style="list-style-type: none"> 1. Supervisor terautentifikasi oleh sistem 2. Supervisor memiliki data akun <i>Medical Representative</i>
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi listOfMedicalRepresentative dari ListOfMedicalRepresentative terbentuk

Tabel 66 *Contract CO2 requestFormAddMedicalRepresentativeAccount*

<i>Operation</i>	requestFormAddMedicalRepresentativeAccount ()
<i>Cross References</i>	Use Case <i>Manage Medical Representative Account</i>
<i>Preconditions</i>	<ol style="list-style-type: none"> 1. Supervisor terautentifikasi oleh sistem 2. Supervisor memiliki data akun <i>Medical Representative</i>
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Instansiasi medicalRepresentative dari MedicalRepresentative terbentuk

Tabel 67 *Contract CO3 enterMedicalRepresentativeAcountData*

<i>Operation</i>	enterMedicalRepresentativeAcountData (medrepId, medrepName, medrepContact, password)
<i>Cross References</i>	Use Case <i>Manage Medical Representative Account</i>
<i>Preconditions</i>	<ol style="list-style-type: none"> 1. Supervisor terautentifikasi oleh sistem 2. Supervisor memiliki data akun <i>Medical Representative</i>
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Atribut medicalRepresentative.medrepId diisi medrepId. 2. Atribut medicalRepresentative.medrepName diisi medrepName. 3. Atribut medicalRepresentative.medrepContact diisi medrepContact. 4. Atribut medicalRepresentative.password diisi password.

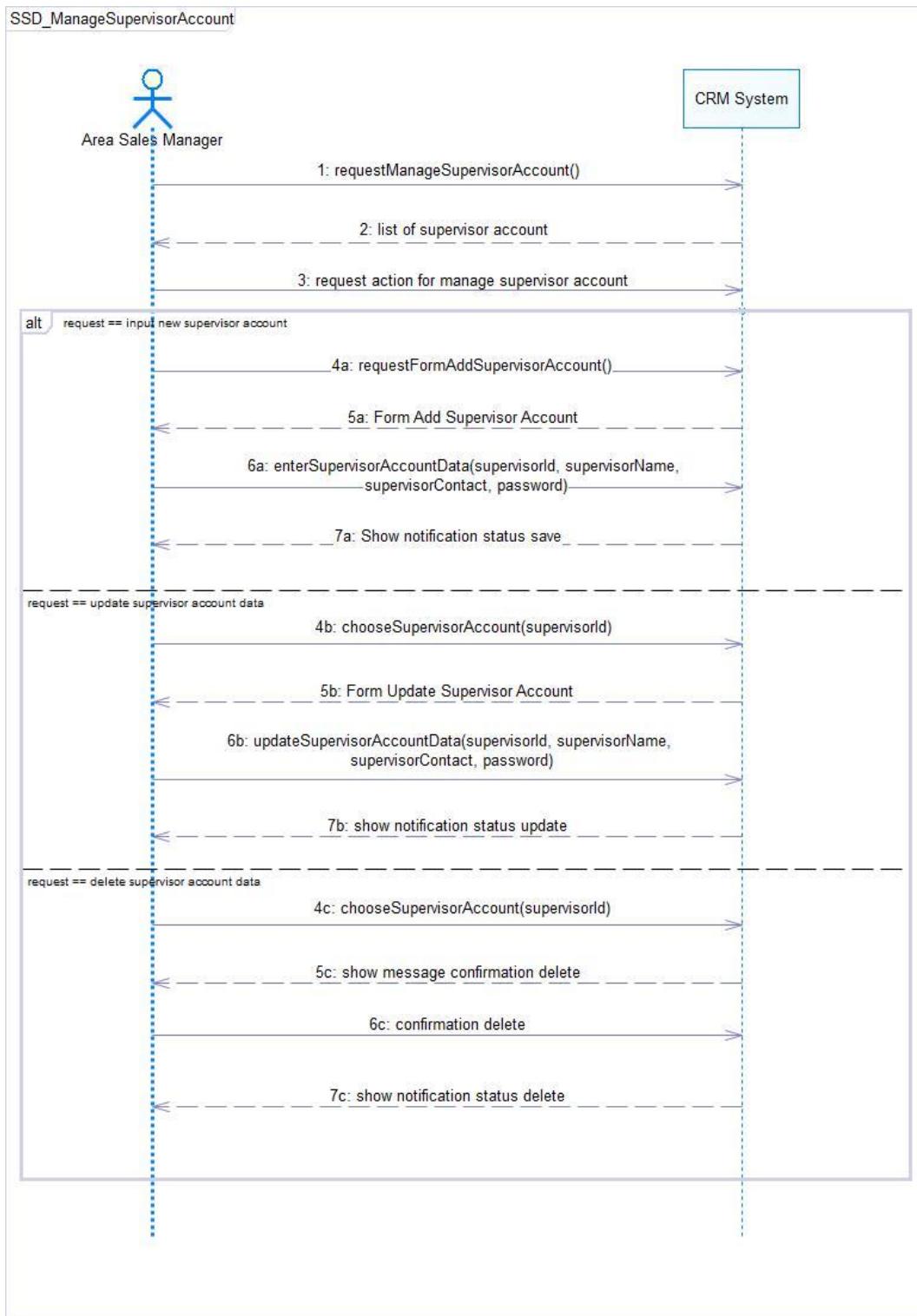
Tabel 68 *Contract CO4 chooseMedicalRepresentativeAccount*

<i>Operation</i>	chooseMedicalRepresentativeAccount(medrepId)
<i>Cross References</i>	Use Case <i>Manage Medical Representative Account</i>
<i>Preconditions</i>	Data akun <i>Medical Representative</i> sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none"> 1. Atribut medicalRepresentative.medrepId diisi medrepId

Tabel 69 *Contract CO5 updateMedicalRepresentativeAccountData*

<i>Operation</i>	updateMedicalRepresentativeAccountData(medrepId, medrepName, medrepContact, password)
<i>Cross References</i>	Use Case <i>Manage Medical Representative Account</i>
<i>Preconditions</i>	Data akun <i>Medical Representative</i> sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none">1. Atribut medicalRepresentative.medrepId diisi medrepId.2. Atribut medicalRepresentative.medrepName diisi medrepName.3. Atribut medicalRepresentative.medrepContact diisi medrepContact.4. Atribut medicalRepresentative.password diisi password.

14. SSD dan OC *Manage Supervisor Account* (SSD-14)



Gambar 47 SSD *Manage Supervisor Account*

Operation contract dari *system sequence diagram Manage Supervisor Account* dijelaskan pada Tabel 70, Tabel 71, Tabel 72, Tabel 73 dan Tabel 74.

Tabel 70 *Contract CO1 requestManageSupervisorAccount*

<i>Operation</i>	requestManageSupervisorAccount()
<i>Cross References</i>	Use Case <i>Manage Supervisor Account</i>
<i>Preconditions</i>	1. <i>Area sales manager</i> terautentifikasi oleh sistem 2. <i>Area sales manager</i> memiliki data akun Supervisor
<i>Postcondition</i>	1. Instansiasi listOfSupervisor dari ListOfSupervisor terbentuk

Tabel 71 *Contract CO2 requestFormAddSupervisorAccount*

<i>Operation</i>	requestFormAddSupervisorAccount ()
<i>Cross References</i>	Use Case <i>Manage Supervisor Account</i>
<i>Preconditions</i>	1. <i>Area sales manager</i> terautentifikasi oleh sistem 2. <i>Area sales manager</i> memiliki data akun Supervisor
<i>Postcondition</i>	1. Instansiasi supervisor dari Supervisor terbentuk

Tabel 72 *Contract CO3 enterSupervisorAcountData*

<i>Operation</i>	enterSupervisorAcountData (supervisorId, supervisorName, supervisorContact, password)
<i>Cross References</i>	Use Case <i>Manage Supervisor Account</i>
<i>Preconditions</i>	1. <i>Area sales manager</i> terautentifikasi oleh sistem 2. <i>Area sales manager</i> memiliki data akun Supervisor
<i>Postcondition</i>	1. Atribut supervisor.supervisorId diisi supervisorId. 2. Atribut supervisor.supervisorName diisi supervisorName. 3. Atribut supervisor.supervisorContact diisi supervisorContact. 4. Atribut supervisor.password diisi password.

Tabel 73 *Contract CO4 chooseSupervisorAccount*

<i>Operation</i>	chooseSupervisorAccount(supervisorId)
<i>Cross References</i>	Use Case <i>Manage Supervisor Account</i>
<i>Preconditions</i>	Data akun Supervisor sudah ada
<i>Postcondition</i>	1. Atribut supervisor.supervisorId diisi supervisorId

Tabel 74 *Contract CO5 updateSupervisorAccountData*

<i>Operation</i>	updateSupervisorAccountData(supervisorId, supervisorName, supervisorContact, password)
<i>Cross References</i>	Use Case <i>Manage Supervisor Account</i>
<i>Preconditions</i>	Data akun Supervisor sudah ada
<i>Postcondition</i>	<ol style="list-style-type: none">1. Atribut supervisor.supervisorId diisi supervisorId.2. Atribut supervisor.supervisorName diisi supervisorName.3. Atribut supervisor.supervisorContact diisi supervisorContact.4. Atribut supervisor.password diisi password.

BAB IV

PERANCANGAN

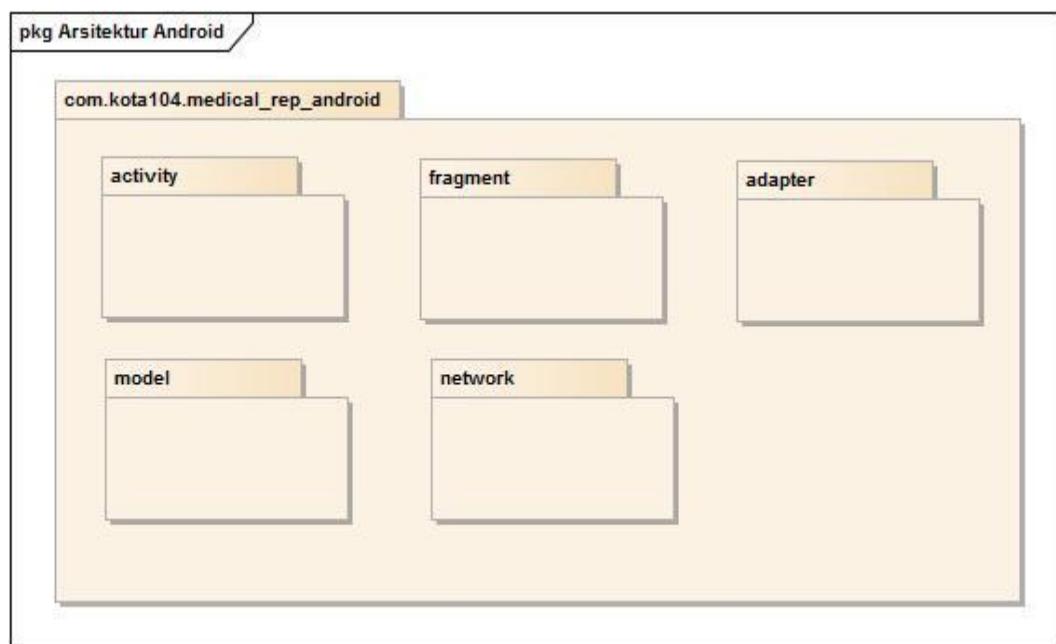
Pada bab ini dijelaskan mengenai rancangan aplikasi yang dikembangkan berdasarkan hasil analisis yang telah dilakukan pada bab sebelumnya. Rancangan ini meliputi perancangan *mobile application*, *web frontend*, *backend* dan *database*.

4.1 Perancangan *Mobile Application*

Sub bab ini dijelaskan mengenai perancangan *mobile application* yang dibangun meliputi perancangan arsitektur, *class*, interaksi antar objek yang digambarkan dengan *sequence diagram* dan *user interface*.

4.1.1 Perancangan Arsitektur

Perancangan ini dilakukan untuk memberikan gambaran mengenai arsitektur *mobile application* pada pengembangan aplikasi CRM. Arsitektur aplikasi diilustrasikan menggunakan *package diagram* seperti terlihat pada Gambar 48.



Gambar 48 Arsitektur CRM *Mobile Application/Android*

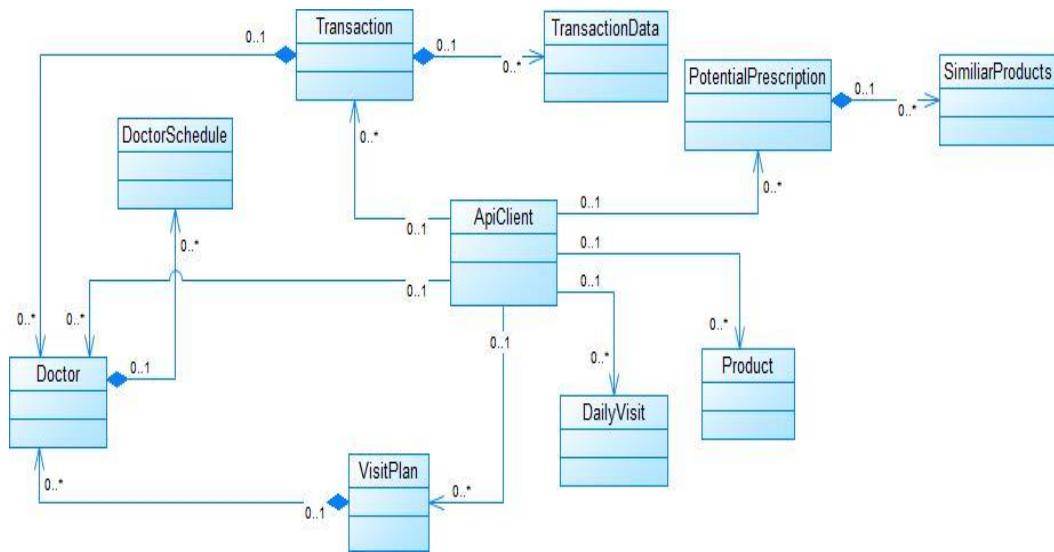
Keterangan *package diagram* pada *frontend mobile application/Android* dijelaskan pada Tabel 75

Tabel 75 Penjelasan Arsitektur CRM *Mobile Application/Android*

Nama Package	Deskripsi
activity	Berisi <i>class – class activity</i> pada <i>Android</i>
fragment	Berisi <i>class – class fragment</i> pada <i>Android</i>
adapter	Berisi <i>class – class adapter</i> untuk menampilkan data ke komponen <i>recyclerview</i>
model	Berisi <i>class – class model</i> untuk membantu pengolahan data yang akan dikirimkan atau yang akan diterima dari <i>backend</i>
network	Berisi <i>class – class</i> sebagai penunjang hubungan komunikasi dengan <i>backend</i>

4.1.2 Perancangan Class

Perancangan *class* digambarkan melalui *class diagram* untuk mengilustrasikan *class* pada aplikasi yang dikembangkan dari *class conceptual (domain model)* dengan menggambarkan *method* dan atribut dari setiap *class*. Berikut perancangan *class* untuk pengembangan aplikasi CRM pada Gambar 49 untuk *mobile application*.



Gambar 49 Class Diagram *Frontend Android*

Nama Class	Doctor		
Deskripsi	<i>Class Doctor berfungsi untuk mengelola data dokter</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorId	Private	String	Variabel untuk menampung data id dokter
medrepId	Private	String	Variabel untuk menampung data id <i>medical rep</i>
doctorName	Private	String	Variabel untuk menampung nama dokter
doctorSpecialist	Private	String	Variabel untuk menampung spesialis dokter
doctorStatus	Private	String	Variabel untuk menampung status dokter
doctorClassification	Private	String	Variabel untuk menampung kelas dokter
Method			
Nama	Visibility	Return Value	Deskripsi
Doctor(String doctorName, String doctorSpecialist, String doctorStatus, String medrepId)	Public	void	Method constructor untuk memasukkan data dokter
setDoctorName(String doctorName)	Public	void	Method untuk mengisi data nama dokter
setDoctorSpecialist(String doctorSpecialist)	Public	void	Method untuk mengisi data specialist
setDoctorStatus(String doctorStatus)	Public	void	Method untuk mengisi data status dokter
setMedrepId(String medrepId)	Public	void	Method untuk mengisi data id <i>medical rep</i>
setDoctorId(String doctorId)	Public	void	Method untuk mengisi data id dokter
setDoctorClassification(String doctorClassification)	Public	void	Method untuk mengisi data kelas dokter
getDoctorName()	Public	String	Method untuk mengambil isi dari data doctorName
getDoctorStatus()	Public	String	Method untuk mengambil isi dari data status dokter
getDoctorSpecialist()	Public	String	Method untuk mengambil isi dari data spesialis dokter
getMedrepId()	Public	String	Method untuk mengambil isi dari data id <i>medical rep</i>
getDoctorId()	Public	String	Method untuk mengambil isi dari data id dokter
getDoctorClassification()	Public	String	Method untuk megambil isi dari data kelas dokter

Nama Class	DoctorSchedule		
Deskripsi	<i>Class DoctorSchedule berfungsi untuk mengelola data jadwal dokter</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
practiceId	Private	String	Variabel untuk menampung data id jadwal praktik dokter
practiceDay	Private	String	Variabel untuk menampung data hari praktik dokter
practiceStartTime	Private	Time	Variabel untuk menampung data waktu awal praktik dokter
practiceEndTime	Private	Time	Variabel untuk menampung data waktu akhir praktik dokter
practiceLocation	Private	String	Variabel untuk menampung data lokasi praktik dokter
Method			
Nama	Visibility	Return Value	Deskripsi
DoctorSchedule(String practiceDay, String practiceStartTime, String practiceEndTime, String practiceLocation)	Public	Void	Method constructor untuk memasukkan data jadwal dokter
setPracticeDay(String practiceDay)	Public	Void	Method untuk mengisi data hari jadwal dokter
setPracticeStartTime(String practiceStartTime)	Public	Void	Method untuk mengisi data waktu awal praktik
setPracticeEndTime(String practiceEndTime)	Public	Void	Method untuk mengisi data waktu akhir praktik
setPracticeLocation(String location)	Public	Void	Method untuk mengisi data alamat praktik dokter

Nama Class	ApiClient		
Deskripsi	<i>Class ApiClient berfungsi sebagai penghubung untuk mengakses aksi pada backend</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
getDoctorsByMedrepAndStatus(String medrepID, String doctorStatus)	Public	Call<ResponseServer>	Method untuk mengakses aksi dalam mengambil data dokter berdasarkan data <i>medical rep</i> dan status dokter
deleteDoctorData(String doctorId)	Public	Call<ResponseServer>	Method untuk mengakses aksi delete
saveNewLeadDoctor(Doctor doctorData)	Public	Call<ResponseServer>	Method untuk mengakses aksi save new lead dokter

getDoctorForVisitByMedrep(String medrepId)	Public	Call<ResponseServer>	Method untuk mengakses aksi dalam mengambil data dokter berdasarkan medrepId
saveDailyVisit(DailyVisit dailyVisit)	Public	Call<ResponseServer>	Method untuk mengakses aksi simpan hasil kunjungan harian
getProductMedrepByTarget(String targetId)	Public	Call<ResponseServer>	Method untuk mengakses aksi dalam mengambil data produk berdasarkan targetId
saveTransaction(Transaction transaction)	Public	Call<ResponseServer>	Method untuk mengakses aksi simpan transaksi
savePotentialPrescription(Potential Prescription potentialPrescription)	Public	Call<ResponseServer>	Method untuk mengakses aksi simpan data besar potensi peresepatan dokter
generateReport(Date startDate, Date endDate, String medrepId)	Public	Call<ResponseServer>	Method untuk mengakses aksi generate laporan
getVisitPlanByMedrep(String medrepId)	Public	Call<ResponseServer>	Method untuk mengakses aksi dalam mengambil data rencana kunjungan berdasarkan medrepId
getDoctorByMedrepAndPracticeDay(String medrepId, String day)	Public	Call<ResponseServer>	Method untuk mengakses aksi dalam mengambil data dokter berdasarkan medrepId dan hari praktiknya
saveVisitPlan(VisitPlan visitPlan)	Public	Call<ResponseServer>	Method untuk mengakses aksi simpan data rencana kunjungan

Nama Class	DailyVisit		
Deskripsi	<i>Class DailyVisit berfungsi untuk mengelola data kunjungan harian</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
medrepId	Private	String	Variabel ini berfungsi sebagai penampung data id <i>medical rep</i>
doctorId	Private	String	Variabel ini berfungsi sebagai penampung data id dokter
visitResult	Private	String	Variabel ini berfungsi sebagai penampung data hasil kunjungan
Method			
Nama	Visibility	Return Value	Deskripsi
DailyVisit(String medrepId, String doctorId, String visitResult)	Public	Void	Method constructor ini berfungsi untuk mengisi data kunjungan harian

Nama Class	Transaction		
Deskripsi	<i>Class Transaction berfungsi untuk mengelola data transaksi</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorId	Private	String	Variabel untuk menampung data id dokter
period	Private	Int	Variabel untuk menampung data period hasil transaksi
Method			
Nama	Visibility	Return Value	Deskripsi
Transaction(doctorId, period)	Public	Void	Method constructor ini berfungsi untuk mengisi data transaksi
setTransactionData(ArrayList<TransactionData> listTransactionData)	Public	Void	Method ini berfungsi untuk mengisi list data detail transaksi

Nama Class	TransactionData		
Deskripsi	<i>Class TransactionData berfungsi untuk mengelola data detail transaksi</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productId	Private	String	Variabel untuk menampung data id produk
prescriptionTarget	Private	Int	Variabel untuk menampung data hasil kesepakatan transaksi
Method			
Nama	Visibility	Return Value	Deskripsi
TransactionData(productId, prescriptionTarget)	Public	Void	Method constructor ini berfungsi untuk mengisi data detail transaksi

Nama Class	PotentialPrescription		
Deskripsi	<i>Class PotentialPrescription berfungsi untuk mengelola data potensi peresepan dokter</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorId	Private	String	Variabel ini berfungsi untuk menampung data id dokter
numberOfPractices	Private	Integer	Variabel ini berfungsi untuk menampung data jumlah hari praktik dalam 1 minggu
numberOfPatient	Private	Integer	Variabel ini berfungsi untuk menampung data

			jumlah rata-rata pasien dalam 1 hari
Method			
Nama	Visibility	Return Value	Deskripsi
PotentialPrescription(String doctorId, int numberOfPractice, int numberOfPatient)	Public	Void	Method constructor ini berfungsi untuk mengisi data potensi peresepan dokter
setSimiliarProducts(ArrayList<SimiliarProducts> listSimiliarProduct)	Public	Void	Method ini berfungsi untuk mengisi data produk sejenis yang diresepkan dokter.
setTotalPotentialPrescription(int totalPotentialPrescription)	Public	Void	Method ini berfungsi untuk menampung data besar potensial dokter

Nama Class	SimiliarProducts		
Deskripsi	<i>Class SimiliarProduct</i> berfungsi untuk mengelola data produk sejenis		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productId	Private	String	Variabel ini berfungsi untuk menampung data id produk
quantity	Private	Integer	Variabel ini berfungsi untuk menampung data <i>quantity</i> yang tersisa atas product yang sejenis
productPrice	Private	Integer	Variabel ini berfungsi untuk menampung data harga <i>product</i> sejenis tersebut
competitor	Private	String	Variabel ini berfungsi untuk menampung data kompetitor
Method			
Nama	Visibility	Return Value	Deskripsi
SimiliarProducts(String productId, int quantity, int productPrice, String competitor)	Public	Void	Method constructor ini berfungsi untuk mengisi data obat sejenis yang diresepkan dokter

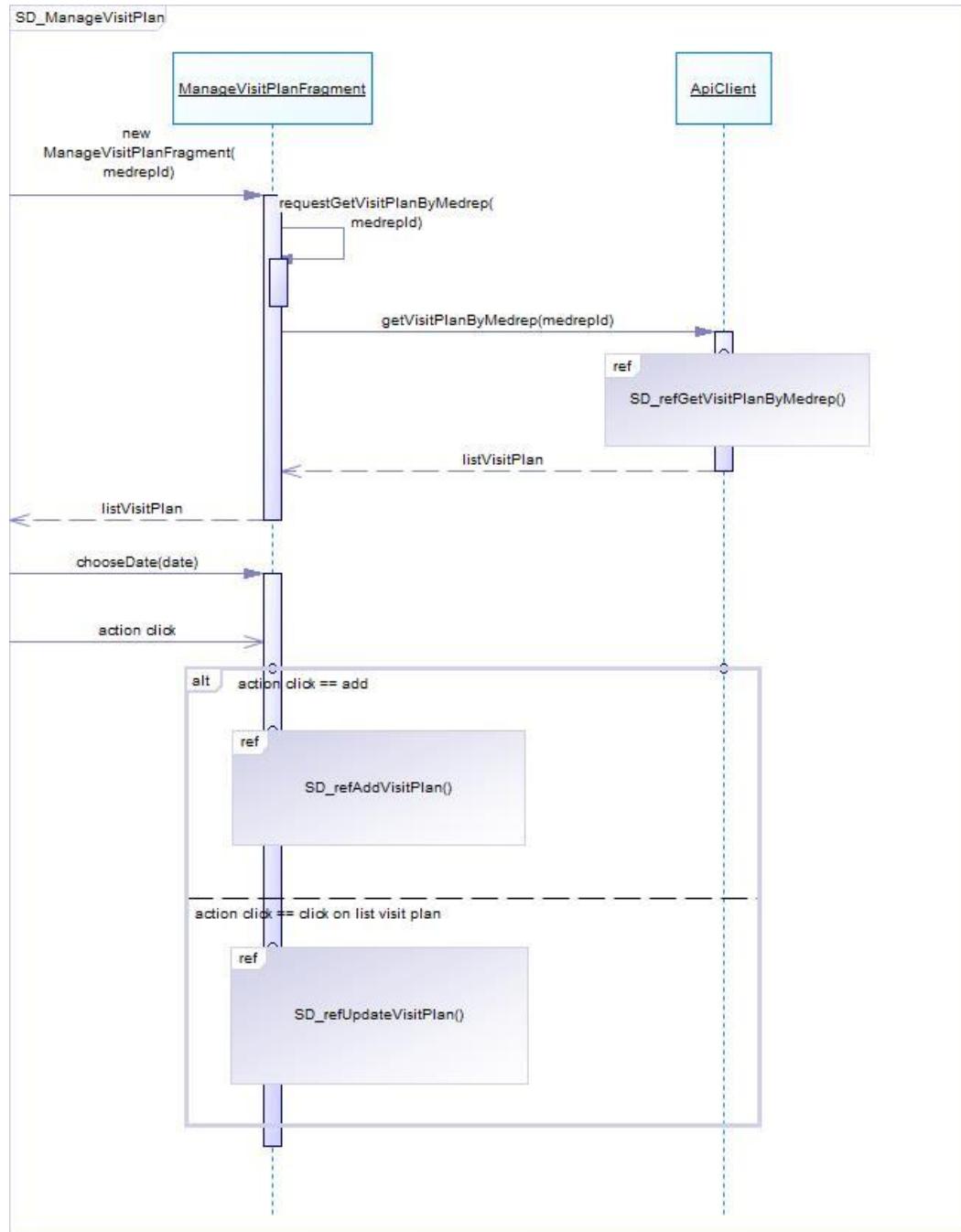
Nama Class	VisitPlan		
Deskripsi	<i>Class VisitPlan</i> berfungsi untuk mengelola data rencana kunjungan		
Attribut			
Nama	Visibility	Type Data	Deskripsi
medrepId	Private	String	Variabel ini berfungsi sebagai penampung data id <i>medical rep</i>
doctorId	Private	String	Variabel ini berfungsi sebagai penampung data id dokter

visitDate	Private	Date	Variabel ini berfungsi sebagai penampung data tanggal rencana kunjungan
visitType	Private	String	Variabel ini berfungsi sebagai penampung data tipe kunjungan
visitTime	Private	Time	Variabel ini berfungsi sebagai penampung data waktu kunjungan
reminderVisitTime	Private	Time	Variabel ini berfungsi sebagai penampung data pengingat rencana kunjungan
note	Private	String	Variabel ini berfungsi sebagai penampung data catatan dari rencana kunjungan
Method			
Nama	Visibility	Return Value	Deskripsi
VisitPlan(String doctorId, String medrepId, Date startDate, String visitType, Time visitTime, Time reminderVisitTime, String note)	Public	Void	Method constructor ini berfungsi untuk mengisi data rencana kunjungan
VisitPlan(String doctorId, String medrepId, String visitPlanId, Date startDate, String visitType, Time visitTime, Time reminderVisitTime, String note)	Public	Void	Method constructor ini berfungsi untuk mengisi data rencana kunjungan dengan tambahan attribut visitPlanId

4.1.3 Perancangan Interaksi Antarobjek

Pada sub bab ini dijelaskan mengenai perancangan interaksi antarobjek. Tujuannya untuk menggambarkan proses interaksi atau komunikasi yang terjadi diantara objek-objek pada *mobile application* dari pengembangan aplikasi CRM. Perancangan proses digambarkan dengan *sequence diagram* seperti pada Gambar 50-83 dan untuk setiap *sequence diagram* dijelaskan pada Tabel 76-108.

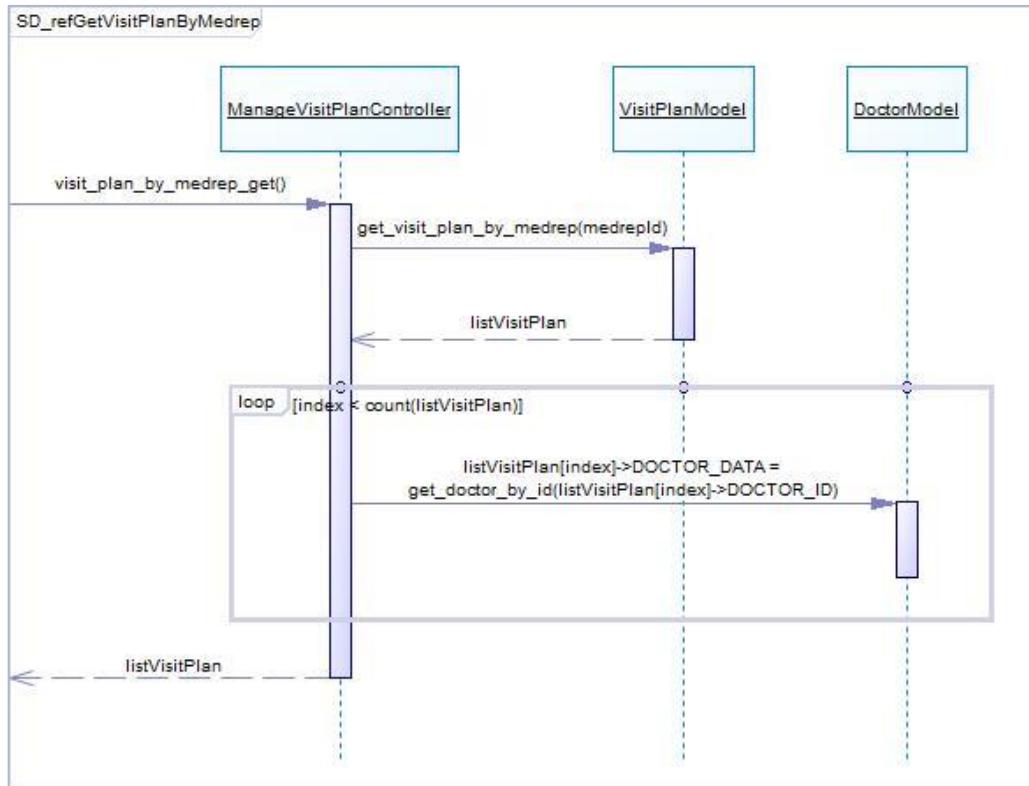
1. SD – Manage Visit Plan



Gambar 50 SD Manage Visit Plan

Tabel 76 Keterangan SD_ManageVisitPlan

ID Sequence diagram	SD-MVP-01
Nama Sequence diagram	SD_ManageVisitPlan
Nama Class Terkait	ManageVisitPlanFragment, ApiClient
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses mengelola rencana kunjungan yang dilakukan oleh <i>medical representative</i> .
Method yang terkait	requestGetVisitPlanByMedrep(medrepId), getVisitPlanByMedrep(medrepId)
Sequence diagram yang terkait	SD_refGetVisitPlanByMedrep, SD_refAddVisitPlan, SD_refUpdateVisitPlan
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String listVisitPlan : variabel arraylist bertipe VisitPlan apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer</pre> <p><u>PROCEDURE</u> requestGetVisitPlanByMedrep (medrepId)</p> <p><u>BEGIN</u></p> <p style="padding-left: 2em;">responseServer <- apiClient.getVisitPlanByMedrep (medrepId) listVisitPlan <- responseServer.getListVisitPlan()</p> <p><u>END PROCEDURE</u></p>

Gambar 51 Referensi *Visit Plan By Medrep*

Tabel 77 Keterangan SD_refGetVisitPlanByMedrep

ID Sequence diagram	SD-MVP-02
Nama Sequence diagram	SD_refGetVisitPlanByMedrep
Nama Class Terkait	ManageVisitPlanController, VisitPlanModel, DoctorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data rencana kunjungan dari database.
Method yang terkait	<code>visit_plan_by_medrep_get()</code> , <code>get_visit_plan_by_medrep(medrepId)</code> , <code>get_doctor_by_id(doctorId)</code>
Sequence diagram yang terkait	-

Algoritma

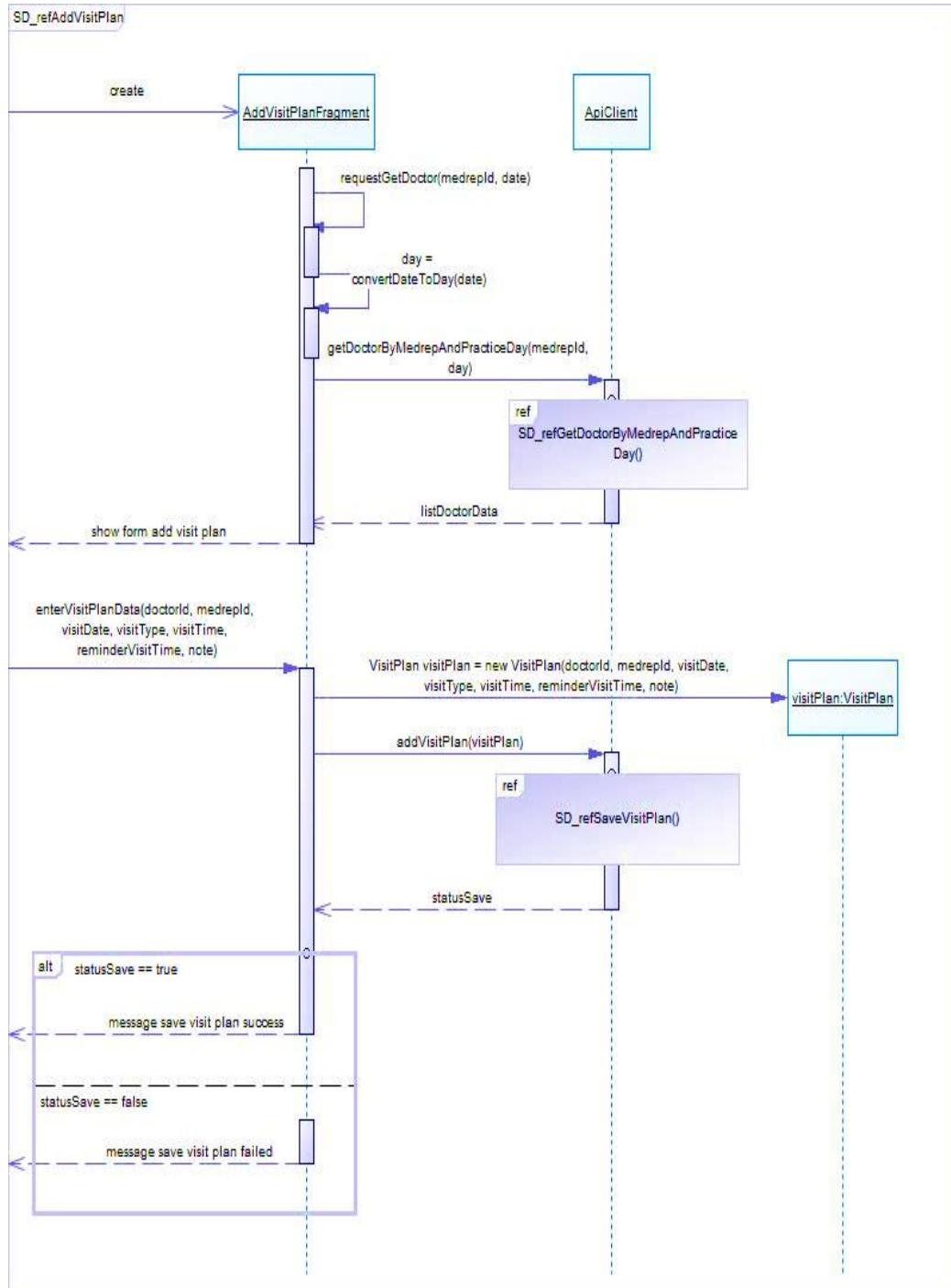
```
Kamus data
    medrepId : variabel tunggal bertipe String
    listVisitPlan : variabel array objek

PROCEDURE visit_plan_by_medrep_get()
BEGIN
    medrepId <- $this->get('medrepId')
    listVisitPlan <- $this->visitplanmodel-
        >get_visit_plan_by_medrep(medrepId)

    FOR EACH listVisitPlan AS visitPlan DO
        visitPlan->DOCTOR_DATA <-
            $this->doctorschedulemodel-
                >get_doctor_by_id(visitPlan->DOCTOR_ID)
    END FOR
END PROCEDURE

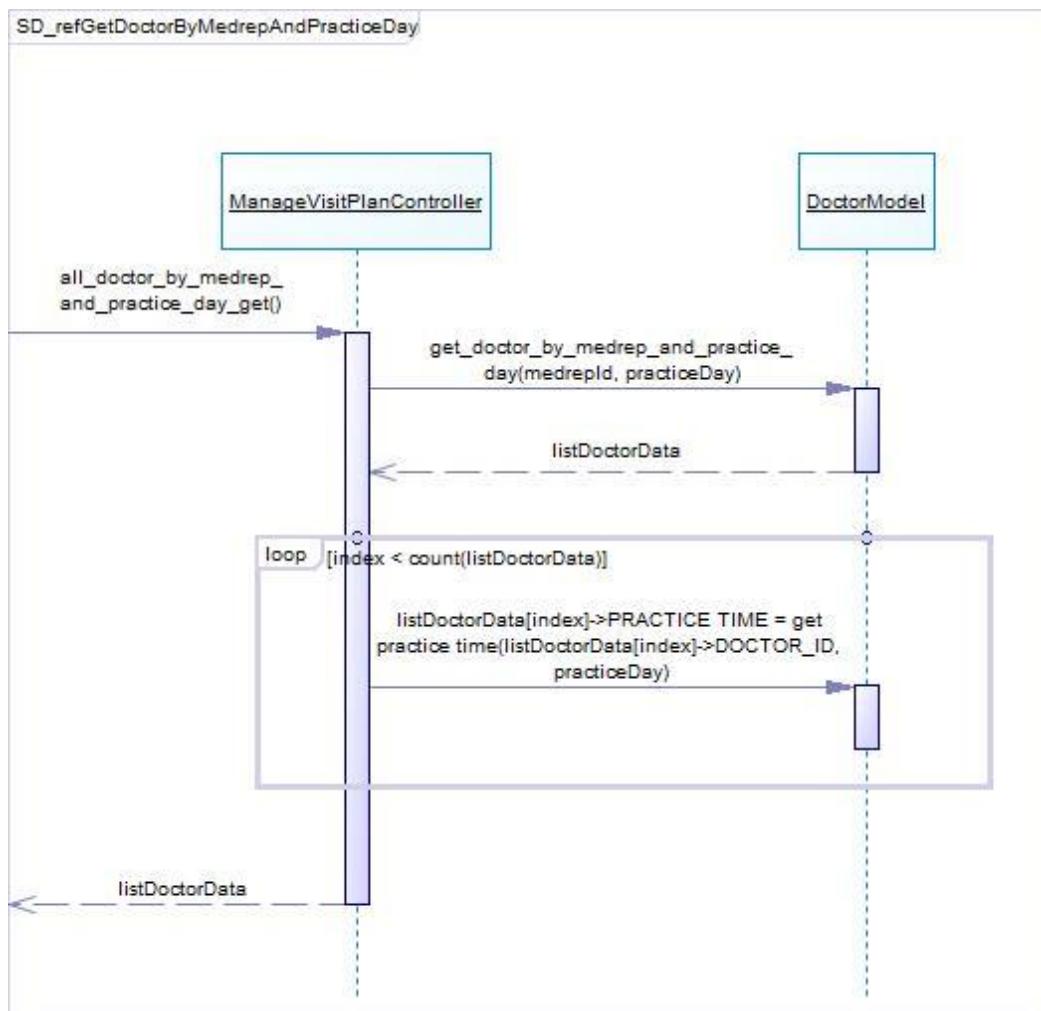
FUNCTION get_visit_plan_by_medrep(medrepId) -> array objek
BEGIN
    return $this->db->query("SELECT * FROM WHERE MEDREP_ID
        = medrepId ORDER BY VISIT_TIME")->result()
END FUNCTION

FUNCTION get_doctor_by_id(doctorId) -> array objek
BEGIN
    return $this->db->query("SELECT * FROM doctor WHERE
        DOCTOR_ID = doctorId AND DELETED = 0")->result()
END FUNCTION
```

Gambar 52 SD Referensi *Add Visit Plan*

Tabel 78 Keterangan SD_refAddVisitPlan

ID Sequence diagram	SD-MVP-03
Nama Sequence diagram	SD_refAddVisitPlan
Nama Class Terkait	AddVisitPlanFragment, ApiClient, VisitPlan
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan data rencana kunjungan ke database.
Method yang terkait	requestGetDoctor(medrepId, date), convertDateToDay(date), getDoctorByMedrepAndPracticeDay(medrepId, day), enterVisitPlanData(doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note), saveVisitPlan(visitPlan)
Sequence diagram yang terkait	SD_refGetDoctorByMedrepAndPracticeDay, SD_refSaveVisitPlan
Algoritma	
Kamus data	<pre> medrepId : variabel tunggal bertipe String day : variabel tunggal bertipe String visitPlan : objek bertipe VisitPlan listDoctorData : variabel arraylist bertipe Doctor responseServer : objek bertipe ResponseServer apiClient : objek bertipe ApiClient </pre>
	<pre> PROCEDURE requestGetDoctor (medrepId, date) BEGIN day <- convertDateToDay (date) responseServer <- apiClient.getDoctorByMedrepAndPracticeDay (medrepId, day) listDoctorData <- responseServer.getDoctorData() END PROCEDURE </pre>
	<pre> PROCEDURE enterVisitPlanData (doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note) BEGIN visitPlan <- new VisitPlan (doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note) responseServer <- apiClient.saveVisitPlan (visitPlan) IF responseServer.getStatusSave () == TRUE THEN WRITE (screen) "save visit plan success" ELSE WRITE (screen) "save visit plan failed" END IF END PROCEDURE </pre>

Gambar 53 SD Referensi *Get Doctor By Medrep And Practice Day*

Tabel 79 Keterangan SD_refGetDoctorByMedrepAndPracticeDay

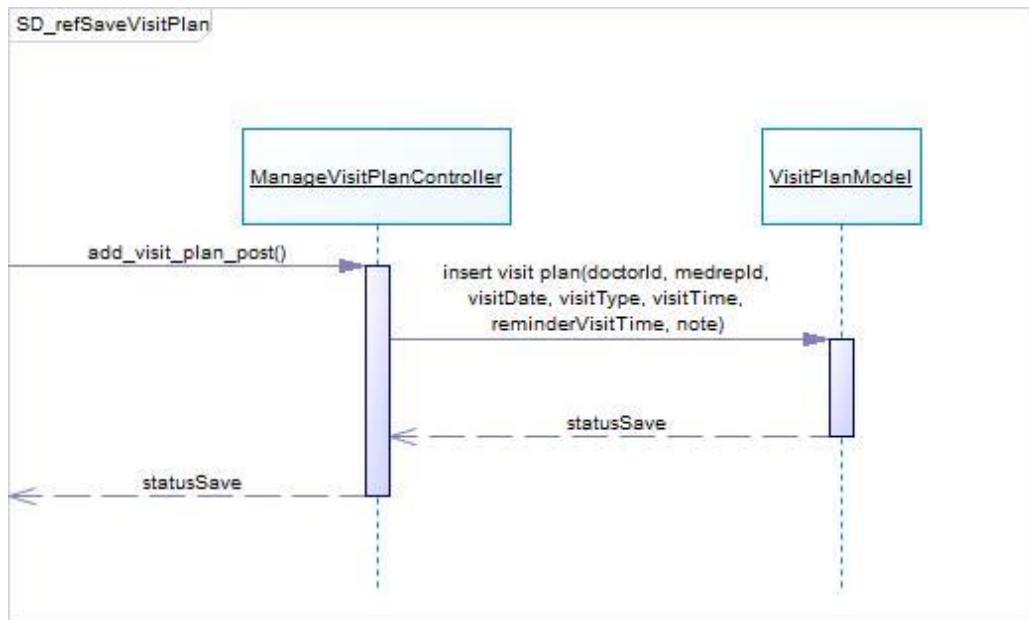
ID Sequence diagram	SD-MVP-04
Nama Sequence diagram	SD_refGetDoctorByMedrepAndPracticeDay
Nama Class Terkait	DoctorController, DoctorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data dokter ke database berdasarkan medrepId dan hari kunjungan.
Method yang terkait	<code>all_doctor_by_medrep_and_practice_day_get()</code> , <code>get_doctor_by_medrep_and_practice_day(medrepId, practiceDay)</code> , <code>get_practice_time(doctorId, practiceDay)</code>
Sequence diagram yang terkait	-
Algoritma	
Kamus data	
	<code>medrepId : variabel tunggal bertipe String</code>

```
practiceDay : variabel tunggal bertipe String
listDoctorData : variabel arraylist bertipe Doctor

PROCEDURE all_doctor_by_medrep_and_practice_day_get()
BEGIN
    medrepId <- $this->get('medrepId')
    practiceDay <- $this->get('day')
    listDoctorData <- $this->doctormodel-
        >get_doctor_by_medrep_and_practice_day(medrepId,
                                                practiceDay)
    FOR EACH listDoctorData AS doctorData DO
        doctorData->PRACTICE_TIME <- $this->doctormodel-
            >get_practice_time(doctorData->DOCTOR_ID,
                                practiceDay)
    END FOR
END PROCEDURE

FUNCTION get_doctor_by_medrep_and_practice_day(medrepId,
                                                practiceDay) -> array objek
BEGIN
    return $this->db->query("SELECT d.* FROM doctorschedule
                                ds, doctor d WHERE ds.DOCTOR_ID = d.DOCTOR_ID AND
                                ds.PRACTICE_DAY = practiceDay AND d.medrepId")-
        >result()
END FUNTION

FUNCTION get_practice_time(doctorId, practiceDay) -> array objek
BEGIN
    return $this->db->query("SELECT SCHEDULE_ID,
                                PRACTICE_DAY, PRACTICE_START_TIME, PRACTICE_END_TIME,
                                PRACTICE_LOCATION
                                FROM doctorschedule WHERE DOCTOR_ID = doctorId AND
                                PRACTICE_DAY = practiceDay")->result()
END FUNTION
```



Gambar 54 SD Referensi Save Visit Plan

Tabel 80 Keterangan SD_refSaveVisitPlan

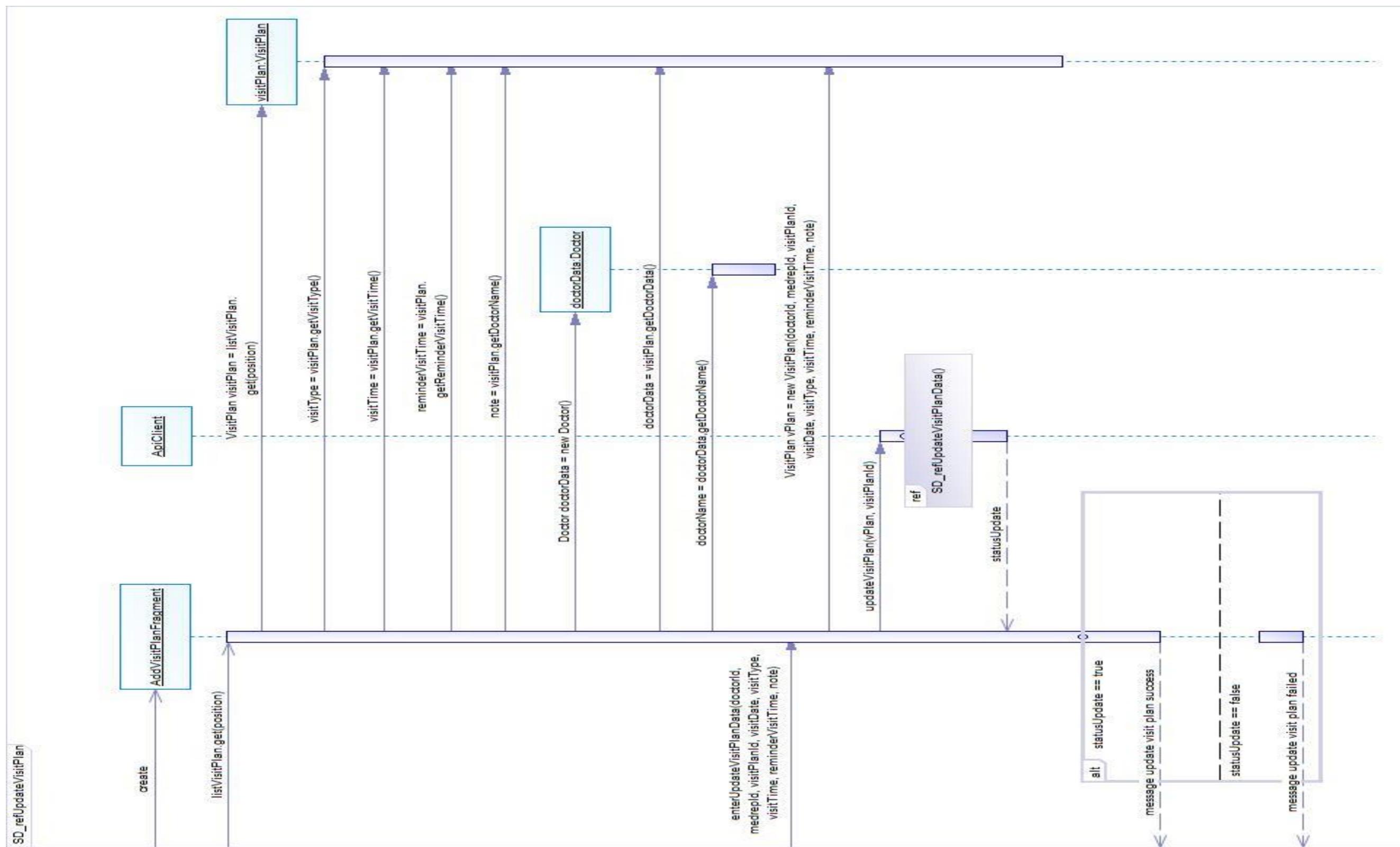
ID Sequence diagram	SD-MVP-05
Nama Sequence diagram	SD_refSaveVisitPlan
Nama Class Terkait	ManageVisitPlanController, VisitPlanModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menyimpan data visit plan ke database.
Method yang terkait	add_visit_plan_post(), insert_visit_plan(doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> visitPlan : variabel array objek </pre> <p>PROCEDURE add_visit_plan_post()</p> <p>BEGIN</p> <pre> doctorId <- \$this->post('doctorId') medrepId <- \$this->post('medrepId') visitDate <- \$this->post('visitDate') visitType <- \$this->post('visitType') visitTime <- \$this->post('visitTime') reminderVisitTime <- \$this->post('reminderVisitTime') note <- \$this->post('note') </pre>

```
statusSave <- $this->visitplanmodel->insert_visit_plan(doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note)

END PROCEDURE

FUNCTION save_visit_plan(doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note) -> boolean
BEGIN
    visitPlan <- array(
        'DOCTOR_ID' <- doctorId,
        'MEDREP_ID' <- medrepId,
        'VISIT_DATE' <- visitDate,
        'VISIT_TYPE' <- visitType,
        'VISIT_TIME' <- visitTime,
        'REMINDER_VISIT_TIME' <- reminderVisitTime,
        'NOTE' <- note
    )

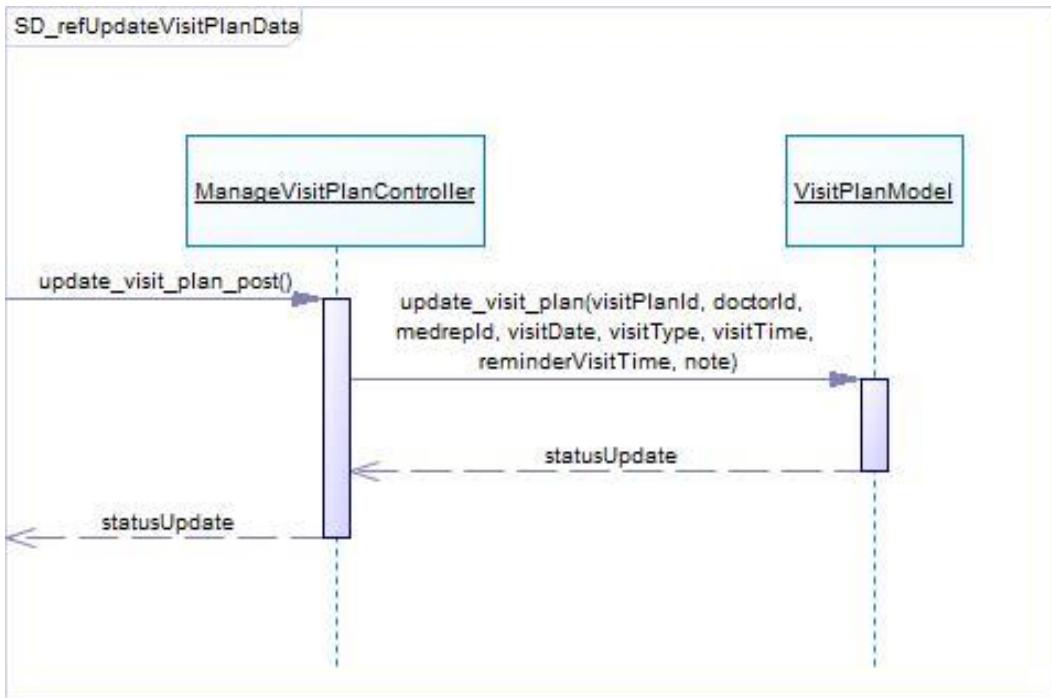
    query <- $this->db->insert('visitplan', visitPlan)
    return query
END FUNCTION
```



Gambar 55 SD Referensi Update Visit Plan

Tabel 81 Keterangan SD_refUpdateVisitPlan

ID Sequence diagram	SD-MVP-06
Nama Sequence diagram	SD_refUpdateVisitPlan
Nama Class Terkait	AddVisitPlanFragment, ApiClient, Doctor, VisitPlan
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses mengubah data rencana kunjungan ke <i>database</i> .
Method yang terkait	enterUpdateVisitPlanData(doctorId, medrepId, visitPlanId, visitDate, visitType, visitTime, reminderVisitTime, note), updateVisitPlan(visitPlan)
Sequence diagram yang terkait	SD_refUpdateVisitPlanData
Algoritma	
<p>Kamus data</p> <pre> visitPlan : objek bertipe VisitPlan responseServer : objek bertipe ResponseServer apiClient : objek bertipe ApiClient PROCEDURE enterVisitPlanData(doctorId, medrepId, visitPlanId, visitDate, visitType, visitTime, reminderVisitTime, note) BEGIN visitPlan <- new VisitPlan(doctorId, medrepId, visitPlanId, visitDate, visitType, visitTime, reminderVisitTime, note) responseServer <- apiClient.updateVisitPlan(visitPlan) IF responseServer.getStatus() == TRUE THEN WRITE (screen) "update visit plan success" ELSE WRITE (screen) "update visit plan failed" END IF END PROCEDURE </pre>	

Gambar 56 Referensi *Update Visit Plan Data*

Tabel 82 Keterangan SD_refUpdateVisitPlanData

ID <i>Sequence diagram</i>	SD-MVP-07
Nama <i>Sequence diagram</i>	SD_refUpdateVisitPlanData
Nama <i>Class</i> Terkait	ManageVisitPlanController, VisitPlanModel
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses menyimpan data <i>visit plan</i> ke <i>database</i> .
Method yang terkait	update_visit_plan_post(), update_visit_plan(visitPlanId, doctorId, medrepId, visitDate, visitType, visitTime, reminderVisitTime, note)
<i>Sequence diagram</i> yang terkait	-
Algoritma	<p>Kamus data</p> <pre> visitPlanData : variabel array objek visitPlanId : variabel tunggal bertipe integer </pre> <p>PROCEDURE save_visit_plan_post()</p> <p>BEGIN</p> <pre> visitPlanId <- \$this->post('visitPlanId') doctorId <- \$this->post('doctorId') medrepId <- \$this->post('medrepId') visitDate <- \$this->post('visitDate') visitType <- \$this->post('visitType') visitTime <- \$this->post('visitTime') </pre>

```

reminderVisitTime <- $this->post('reminderVisitTime')
note <- $this->post('note')

statusUpdate <- $this->visitplanmodel-
    >update_visit_plan(visitPlanId, doctorId, medrepId,
    visitDate, visitType, visitTime, reminderVisitTime,
    note)
END PROCEDURE

FUNCTION update_visit_plan(visitPlanId, doctorId, medrepId,
visitDate, visitType, visitTime, reminderVisitTime, note) ->
boolean
BEGIN
    visitPlan <- array(
        'DOCTOR_ID' <- doctorId,
        'MEDREP_ID' <- medrepId,
        'VISIT_DATE' <- visitDate,
        'VISIT_TYPE' <- visitType,
        'VISIT_TIME' <- visitTime,
        'REMINDER_VISIT_TIME' <- reminderVisitTime,
        'NOTE' <- note
    )

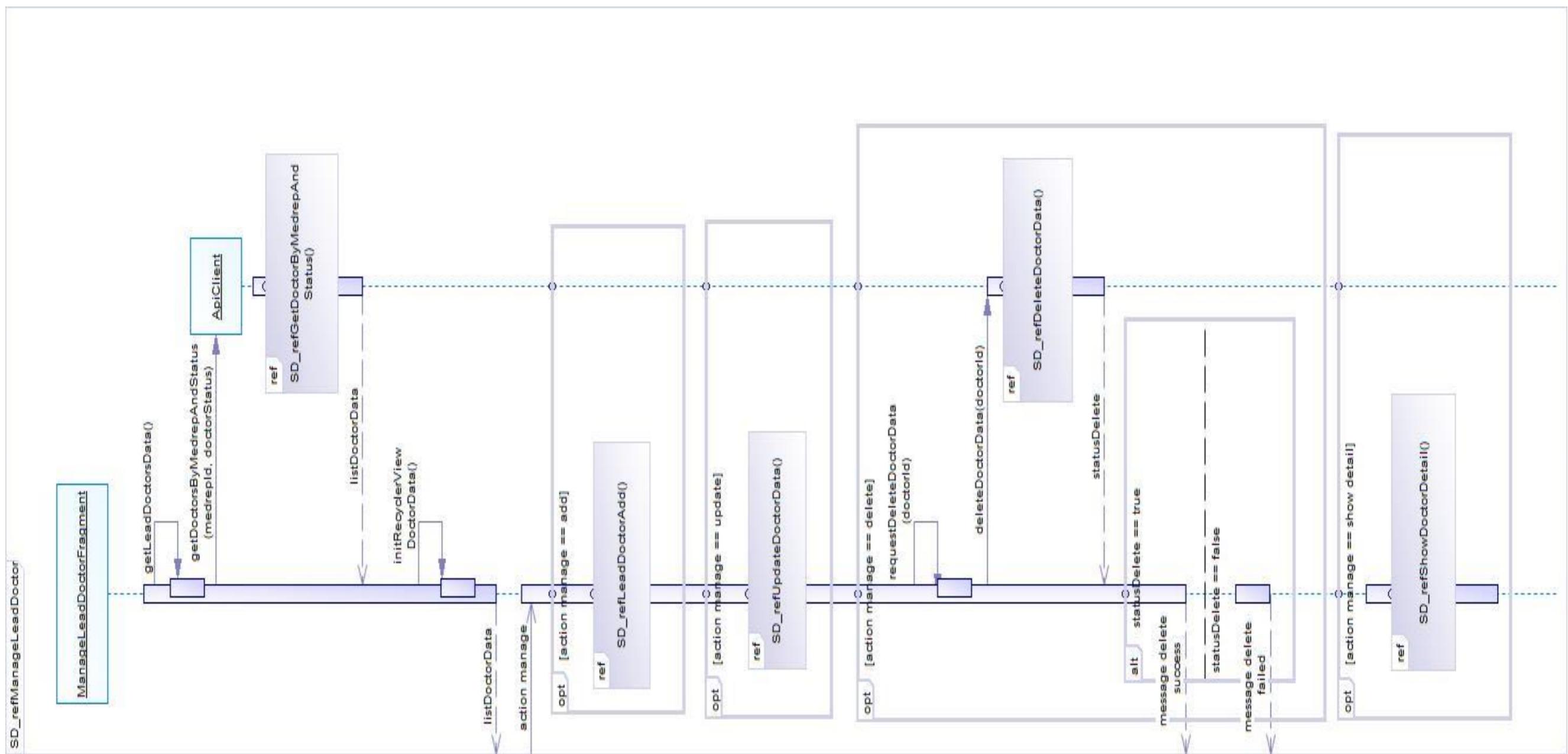
    $this->db->where('VISIT_PLAN_ID', visitPlanId)
    query <- $this->db->update('visitplan', visitPlanData)
    return query
END FUNCTION

```

2. SD – Manage Doctor Data

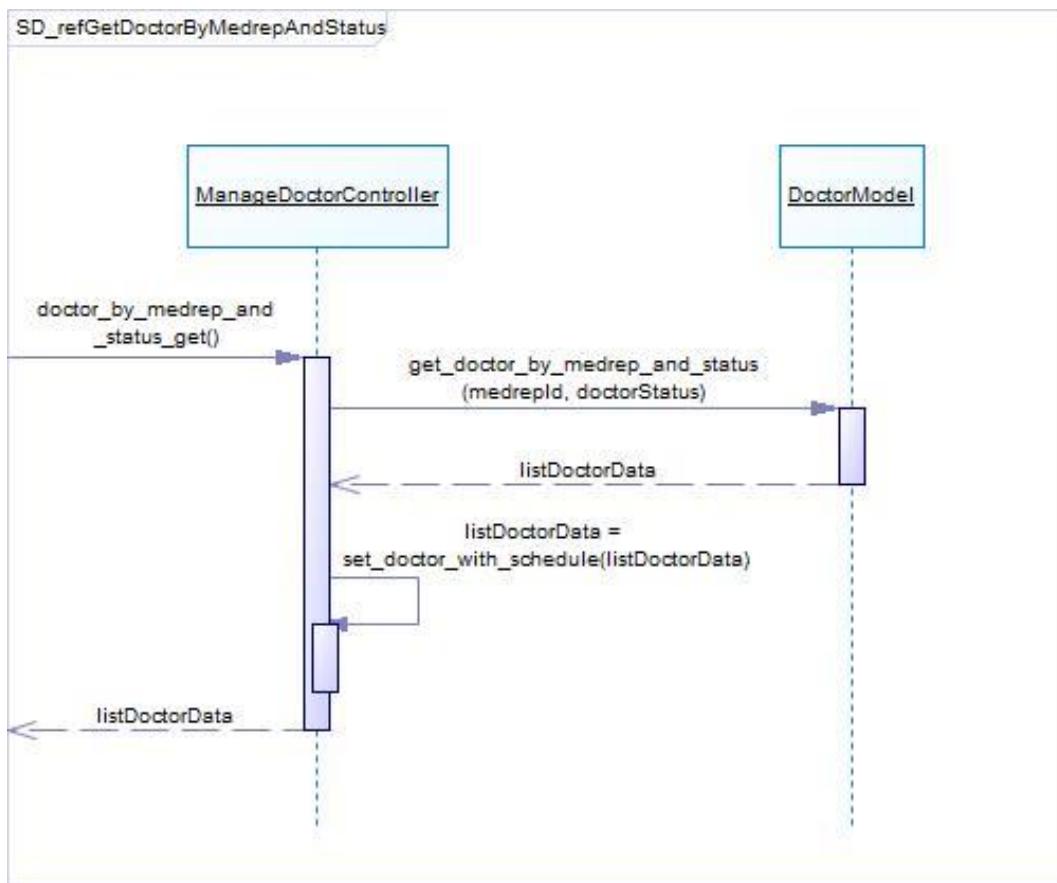


Gambar 57 SD *Manage Doctor Data*

Gambar 58 SD Referensi *Manage Lead Doctor*

Tabel 83 Keterangan SD_refManageLeadDoctor

ID Sequence diagram	SD-MD-02
Nama Sequence diagram	SD_refManageLeadDoctor
Nama Class Terkait	ManageLeadDoctorFragment, ApiClient
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses mengelola dokter yang mempunyai status “lead”.
Method yang terkait	getLeadDoctorsData(), getDoctorsByMedrepAndStatus(medrepId, doctorStatus), requestDeleteDoctor(doctorId), deleteDoctorData(doctorId)
Sequence diagram yang terkait	SD_refGetDoctorByMedrepAndStatus, SD_refLeadDoctorAdd, SD_refUpdateDoctorData, SD_refDeleteDoctorData, SD_refShowDoctorDetail
Algoritma	<p>Kamus data</p> <pre> listDoctorData : variabel arraylist bertipe Doctor medrepId : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre> <p>PROCEDURE getLeadDoctorsData()</p> <p><u>BEGIN</u></p> <ul style="list-style-type: none"> doctorStatus <- "Lead" responseServer <- apiClient.getDoctorsByMedrepAndStatus (medrepId, doctorStatus) listDoctorData <- responseServer.getDoctorsData() <p><u>END PROCEDURE</u></p> <p>PROCEDURE requestDeleteDoctorData (doctorId)</p> <p><u>BEGIN</u></p> <ul style="list-style-type: none"> responseServer <- apiClient.deleteDoctorData (doctorId) <u>IF</u> responseServer.isStatusResponse() == TRUE <u>THEN</u> <u>WRITE</u> (Screen) "delete success" getDoctorsData() <u>ELSE</u> <u>WRITE</u> (Screen) "delete failed" <u>END IF</u> <p><u>END PROCEDURE</u></p>

Gambar 59 SD Referensi *Get Doctor By Medrep and Status*

Tabel 84 Keterangan SD_refGetDoctorByMedrepAndStatus

ID Sequence diagram	SD-MD-07
Nama Sequence diagram	SD_refGetDoctorByMedrepAndStatus
Nama Class Terkait	ManageDoctorController, DoctorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data dokter dari database.
Method yang terkait	doctor_by_medrep_and_status_get(), get_doctor_by_medrep_and_status(medrepId, doctorStatus), set_doctor_with_schedule(listDoctorData)
Sequence diagram yang terkait	-
Algoritma	
Kamus data	<pre> medrepId : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String listDoctorData : variabel array objek doctor : variabel array objek index : variabel tunggal bertipe integer </pre>

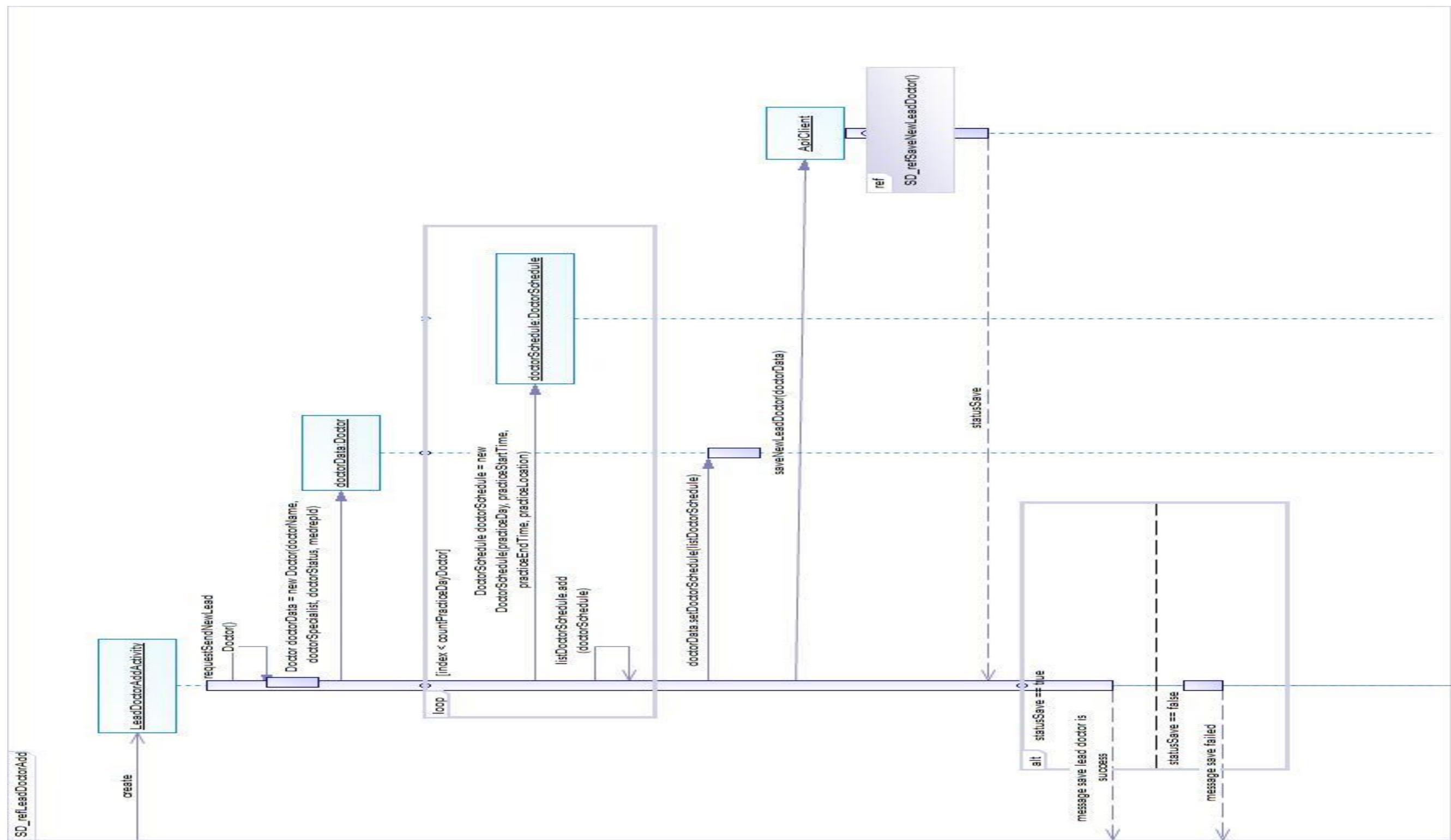
```
PROCEDURE doctor_by_medrep_and_status_get()
BEGIN
    medrepId <- $this->post("medrepId")
    doctorStatus <- $this->post("doctorStatus")
    listDoctorData <- $this->doctormodel->
        get_doctor_by_medrep_and_status(medrepId,
        doctorStatus)
    listDoctorData <- $this->
        set_doctor_with_schedule(listDoctorData)
END PROCEDURE

FUNCTION set_doctor_with_schedule(listDoctorData) -> objek list
doctor
BEGIN
    index <- 0
    FOR (index TO count(listDoctorData), index+1) DO
        listDoctorData->DOCTOR_SCHEDULE <- $this-
            >doctorschedulemodel->
            get_doctor_schedule(listDoctorData[index]->
            DOCTOR_ID)
    END FOR

    return doctor
END FUNCTION

FUNCTION get_doctor_by_medrep_and_status(medrepId, doctorStatus)
-> objek list doctor
BEGIN
    return $this->db->query("SELECT * FROM doctor WHERE
    MEDREP_ID = medrepId AND DOCTOR_STATUS = doctorStatus
    AND DELETED = 0")->result()
END FUNCTION

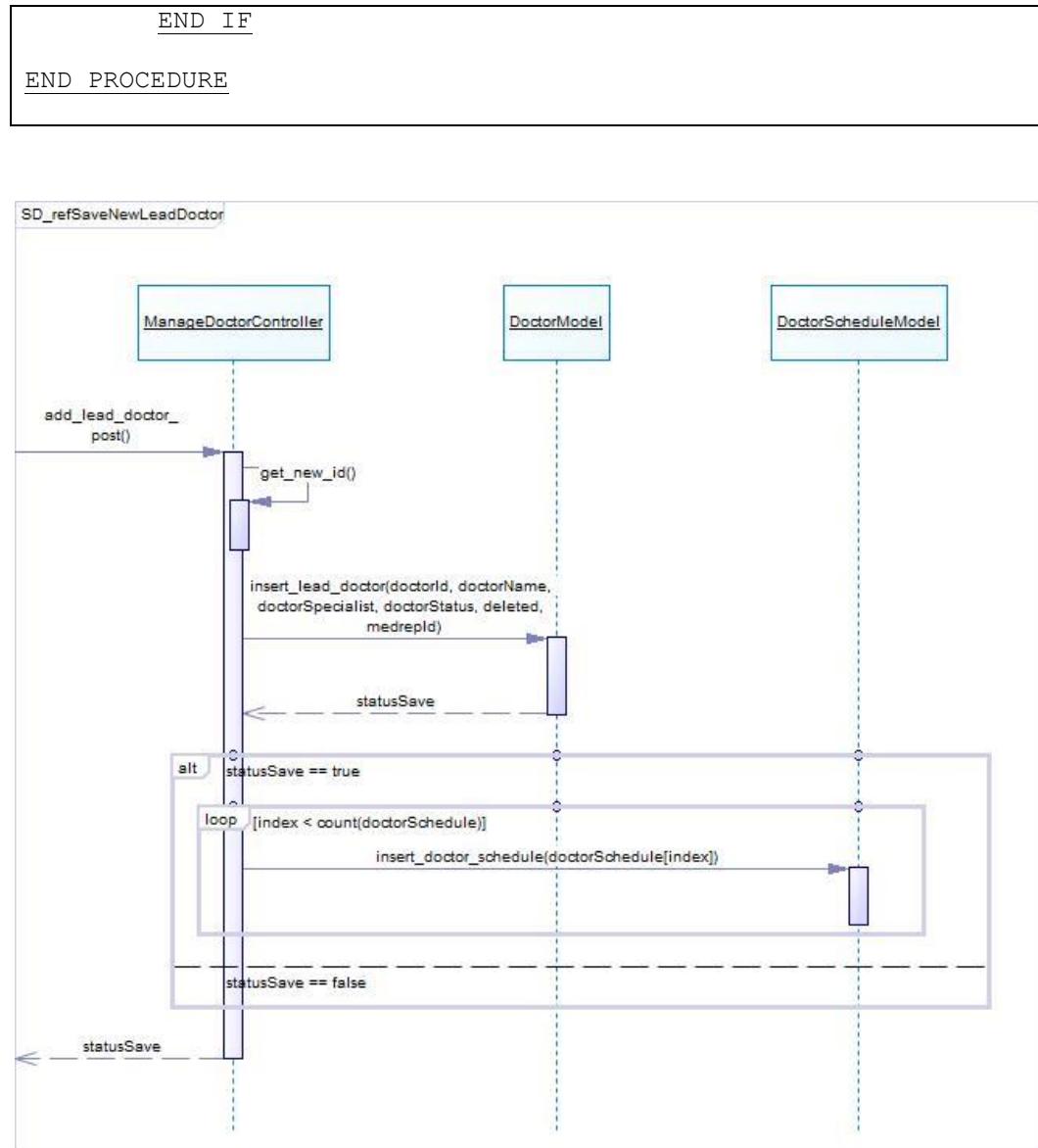
FUNCTION get_doctor_schedule(doctorId) -> objek list doctor
BEGIN
    return $this->db->query("SELECT * FROM doctorschedule
    WHERE DOCTOR_ID = doctorId")->result()
END FUNCTION
```



Gambar 60 SD Referensi Add Lead Doctor Data

Tabel 85 Keterangan SD_refLeadDoctorAdd

ID Sequence diagram	SD-MD-03
Nama Sequence diagram	SD_refLeadDoctorAdd
Nama Class Terkait	LeadDoctorAddActivity, Doctor, DoctorSchedule, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan lead dokter baru.
Method yang terkait	requestSendNewLeadDoctor(), saveNewLeadDoctor(doctorData)
Sequence diagram yang terkait	SD_refSaveNewLeadDoctor
Algoritma	<p>Kamus data</p> <pre> doctorData : objek bertipe Doctor doctorSchedule : objek bertipe DoctorSchedule listDoctorSchedule : variabel arraylist bertipe DoctorSchedule apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer doctorName : variabel tunggal bertipe String doctorSpecialist : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String practiceDays : variabel array bertipe String practiceStartTimes : variabel array bertipe String practiceEndTimes : variabel array bertipe String practiceLocations : variabel array bertipe String index : variabel tunggal bertipe integer </pre> <p>PROCEDURE requestSendNewLeadDoctor()</p> <p><u>BEGIN</u></p> <pre> index <- 0 doctorData <- new DoctorData(doctorName, doctorSpecialist, doctorStatus, medrepId) FOR (index TO practiceDays.length, index+1) DO doctorSchedule <- new DoctorSchedule(practiceDays[i], practiceStartTimes[i], practiceEndTimes[i], practiceLocations[i]) listDoctorSchedule.add(doctorSchedule) </pre> <p><u>END FOR</u></p> <pre> doctorData.setDoctorSchedules(listDoctorSchedule) responseServer <- apiClient.saveNewLeadDoctor(doctorData) IF responseServer.isStatusResponse() == TRUE THEN WRITE (Screen) "save lead doctor success" getDoctorsData() ELSE WRITE (Screen) "save failed" </pre>



Gambar 61 SD Referensi Save New Lead Doctor

Tabel 86 Keterangan SD_refSaveNewLeadDoctor

ID Sequence diagram	SD-MD-08
Nama Sequence diagram	SD_refSaveNewLeadDoctor
Nama Class Terkait	ManageDoctorController, DoctorModel, DoctorScheduleModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menyimpan data dokter ke database.
Method yang terkait	<code>add_lead_doctor_post(),</code> <code>get_new_id(),</code>

	insert_lead_doctor(doctorId, doctorName, doctorSpecialist, doctorStatus, deleted, medrepId), insert_doctor_schedule(doctorSchedule)
Sequence diagram yang terkait	-

Algoritma

Kamus data

```

statusSave : variabel tunggal bertipe boolean
doctorSchedules : variabel array objek
doctorId : variabel tunggal bertipe String
doctorName : variabel tunggal bertipe String
doctorSpecialist : variabel tunggal bertipe String
doctorStatus : variabel tunggal bertipe String
deleted : variabel tunggal bertipe integer
medrepId : variabel tunggal bertipe String

```

PROCEDURE save_lead_doctor_post()

BEGIN

```

statusSave <- FALSE
doctorId <- $this->get_new_id()
doctorName <- $this->post("DOCTOR_NAME")
doctorSpecialist <- $this-> post("DOCTOR_SPECIALIST")
doctorStatus <- "Lead"
deleted <- 0
medrepId <- $this->post("MEDREP_ID")

statusSave = $this->doctormodel-
>insert_lead_doctor(doctorId, doctorName,
doctorSpecialist, doctorStatus, deleted, medrepId)

IF statusSave == TRUE THEN
    doctorSchedules <- $this->post("DOCTOR_SCHEDULE")
    IF doctorSchedules != null THEN
        FOR EACH doctorSchedules AS schedule
            doctorSchedule <- array(
                'PRACTICE_DAY' <-
                    schedule['PRACTICE_DAY']
                'PRACTICE_START_TIME' <-
                    schedule['PRACTICE_START_TIME']
                'PRACTICE_END_TIME' <-
                    schedule['PRACTICE_END_TIME']
                'PRACTICE_LOCATION' <-
                    schedule['PRACTICE_LOCATION']
                'DOCTOR_ID' <- doctor_id
            )
            $this->doctorschedulemodel->
                insert_doctor_schedule(doctorSchedule)
        END FOR
    END IF
END IF
END PROCEDURE

```

FUNCTION save_lead_doctor(doctorId, doctorName, doctorSpecialist, doctorStatus, deleted, medrepId) -> boolean

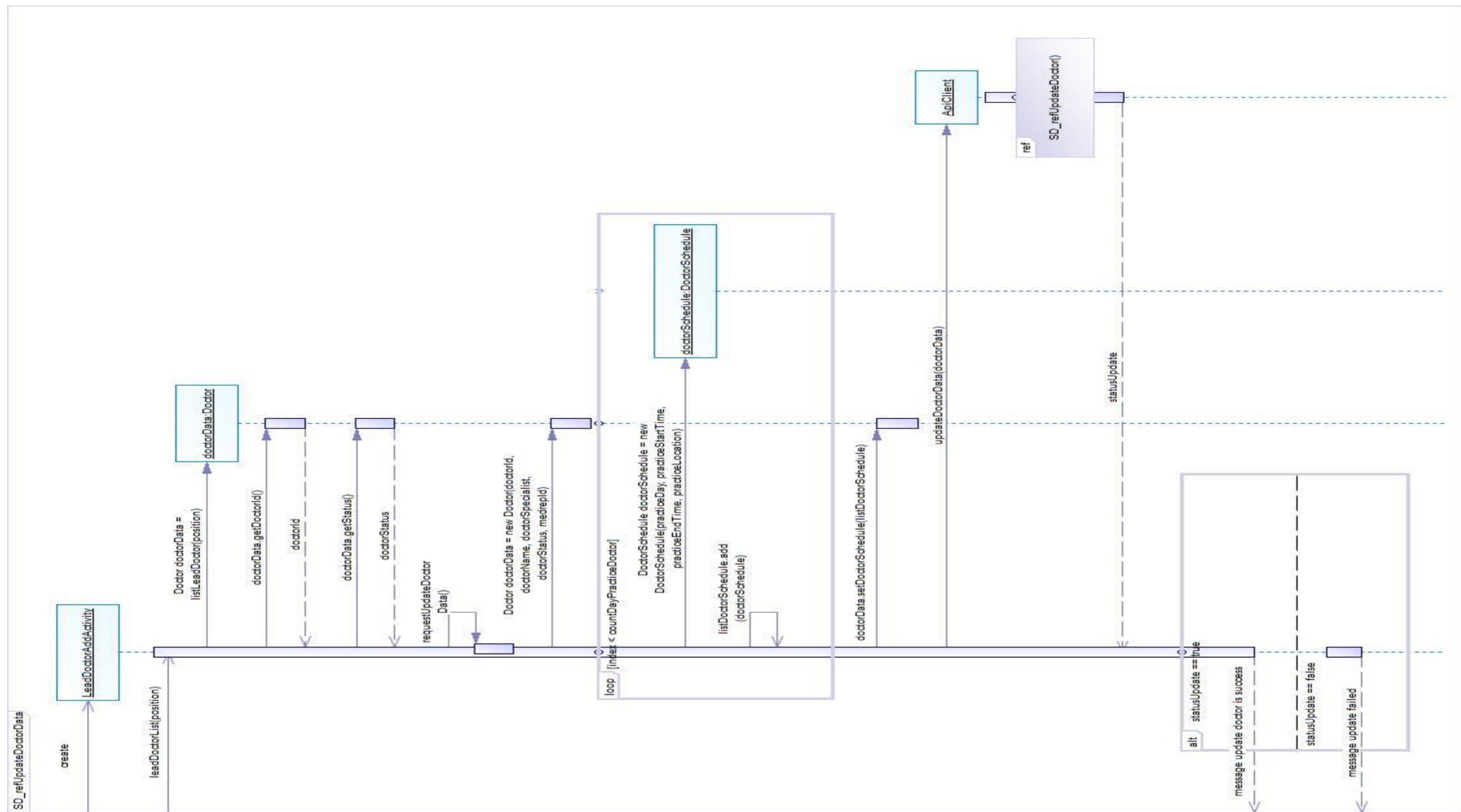
BEGIN

```
newLeadDoctorData <- array(
    'DOCTOR_ID' <- doctorId,
    'DOCTOR_NAME' <- doctorName,
    'DOCTOR_SPECIALIST' <- doctorSpecialist,
    'DOCTOR_STATUS' <- doctorStatus,
    'DELETED' <- deleted,
    'MEDREP_ID' <- medrepId
)

query <- $this->db->insert('doctor', newLeadDoctorData)
IF query == true THEN
    return TRUE
ELSE
    return FALSE
END IF
END FUNCTION

PROCEDURE insert_doctor_schedule(doctorSchedule)
BEGIN
    $this->db->insert('doctorschedule', doctorSchedule)
END PROCEDURE

FUNCTION get_new_id() -> String
BEGIN
    last_id <- $this->doctormodel->count()
    return "DR" . (last_id + 1)
END FUNCTION
```

Gambar 62 SD Referensi *Update Doctor Data*

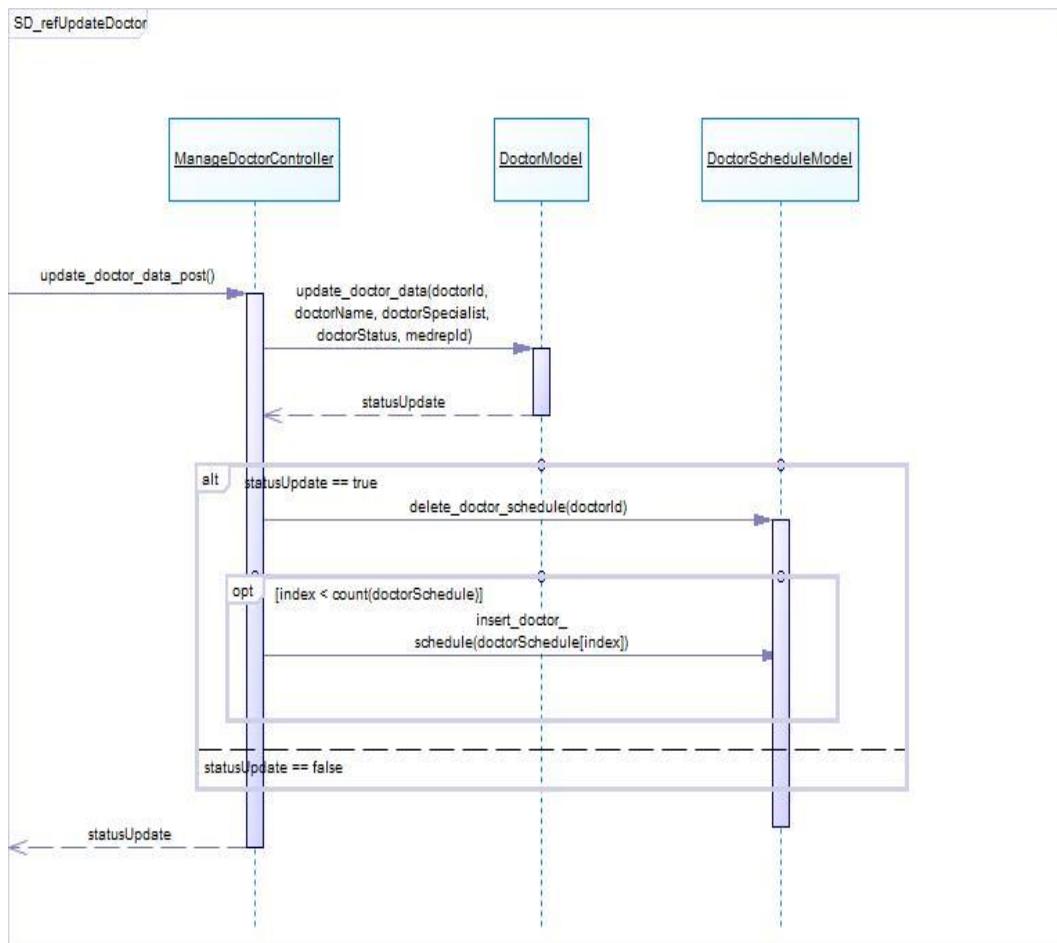
Tabel 87 Keterangan SD_refUpdateDoctorData

ID Sequence diagram	SD-MD-04
Nama Sequence diagram	SD-refUpdateDoctorData
Nama Class Terkait	LeadDoctorAddActivity, Doctor, DoctorSchedule, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengubah data dokter.
Method yang terkait	requestUpdateDoctorData() updateDoctorData(doctorData)
Sequence diagram yang terkait	SD_refUpdateDoctor
Algoritma	
<p>Kamus data</p> <pre> doctorData : objek bertipe Doctor doctorSchedule : objek bertipe DoctorSchedule listDoctorSchedule : variabel arraylist bertipe DoctorSchedule apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer doctorId : variabel tunggal bertipe String doctorName : variabel tunggal bertipe String doctorSpecialist : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String practiceDays : variabel array bertipe String practiceStartTimes : variabel array bertipe String practiceEndTimes : variabel array bertipe String practiceLocations : variabel array bertipe String index : variabel tunggal bertipe integer </pre> <p>PROCEDURE requestUpdateDoctorData()</p> <p><u>BEGIN</u></p> <pre> doctorId <- leadDoctorList.get(position).getDoctorId() doctorStatus <- leadDoctorList.get(position).getDoctorStatus() index <- 0 doctorData <- new DoctorData(doctorId, doctorName, doctorSpecialist, doctorStatus, medrepId) <u>FOR</u> (index <u>TO</u> practiceDays.length, index+1) <u>DO</u> doctorSchedule <- new DoctorSchedule(practiceDays[i], practiceStartTimes[i], practiceEndTimes[i], practiceLocations[i]) listDoctorSchedule.add(doctorSchedule) <u>END FOR</u> doctorData.setDoctorSchedules(listDoctorSchedule) responseServer <- apiClient.updateDoctorData(doctorData) <u>IF</u> responseServer.isStatusResponse() == TRUE <u>THEN</u> </pre>	

```

    WRITE (Screen) "update doctor success"
    getDoctorsData()
ELSE
    WRITE (Screen) "update failed"
END IF
END PROCEDURE

```

Gambar 63 SD Referensi *Update Doctor*

Tabel 88 Keterangan SD_refUpdateDoctor

ID Sequence diagram	SD-MD-09
Nama Sequence diagram	SD_refUpdateDoctor
Nama Class Terkait	ManageDoctorController, DoctorModel, DoctorScheduleModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menyimpan perubahan data dokter ke database.

<i>Method</i> yang terkait	update_doctor_data_post(), update_doctor_data(doctorId, doctorName, doctorSpecialist, doctorStatus, medrepId), delete_doctor_schedule(doctorId), insert_doctor_schedule(doctorSchedule)
<i>Sequence diagram</i> yang terkait	-
Algoritma	

Kamus data

```

statusUpdate : variabel tunggal bertipe boolean
doctorData : variabel array objek
doctorSchedules : variabel array objek
doctorId : variabel tunggal bertipe String
doctorName : variabel tunggal bertipe String
doctorSpecialist : variabel tunggal bertipe String
doctorStatus : variabel tunggal bertipe String
medrepId : variabel tunggal bertipe String

```

PROCEDURE update_doctor_data_post()

BEGIN

```

statusUpdate <- FALSE
doctorId <- $this->post("DOCTOR_ID")
doctorName <- $this->post("DOCTOR_NAME")
doctorSpecialist <- $this->post("DOCTOR_SPECIALIST")
doctorStatus <- $this->post("DOCTOR_STATUS")
medrepId <- $this->post("MEDREP_ID")

statusUpdate <- $this->doctormodel->
    update_doctor_data(doctorId, doctorName,
    doctorSpecialist, doctorStatus,
    medrepId)
IF statusUpdate == TRUE THEN
    doctorSchedules <- $this->post("DOCTOR_SCHEDULE")
    IF doctorSchedules != null THEN
        $this->doctormodel-
        >delete_doctor_schedule(doctorId)
        FOR EACH doctorSchedules AS schedule
            doctorSchedule <- array(
                'PRACTICE_DAY' <-
                    schedule['PRACTICE_DAY']
                'PRACTICE_START_TIME' <-
                    schedule['PRACTICE_START_TIME']
                'PRACTICE_END_TIME' <-
                    schedule['PRACTICE_END_TIME']
                'PRACTICE_LOCATION' <-
                    schedule['PRACTICE_LOCATION']
                'DOCTOR_ID' <- doctor_id
            )
            $this->doctorschedulemodel->
                insert_doctor_schedule(doctorSchedule)
        END FOR
    END IF
END IF

```

END PROCEDURE

```

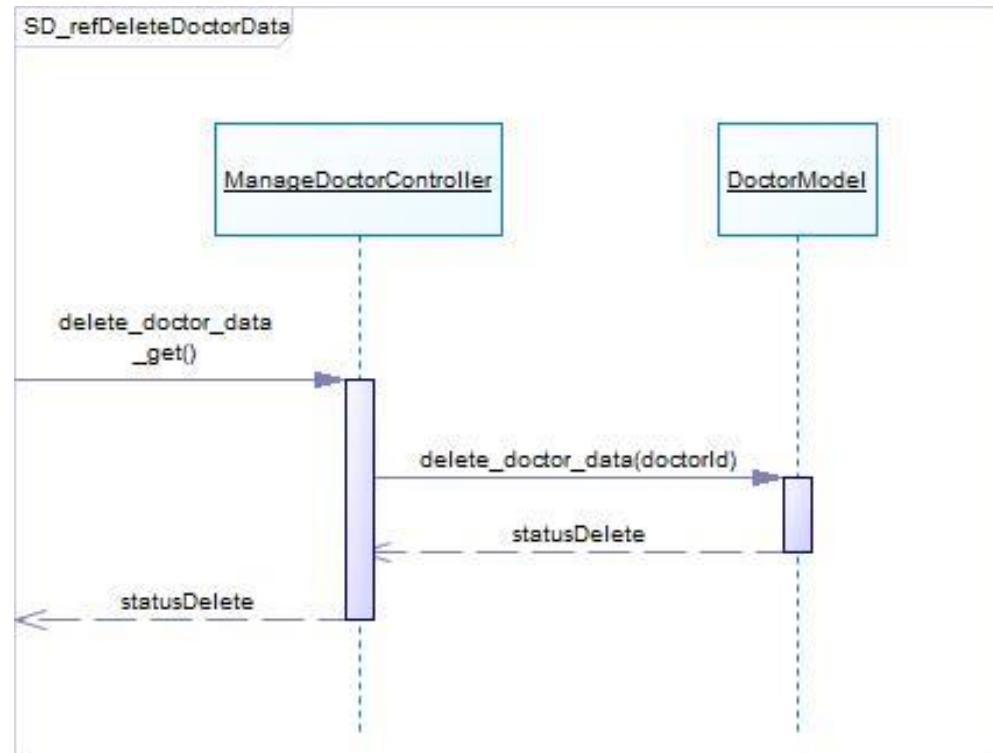
FUNCTION update_doctor_data(doctorId, doctorName,
doctorSpecialist, doctorStatus, medrepId) -> boolean
BEGIN
    doctorData <- array(
        'DOCTOR_NAME' <- doctorName,
        'DOCTOR_SPECIALIST' <- doctorSpecialist,
        'DOCTOR_STATUS' <- doctorStatus,
        'MEDREP_ID' <- medrepId
    )

    $this->db->where('DOCTOR_ID', doctorId)
    query <- $this->db->update('doctor', doctorData)
    IF query == true THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

PROCEDURE insert_doctor_schedule(doctorSchedule)
BEGIN
    $this->db->insert('doctorschedule', doctorSchedule)
END PROCEDURE

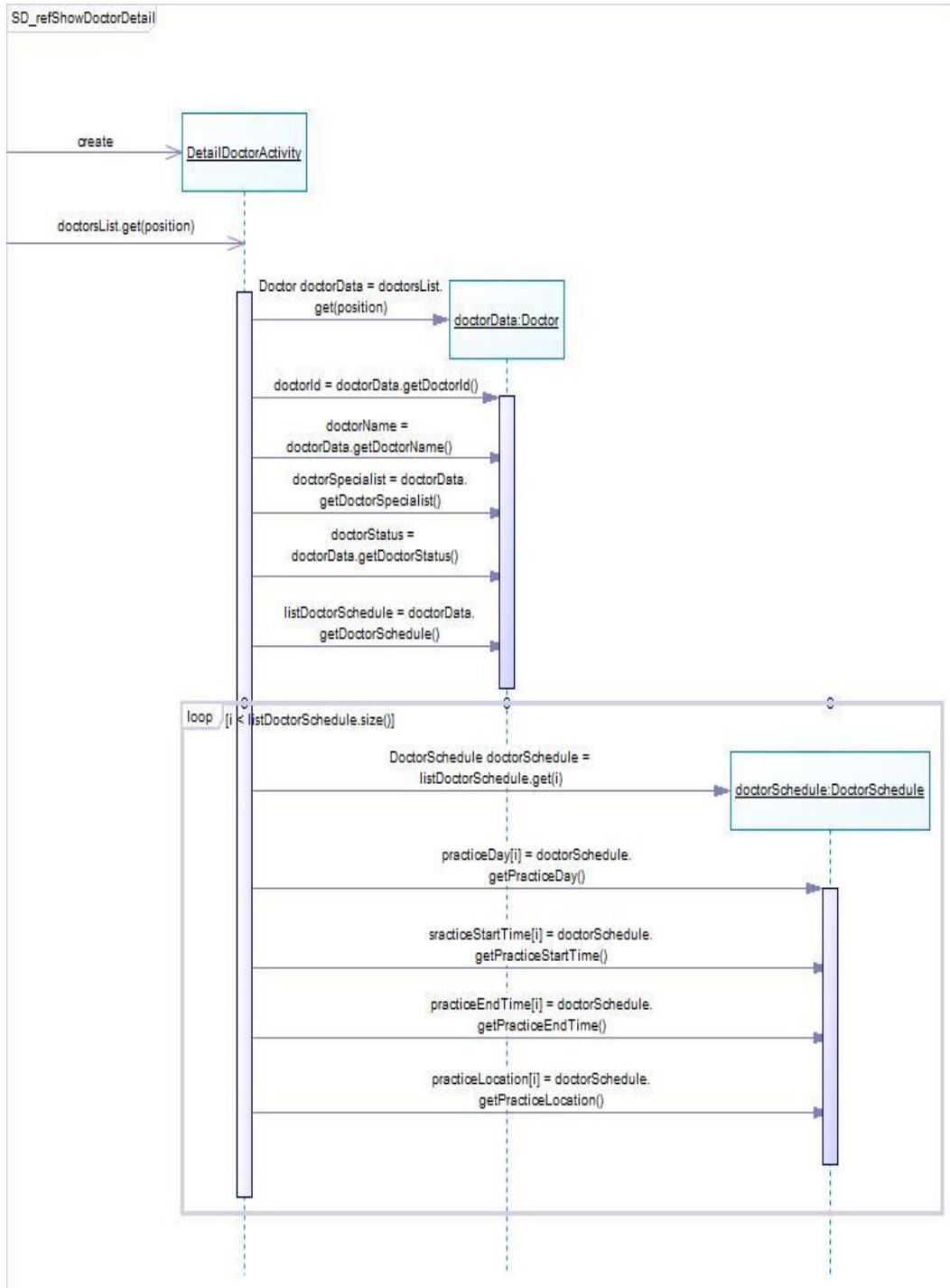
PROCEDURE delete_doctor_schedule(doctorId)
BEGIN
    $this->db->query("DELETE FROM doctorschedule WHERE
        DOCTOR_ID = doctorId")
END PROCEDURE

```

Gambar 64 Referensi *Delete Doctor Data*

Tabel 89 Keterangan SD_refDeleteDoctorData

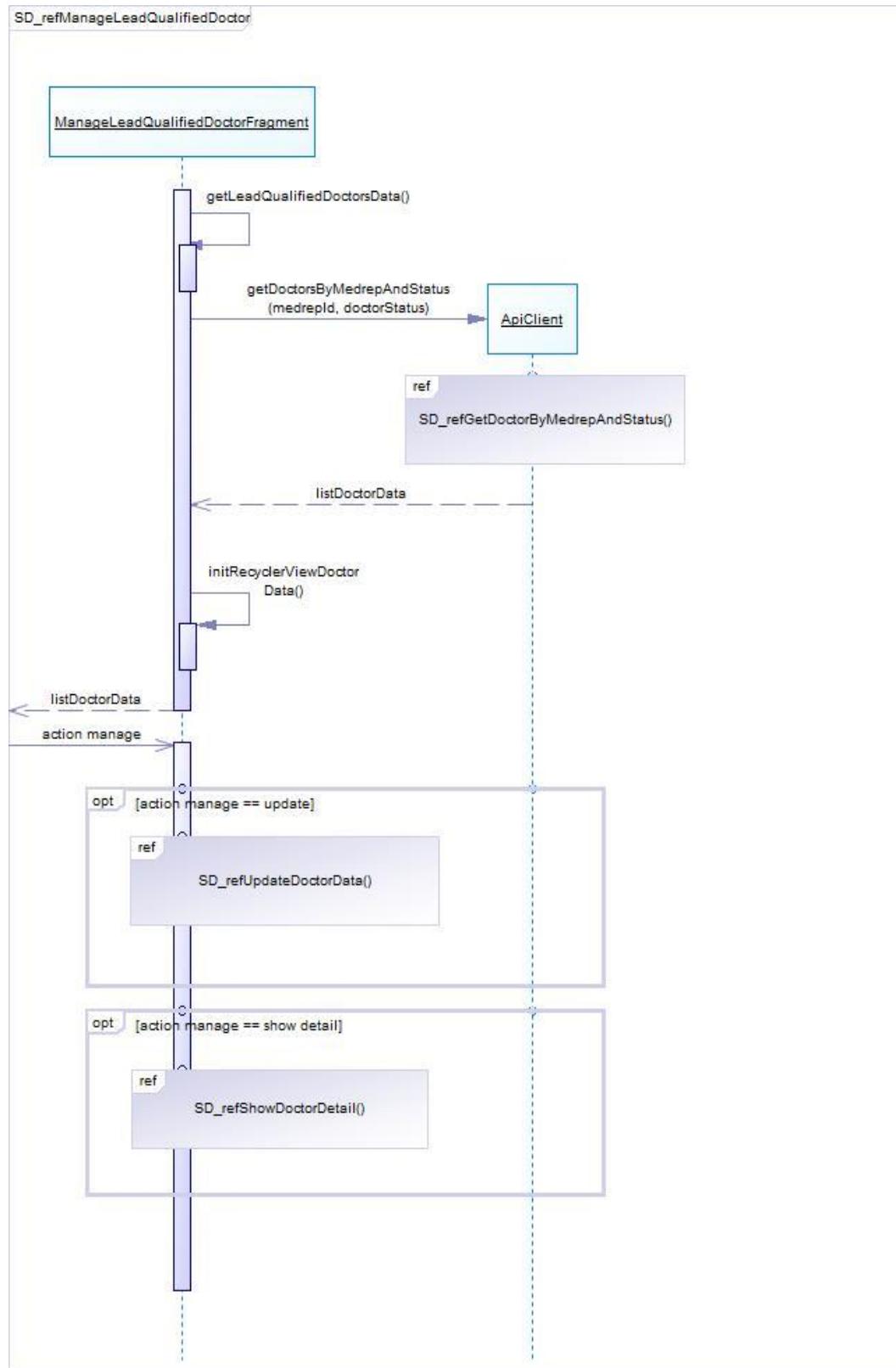
ID Sequence diagram	SD-MD-10
Nama Sequence diagram	SD_refDeleteDoctorData
Nama Class Terkait	ManageDoctorController, DoctorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses manipulasi hapus data dokter pada database.
Method yang terkait	delete_doctor_data_get(), delete_doctor_data(doctorId)
Sequence diagram yang terkait	-
Algoritma	
<pre> Kamus data statusDelete : variabel tunggal bertipe boolean doctorId : variabel tunggal bertipe String PROCEDURE delete_doctor_data_get() BEGIN doctorId <- \$this->post("DOCTOR_ID") statusDelete <- \$this->doctormodel-> delete_doctor_data(doctorId) END PROCEDURE FUNCTION delete_doctor_data(doctorId) -> boolean BEGIN query <- \$this->db->query("UPDATE doctor SET DELETED = 1 WHERE DOCTOR_ID = doctorId") IF query == true THEN return TRUE ELSE return FALSE END IF END FUNCTION </pre>	



Gambar 65 Referensi Show Doctor's Detail

Tabel 90 Keterangan SD_refShowDoctorDetail

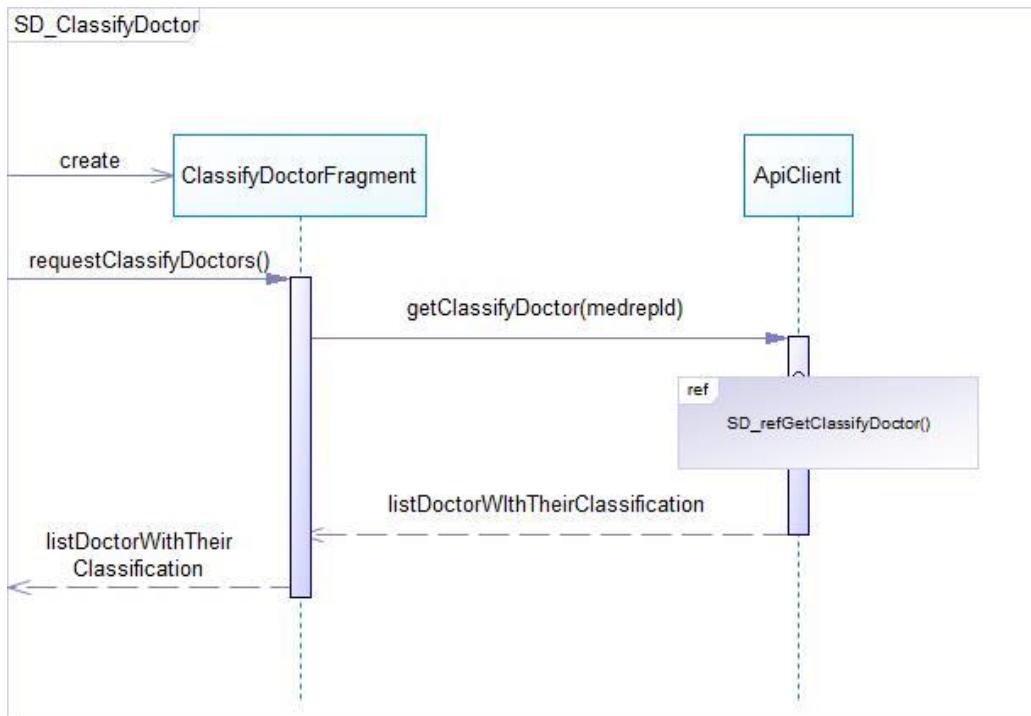
ID Sequence diagram	SD-MD-05
Nama Sequence diagram	SD_refShowDoctorDetail
Nama Class Terkait	DetailDoctorActivity, Doctor, DoctorSchedule
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan detail data dokter yang dipilih oleh user.
Method yang terkait	showDetailDoctor()
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> doctorData : objek bertipe Doctor doctorSchedule : objek bertipe DoctorSchedule listDoctorSchedule : variabel arraylist bertipe DoctorSchedule doctorId : variabel tunggal bertipe String doctorName : variabel tunggal bertipe String doctorSpecialist : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String practiceDays : variabel array bertipe String practiceStartTimes : variabel array bertipe String practiceEndTimes : variabel array bertipe String practiceLocations : variabel array bertipe String index : variabel tunggal bertipe integer </pre> <p>PROCEDURE showDetailDoctor()</p> <p><u>BEGIN</u></p> <pre> doctorData <- doctorList.get(position) doctorId <- doctorData.getDoctorId() doctorName <- doctorData.getDoctorName() doctorSpecialist <- doctorData.getDoctorSpecialist() doctorStatus <- doctorData.getDoctorStatus() listDoctorSchedule <- doctorData.getDoctorSchedules() index <- 0 FOR (index TO listDoctorSchedule.size(), index+1) DO doctorSchedule <- listDoctorSchedules.get(index) practiceDays[index] <- doctorSchedule.getPracticeDay() practiceStartTimes[index] <- doctorSchedule.getPracticeStartTime() practiceEndTimes[index] <- doctorSchedule.getPracticeEndTime() practiceLocations[index] <- doctorSchedule.getPracticeLocation() </pre> <p><u>END FOR</u></p> <p><u>END PROCEDURE</u></p>

Gambar 66 Referensi *Manage Lead Qualified Doctor*

Tabel 91 Keterangan SD_refManageLeadQualifiedDoctor

ID Sequence diagram	SD-MD-06
Nama Sequence diagram	SD_refManageLeadQualifiedDoctor
ID Use Case	
Nama Class Terkait	ManageLeadQualifiedDoctorFragment, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengelola data dokter yang mempunyai status “lead qualified”.
Method yang terkait	getLeadQualifiedDoctorsData(), getDoctorsByMedrepAndStatus
Sequence Diagram yang terkait	-
Algoritma	
<p>Kamus data</p> <pre> listDoctorData : variabel arraylist bertipe Doctor medrepId : variabel tunggal bertipe String doctorStatus : variabel tunggal bertipe String apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre> <p>PROCEDURE getLeadQualifiedDoctorsData()</p> <p><u>BEGIN</u></p> <pre> doctorStatus <- "Lead Qualified" responseServer <- apiClient.getDoctorsByMedrepAndStatus(medrepId, doctorStatus) listDoctorData <- responseServer.getDoctorsData() </pre> <p><u>END PROCEDURE</u></p>	

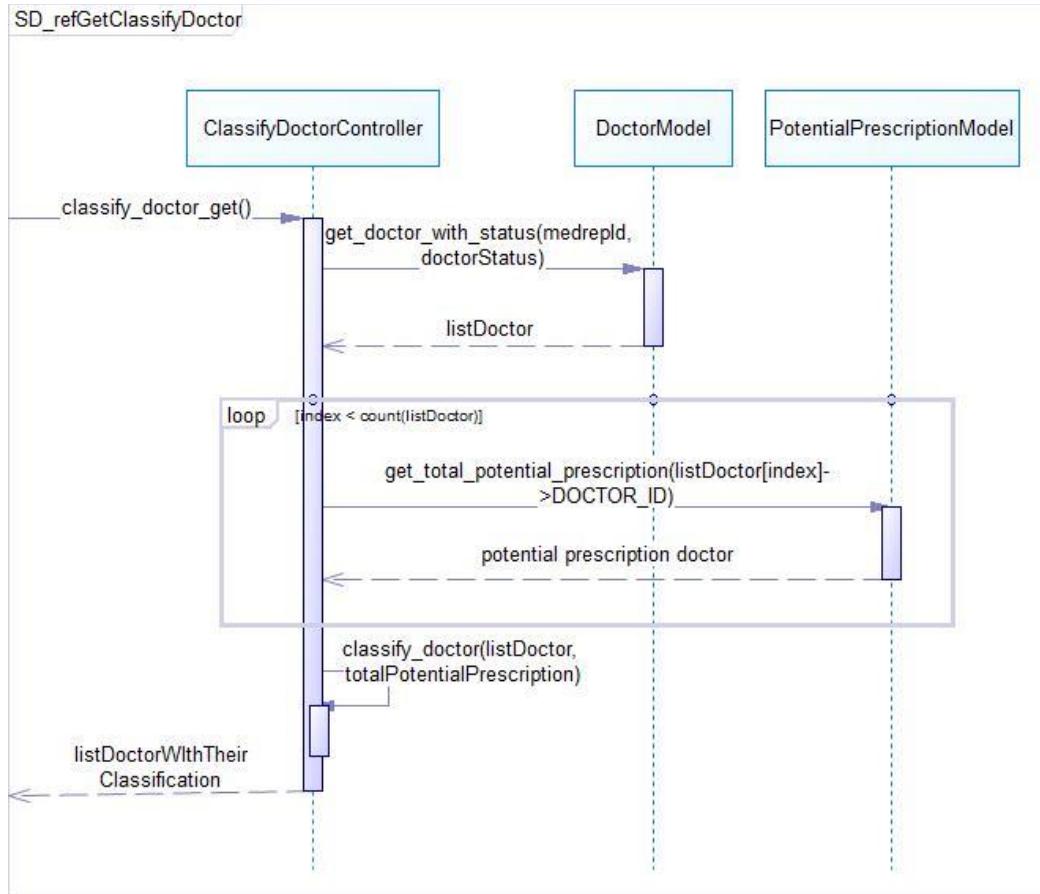
3. SD – Classify Doctor



Gambar 67 SD *Classify Doctor*

Tabel 92 Keterangan SD *ClassifyDoctor*

ID Sequence Diagram	SD-CD-01
Nama Sequence Diagram	SD <i>ClassifyDoctor</i>
Nama Class Terkait	<code>ClassifyDoctorFragment</code> , <code>ApiClient</code>
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses pengklasifikasian dokter.
Method yang terkait	<code>requestClassifyDoctor()</code> , <code>getClassifyDoctor(medrepId)</code>
Sequence Diagram yang terkait	SD <i>_refGetClassifyDoctor</i>
Algoritma	
Kamus data	
	<pre> listDoctor : variabel arrayList bertipe Doctor responseServer : variabel objek bertipe ResponseServer apiClient : variabel objek bertipe ApiClient </pre>
PROCEDURE <code>requestClassifyDoctor()</code>	
BEGIN	
	<pre> responseServer <- apiClient.getClassifyDoctor(medrepId) listDoctor <- responseServer.getDoctorData() </pre>
END PROCEDURE	

Gambar 68 Referensi *Get Classify Doctor*

Tabel 93 Keterangan SD_refGetClassifyDoctor

ID Sequence Diagram	SD-CD-02
Nama Sequence Diagram	SD_refGetClassifyDoctor
Nama Class Terkait	<i>ClassifyDoctorController, DoctorModel, PotentialPrescriptionModel</i>
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses pengklasifikasian dokter.
Method yang terkait	<i>classify_doctor_get(), get_doctor_by_medrep_and_status(medrepId, doctorStatus), get_total_potential_prescription(doctorId), classify_doctor(listDoctor)</i>
Sequence Diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> listDoctor : variable array objek index : variabel tunggal bertipe integer totalPotentialPrescription : variabel tunggal bertipe integer classParameter <- variabel array bertipe integer value <- variabel array bertipe String PROCEDURE classify_doctor_get() BEGIN listDoctor <- \$this->doctormodel- >get_doctor_by_medrep_and_status(medrepId, doctorStatus) totalPotentialPrescription <- 0 index <- 0 FOR EACH listDoctor AS doctor DO doctor->POTENTIAL_PRESCRIPTION <- \$this- >potentialprescriptionmodel- >get_total_potential_prescription(doctor->DOCTOR_ID) totalPotentialPrescription <- (totalPotentialPrescription + doctor- >POTENTIAL_PRESCRIPTION) END FOR listDoctor = \$this->classify_doctor(listDoctor, totalPotentialPrescription) END PROCEDURE FUNCTION get_total_potential_prescription(doctorId) -> array objek BEGIN return \$this->db->query("SELECT TOTAL_POTENTIAL_PRESCRIPTION FROM potentialprescription WHERE DOCTOR_ID = doctorId ORDER BY CHECKING_DATE DESC LIMIT 1")->row() </pre>

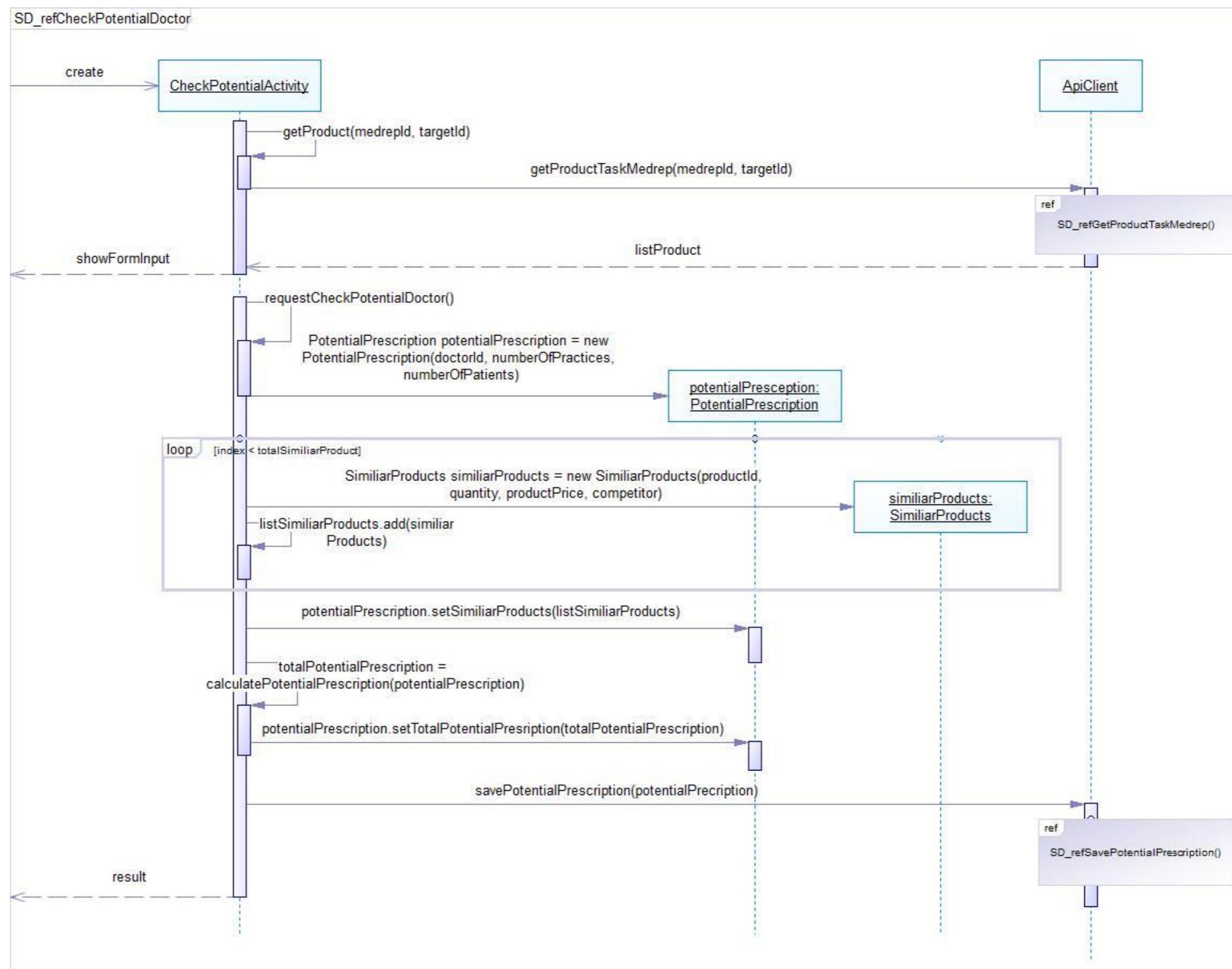
```
END PROCEDURE

FUNCTION classify_doctor(listDoctor, totalPotentialPrescription)
-> array objek
BEGIN
    tempPercentage <- 0
    classParameter <- array(70, 20, 10)
    value <- array('A', 'B', 'C')
    index <- 0
    listDoctor <- sort(listDoctor)
    FOR EACH listDoctor AS doctor DO
        doctor->PERCENTAGE <- (doctor->POTENTIAL_PRESCRIPTION
        / totalPotentialPrescription)
        tempPercentage <- doctor->PERCENTAGE

        doctor->CLASS <- value[index]
        IF tempPercentage >= classParameter[index] THEN
            index++;
            tempPercentage <- 0
        ENDIF
    END FOR

    return listDoctor
END PROCEDURE
```

4. SD – Check Potential Doctor



Gambar 69 Referensi Check Potential Doctor

Tabel 94 Keterangan SD_refCheckPotentialDoctor

ID Sequence Diagram	SD-CPD-01
Nama Sequence Diagram	SD_refCheckPotentialDoctor
Nama Class Terkait	CheckPotentialActivity, PotentialPrescription, SimiliarProducts, ApiClient
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses perhitungan besar potensi peresepan dokter.
Method yang terkait	getProduct(medrepId, targetId), getProductTaskMedrep(medrepId, targetId), requestCheckPotentialDoctor(), calculatePotentialPrescription(potentialPrescription), savePotentialPrescription(potentialPrescription)
Sequence Diagram yang terkait	SD_refGetProductTaskMedrep, SD_refSavePotentialPrescription
Algoritma	<p>Kamus data</p> <pre> targetId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String doctorId : variabel tunggal bertipe String listProduct : variabel arraylist bertipe Product potentialPrescription : objek bertipe PotentialPrescription numberOfPractices : variabel tunggal bertipe integer numberOfPatients : variabel tunggal bertipe integer index : variabel tunggal bertipe integer totalSimiliarProduct : variabel tunggal bertipe integer similiarProducts : objek bertipe SimiliarProducts productId : variabel tunggal bertipe String quantity : variabel tunggal bertipe integer productPrice : variabel tunggal bertipe integer listSimiliarProducts : variabel arraylist bertipe SimiliarProducts total : variabel tunggal bertipe integer totalProduct : variabel tunggal bertipe integer apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre> <p>PROCEDURE getProduct (medrepId, targetId)</p> <p><u>BEGIN</u></p> <pre> responseServer ← apiClient.getProductTaskMedrep (medrepId, targetId) listProduct ← responseServer.getProduct() </pre> <p><u>END PROCEDURE</u></p> <p>PROCEDURE requestCheckPotentialDoctor()</p> <p><u>BEGIN</u></p> <pre> potentialPrescription ← new PotentialPrescription(doctorId, numberOfPractices, numberOfPatients) index ← 0 </pre> <p><u>FOR</u> (index TO totalSimiliarProducts-1, index+1) <u>DO</u></p>

```

similiarProducts ← new SimiliarProducts(productId,
    quantity, productPrice, competitor)
listSimiliarProducts.add(similiarProducts)
END FOR

potentialPrescription.setSimiliarProducts(listSimiliarPr
oducts)
totalPotentialPrescription ←
    calculatePotentialPrescription(potentialPrescription)
potentialPrescription.setTotalPotentialPrescription(tota
lPotentialPrescription)
responseServer ←
    apiClient.savePotentialPrescreption(potentialPrescript
ion)

END PROCEDURE

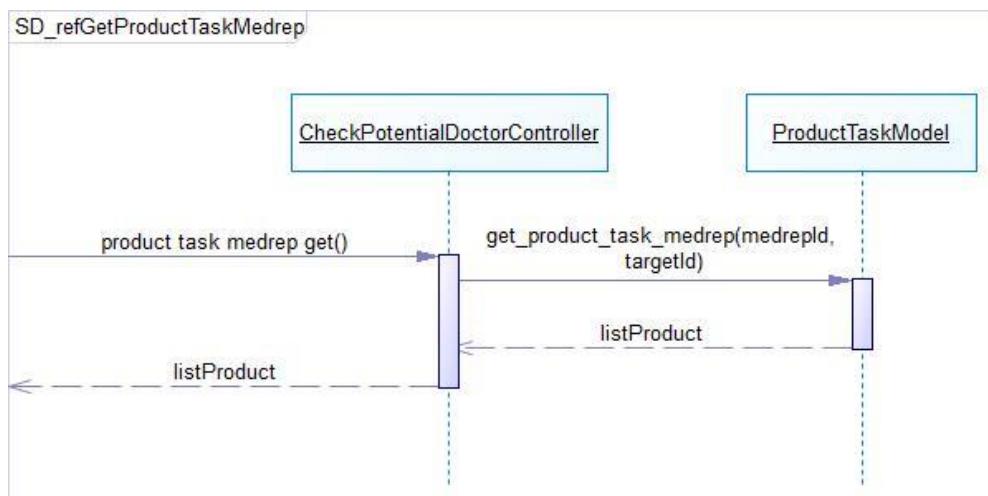
FUNCTION calculatePotentialPrescription(potentialPrescription) -> integer
BEGIN
total ← potentialPrescription.getNumberOfPractices() *
    potentialPrescription.getNumberOfPatients() * 4

FOR EACH potentialPrescription.getSimilarProducts() AS
prescription DO
totalProduct ← totalProduct + (prescription.getQuantity() *
    prescription.getProductPrice())
ENDFOR

total ← total * totalproduct

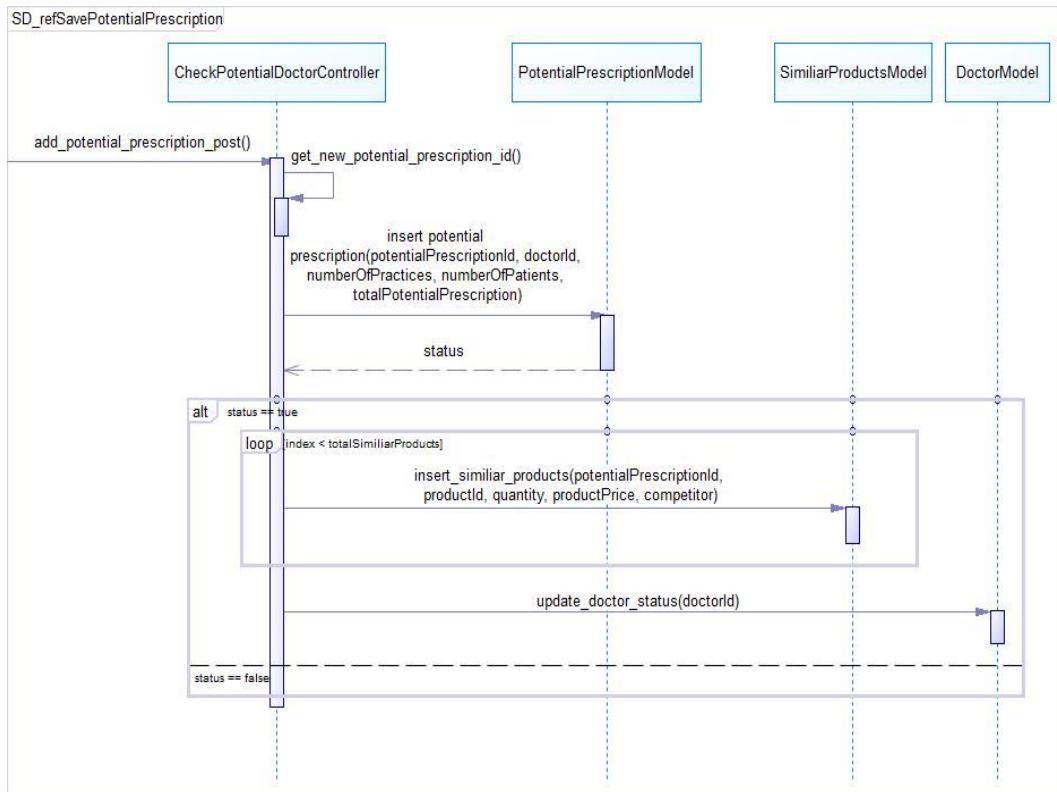
return total
END FUNCTION

```

Gambar 70 SD Referensi *Get Product Task Medrep*

Tabel 95 Keterangan SD_refGetProductTaskMedrep

ID Sequence Diagram	SD-CPD-02
Nama Sequence Diagram	SD_refGetProductTaskMedrep
Nama Class Terkait	CheckPotentialDoctorController, ProductTaskModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data produk dari database.
Method yang terkait	product_medrep_by_target_get(), get_product_medrep_by_target(target_id)
Sequence Diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> targetId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String listProduct : variabel array objek PROCEDURE product_task_medrep_get() BEGIN targetId ← \$this->post("TARGET_ID") medrepId ← \$this->post("MEDREP_ID") listProduct ← \$this->producttaskmodel- >get_product_task_medrep(medrepId, targetId) END PROCEDURE FUNCTION get_product_task_medrep(medrepId, targetId) -> objek list product BEGIN return \$this->db->query("SELECT * FROM producttask pt, product p WHERE pt.TARGET_ID == targetId AND pt.PRODUCT_ID == p.PRODUCT_ID AND DELETED = 0")->result() END FUNCTION </pre>



Gambar 71 SD Referensi Save Potential Prescription

Tabel 96 Keterangan SD_refSavePotentialPrescription

ID Sequence Diagram	SD-CPD-03
Nama Sequence Diagram	SD_refSavePotentialPrescription
Nama Class Terkait	CheckPotentialDoctorController, PotentialPrescriptionModel, SimiliarProductsModel, DoctorModel
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses menyimpan data potensi peresepan dokter ke <i>database</i> .
Method yang terkait	add_potential_prescription_post(), get_new_potential_prescription_id(), insert_potential_prescription(potentialPrescriptionId, doctorId, numberOffPractices, numberofPatients, totalPotentialPrescription), insert_similiar_products(potentialPrescriptionId, productId, quantity, productPrice, competitor), update_doctor_status(doctorId)
Sequence Diagram yang terkait	-
Algoritma	<p>Kamus data</p> <p>potentialPrescription : variabel array objek similarProducts : variabel array objek potentialPrescriptionId : variabel tunggal bertipe String</p> <pre> PROCEDURE add_potential_prescription_post() </pre>

```

BEGIN
    status ← FALSE
    potentialPrescriptionId ← $this-
        >get_new_potential_prescription_id()

    doctorId ← $this->post("DOCTOR_ID")
    number_of_practices ← $this->post("NUMBER_OF_PRACTICES")
    number_of_patients ← $this->post("NUMBER_OF_PATIENTS")
    totalPotentialPrescription ← $this->
        post("TOTAL_POTENTIAL_PRESCRIPTION")

    status ← $this->potentialprescriptionmodel-
        >insert_potential_prescription(potentialPrescriptionId,
            doctorId, number_of_practices, number_of_patients,
            totalPotentialPrescription)

    IF status == TRUE THEN
        similiarProducts ← $this->post("SIMILIAR_PRODUCTS")
        IF similiarProducts != null THEN
            FOR EACH similiarProducts AS product
                productId ← product["PRODUCT_ID"]
                quantity ← product["QUANTITY"]
                productPrice ← product["PRODUCT_PRICE"]
                competitor ← $this->post("COMPETITOR")

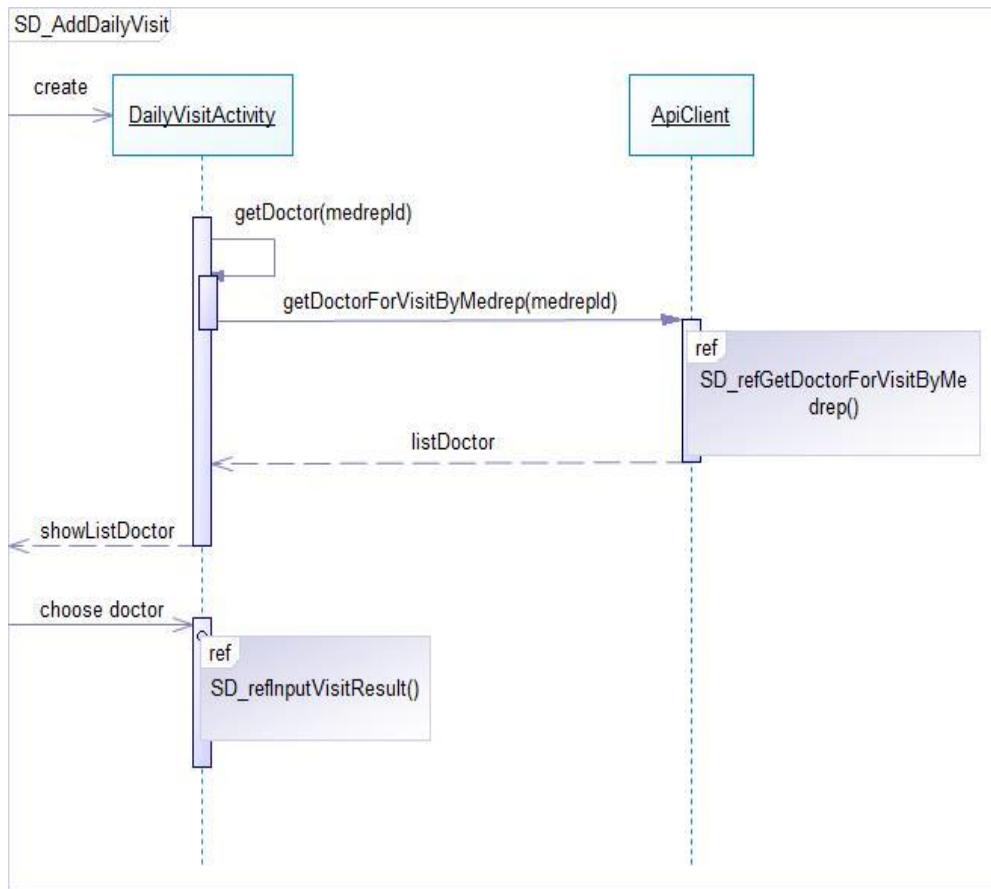
                $this->similiarproductsmodel-
                    >insert_similiar_products(potentialPrescripti
                        onId, productId, quantity, productPrice,
                        competitor)
                $this->doctormodel-
                    >update_doctor_status(doctorId)
            END FOR
        END IF
    END IF
END PROCEDURE

FUNCTION get_new_potential_prescription_id() -> String
BEGIN
    lastId ← $this->potentialprescriptionmodel->count()

    return "PP" . (lastId + 1)
END FUNCTION

```

5. SD – Add Daily Visit



Gambar 72 SD Add Daily Visit

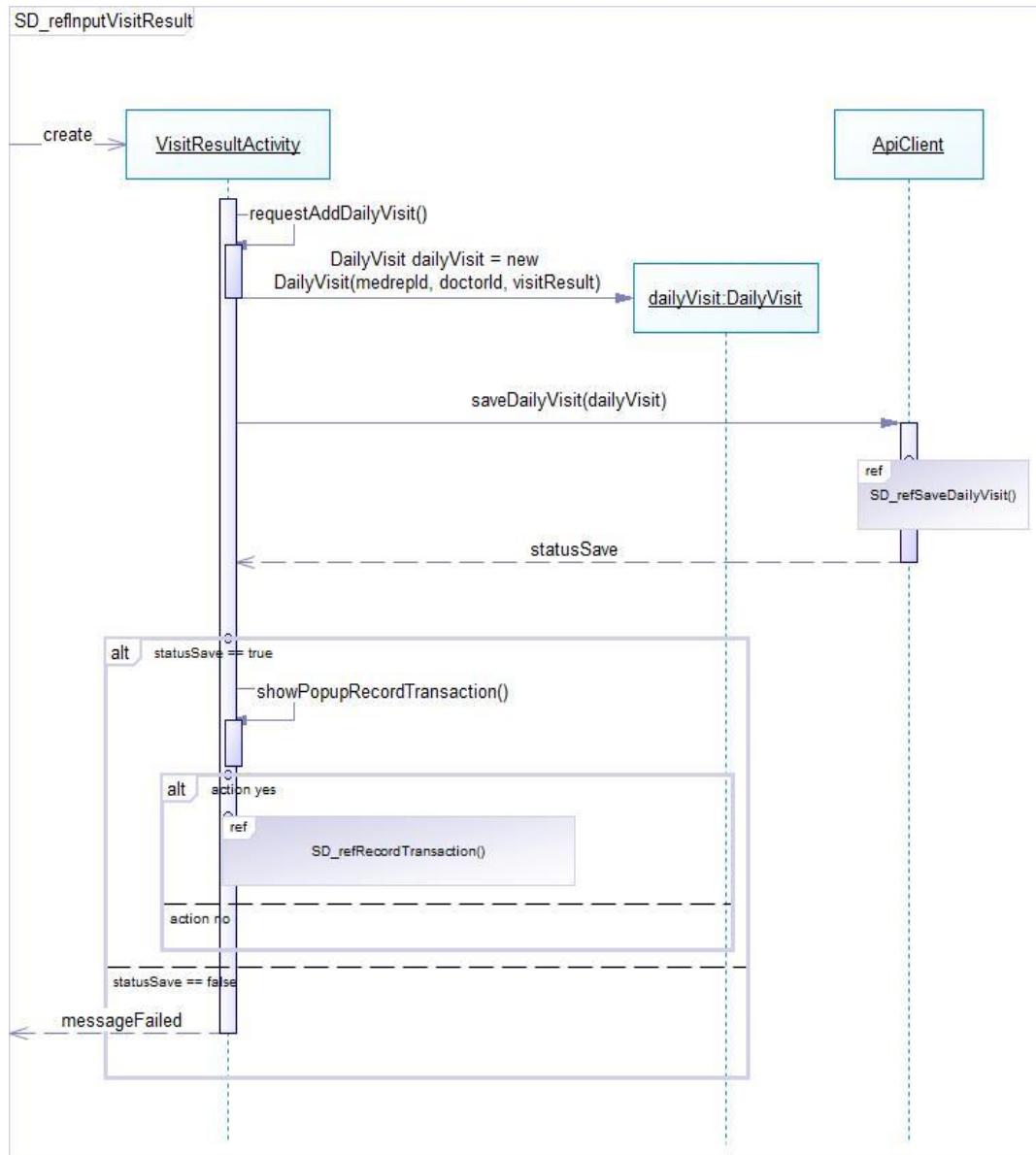
Tabel 97 Keterangan SD_AddDailyVisit

ID Sequence diagram	SD-DV-01
Nama Sequence diagram	SD_AddDailyVisit
Nama Class Terkait	DailyVisitActivity, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses pencatatan hasil kunjungan Medical Representative.
Method yang terkait	getDoctor(medrepId), getDoctorForVisitByMedrep(medrepId)
Sequence diagram yang terkait	SD_refGetDoctorForVisitByMedrep, SD_refInputVisitResult
Algoritma	<p>Kamus data</p> <pre> listDoctorData : variabel arraylist bertipe Doctor medrepId : variabel tunggal bertipe String apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre>

```

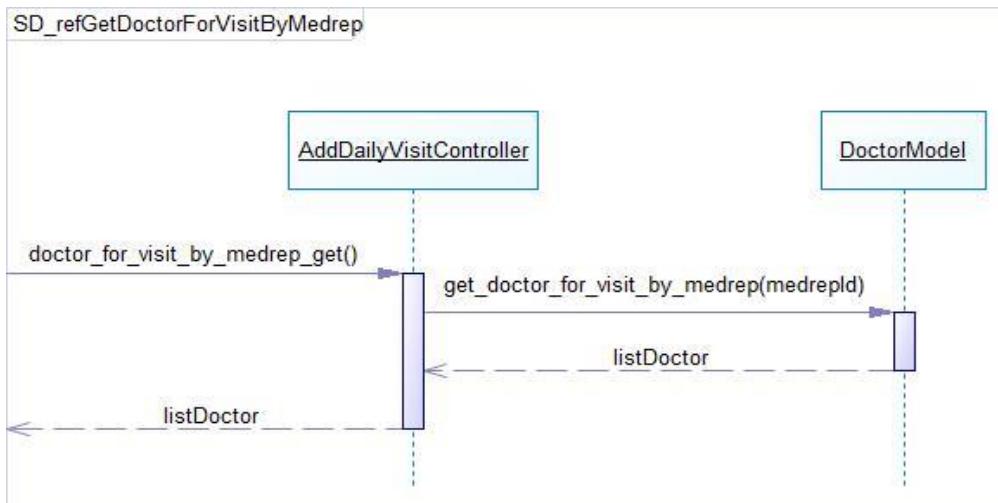
PROCEDURE getDoctor (medrepId)
BEGIN
    responseServer ←
        apiClient.getDoctorForVisitByMedrep (medrepId)
    listDoctorData ← responseServer.getDoctorData ()
END PROCEDURE

```

Gambar 73 SD Referensi *Input Visit Result*

Tabel 98 Keterangan SD_refInputVisitResult

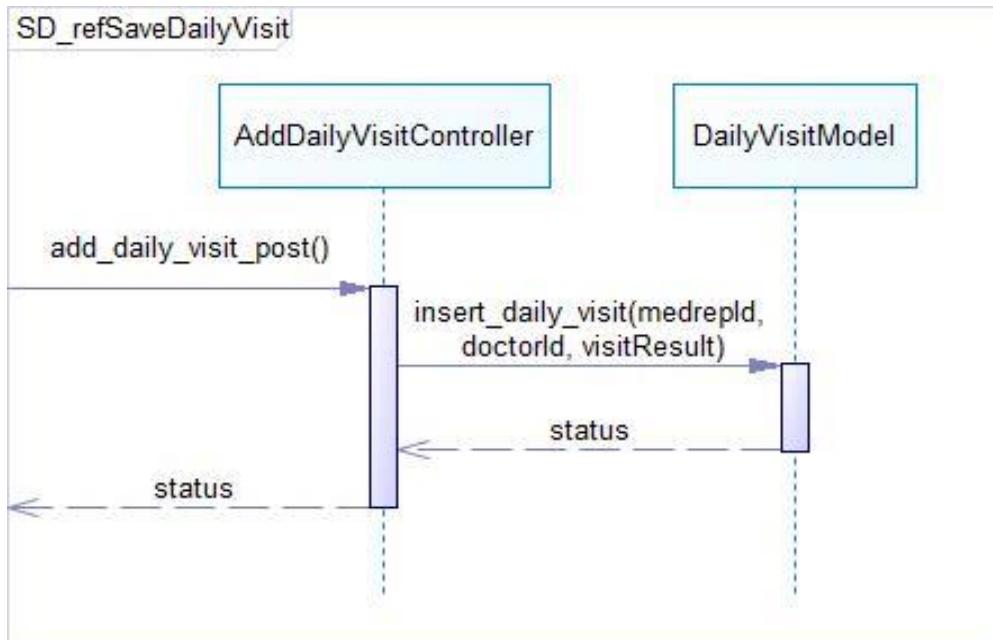
ID Sequence diagram	SD-DV-02
Nama Sequence diagram	SD_refInputVisitResult
Nama Class Terkait	VisitResultActivity, DailyVisit, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses pencatatan hasil kunjungan <i>Medical Representative</i> .
Method yang terkait	requestAddDailyVisit(), saveDailyVisit(dailyVisit), showPopupRecordTransaction()
Sequence diagram yang terkait	SD_saveDailyVisit(), SD_refRecordTransaction()
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String doctorId : variabel tunggal bertipe String dailyVisit : objek bertipe DailyVisit visitResult : variabel tunggal bertipe String apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer PROCEDURE requestAddDailyVisit() BEGIN doctorId ← listDoctorData.get(position).getDoctorId() dailyVisit ← new DailyVisit(medrepId, doctorId, visitResult) responseServer ← apiClient.saveDailyVisit(dailyVisit) IF responseServer.isStatusResponse() == TRUE THEN showPopupRecordTransaction() ELSE WRITE (Screen) "save failed" END IF END PROCEDURE </pre>



Gambar 74 SD Referensi Get Doctor For Visit By Medrep

Tabel 99 Keterangan SD_refGetDoctorForVisitByMedrep

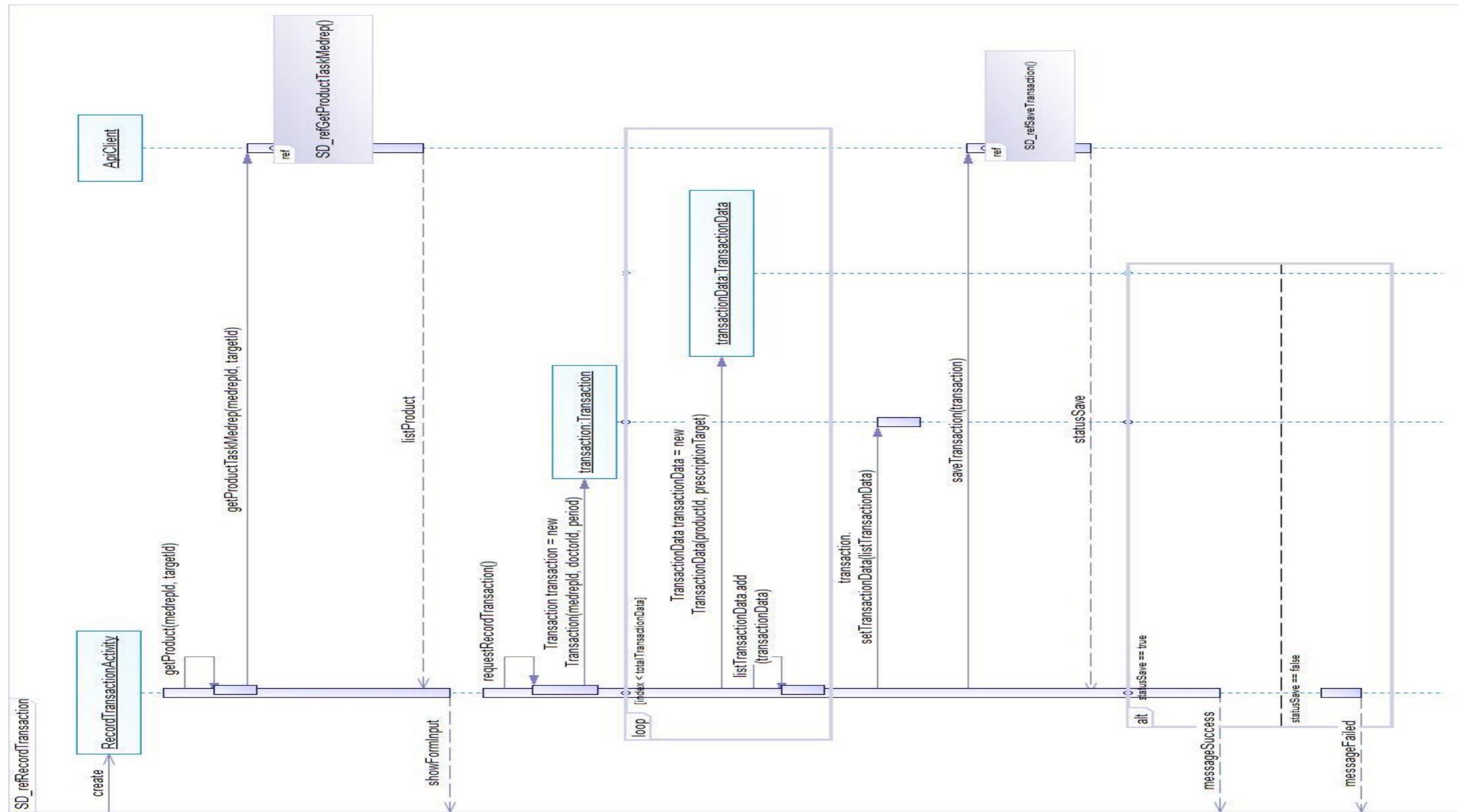
ID Sequence diagram	SD-DV-03
Nama Sequence diagram	SD_refGetDoctorForVisitByMedrep
Nama Class Terkait	AddDailyVisitController, DoctorModel
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses mengambil data dokter dari <i>database</i> . Data dokter yang diambil adalah data dokter yang mempunyai status ‘Lead Qualified’ dan ‘Contact’.
Method yang terkait	doctor_for_visit_by_medrep_get(), get_doctor_for_visit_by_medrep (medrepId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String listDoctor : variabel array objek doctor : variabel array objek PROCEDURE doctor_for_visit_by_medrep _get() BEGIN medrepId ← \$this->post("MEDREP_ID") list_doctor_data ← \$this->doctormodel-> get_doctor_for_visit_by_medrep(medrepId) END PROCEDURE FUNCTION get_doctor_for_visit_by_medrep(medrepId) -> objek list doctor BEGIN return \$this->db->query("SELECT * FROM doctor WHERE MEDREP_ID = medrepId AND DELETED = 0 AND DOCTOR_STATUS != 'LEAD'")->result() END FUNCTION </pre>

Gambar 75 SD Referensi *Save Daily Visit*

Tabel 100 Keterangan SD_refSaveDailyVisit

ID Sequence diagram	SD-DV-04
Nama Sequence diagram	SD_refSaveDailyVisit
Nama Class Terkait	AddDailyVisitController, DailyVisitModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menyimpan data kunjungan ke database.
Method yang terkait	add_daily_visit_post(), insert_daily_visit(medrepId, doctorId, visitResult)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String doctorId : variabel tunggal bertipe String visitResult : variabel tunggal bertipe String status : variabel tunggal bertipe boolean </pre> <p>PROCEDURE add_daily_visit_post()</p> <p>BEGIN</p> <ul style="list-style-type: none"> medrepId ← \$this->post("MEDREP_ID") doctorId ← \$this->post("DOCTOR_ID") visitResult ← \$this->post("VISIT_RESULT") status ← \$this->dailyvisitmodel ->insert_daily_visit(medrepId, doctorId, visitResult) <p>END PROCEDURE</p>

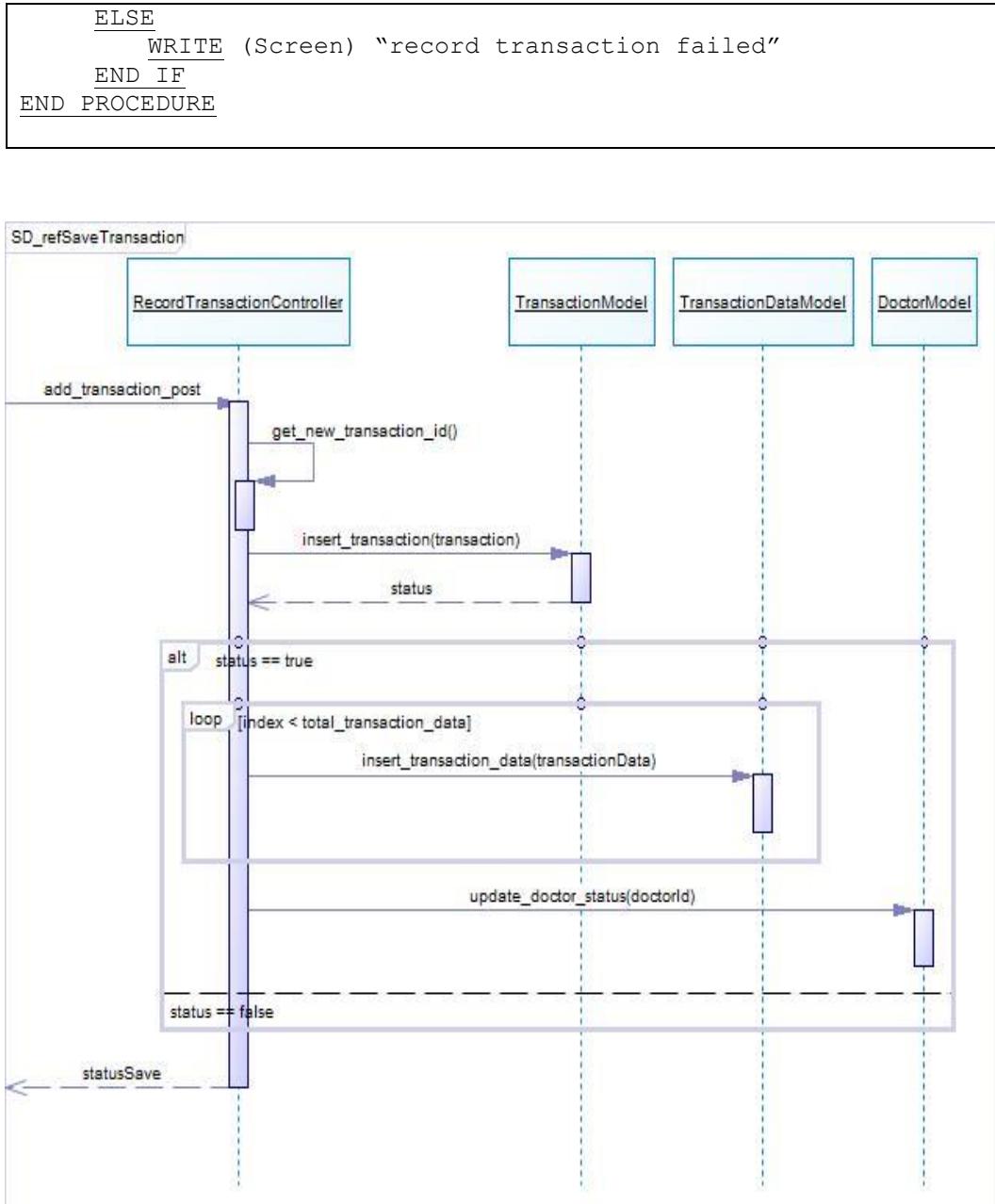
6. SD – Record Transaction



Gambar 76 SD Referensi Record Transaction

Tabel 101 Keterangan SD_refRecordTransaction

ID Sequence diagram	SD-RT-01
Nama Sequence diagram	SD_refRecordTransaction
Nama Class Terkait	RecordTransactionActivity, Transaction, TransactionData, ApiClient
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses pencatatan hasil transaksi <i>Medical Representative</i> .
Method yang terkait	getProduct(medrepId, targetId), getProductTaskMedrep (medrepId, targetId), requestRecordTransaction(), saveTransaction(transaction)
Sequence diagram yang terkait	SD_refGetProductTaskMedrep, SD_refSaveTransaction
Algoritma	<p>Kamus data</p> <pre> targetId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String doctorId : variabel tunggal bertipe String listProduct : variabel arraylist bertipe Product transaction : objek bertipe Transaction period : variabel tunggal bertipe integer index : variabel tunggal bertipe integer totalTransactionData : variabel tunggal bertipe integer transactionData : objek bertipe TransactionData productId : variabel tunggal bertipe String presceptionTarget : variabel tunggal bertipe integer listTransactionData : variabel arraylist bertipe TransactionData apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre> <p>PROCEDURE getProduct (medrepId, targetId)</p> <p><u>BEGIN</u></p> <pre> responseServer ← apiClient.getProductTaskMedrep (medrepId, targetId) listProduct ← responseServer.getProduct() </pre> <p><u>END PROCEDURE</u></p> <p>PROCEDURE requestRecordTransaction()</p> <p><u>BEGIN</u></p> <pre> transaction ← new Transaction (medrepId, doctorId, period) index ← 0 FOR (index TO totalTransactionData-1, index+1) DO transactionData ← new TransactionData (productId, presceptionTarget) listTransactionData.add (transactionData) </pre> <p><u>END FOR</u></p> <pre> transaction.setTransactionData (listTransactionData) responseServer ← apiClient.saveTransaction (transaction) IF responseServer.isStatusResponse() == TRUE THEN WRITE (Screen) "record transaction success" </pre>



Gambar 77 SD Referensi Save Transaction

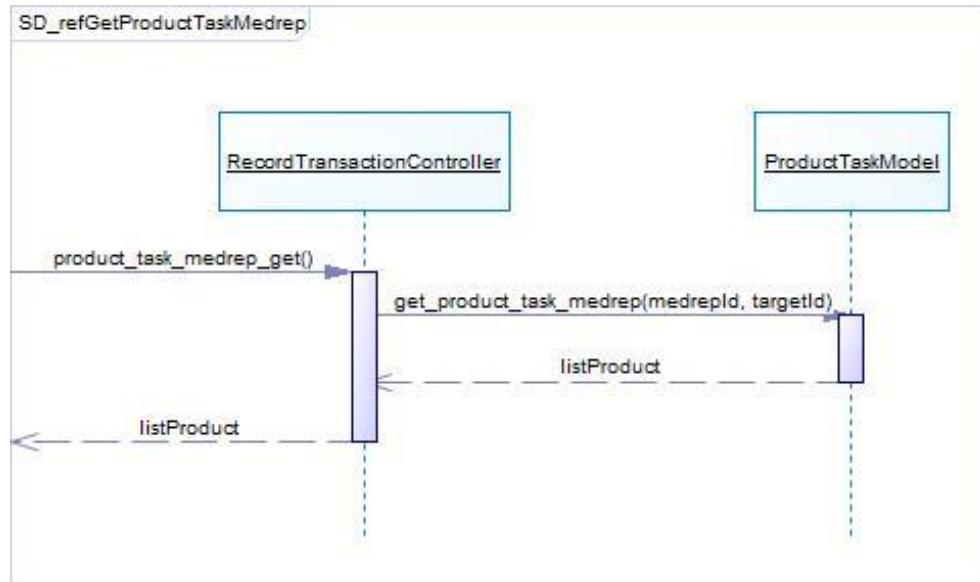
Tabel 102 Keterangan SD_refSaveTransaction

ID Sequence diagram	SD-RT-02
Nama Sequence diagram	SD_refSaveTransaction
Nama Class Terkait	RecordTransactionController, TransactionModel, TransactionDataModel, DoctorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menyimpan data transaksi ke database.
Method yang terkait	add_transaction_post(), get_new_transaction_id(), insert_transaction(transactionId, medrepId, doctorId, period), insert_transaction_data(transactionId, productId, prescriptionTarget), update_doctor_status(doctorId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> status : variabel tunggal bertipe boolean transaction : variabel array objek transactionData : variabel array objek transactionId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String doctorId : variabel tunggal bertipe String period : variabel tunggal bertipe Integer productId : variabel tunggal bertipe String prescriptionTarget : variabel tunggal bertipe Integer </pre> <p>PROCEDURE add_transaction_post()</p> <p>BEGIN</p> <pre> status ← FALSE medrepId ← \$this->post("DOCTOR_ID") doctorId ← \$this->post("DOCTOR_ID") period ← \$this->post("DOCTOR_ID") transactionId ← \$this->get_new_transaction_id() status ← \$this->transactionmodel-> insert_transaction(transactionId, medrepId, doctorId, period) IF status_save == TRUE THEN transactionData ← \$this->post("TRANSACTION_DATA") IF transactionData != null THEN FOR EACH transactionData AS detailTransaction productId ← detailTransaction ['PRODUCT_ID'] prescriptionTarget ← detailTransaction ['PRESCEPTION_TARGET'] \$this->transactiondatamodel-> insert_transaction_data(transactionId, productId, prescriptionTarget) \$this->doctormodel-> update_doctor_status(doctorId) END FOR END IF END PROCEDURE </pre>

```

FUNCTION get_new_transaction_id() -> String
BEGIN
    lastId ← $this->transactionmodel->count()
    return "TR" . (lastId + 1)
END FUNCTION

```



Gambar 78 SD Referensi Get Product Task Medrep

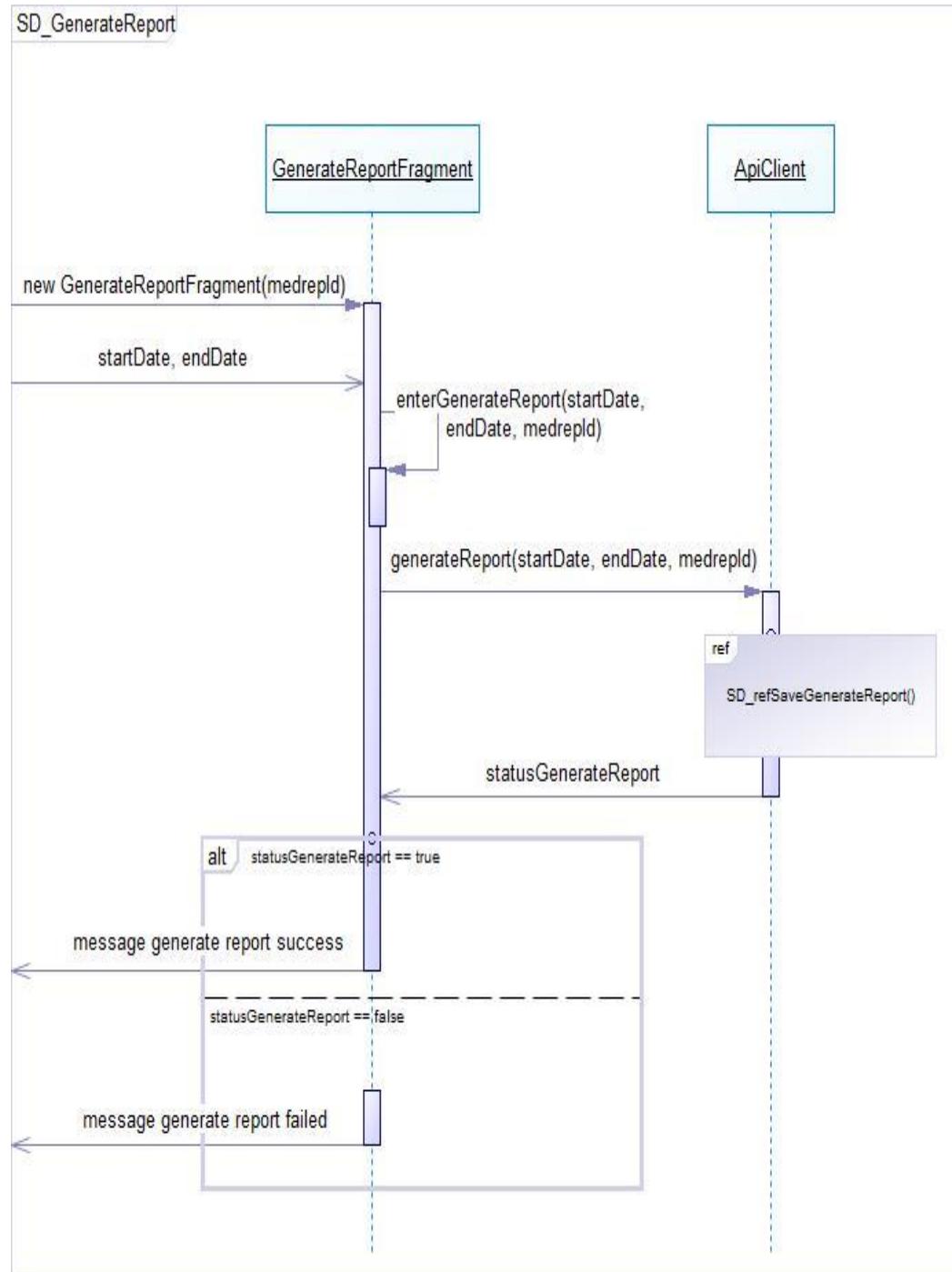
Tabel 103 Keterangan SD_refGetProductTaskMedrep

ID Sequence diagram	SD-RT-03
Nama Sequence diagram	SD_refGetProductTaskMedrep
Nama Class Terkait	RecordTransactionController, ProductTaskModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data produk dari database. Produk yang diambil adalah produk yang merupakan tugas <i>Medical Representative</i> .
Method yang terkait	product_task_medrep_get(), get_product_task_medrep (medrepId, targetId)
Sequence diagram yang terkait	-
Algoritma	
Kamus data	
	targetId : variabel tunggal bertipe String
	medrepId : variabel tunggal bertipe String
	listProduct : variabel array objek
PROCEDURE product_task_medrep_get()	
BEGIN	
	targetId ← \$this->post("TARGET_ID")

```
medrepId ← $this->post("MEDREP_ID")
listProduct ← $this->producttaskmodel-
>get_product_task_medrep(medrepId, targetId)
END PROCEDURE

FUNCTION get_product_task_medrep(medrepId, targetId) -> objek
list product
BEGIN
    return $this->db->query("SELECT * FROM producttask pt,
product p WHERE pt.TARGET_ID == targetId AND pt.PRODUCT_ID
== p.PRODUCT_ID AND DELETED = 0")->result()
END FUNCTION
```

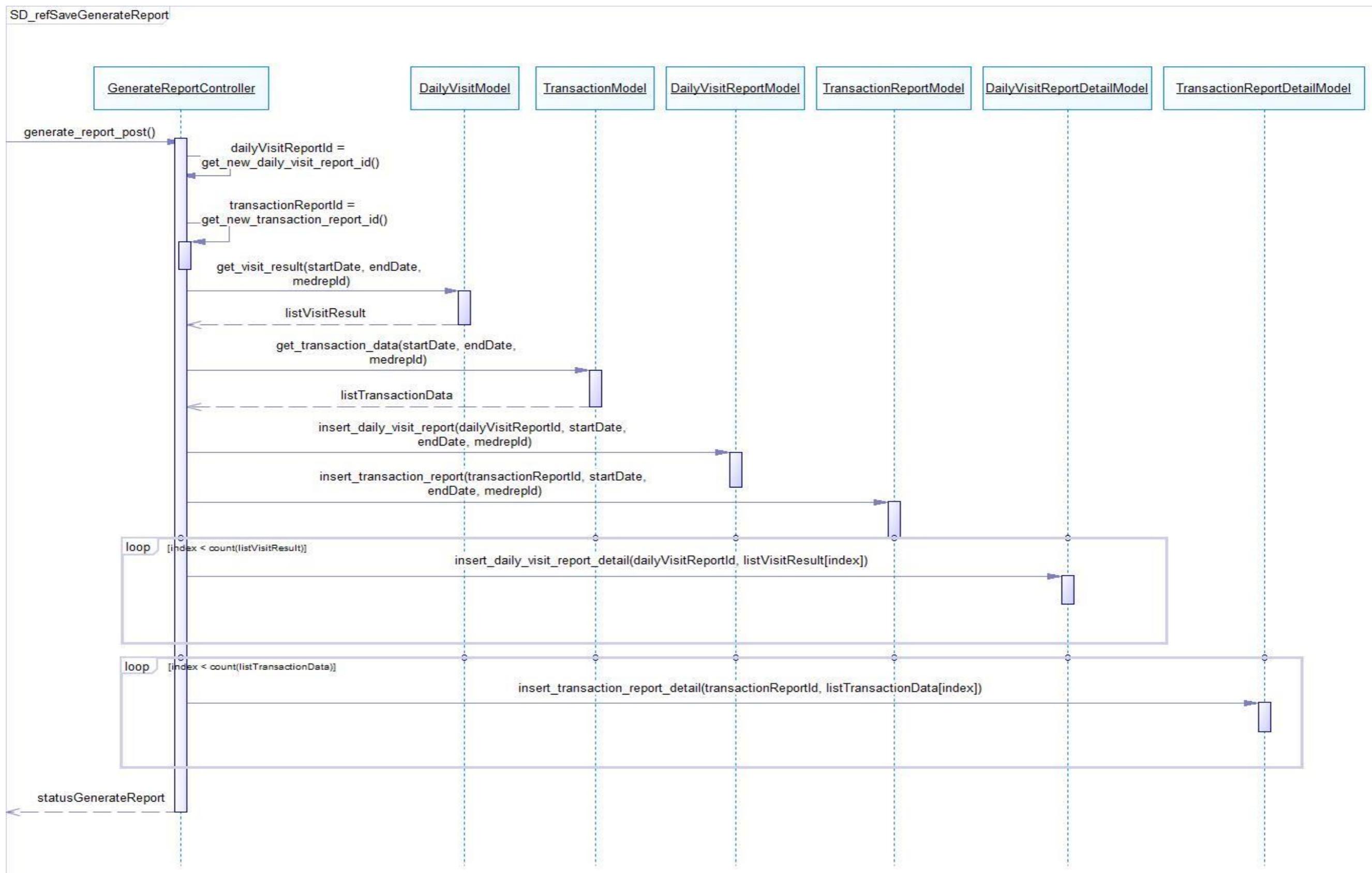
7. SD – Generate Report



Gambar 79 SD *Generate Report*

Tabel 104 Keterangan SD_GenerateReport

ID Sequence diagram	SD-GR-01
Nama Sequence diagram	SD_GenerateReport
Nama Class Terkait	GenerateReportFragment, ApiClient
Deskripsi	Sequence ini digunakan untuk menjelaskan proses <i>generate</i> laporan <i>Medical Representative</i> .
Method yang terkait	enterGenerateReport(startDate, endDate, medrepId), generateReport(startDate, endDate, medrepId)
Sequence diagram yang terkait	SD_refSaveGenerateReport
Algoritma	<p>Kamus data</p> <pre> startDate : variabel objek bertipe Date endDate : variabel objek bertipe Date apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer medrepId : variabel tunggal bertipe String </pre> <p>PROCEDURE enterGenerateReport(startDate, endDate, medrepId)</p> <p><u>BEGIN</u></p> <pre> responseServer <- apiClient.generateReport(startDate, endDate, medrepId) IF responseServer.isStatusResponse() == TRUE <u>THEN</u> <u>WRITE</u> (Screen) "generate report success" ELSE <u>WRITE</u> (Screen) "generate report failed" <u>END IF</u> </pre> <p><u>END PROCEDURE</u></p>



Gambar 80 SD Referensi Save Generate Report

Tabel 105 Keterangan SD_refSaveGenerateReport

ID Sequence diagram	SD-GR-02
Nama Sequence diagram	SD_refSaveGenerateReport
Nama Class Terkait	ManageReportController, DailyVisitModel, TransactionModel, DailyVisitReportModel, TransactionReportModel, DailyVisitReportDetailModel, TransactionReportDetailModel
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses menyimpan <i>request generate laporan</i> .
Method yang terkait	generate_report_post(), get_new_daily_visit_report_id(), get_new_transaction_report_id(), get_visit_result(startDate, endDate, medrepId), get_transaction_data(startDate, endDate, medrepId), insert_daily_visit_report(dailyVisitReportId, startDate, endDate, medrepId), insert_transaction_report(transactionReportId, startDate, endDate, medrepId), insert_detail_visit_report_detail(dailyVisitReportId, visitResult), insert_transaction_report_detail(transactionReportId, transactionData)
Sequence diagram yang terkait	-
Algoritma	
<p>Kamus data</p> <pre> startDate : variabel tunggal bertipe String endDate : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String dailyVisitReportId : variabel tunggal bertipe String transactionReportId : variabel tunggal bertipe String listVisitResult : variabel array objek listTransactionData : variabel array objek statusGenerateReport : variabel tunggal bertipe boolean index : variabel tunggal bertipe integer </pre> <p>PROCEDURE generate_report_post()</p> <p><u>BEGIN</u></p> <pre> startDate <- \$this->post("startDate") endDate <- \$this->post("endDate") medrepId <- \$this->post("medrepId") dailyVisitReportId <- \$this-> get_new_daily_visit_report_id() transactionReportId <- \$this-> get_new_transaction_report_id() listVisitResult <- \$this->dailyvisitmodel-> get_visit_result(startDate, endDate, medrepId) listTransactionData <- \$this->transactionmodel-> get_transaction_data(startDate, endDate, medrepId) statusSaveReport <- \$this->dailyvisitreportmodel- >insert_daily_visit_report(dailyVisitReportId, startDate, endDate, medrepId) </pre>	

```

statusSaveReport <- $this->transactionreportmodel->
insert_transaction_report(transactionReportId,
startDate, endDate, medrepId)

IF statusSaveReport == TRUE THEN
    statusGenerateReport = TRUE
    index <- 0
    FOR (index TO (count(listVisitResult) - 1), index +
1) DO
        $this->dailyvisitreportdetailmodel->insert_
        visit_report_detail(dailyVisitReportId ,
        listVisitResult[index])
    END FOR

    index <- 0
    FOR(index TO (count(listTransactionData) - 1),
    index+1) DO
        $this->transactionreportdetailmodel->insert_
        transaction_report_detail(transactionReportId,
        listTransactionData[index])
    END FOR
ELSE
    statusGenerateReport = false
END IF
END PROCEDURE

FUNCTION get_new_daily_visit_report_id() -> String
BEGIN
    last_id <- $this->dailyvisitreportmodel-
    >count_visit_report()

    return "RDV" . last_id
END FUNCTION

FUNCTION get_new_transaction_report_id() -> String
BEGIN
    last_id <- $this->transactionreportmodel->count_
    transaction_report()

    return "RTD" . last_id
END FUNCTION

FUNCTION get_visit_result(startDate, endDate, medrepId) -> list
objek array visit result
BEGIN
    listVisitResult <- $this->db->query("SELECT * FROM
    dailyvisit WHERE MEDREP_ID = medrepId AND
    VISIT_DATE_TIME >= startDate AND VISIT_DATE_TIME <=
    endDate")->result()

    return listVisitResult
END FUNCTION

FUNCTION get_transaction_data(startDate, endDate, medrepId) ->
list objek array transaction data
BEGIN
    listTransactionData <- $this->db->query("SELECT * FROM
    transaction WHERE MEDREP_ID = medrepId AND

```

```

TRANSACTION_DATE >= startDate AND TRANSACTION_DATE <=
endDate")->result()

return listTransactionData
END FUNCTION

FUNCTION insert_daily_visit_report(dailyVisitReportId,
startDate, endDate, medrepId) -> boolean
BEGIN
    reportStatus <- FALSE
    visitReport <- array(
        'REPORT_VISIT_ID' <- dailyVisitReportId,
        'MEDREP_ID' <- medrepId
        'START_DATE_REPORT_PERIOD' <- startDate
        'END_DATE_REPORT_PERIOD' <- endDate
    )

    insertVisitReport <- $this->db->insert('visitreport',
visitReport)
    IF insertVisitReport == TRUE THEN
        reportStatus = TRUE
    END IF

    return reportStatus
END FUNCTION

FUNCTION insert_transaction_report(transactionReportId,
startDate, endDate, medrepId) -> boolean
BEGIN
    reportStatus <- FALSE
    transactionReport <- array(
        'REPORT_TRANSACTION_ID' <- dailyVisitReportId,
        'MEDREP_ID' <- medrepId
        'START_DATE_REPORT_PERIOD' <- startDate
        'END_DATE_REPORT_PERIOD' <- endDate
    )

    insertTransactionReport <- $this->db-
>insert('transactionreport', transactionReport)
    IF insertVisitReport == TRUE THEN
        reportStatus = TRUE
    END IF

    return reportStatus
END FUNCTION

PROCEDURE insert_daily_visit_report_detail(dailyVisitReportId ,
visitResult)
BEGIN
    visitDetail <- array(
        'REPORT_VISIT_DETAIL_ID' <- visitResult->
DAILY_VISIT_ID,
        'REPORT_VISIT_ID' <- dailuVisitReportid
    )

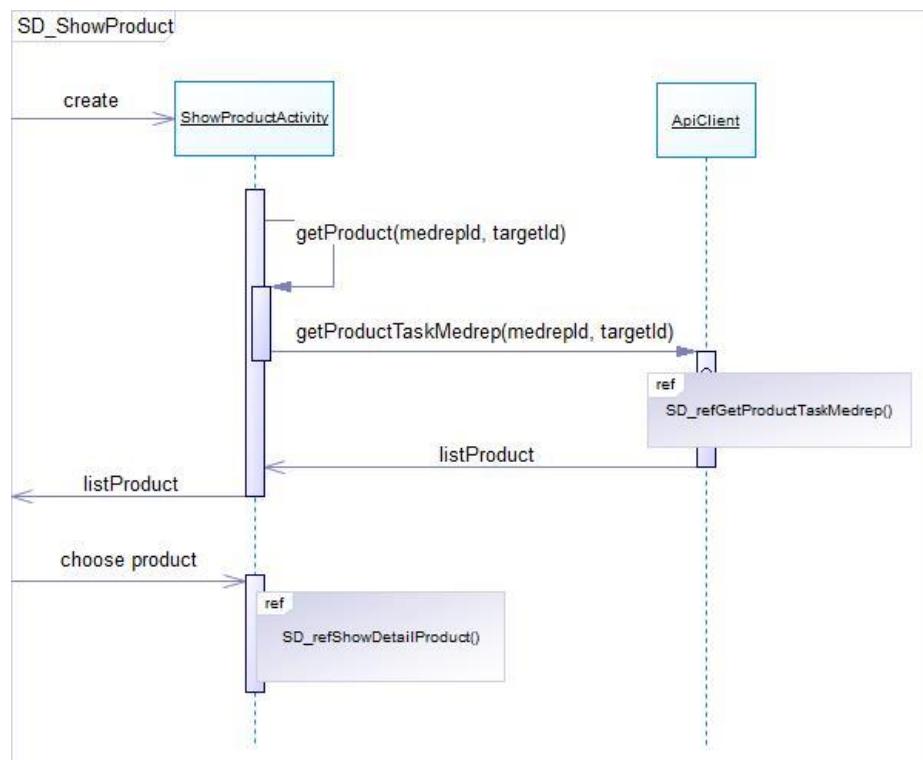
    $this->db->insert('detailvisitreport', visitDetail)
END PROCEDURE

```

```
PROCEDURE insert_transaction_report_detail(transactionReportId,
transactionData)
BEGIN
    transactionDetail <- array(
        'REPORT_TRANSACTION_DETAIL_ID' <- transactionData->
TRANSACTION_ID,
        'REPORT_TRANSACTION_ID' <- transactionReportId
    )

    $this->db->insert('detailtransactionreport',
transactionDetail)
END PROCEDURE
```

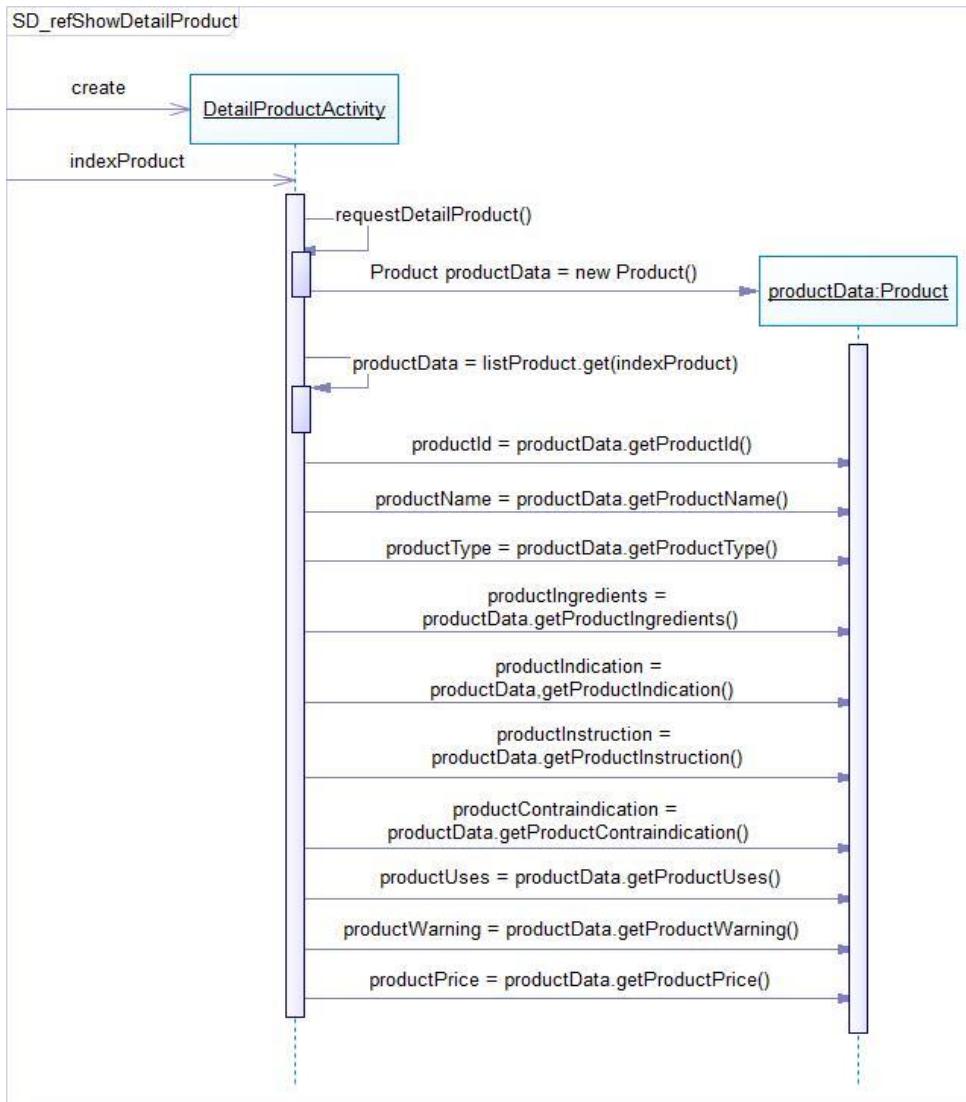
8. SD – Show Product



Gambar 81 SD Show Product

Tabel 106 Keterangan SD_ShowProduct

ID Sequence diagram	SD-SP-01
Nama Sequence diagram	SD_ShowProduct
Nama Class Terkait	ShowProductActivity, ApiClient
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses menampilkan <i>list product</i> .
Method yang terkait	getProduct(medrepId, targetId), getProductTaskMedrep(medrepId, targetId)
Sequence diagram yang terkait	SD_refGetProductTaskMedrep, SD_refShowDetailProduct
Algoritma	<p>Kamus data</p> <pre> targetId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String listProduct : variabel arraylist bertipe Product apiClient : objek bertipe ApiClient responseServer : objek bertipe ResponseServer </pre> <p>PROCEDURE getProduct (medrepId, targetId)</p> <p><u>BEGIN</u></p> <pre> responseServer ← apiClient.getProductTaskMedrep(targetId) listProduct ← responseServer.getProduct() </pre> <p><u>END PROCEDURE</u></p>

Gambar 82 SD Referensi *Show Detail Product*

Tabel 107 Keterangan SD_refShowDetailProduct

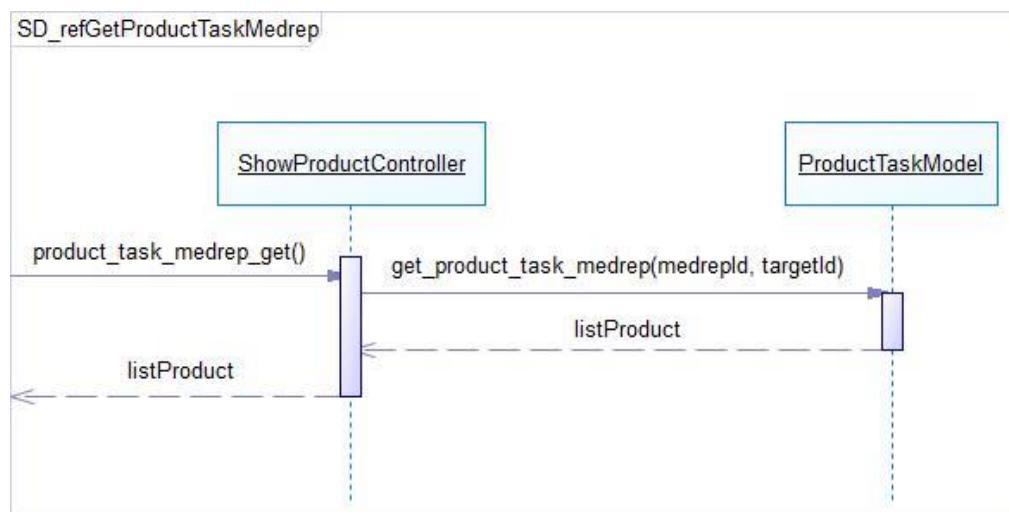
ID Sequence diagram	SD-SP-02
Nama Sequence diagram	SD_refShowDetailProduct
Nama Class Terkait	DetailProductActivity, Product
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan detail product yang dipilih.
Method yang terkait	requestDetailProduct()
Sequence diagram yang terkait	-
Algoritma	

```

Kamus data
medrepId : variabel tunggal bertipe String
productData : objek bertipe Product
indexProduct : variabel tunggal bertipe integer
listProduct : variabel arraylist bertipe Product
productId : variabel tunggal bertipe String
productName : variabel tunggal bertipe String
productType : variabel tunggal bertipe String
productIngredients : variabel tunggal bertipe String
productIndication : variabel tunggal bertipe String
productInstruction : variabel tunggal bertipe String
productContraindication : variabel tunggal bertipe String
productUses : variabel tunggal bertipe String
productWarning : variabel tunggal bertipe String
productPrice : variabel tunggal bertipe integer

PROCEDURE requestDetailProduct()
BEGIN
    productData ← listProduct.get(indexProduct)
    productId ← productData.getProductId()
    productName ← productData.getProductName()
    productType ← productData.getProductType()
    productIngredients ← productData.getProductIngredients()
    productIndication ← productData.getProductIndication()
    productInstruction ← productData.getProductInstruction()
    productContraindication ←
        productData.getProductIndication()
    productUses ← productData.getProductUses()
    productWarning ← productData.getProductWarning()
    productPrice ← productData.getProductPrice()
END PROCEDURE

```



Gambar 83 SD Referensi Get Product Task Medrep

Tabel 108 Keterangan SD_refGetProductTaskMedrep

ID Sequence diagram	SD-SP-03
Nama Sequence diagram	SD_refGetProductTaskMedrep
Nama Class Terkait	ShowProductController, ProdukTaskModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data produk dari database.
Method yang terkait	product_task_medrep_get(), get_product_task_medrep (medrepId, targetId)
Sequence diagram yang terkait	-
Algoritma	
<pre> Kamus data targetId : variabel tunggal bertipe String medrepId : variabel tunggal bertipe String listProduct : variabel array objek PROCEDURE product_task_medrep_get() BEGIN targetId ← \$this->post("TARGET_ID") medrepId ← \$this->post("MEDREP_ID") listProduct ← \$this->producttaskmodel- >get_product_task_medrep(medrepId, targetId) END PROCEDURE FUNCTION get_product_task_medrep(medrepId, targetId) -> objek list product BEGIN return \$this->db->query("SELECT * FROM producttask pt, product p WHERE pt.TARGET_ID == targetId AND pt.PRODUCT_ID == p.PRODUCT_ID AND DELETED = 0")->result() END FUNCTION </pre>	

4.1.4 Perancangan User Interface

Rancangan *user interface* yang dijelaskan pada sub bab ini dibuat untuk menggambarkan tampilan *mobile application* yang digunakan oleh pengguna untuk berinteraksi dengan aplikasi. Rancangan ini berdasarkan *system sequence diagram* yang telah dibuat.

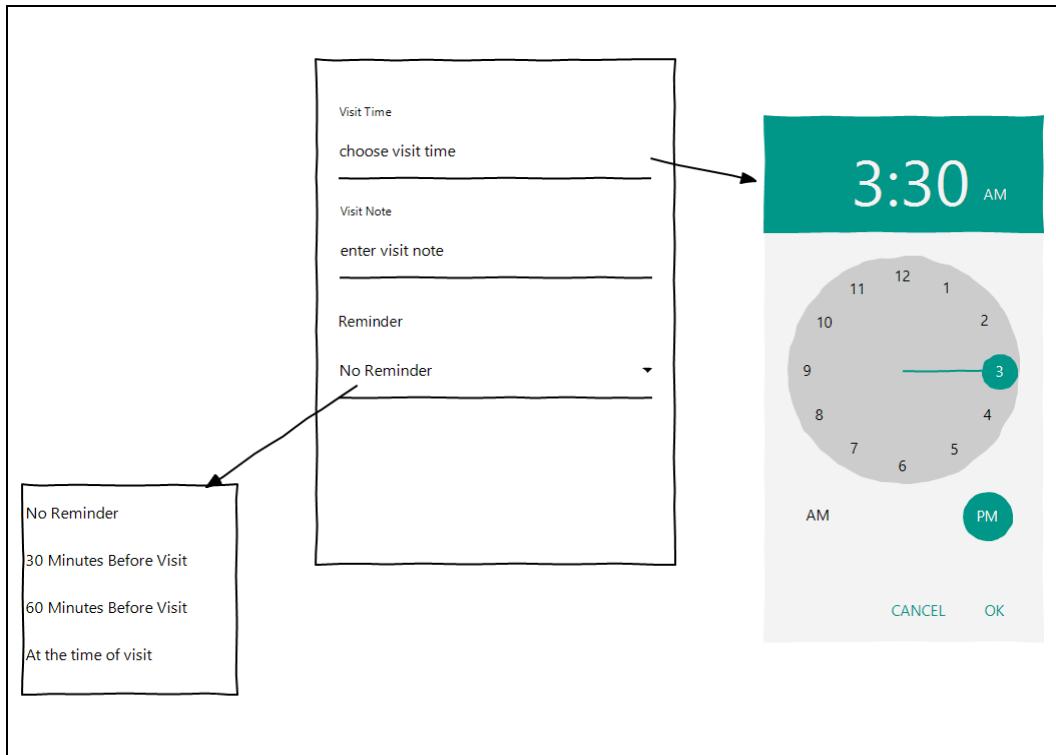
1. UI – Manage Visit Plan

Tabel 109 Tampilan Daftar Visit Plan

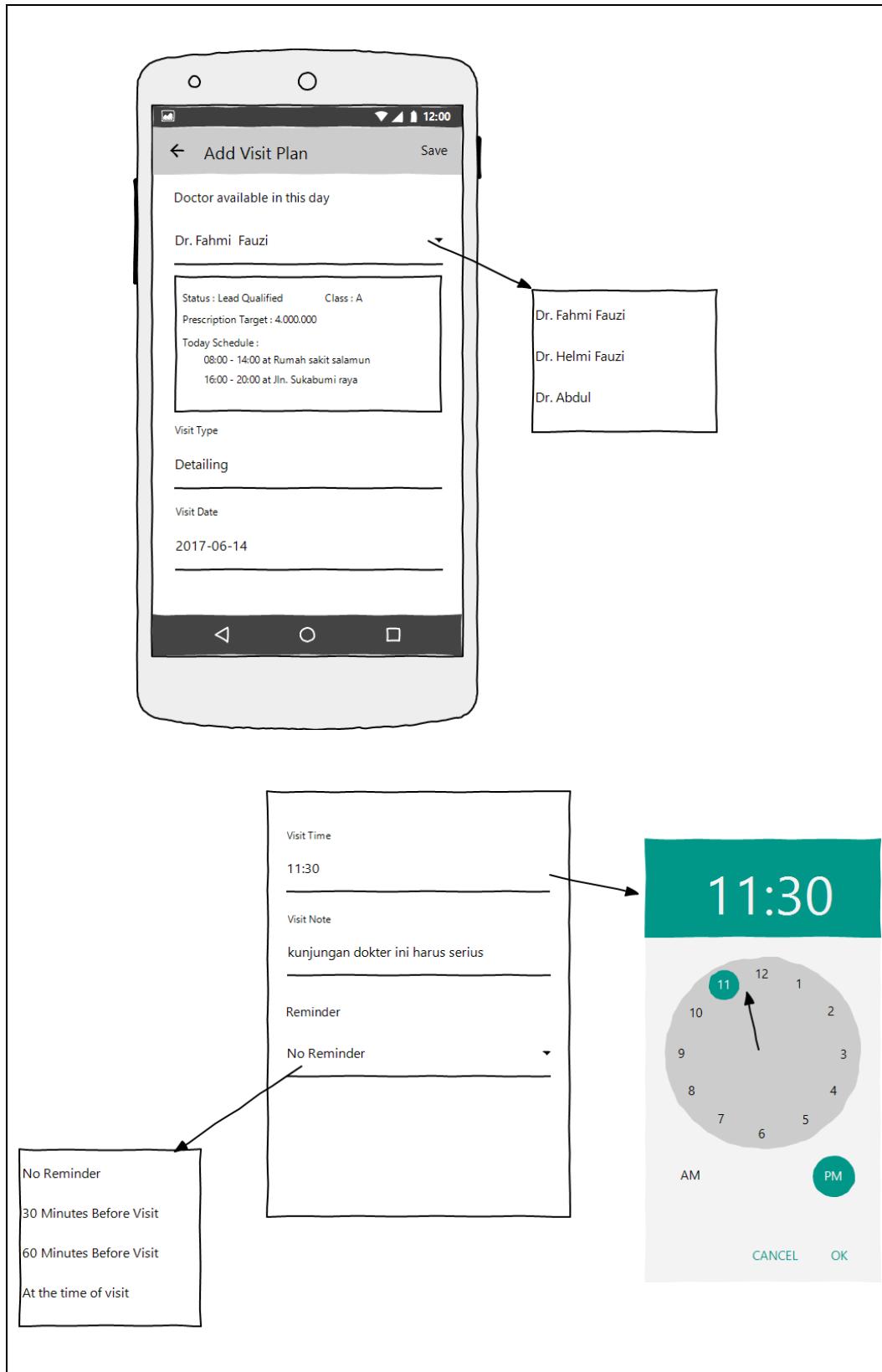
No. UI	GUI-VP.01
Nama UI	Tampilan Daftar Visit Plan
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>manage visit plan</i>.
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan semua data <i>visit plan</i> yang telah dibuat oleh <i>medical rep</i> . Tampilan untuk menampilkan daftar <i>visit plan</i> terdiri dari komponen <i>calendar view</i> untuk menampilkan kalender, <i>list view/recycler view</i> untuk menampilkan daftar <i>visit plan</i> sesuai tanggal yang dipilih pada <i>calendar view</i> , <i>button add</i> untuk masuk ke halaman <i>add visit plan</i> .
Gambar	

Tabel 110 Tampilan Add Visit Plan

No. UI	GUI-VP.02
Nama UI	Tampilan Add Visit Plan
Pengguna	Medical Representative
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>manage visit plan</i>. - Pengguna telah memilih salah satu tanggal pada <i>calendar view</i> di GUI-VP.01
Deskripsi	<p>Tampilan ini merupakan halaman untuk menampilkan <i>form</i> menambahkan <i>visit plan</i>. Data yang harus diisi pada <i>form</i> ini adalah nama dokter, tipe kunjungan (<i>visit type</i>), tanggal kunjungan (<i>visit date</i>), waktu kunjungan (<i>visit time</i>) note dan pengingat (<i>reminder</i>).</p> <p>Data <i>visit type</i> terisi secara otomatis sesuai dokter yang dipilih, jika dokter berstatus Lead Qualified, maka data <i>visit type</i> di isi <i>detailing</i>, jika dokter berstatus Contact, maka data <i>visit type</i> di isi <i>follow up</i>.</p> <p>Data <i>visit date</i> terisi secara otomatis sesuai pilihan tanggal pada tampilan GUI-VP.01.</p> <p>Komponen pada tampilan ini terdiri dari <i>edittext</i>, <i>spinner</i> untuk menampilkan data <i>dropdown</i> dokter dan <i>reminder</i> serta <i>timepicker</i>.</p>
Gambar	

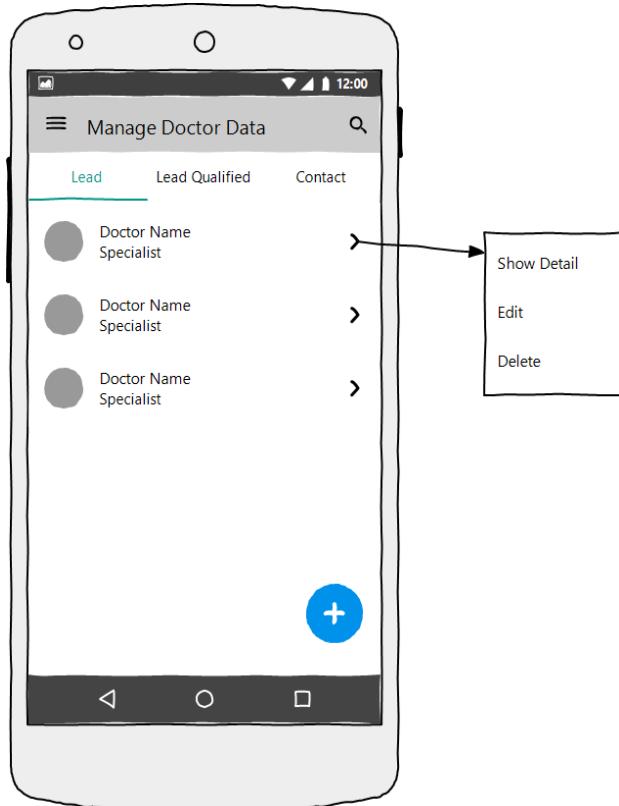
Tabel 111 Tampilan *Update Visit Plan*

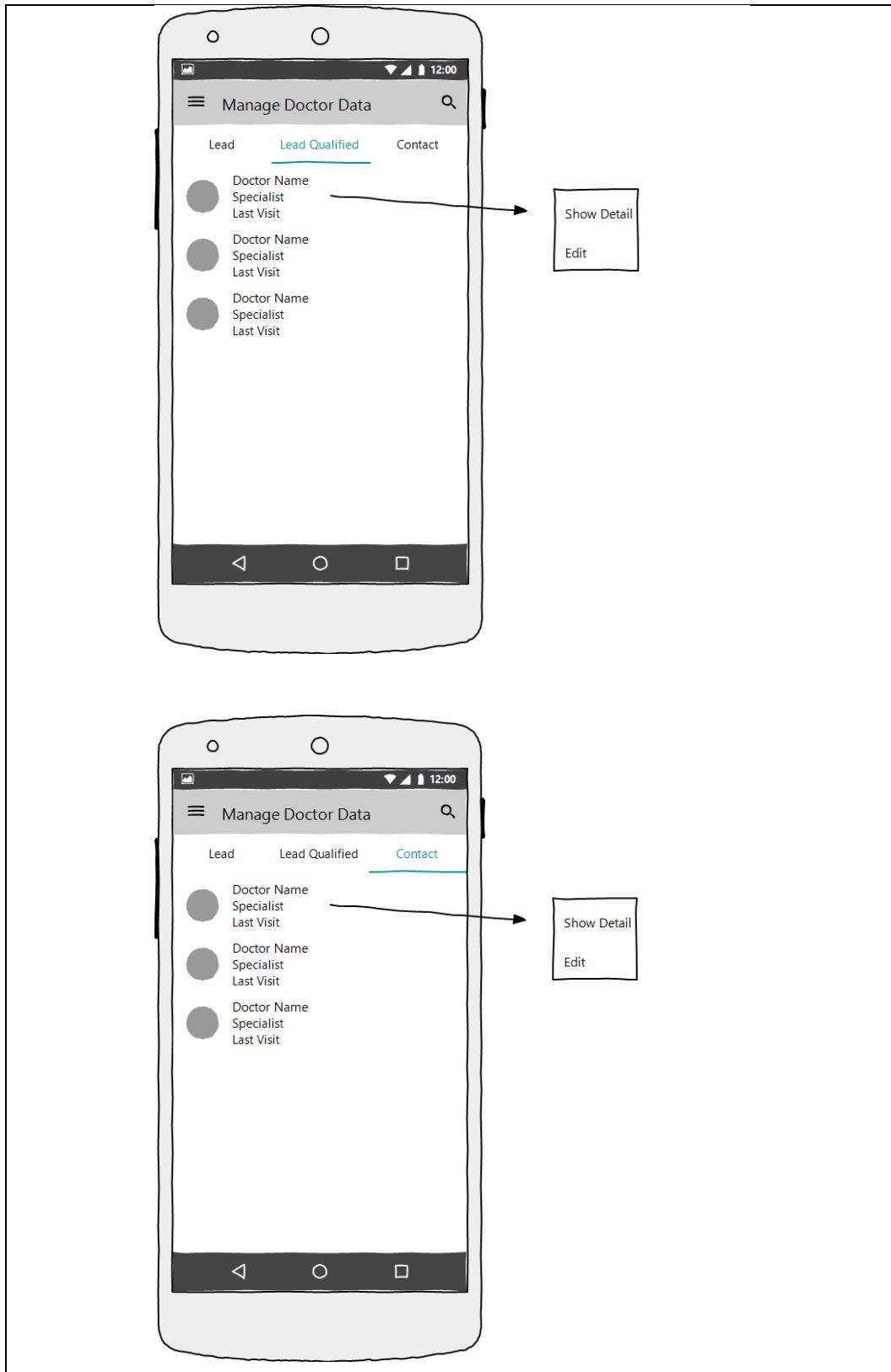
No. UI	GUI-VP.03
Nama UI	Tampilan <i>Update Visit Plan</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>manage visit plan</i>. - Pengguna telah memilih salah satu <i>visit plan</i> di GUI-VP.01
Deskripsi	<p>Tampilan ini merupakan halaman untuk menampilkan <i>form</i> mengubah <i>visit plan</i>. Data yang harus diisi pada <i>form</i> ini adalah nama dokter, tipe kunjungan (<i>visit type</i>), tanggal kunjungan (<i>visit date</i>), waktu kunjungan (<i>visit time</i>) note dan pengingat (<i>reminder</i>).</p> <p>Pada awal masuk ke tampilan, semua data otomatis terisi sesuai pilihan <i>visit plan</i> yang di pilih pada GUI-VP.01.</p> <p>Komponen pada tampilan ini terdiri dari <i>edittext</i>, <i>spinner</i> untuk menampilkan data <i>dropdown</i> dokter dan <i>reminder</i> serta <i>timepicker</i>.</p>
Gambar	



2. UI – Manage Doctor Data

Tabel 112 Tampilan Daftar Data Dokter *Lead* dan Dokter *Lead Qualified*

No. UI	GUI-MD.01
Nama UI	Tampilan Daftar Data Dokter <i>Lead</i> dan Dokter <i>Lead Qualified</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu manage doctor data.
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan semua data dokter yang menjadi tugas <i>medical rep</i> tersebut. Tampilan untuk menampilkan data dokter dibagi menjadi dua tab, yang mana masing-masing tab merepresentasikan status dokter (<i>lead</i> dan <i>lead qualified</i> ,)
Gambar	



Tabel 113 Tampilan Form add New Lead Doctor

No. UI	GUI-MD.02
Nama UI	Tampilan Form add New Lead Doctor
Pengguna	Medical Representative
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu manage doctor data. - Pengguna berada pada tab Lead - Pengguna telah memilih aksi “Add” yang direpresentasikan menggunakan icon +
Deskripsi	<p>Tampilan ini merupakan halaman untuk menampilkan form membuat data dokter lead baru yang terdiri dari komponen <i>textfield</i>, <i>combobox</i>, <i>timepicker</i> dan <i>button</i>. Jika data nama dokter, spesialist tidak diisi, maka ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice start time</i> telah diisi dan data <i>practice end time</i> atau data <i>practice location</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice end time</i> sudah terisi dan data <i>practice start time</i> atau data <i>practice location</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice location</i> sudah terisi dan data <i>practice start time</i> atau data <i>practice end time</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut.</p>
Gambar	

Tabel 114 Tampilan Form Update Doctor Data

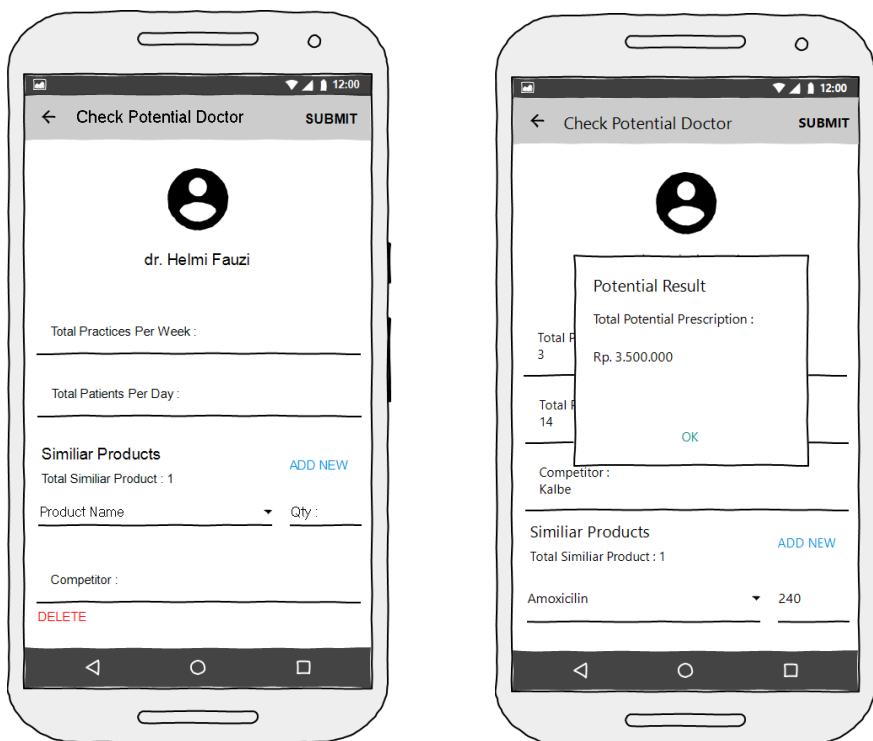
No. UI	GUI-MD.03
Nama UI	Tampilan Form Update Doctor Data
Pengguna	Medical Representative
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>manage doctor data</i>. - Pengguna menekan hingga 2 detik ke salah satu data dokter. - Pengguna telah memilih aksi “Update”
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan form mengubah data dokter yang terdiri dari komponen <i>textfield</i> , <i>combobox</i> , <i>timepicker</i> dan <i>button</i> . Jika data nama dokter, spesialist tidak di isi, maka ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice start time</i> telah di isi dan data <i>practice end time</i> atau data <i>practice location</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice end time</i> sudah terisi dan data <i>practice start time</i> atau data <i>practice location</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut. Jika data <i>practice location</i> sudah terisi dan data <i>practice start time</i> atau data <i>practice end time</i> belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut.
Gambar	

Tabel 115 Tampilan Detail Doctor Data

No. UI	GUI-MD.04
Nama UI	Tampilan <i>Detail Doctor Data</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>manage doctor data</i>. - Pengguna menekan salah satu data dokter atau menekan hingga dua detik ke salah satu data dokter. - Pengguna telah memilih aksi “Show Detail”
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan <i>detail</i> data dokter yang dipilih yang terdiri dari komponen <i>textview</i> dan <i>button</i> .
Gambar	

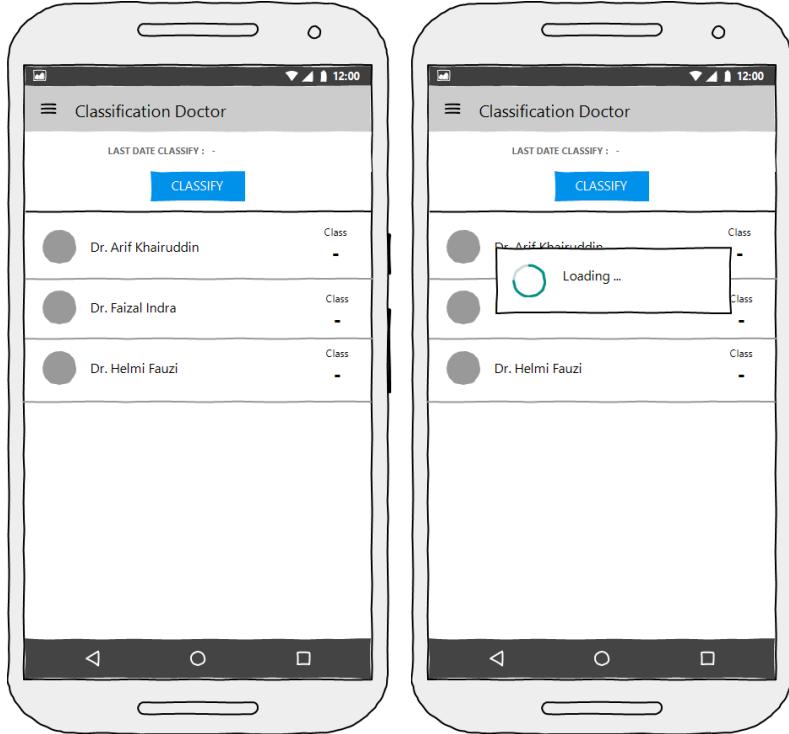
3. UI – Check Potential Doctor

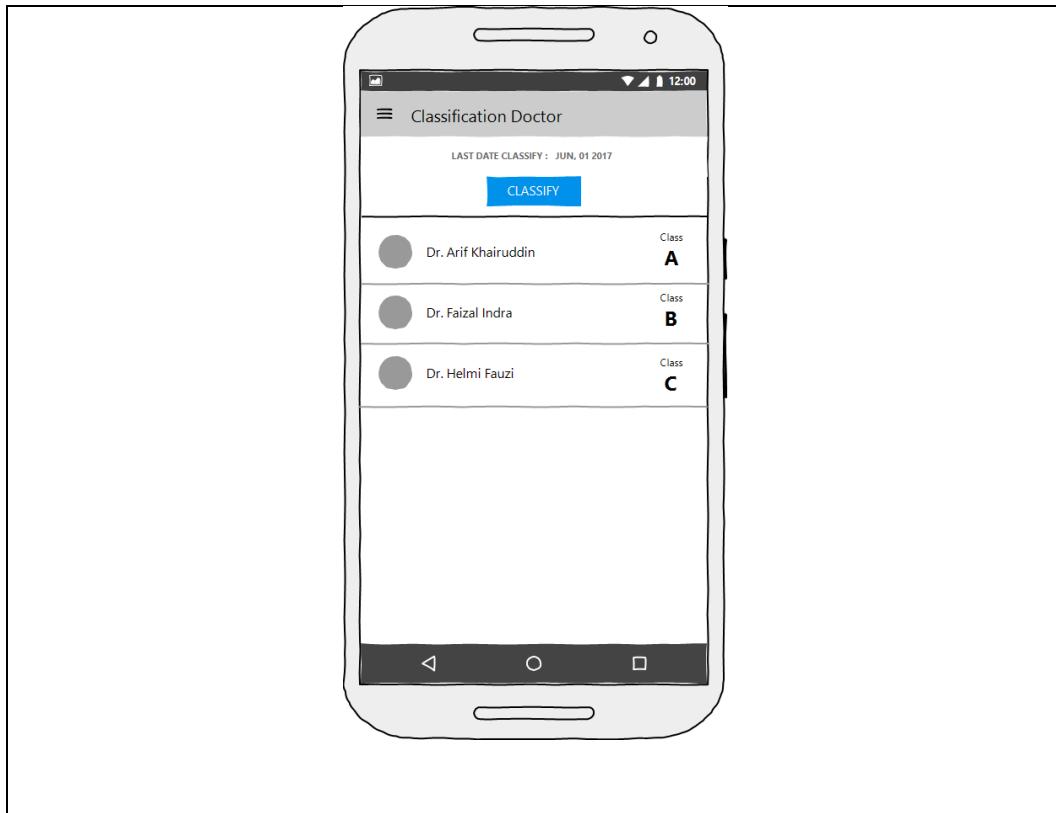
Tabel 116 Tampilan Form Check Potential Doctor

No. UI	GUI-CP.01
Nama UI	Tampilan Form Check Potential Doctor
Pengguna	Medical Representative
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih dokter. - Pengguna telah memilih aksi <i>check potential doctor</i>.
Deskripsi	<p>Tampilan ini merupakan halaman yang menampilkan <i>form</i> untuk mencatat data peresepan dokter. Tampilan ini terdiri dari komponen <i>textview</i>, <i>imageview</i>, <i>button</i>, <i>spinner/dropdown</i>, dan <i>edittext</i>.</p> <p>Jika pengguna melakukan submit, maka akan tampil <i>alertdialog</i> yang menampilkan informasi besar potensi peresepan dokter.</p>
Gambar	

4. UI – *Classify Doctor*

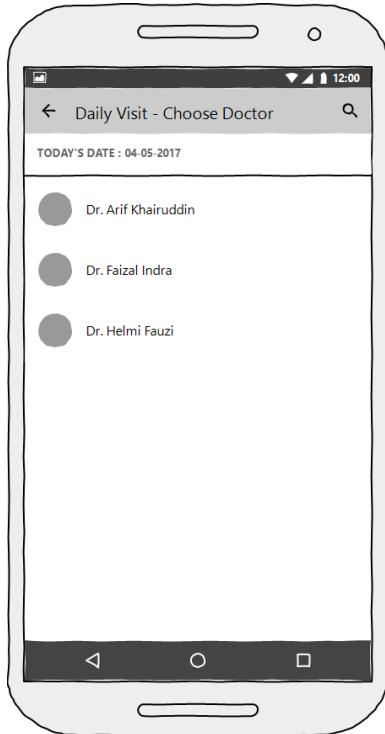
Tabel 117 Tampilan *Classify Doctor*

No. UI	GUI-CD.01
Nama UI	Tampilan <i>Classify Doctor</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none">- Pengguna telah masuk ke dalam aplikasi.- Pengguna telah memilih dokter dengan status ‘Lead Qualified’.
Deskripsi	Tampilan ini merupakan halaman yang menampilkan <i>list</i> dokter beserta klasifikasinya. Tampilan ini terdiri dari komponen <i>textview</i> , <i>imageview</i> , <i>button</i> , dan <i>listview</i> .
Gambar	

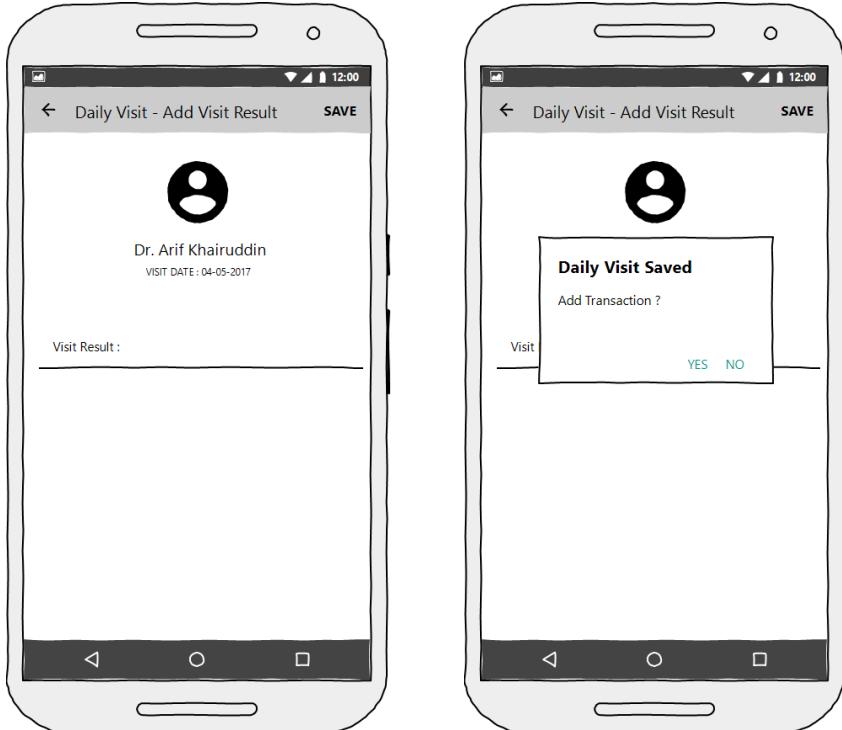


5. UI – Add Daily Visit

Tabel 118 Tampilan Pilihan Dokter yang Akan Dikunjungi

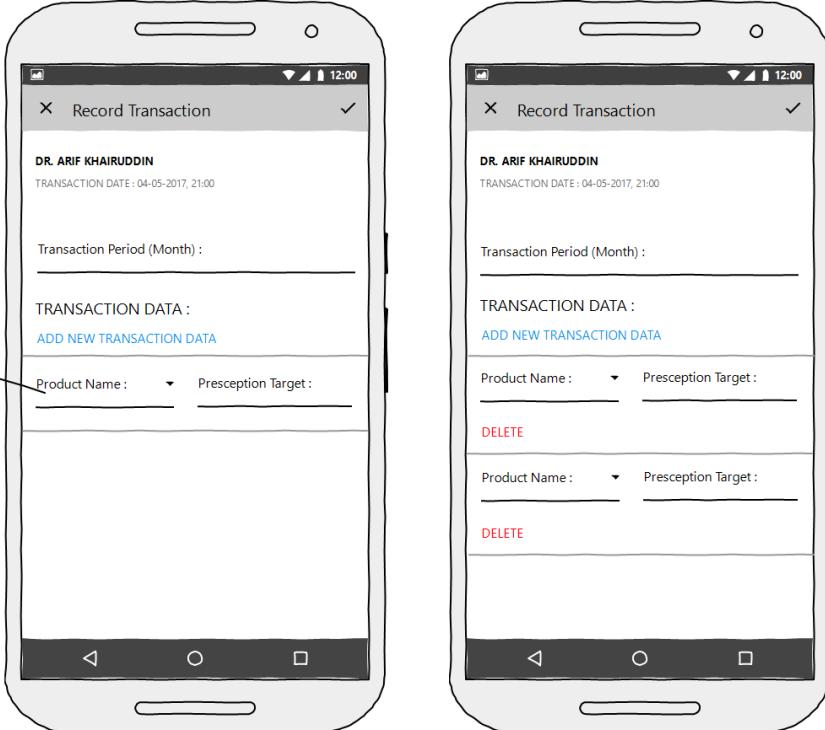
No. UI	GUI-DV.01
Nama UI	Tampilan Pilihan Dokter yang Akan Dikunjungi
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none">- Pengguna telah masuk ke dalam aplikasi.- Pengguna telah memilih menu <i>add daily visit</i>.
Deskripsi	Tampilan ini merupakan halaman yang menampilkan semua data dokter dengan status <i>lead qualified</i> dan <i>contact</i> . Tampilan ini terdiri dari komponen <i>textview</i> , <i>imageview</i> , dan <i>list view/recycler view</i> untuk menampilkan <i>list</i> dokter.
Gambar	

Tabel 119 Tampilan Add Daily Visit

No. UI	GUI-DV.02
Nama UI	Tampilan Add Daily Visit
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none">- Pengguna telah masuk ke dalam aplikasi.- Pengguna telah memilih menu <i>add daily visit</i>.- Pengguna telah memilih dokter.
Deskripsi	Tampilan ini merupakan halaman yang menampilkan <i>form</i> untuk data hasil kunjungan. Data yang harus diisi pada <i>form</i> ini adalah hasil kunjungan. Tampilan ini terdiri dari komponen <i>textview</i> , <i>imageview</i> , <i>button</i> , <i>edittext</i> , dan <i>alertdialog</i> (apabila data hasil kunjungan telah tersimpan).
Gambar	 The image shows two side-by-side screenshots of a mobile application. Both screens have a header bar with a back arrow, the title 'Daily Visit - Add Visit Result', and a 'SAVE' button. The left screen displays a placeholder profile icon, the name 'Dr. Arif Khairuddin', and the visit date 'VISIT DATE : 04-05-2017'. Below this is a large empty text area labeled 'Visit Result :'. The right screen shows the same header and a confirmation message 'Daily Visit Saved' in a box. Below the message is the question 'Add Transaction ?' with a 'Visit' button. At the bottom of the right screen is a dialog box with 'YES' and 'NO' buttons.

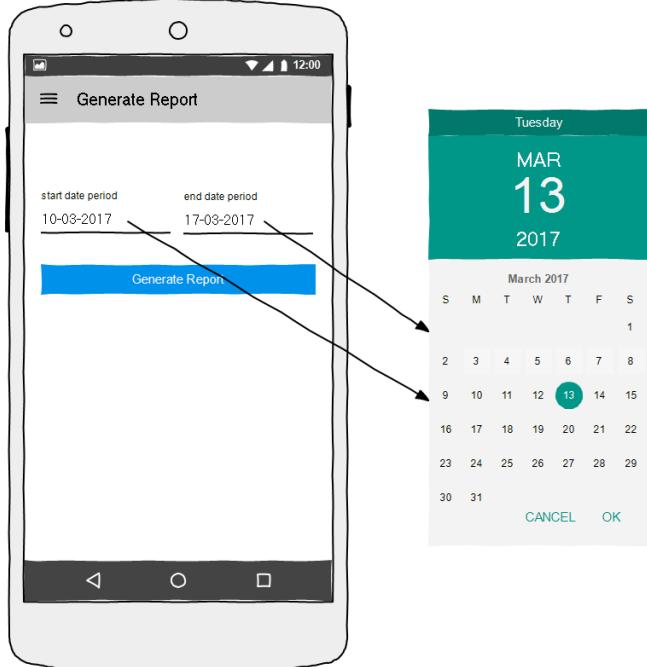
6. UI – Record Transaction

Tabel 120 Tampilan Record Transaction

No. UI	GUI-RT.01
Nama UI	Tampilan Record Transaction
Pengguna	Medical Representative
Precondition	Pengguna telah masuk ke dalam aplikasi. Pengguna telah menambahkan <i>daily visit</i> .
Deskripsi	Tampilan ini merupakan halaman yang menampilkan <i>form</i> untuk menambahkan data transaksi. Data yang di- <i>input</i> pada <i>form</i> ini adalah periode transaksi, data transaksi yang terdiri dari nama produk dan target peresepan. Tampilan ini terdiri dari komponen <i>textview</i> , <i>button</i> , dan <i>edittext</i> .
Gambar	

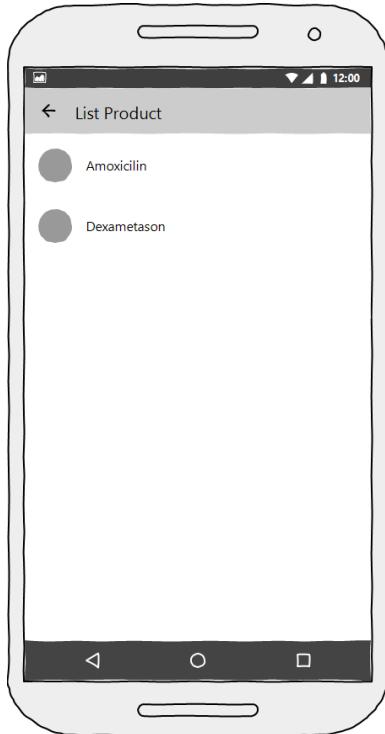
7. UI – Generate Report

Tabel 121 Tampilan *Generate Report*

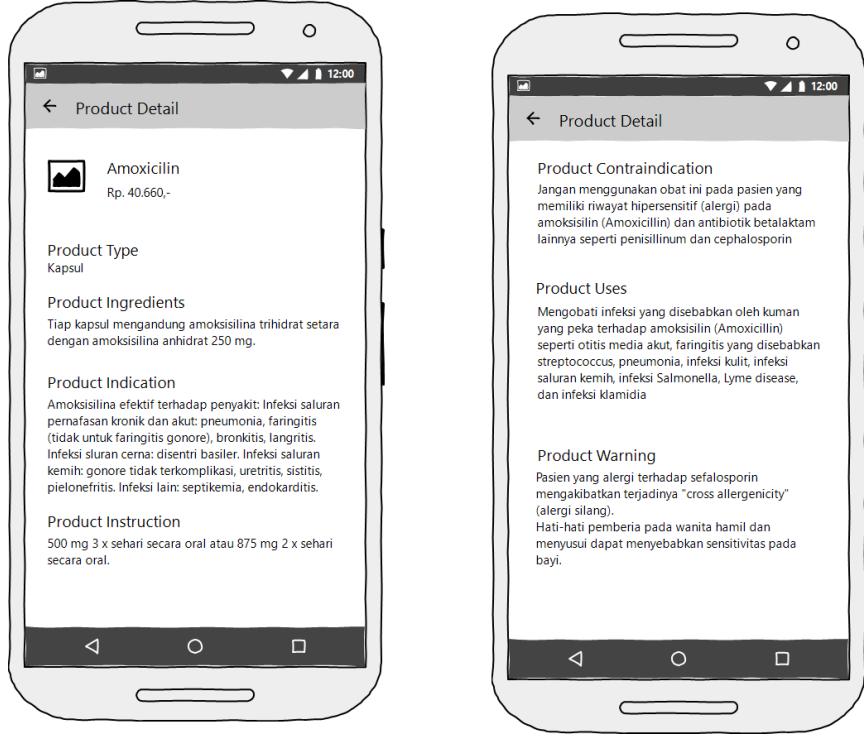
No. UI	GUI-GR.01
Nama UI	Tampilan <i>Generate Report</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>generate report</i>.
Deskripsi	Tampilan ini merupakan halaman <i>form</i> untuk meng- <i>generate</i> laporan yang terdiri dari komponen <i>textfield</i> , <i>datepicker</i> , dan <i>button</i> . Jika terdapat data yang belum terisi, maka akan ada pesan peringatan untuk mengisi data tersebut. Jika isi <i>textfield end date period</i> lebih kecil dari isi <i>textfield start date period</i> , maka akan ada pesan peringatan untuk memilih tanggal ulang. Jika memilih tanggal yang sudah di- <i>generate</i> sebelumnya, maka akan ada pesan pemberitahuan bahwa data pada tanggal tersebut telah di- <i>generate</i>
Gambar	

8. UI – Show Product

Tabel 122 Tampilan *List Product*

No. UI	GUI-SP.01
Nama UI	Tampilan <i>List Product</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none">- Pengguna telah masuk ke dalam aplikasi.- Pengguna telah memilih menu <i>show product</i>.
Deskripsi	Tampilan ini merupakan halaman yang menampilkan daftar produk yang menjadi target pengguna. Tampilan ini terdiri dari komponen <i>textview</i> , <i>imageview</i> , dan <i>list view/ recycler view</i> untuk menampilkan <i>list produk</i> .
Gambar	

Tabel 123 Tampilan *Detail Product*

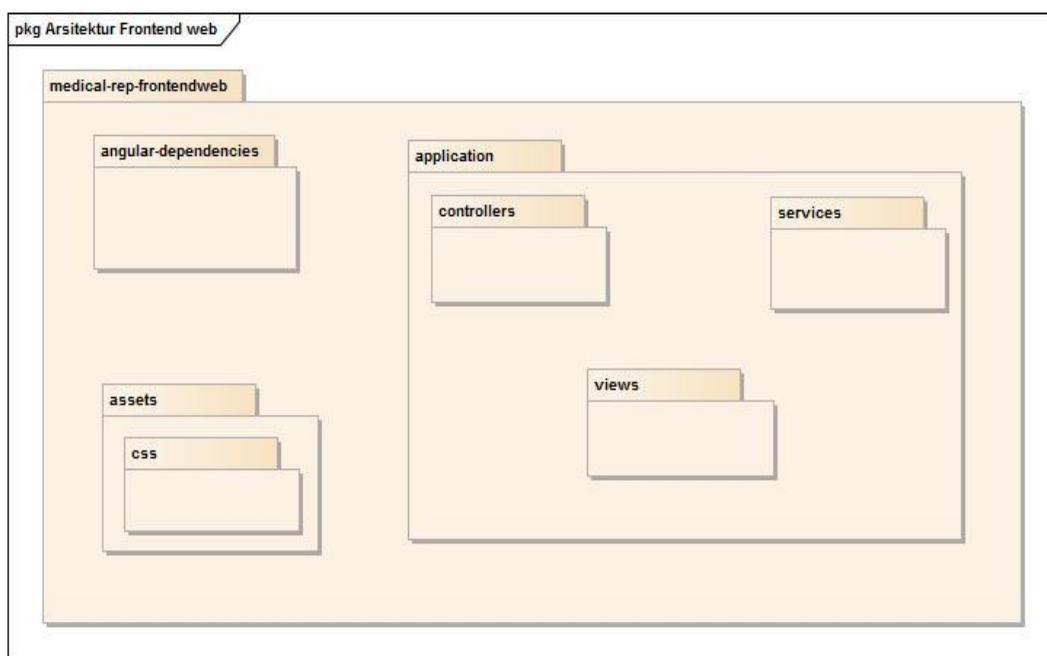
No. UI	GUI-SP.02
Nama UI	Tampilan <i>Detail Product</i>
Pengguna	<i>Medical Representative</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih produk.
Deskripsi	Tampilan ini merupakan halaman yang menampilkan informasi produk sesuai yang dipilih pada GUI-SP.01. Tampilan ini terdiri dari komponen <i>textview</i> , dan <i>imageview</i> .
Gambar	

4.2 Perancangan Web Frontend

Sub bab ini dijelaskan mengenai perancangan *web frontend* yang dibangun meliputi: perancangan arsitektur, *class*, interaksi antar objek yang digambarkan dengan *sequence diagram* dan *user interface*.

4.2.1 Perancangan Arsitektur

Perancangan ini dilakukan untuk memberikan gambaran mengenai arsitektur *Web Frontend* pada pengembangan aplikasi CRM yang digambarkan pada Gambar 84.



Gambar 84 Arsitektur *Frontend Website*

Keterangan *package diagram* pada *frontend website* dijelaskan pada Tabel 124.

Tabel 124 Penjelasan Arsitektur *Frontend Website*

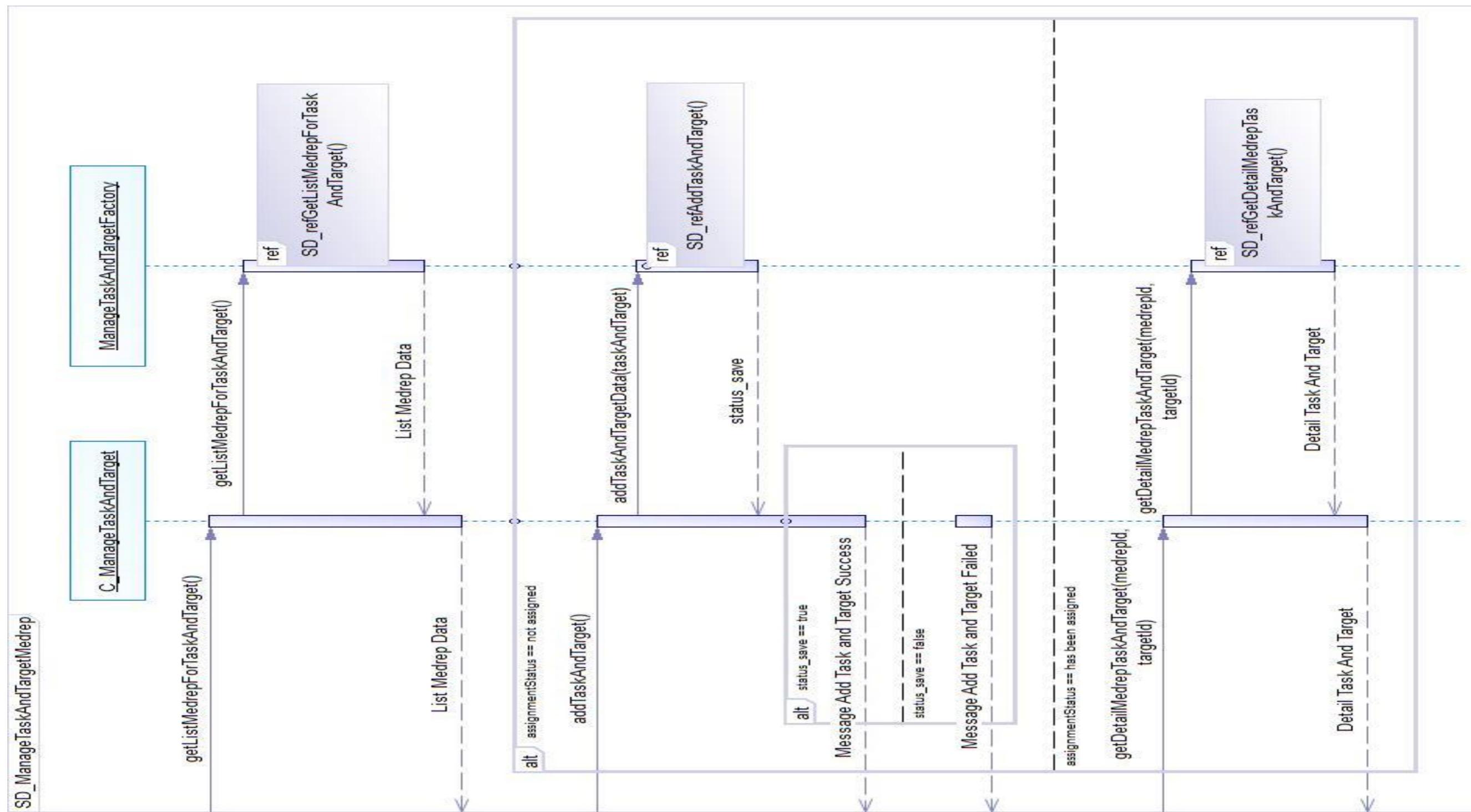
Nama Package	Deskripsi
<i>angular-dependencies</i>	Berisi <i>file – file</i> library <i>dependency Angular</i> yang digunakan untuk kebutuhan aplikasi.
<i>assets</i>	Berisi <i>file – file</i> pendukung untuk membangun aplikasi seperti <i>css</i> , <i>image</i> dan <i>javascript</i>
<i>css</i>	Berisi <i>file – file</i> <i>css</i> untuk digunakan pada aplikasi
<i>application</i>	Berisi <i>file – file</i> buatan yang merupakan konten pada aplikasi
<i>controllers</i>	Berisi <i>file – file</i> <i>controller</i> untuk mengatur alur kerja sistem

<i>services</i>	Berisi <i>file – file</i> service untuk mengakses dan menerima data dari <i>backend</i>
<i>views</i>	Berisi <i>file – file</i> html untuk menampilkan halaman <i>web</i>

4.2.2 Perancangan Interaksi Antarobjek

Pada sub bab ini dijelaskan mengenai perancangan interaksi antarobjek untuk *website application* pada pengembangan aplikasi CRM.

1. SD – Manage Medical Representative's Task and Target



Gambar 85 SD Manage Medical Representative's Task and Target

Tabel 125 Keterangan SD_ManageTaskAndTargetMedrep

ID Sequence diagram	SD-MTT-01
Nama Sequence diagram	SD_ManageTaskAndTargetMedrep
Nama Class Terkait	C_ManageTaskAndTarget, ManageTaskAndTargetFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengelola tugas dan target <i>medical rep</i> pada bagian <i>web frontend</i> .
Method yang terkait	getListMedrepForTaskAndTarget(), addTaskAndtarget(), getDetailMedrepTaskAndTarget(medrepId, targetId), addTaskAndTargetData(taskAndTarget)
Sequence diagram yang terkait	SD_refGetListMedrepForTaskAndTarget, SD_refAddTaskAndTarget, SD_refGetDetailMedrepTaskAndTarget
Algoritma	<pre> Kamus data listMedrep : variabel array objek PROCEDURE getListMedrepForTaskAndTarget() BEGIN manageTaskAndTargetFactory.getMedrepForTaskAndTarget(). then(function(result) { listMedrep ← result.data.listMedrepData }) END PROCEDURE PROCEDURE addTaskAndTarget() BEGIN taskAndTargetData ← array('medrepId' ← \$scope.medrepId, 'medrepTarget' ← \$scope.medrepTarget, 'note' ← \$scope.note, 'startDateTargetPeriod' ← \$scope.startDateTargetPeriod, 'endDateTargetPeriod' ← \$scope.endDateTargetPeriod, 'listProductTask' ← \$scope.products) manageTaskAndTargetFactory.addTaskAndTargetData(taskAnd TargetData).then(function(result) { statusSave ← result.data.status IF statusSave == TRUE THEN WRITE (layar) "save success" ELSE WRITE (layar) "save failed" END IF }) END PROCEDURE PROCEDURE getDetailMedrepTaskAndTarget(medrepId, targetId) BEGIN manageTaskAndTargetFactory.getDetailMedrepTaskAndTarget (medrepId, targetId).then(function(result) { </pre>

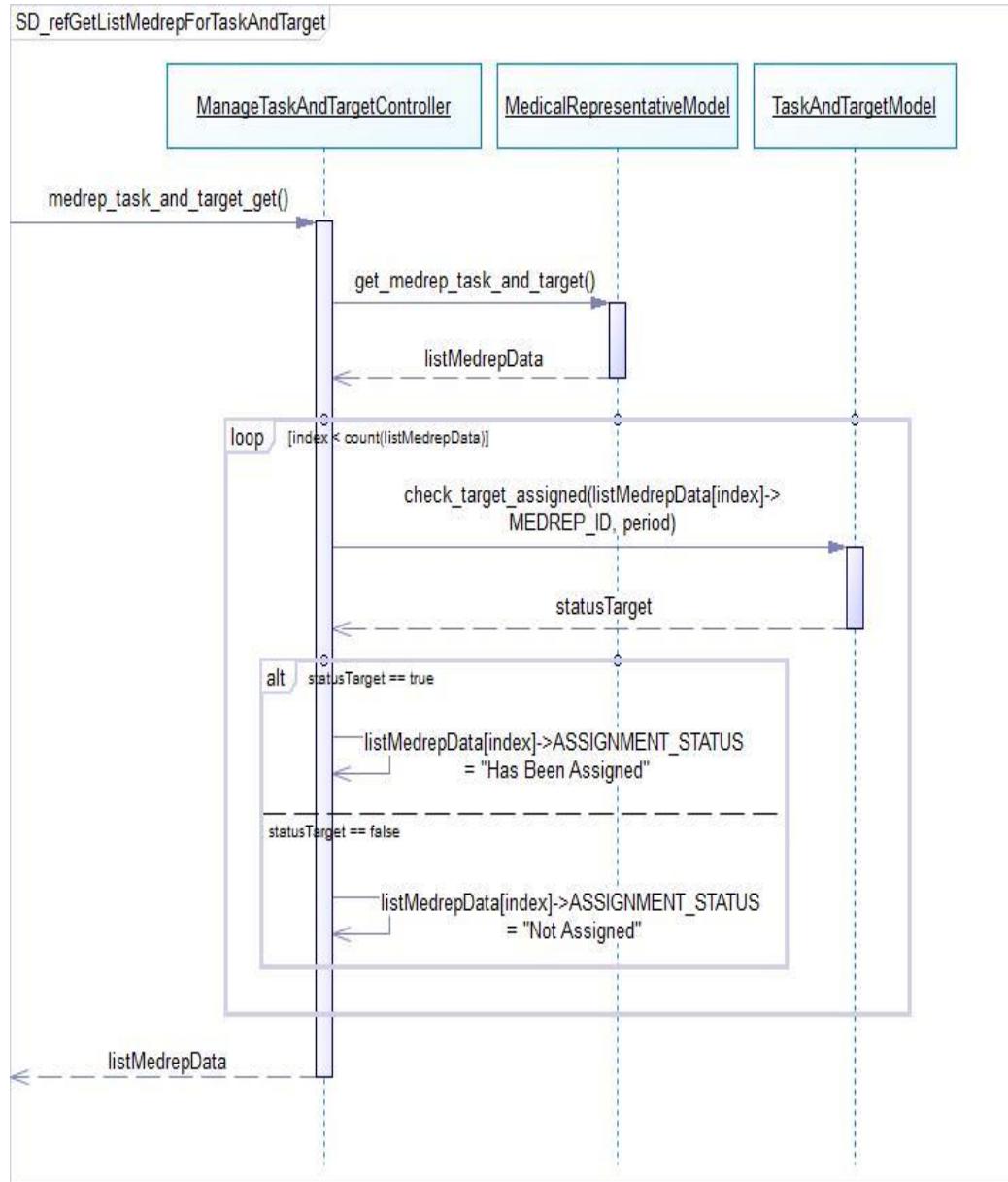
```
        detailTaskAndTarget ←
            result.data.detailTaskAndTarget
        $scope.medrepId ← detailTaskAndTarget.MEDREP_ID
        $scope.medrepName ← detailTaskAndTarget.MEDREP_NAME
        $scope.medrepTarget ←
            detailTaskAndTarget.MEDICAL REPRESENTATIVE TARGET
        $scope.startDateTargetPeriod ←
            detailTaskAndTarget.START_DATE_TARGET_PERIOD
        $scope.endDateTargetPeriod ←
            detailTaskAndTarget.END_DATE_TARGET_PERIOD
        $scope.products ← detailTaskAndTarget.listProductTask
        $scope.note ← detailTaskAndTarget.NOTE
    })
END PROCEDURE

FUNCTION manageTaskAndTargetFactory.getMedrepForTaskAndTarget()
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/managetaskandtargetcontroller/medrep_task_and_t
        arget_get")
END FUNCTION

FUNCTION manageTaskAndTargetFactory.addTaskAndTargetData(
taskAndTargetData)
BEGIN
    return $http.post("http://localhost/medical-rep-
        backend/managetaskandtargetcontroller/add_task_and_targ
        et_post", taskAndTargetData)
END FUNCTION

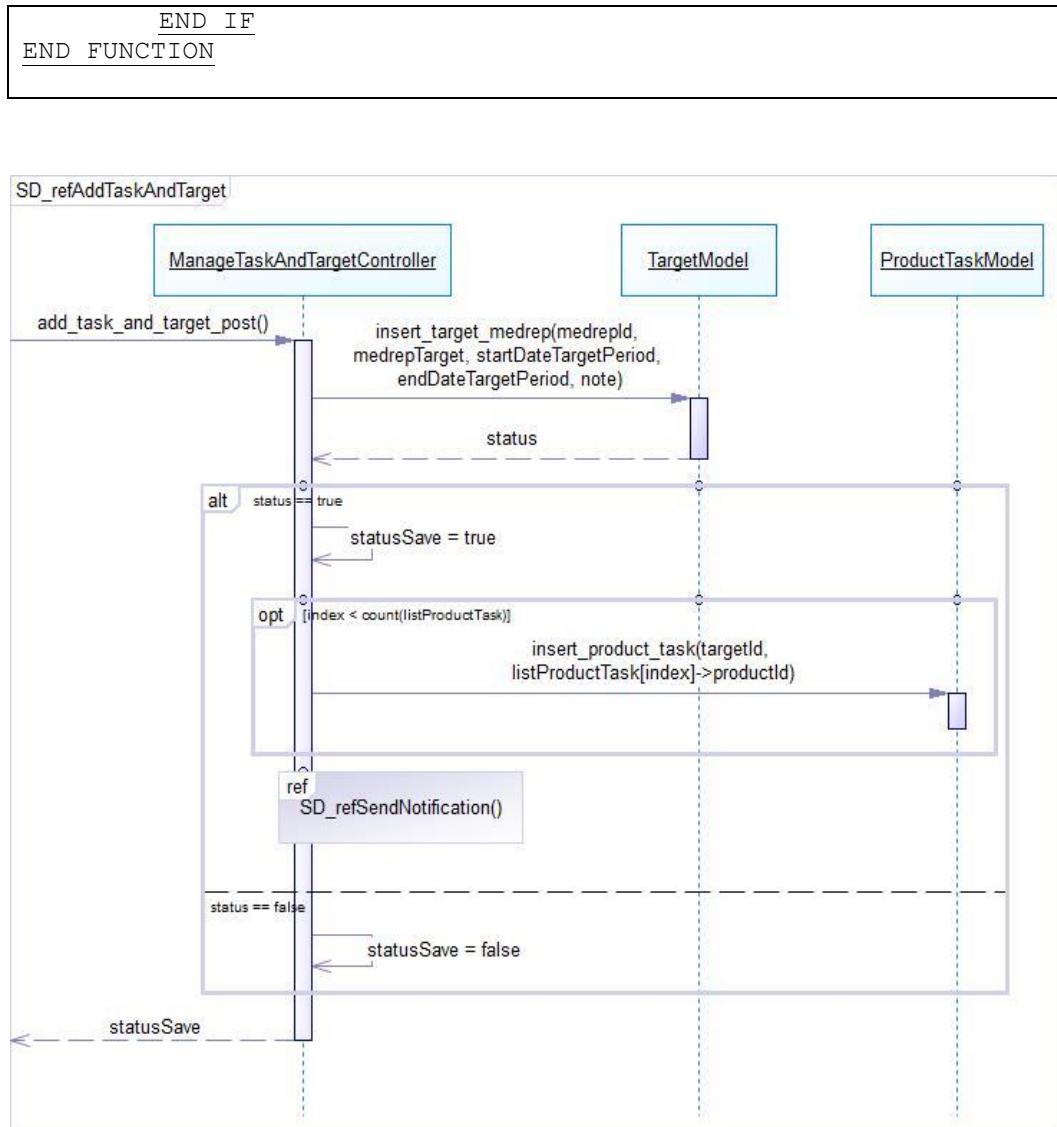
FUNCTION
manageTaskAndTargetFactory.getDetailMedrepTaskAndTarget(medrepId
, targetId)
BEGIN

    return $http.get("http://localhost/medical-rep-
        backend/managetaskandtargetcontroller/detail_task_and_t
        arget_get?medrepId=" + medrepId + "&targetId=" +
        targetId")
END FUNCTION
```

Gambar 86 SD Referensi *Get List Medrep For Task and Target*

Tabel 126 Keterangan SD_refGetListMedrepForTaskAndTarget

ID Sequence diagram	SD-MTT-02
Nama Sequence diagram	SD_refGetListMedrepForTaskAndTarget
Nama Class Terkait	ManageTaskAndTargetController, MedicalRepresentativeModel, TargetModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan <i>list medical rep</i> pada pengelolaan tugas dan target.
Method yang terkait	medrep_task_and_target_get(), get_medrep_task_and_target(), check_target_assigned(medrepId, month, year)
Sequence diagram yang terkait	-
Algoritma	
<pre> Kamus data listMedrepData : variabel array objek PROCEDURE medrep_task_and_target_get() BEGIN listMedrepData ← \$this->medicalrepresentativemodel-> get_medrep_task_and_target() month ← date('m') year ← date('Y') FOR EACH listMedrep AS medrep statusTarget ← \$this->targetmodel-> check_target_assigned(medrep->MEDREP_ID, month, year) IF statusTarget == TRUE THEN medrep->ASSIGNMENT_STATUS ← "Has Been Assigned" ELSE Medrep->ASSIGNMENT_STATUS ← "Not Assigned" END IF END FOR END PROCEDURE FUNCTION get_medrep_task_and_target() BEGIN return \$this->db->query("SELECT * FROM medicalrepresentative")->result() END FUNCTION FUNCTION check_target_assigned(medrepId, month, year) BEGIN query ← \$this->db->query("SELECT * FROM target WHERE MEDREP_ID = medrepId AND MONTH(START_DATE_TARGET_PERIOD) = month AND YEAR(START_DATE_TARGET_PERIOD) = year")->num_rows() IF query > 0 THEN return TRUE ELSE return FALSE END IF END FUNCTION </pre>	



Gambar 87 SD Referensi Add Task and Target

Tabel 127 Keterangan SD_refAddTaskAndTarget

ID Sequence diagram	SD-MTT-03
Nama Sequence diagram	SD_refAddTaskAndTarget
Nama Class Terkait	ManageTaskAndTargetController, TargetModel, ProductTaskModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan tugas dan target untuk <i>medical rep</i> .
Method yang terkait	<code>add_task_and_target_post()</code> , <code>insert_task_and_target(medrepId, medrepTarget, startDateTargetPeriod, endDateTargetPeriod, note)</code> , <code>insert_product_task(targetId, productId)</code>

<i>Sequence diagram yang terkait</i>	SD_refSendNotification
Algoritma	
<pre> Kamus data medrepId : variabel tunggal bertipe String medrepTarget : variabel tunggal bertipe integer startDateTargetPeriode : variabel tunggal bertipe Date endDateTargetPeriode : variabel tunggal beripte Date note : variabel tunggal bertipe String listProductTask : variabel array objek PROCEDURE add_task_and_target_post() BEGIN medrepId ← \$this->post("medrepId") medicalRepresentativeTarget ← \$this-> post("medrepTarget") startDateTargetPeriod ← \$this-> post("startDateTargetPeriod") endDateTargetPeriod ← \$this->post("endDateTargetPeriod") note ← \$this->post("note") listProductTask ← \$this->post("listProduct") statusSave ← \$this->targetmodel-> insert_target_medrep(medrepId, medrepTarget, startDateTargetPeriod, endDateTargetPeriod, note) IF statusSave == TRUE THEN target ← \$this->targetmodel->get_target_id(medrepId, startDateTargetPeriod, endDateTargetPeriod) index ← 0 FOR (index TO count(listProductTask) - 1, index + 1) DO \$this->producttaskmodel- >insert_product_task(target->TARGET_ID, listProductTask[index]) END FOR END IF END PROCEDURE FUNCTION insert_target_medrep(medrepId, startDateTargetPeriod, endDateTargetPeriod, note) -> boolean BEGIN targetData ← array('MEDREP_ID' ← medrepId, 'MEDICAL_REPRESENTATIVE_TARGET' ← medrepTarget, 'START_DATE_TARGET_PERIOD' ← startDateTargetPeriod, 'END_DATE_TARGET_PERIOD' ← endDateTargetPeriod 'NOTE' ← note) query ← \$this->db->insert('target', targetData) </pre>	

```

IF query == TRUE THEN
    return TRUE
ELSE
    return FALSE
END IF

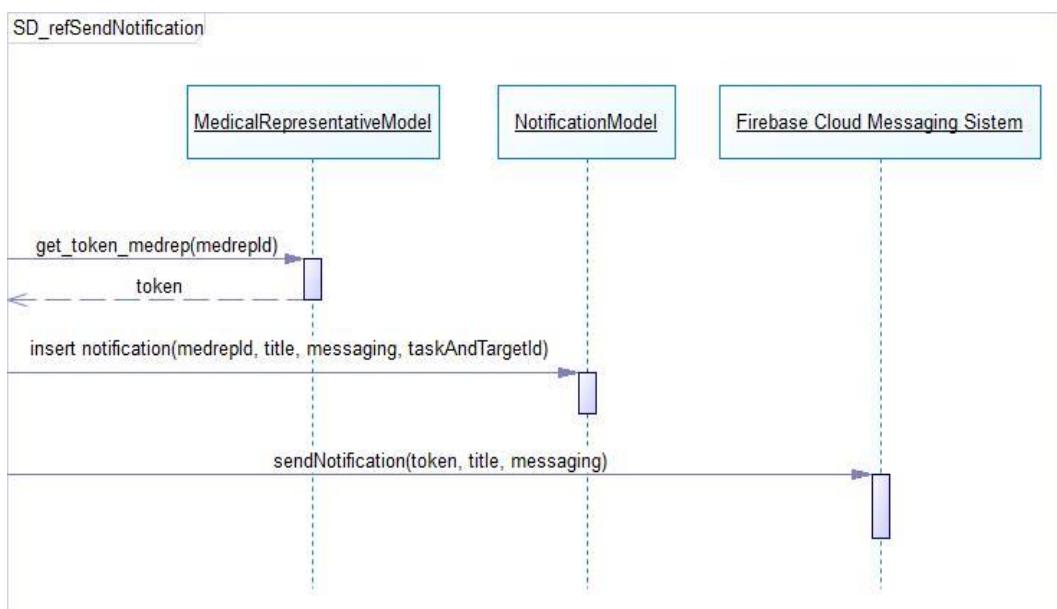
END FUNCTION

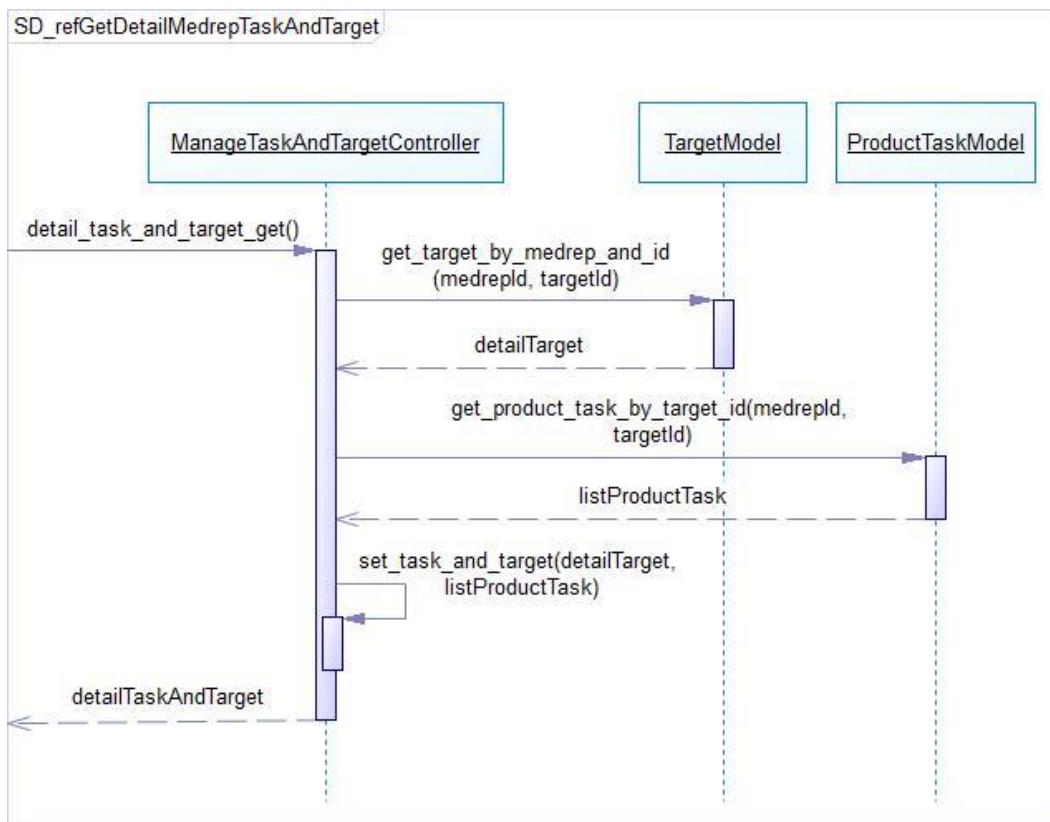
FUNCTION get_target_id(medrepId, startDateTargetPeriod,
endDateTargetPeriod) -> array objek
BEGIN
    return $this->db->query("SELECT TARGET_ID FROM target
WHERE MEDREP_ID = medrepId AND START_DATE_TARGET_PERIOD =
startDateTargetPeriod AND END_DATE_TARGET_PERIOD =
endDateTargetPeriod")->row()

END FUNCTION

PROCEDURE insert_product_task(targetId, productId
BEGIN
    productTask ← array(
        'TARGET_ID' ← targetId,
        'PRODUCT_ID' ← productId
    )
    $this->db->insert('producttask', productTask)
END PROCEDURE

```

Gambar 88 SD Referensi *Send Notification*

Gambar 89 SD Referensi *Get Detail Medrep Task and Target*

Tabel 128 Keterangan SD_refGetDetailMedrepTaskAndTarget

ID Sequence diagram	SD-MTT-05
Nama Sequence diagram	SD_refGetDetailMedrepTaskAndTarget
Nama Class Terkait	ManageTaskAndTargetController, TargetModel, ProductTaskModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan detail tugas dan target berdasarkan Id medical rep.
Method yang terkait	detail_task_and_target_get(), get_target_by_medrep_and_id(medrepId, targetId), get_product_task_by_target_id(targetId)
Sequence diagram yang terkait	-

Algoritma

```
Kamus data
    medrepId : variabel tunggal bertipe string
    targetId : variabel tunggal bertipe integer
    detailTaskAndTarget : variabel array objek

PROCEDURE detail_task_and_target_get()
BEGIN
    medrepId ← $this->get("medrepId")
    targetId ← $this->get("targetId")

    detailTaskAndTarget ← $this->targetmodel-
        get_target_by_medrep_and_id(medrepId, targetId)

    detailTaskAndTarget->listProductTask ← $this-
        >producttaskmodel-
        >get_product_task_by_target_id(targetId)

END PROCEDURE

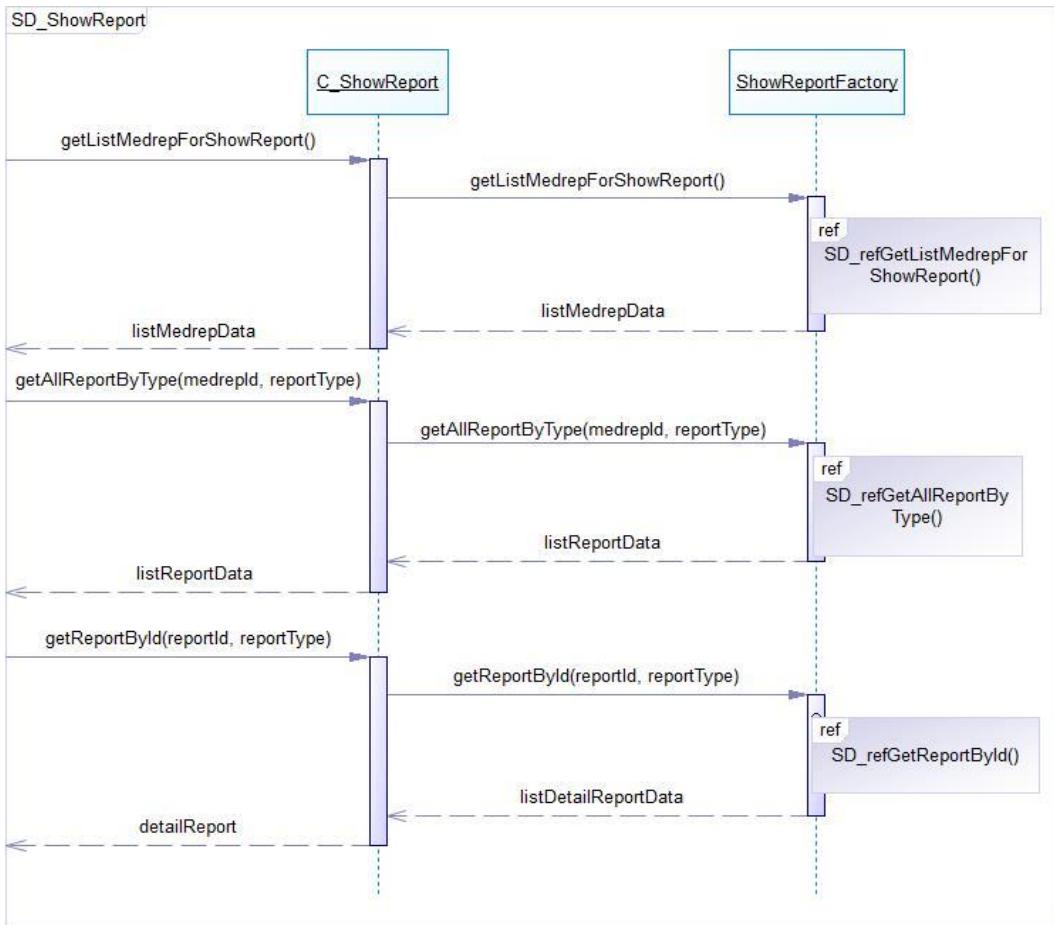
FUNCTION get_target_by_medrep_and_id(medrepId, targetId) ->
array objek
BEGIN
    return $this->db->query("SELECT t.*, m.MEDREP_NAME FROM
target t, medicalrepresentative m WHERE t.MEDREP_ID =
m.MEDREP_ID AND t.MEDREP_ID = 'medrepId' AND t.TARGET_ID =
'targetId'")->row()

END FUNCTION

FUNCTION get_product_task_by_target_id(targetId) -> array objek
BEGIN
    return $this->db->query("SELECT pt.PRODUCT_ID,
p.PRODUCT_NAME FROM producttask pt, product p WHERE
pt.PRODUCT_ID = p.PRODUCT_ID AND pt.TARGET_ID =
'targetId'")->result()

END FUNCTION
```

2. SD – Show Report



Gambar 90 SD Show Report

Tabel 129 Keterangan SD_ShowReport

ID Sequence diagram	SD-SR-01
Nama Sequence diagram	SD_ShowReport
Nama Class Terkait	C_ShowReport, ShowReportFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses untuk menampilkan laporan kunjungan ataupun laporan transaksi <i>medical rep</i> pada bagian <i>web frontend</i> .
Method yang terkait	<code>getListMedrepForShowReport()</code> , <code>getAllReportByType(medrepId, reportType)</code> , <code>getReportById(reportId, reportType)</code>
Sequence diagram yang terkait	SD_refGetListMedrepForShowReport, SD_refGetAllReportByType, SD_refGetReportById
Algoritma	<p>Kamus data</p> <ul style="list-style-type: none"> listMedrepData : variabel array objek listReportData: variabel array objek listDetailReportData : varibel array objek

```
PROCEDURE getListMedrepForShowReport()
BEGIN
    ShowReportFactory.getMedrepForShowReport().success(function(result))
    {
        listMedrepData ← result.data.listMedrepData
    }

END PROCEDURE

PROCEDURE getAllReportByType(medrepId, reportType)
BEGIN
    ShowReportFactory.getAllReportByType(medrepId, reportType).success(function(result))
    {
        $scope.listReportData ← result.data.listReportData
    }

END PROCEDURE

PROCEDURE getReportById(reportId, reportType)
BEGIN
    ShowReportFactory.getAllReportById(reportId, reportType)
    .success(function(result))
    {
        listDetailReportData ←
            result.data.listDetailReportData
    }

END PROCEDURE

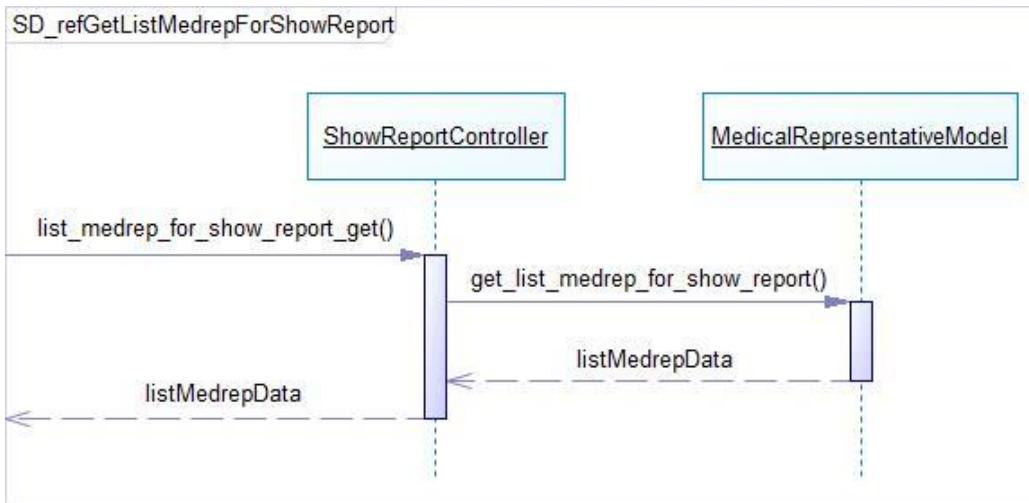
FUNCTION ShowReportFactory.getListMedrepForShowReport()
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/showreportcontroller/list_medrep_for_show_repor-
        t_get")
END FUNCTION

FUNCTION ShowReportFactory.getAllReportByType(medrepId,
    reportType)
BEGIN

    return $http.get("http://localhost/medical-rep-
        backend/showreportcontroller/all_report_by_type_get?med-
        repId=" + medrepId + "&reportType=" + reportType )

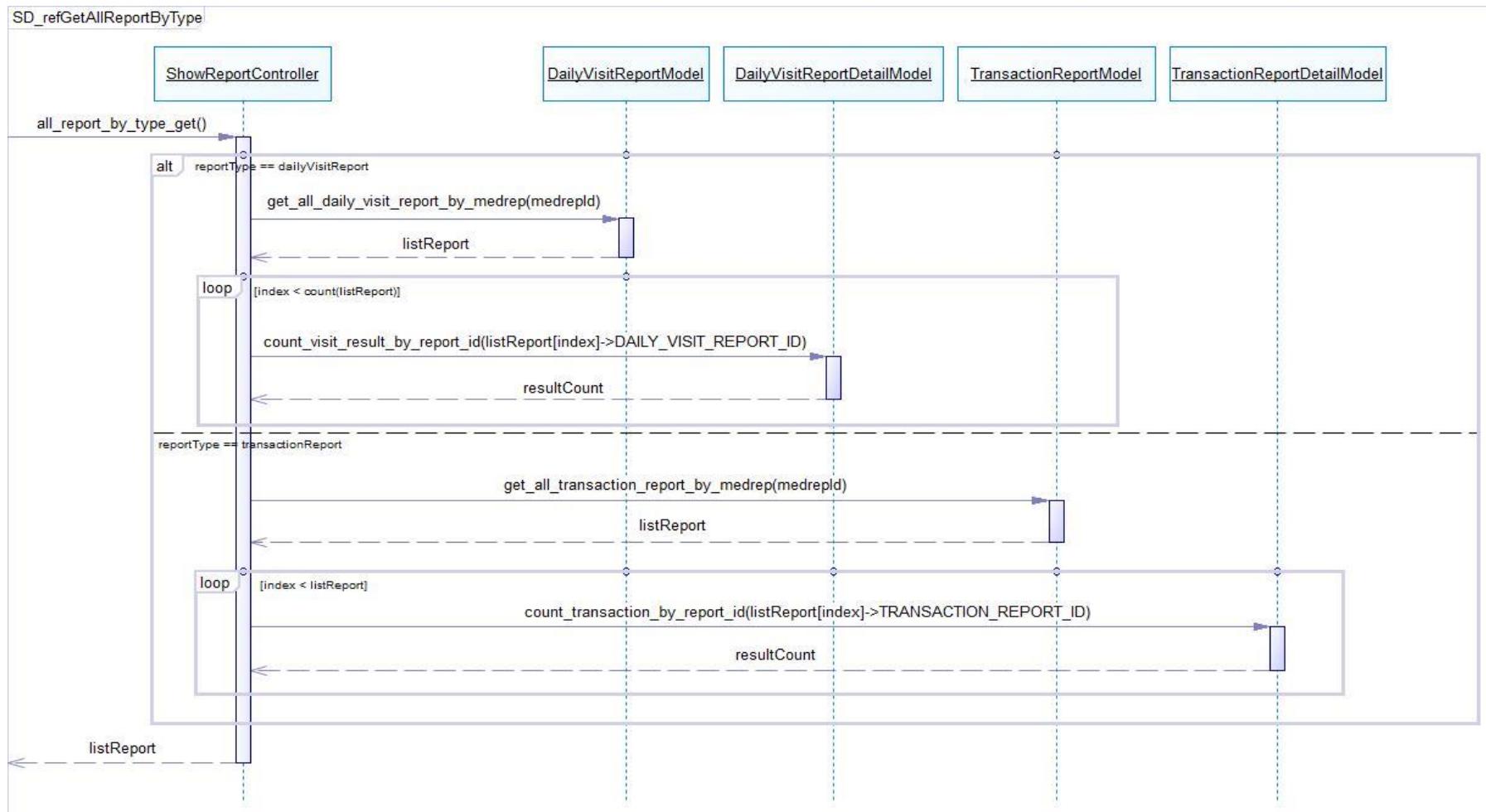
END FUNCTION

PROCEDURE ShowReportFactory.getReportById(reportId, reportType)
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/showreportcontroller/report_by_id_get?reportId=
        " + reportId + "&reportType=" + reportType")
END PROCEDURE
```

Gambar 91 SD Referensi *Get List Medrep For Show Report*

Tabel 130 Keterangan SD_refGetListMedrepForShowReport

ID Sequence diagram	SD-SR-02
Nama Sequence diagram	SD_refGetListMedrepForShowReport
Nama Class Terkait	MedicalRepresentativeController, MedicalRepresentativeModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan data <i>medical rep</i> agar dapat diakses laporan kunjungan dan laporan transaksinya.
Method yang terkait	<code>list_medrep_show_report_get()</code> , <code>get_list_medrep_for_show_report()</code>
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> listMedrepData : variabel array objek PROCEDURE list_medrep_show_report_get() BEGIN listMedrepData ← \$this->medicalrepresentativemodel-> get_list_medrep_for_show_report() END PROCEDURE FUNCTION get_list_medrep_for_show_report() BEGIN return \$this->db->query("SELECT MEDREP_ID, MEDREP_NAME FROM medicalrepresentative")->result() END FUNCTION </pre>



Gambar 92 SD Referensi Get All Report By Type

Tabel 131 Keterangan SD_refGetAllReportByType

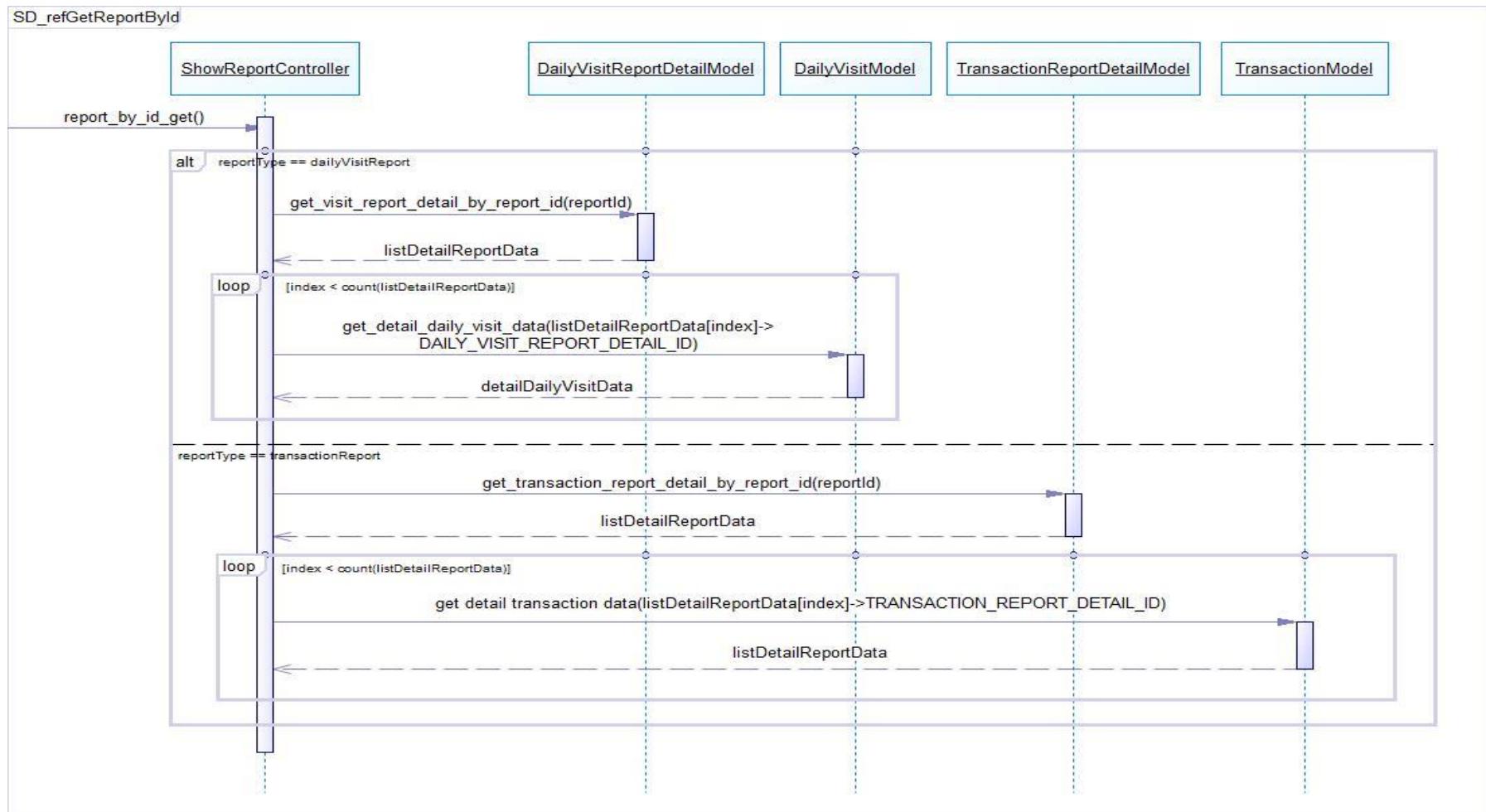
ID Sequence diagram	SD-SR-03
Nama Sequence diagram	SD_refGetAllReportByType
Nama Class Terkait	ShowReportController, DailyVisiReportModel, DailyVisirReportDetailModel, TransactionReportModel, TransactionReportDetailModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan daftar laporan kunjungan ataupun laporan transaksi berdasarkan Id dari medical rep.
Method yang terkait	all_report_by_type_get(), get_daily_visit_report_by_medrep(medrepId), get_transaction_report_by_medrep(medrepId), count_visit_report_by_id(dailyVisitReportId), count_transaction_report_by_id(transactionReportId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String reportType : variabel tunggal bertipe String listReport : variabel array objek PROCEDURE all_report_by_type_get() BEGIN medrepId ← \$this->get("medrepId") reportType ← \$this->get("reportType") IF reportType == "dailyVisitReport" THEN listReport ← \$this->dailyvisitreportmodel- >get_all_daily_visit_report_by_medrep(medrepId) FOR EACH listReport AS report report->TOTAL_VISIT_RESULT ← \$this- >dailyvisitreportdetailmodel- >count_visit_report_by_id(report- >REPORT_VISIT_ID) END FOR ELSE IF reportType == "transactionReport" THEN listReport ← \$this->transactionreportmodel- >get_all_transaction_report_by_medrep(medrepId) FOR EACH listReport AS report report->TOTAL_TRANSACTION ← \$this- >transactionreportdetailmodel- >count_transaction_report_by_id(report- >REPORT_TRANSACTION_ID) END FOR END IF END PROCEDURE </pre>

```
FUNCTION get_all_daily_visit_report_by_medrep(medrepId)
BEGIN
    return $this->db->query("SELECT * FROM visitreport
    WHERE MEDREP_ID = medrepId")->result()
END FUNCTION

FUNCTION get_all_transaction_report_by_medrep(medrepId)
BEGIN
    return $this->db->query("SELECT * FROM
    transactionreport WHERE MEDREP_ID = medrepId")-
    >result()
END FUNCTION

FUNCTION count_visit_report_by_id(dailyVisitReportId)
BEGIN
    return $this->db->query("SELECT * FROM
    detailvisitreport WHERE REPORT_VISIT_ID =
    dailyVisitReportId")->num_rows()
END FUNCTION

FUNCTION count_transaction_report_by_id(transactionReportId)
BEGIN
    return $this->db->query("SELECT * FROM
    detailtransactionreport WHERE REPORT_TRANSACTION_ID =
    transactionReportId")-> num_rows()
END FUNCTION
```



Gambar 93 SD Referensi Get Report By Id

Tabel 132 Keterangan SD_refGetReportById

ID Sequence diagram	SD-SR-04
Nama Sequence diagram	SD_refGetReportById
Nama Class Terkait	ShowReportController, DailyVisitReportDetailModel, DailyVisitModel, TransactionReportDetailModel, TransactionModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menampilkan detail laporan kunjungan ataupun laporan transaksi <i>medical rep</i> berdasarkan Id laporan.
Method yang terkait	report_by_id_get(), get_visit_report_detail_by_report_id(dailyVisitReportId), get_transaction_report_detail_by_report_id(transactionReportId), get_detail_daily_visit_data(dailyVisitReportDetailId), get_detail_transaction_data(transactionReportDetailId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> reportId : variabel tunggal bertipe String reportType : variabel tunggal bertipe String listReportDetailData : variabel array objek PROCEDURE report_by_id_get() BEGIN reportId ← \$this->get("reportId") reportType ← \$this->get("reportType") IF reportType == "dailyVisitReport" THEN listReportDetailData ← \$this- >dailyvisitreportdetailmodel->get_ visit_report_detail_by_report_id(reportId) FOR EACH listReportDetailData AS report report->DETAIL_REPORT_DATA ← \$this- >dailyvisitmodel- >get_detail_daily_visit_data(report- >REPORT_VISIT_DETAIL_ID) END FOR ELSE IF reportType == "transactionReport" THEN listReportDetailData ← \$this- >transactionreportdetailmodel-> get_transaction_report_detail_by_report_id(reportId) FOR EACH listReportDetailData AS report report->DETAIL_REPORT_DATA ← \$this- >transactionmodel->get_detail_transaction(report- >REPORT_TRANSACTION_DETAIL_ID) END FOR END IF </pre>

```

END PROCEDURE

FUNCTION get_visit_report_detail_by_report_id
(dailyVisitReportId)
BEGIN
    return $this->db->query("SELECT * FROM
detailvisitreport WHERE REPORT_VISIT_ID =
dailyVisitReportId")->result()
END FUNCTION

FUNCTION get_transaction_report_detail_by_report_id
(transactionReportId)
BEGIN
    return $this->db->query("SELECT * FROM
detailtransactionreport WHERE REPORT_TRANSACTION_ID =
transactionReportId")->result()
END FUNCTION

FUNCTION get_detail_daily_visit_data(dailyVisitReportDetailId)
BEGIN
    return $this->db->query("SELECT dv.*, m.MEDREP_ID,
m.MEDREP_NAME, d.DOCTOR_NAME FROM dailyvisit dv,
medicalrepresentative m, doctor d WHERE dv.MEDREP_ID =
m.MEDREP_ID AND dv.DOCTOR_ID = d.DOCTOR_ID AND
dv.DAILY_VISIT_ID = dailyVisitReportDetailId")-
>result()
END FUNCTION

FUNCTION get_detail_transaction_data(transactionReportDetailId)
BEGIN
    query ← $this->db->query("SELECT t.*, d.DOCTOR_NAME,
m.MEDREP_ID, m.MEDREP_NAME FROM transaction t,
doctor d, medicalrepresentative m WHERE t.DOCTOR_ID =
d.DOCTOR_ID AND t.MEDREP_ID = m.MEDREP_ID AND
t.TRANSACTION_ID = transactionReportDetailId")-
>result()

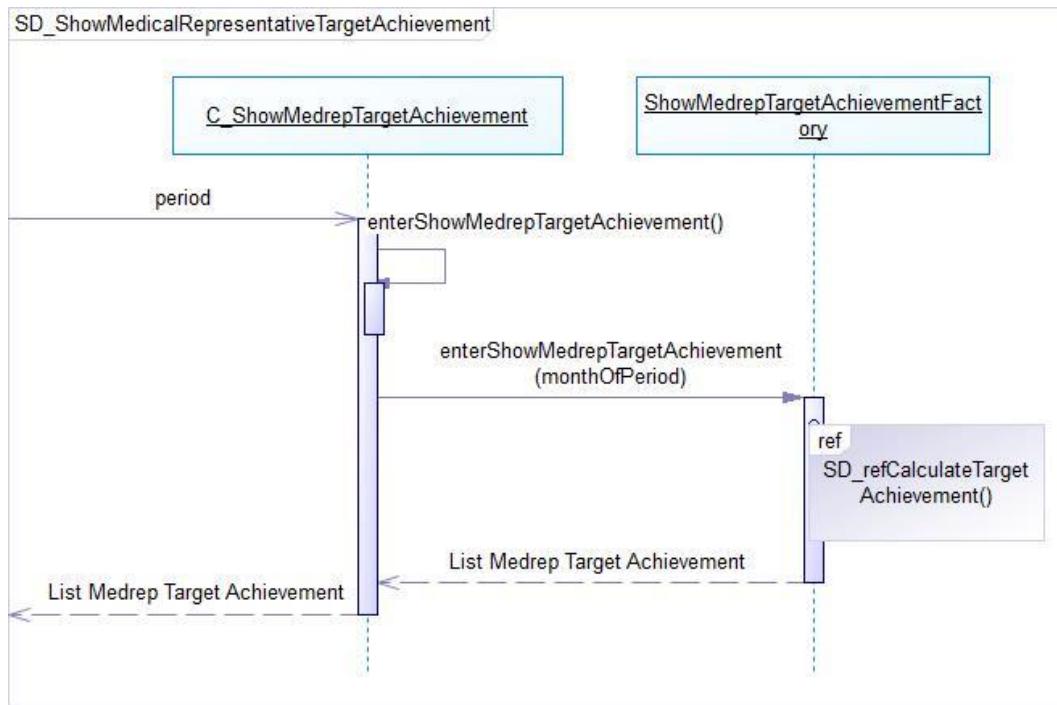
    FOR EACH query AS transactionData
        transactionData->DETAIL_TRANSACTION ← $this-
>get_transaction_detail(transactionData-
>TRANSACTION_ID)

    END FOR
    return query
END FUNCTION

FUNCTION get_transaction_detail(transactionId)
BEGIN
    return $this->db->query("SELECT
dt.PRESCRIPTION_TARGET, p.PRODUCT_NAME FROM
detailtransaction WHERE dt.PRODUCT_ID = p.PRODUCT_ID
AND dt.TRANSACTION_ID = transactionId")
END FUNCTION

```

3. SD – Show Medical Representative's Target Achievement



Gambar 94 SD Show Medical Representative's Target Achievement

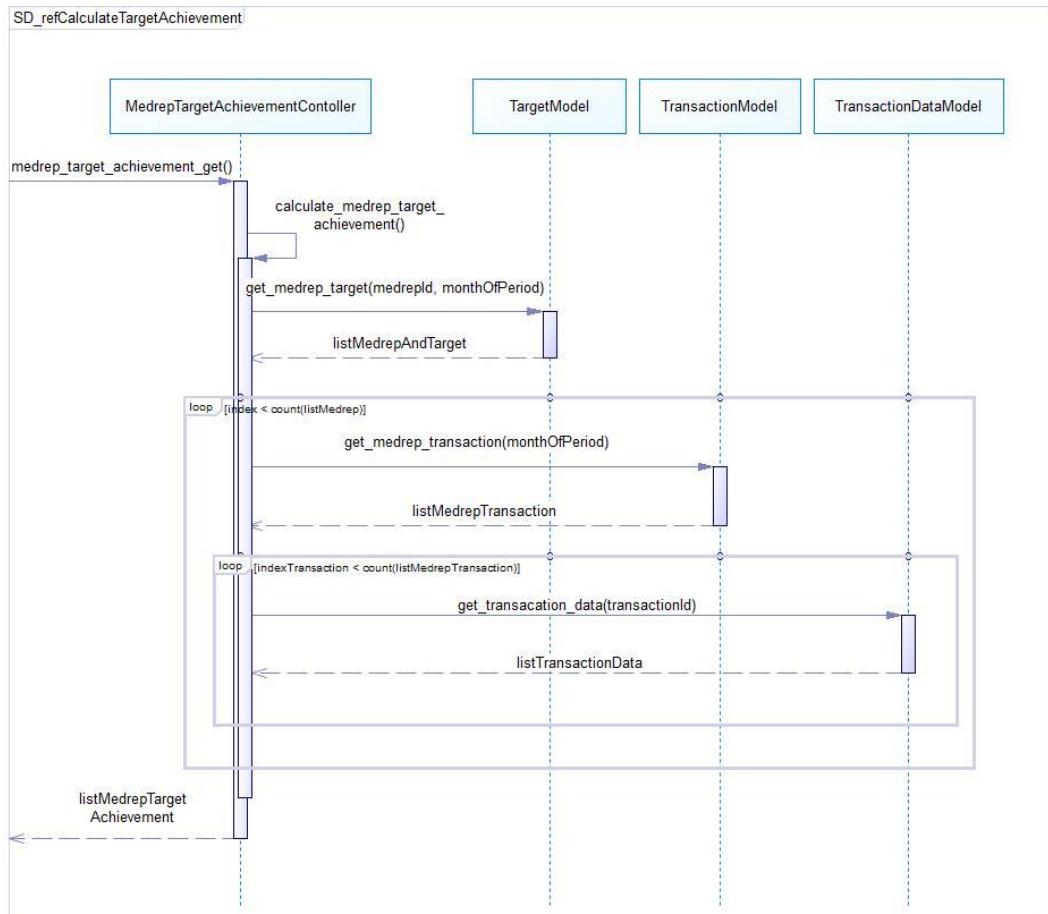
Tabel 133 Keterangan SD_ShowMedicalRepresentativeTargetAchievement

ID Sequence diagram	SD-CT-01
Nama Sequence diagram	SD_ShowMedicalRepresentativeTargetAchievement
Nama Class Terkait	C_ShowMedrepTargetAchievement, ShowMedrepTargetAchievementFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menghitung pencapaian target dari medical rep pada bagian web frontend.
Method yang terkait	enterShowMedrepTargetAchievement(monthOfPeriod)
Sequence diagram yang terkait	SD_refCalculateTargetAchievement
Algoritma	<p>Kamus data</p> <pre> PROCEDURE enterShowMedrepTargetAchievement () BEGIN ShowMedrepTargetAchievementFactory. enterShowMedrepTargetAchievement (\$scope.monthOfPeriod) .success(function(result)) { list_medrep_target_achievement ← result.data.list_medrep_target_achievement </pre>

```

    }
END PROCEDURE
FUNCTION ShowMedrepTargetAchievementFactory.
enterShowMedrepTargetAchievement(monthOfPeriod)
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/medreptargetachievementcontroller/medrep_target_
        _achievement_get?monthOfPeriod="+ monthOfPeriod)
END FUNCTION

```

Gambar 95 SD Referensi *Calculate Target Achievement*

Tabel 134 Keterangan SD_refCalculateTargetAchievement

ID Sequence diagram	SD-CT-02
Nama Sequence diagram	SD_refCalculateTargetAchievement
Nama Class Terkait	MedrepTargetAchievementController, TargetModel, TransactionModel, TransactionDataModel
Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses menghitung pencapaian target dari <i>medical rep</i> pada bagian <i>web backend</i> .
Method yang terkait	medrep_targett_achievement_get(), calculate_medrep_target_achievement(), get_medrep_target(monthOfPeriod), get_medrep_transaction(monthOfPeriod), get_transaction_data(transactionId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> medrepId : variabel tunggal bertipe String monthOfPeriod : varibel tunggal bertipe String PROCEDURE medrep_target_achievement_get() BEGIN \$this->calculate_medrep_target_achievement() END PROCEDURE FUNCTION get_medrep_target(monthOfPeriod) BEGIN return \$this->db->query("SELECT * FROM target WHERE month(START_DATE_PERIOD) = monthOfPeriod")->result() END FUNCTION FUNCTION get_medrep_transaction(medrepId, monthOfPeriod) BEGIN return \$this->db->query("SELECT * FROM transaction WHERE MEDREP_ID = medrepId AND month(TRANSACTION_DATE) = monthOfPeriod")->result() END FUNCTION FUNCTION get_transaction_data(transactionId) BEGIN prescriptionTarget ← \$this->db->query("SELECT SUM(PRESCRIPTION_TARGET) as total_prescription_target FROM transactiondata WHERE TRANSACTION_ID = transactionId")->row() return prescriptionTarget->total_prescription_target END FUNCTION </pre>

```
PROCEDURE calculate_medrep_target_achievement()
BEGIN
    monthOfPeriod ← $this->get("monthOfPeriod")
    listMedrep ← $this->targetmodel-
    >get_medrep_target(monthOfPeriod)

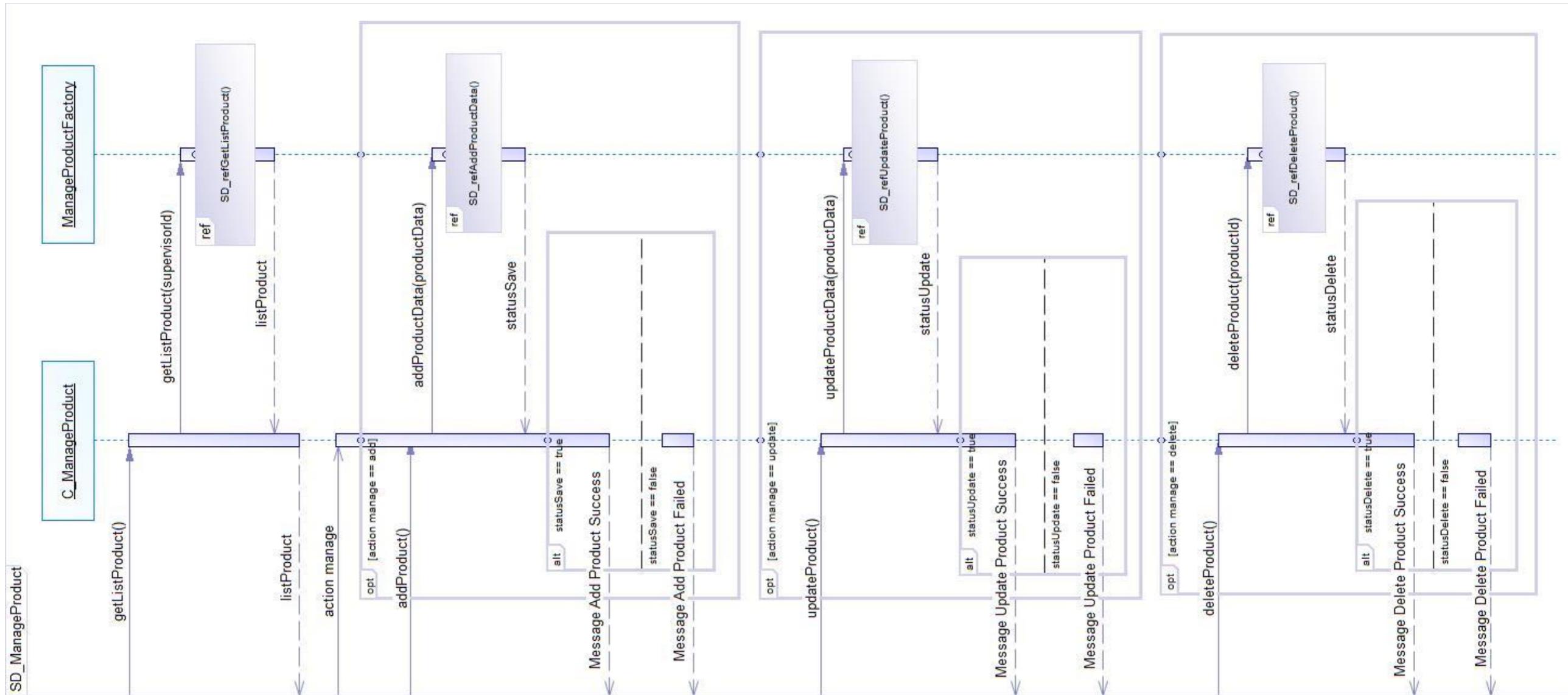
    FOR EACH listMedrep AS medrep DO
        listMedrepTransaction ← $this->transactionmodel-
        >get_medrep_transaction(medrep->MEDREP_ID,
        monthOfPeriod)

        totalTransaction ← 0
        FOR EACH listMedrepTransaction AS transaction DO
            transactionData ← $this-
            >transactiondatamodel-
            >get_transaction_data(transaction-
            >TRANSACTION_ID)

            totalTransaction ← totalTransaction +
            transactionData
        END FOR

        medrep->TOTAL_TRANSACTION ← totalTransaction
        medrep->TARGET_ACHIEVEMENT ← (totalTransaction /
        medrep->TARGET)
    END FOR
END FUNCTION
```

4. SD – Manage Product



Gambar 96 SD Manage Product

Tabel 135 Keterangan SD_ManageProduct

ID Sequence diagram	SD-MP-01
Nama Sequence diagram	SD_ManageProduct
Nama Class Terkait	C_ManageProduct, ManageProductFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengelola product yang akan ditugaskan untuk <i>medical rep</i> .
Method yang terkait	getListProduct(supervisorId), addProduct(), addProductData(productData), updateProduct(), updateProductData(productData), deleteProduct(), deleteProduct(productId).
Sequence diagram yang terkait	SD_refGetListProduct, SD_refAddProductData, SD_refUpdateProduct, SD_refDeleteProduct
Algoritma	<pre> Kamus data listProduct : variabel array objek PROCEDURE getListProduct() BEGIN manageProductFactory.getListProduct(\$scope.supervisorId) .then(function(result){ listProduct ← result.data.listProduct }) END PROCEDURE PROCEDURE addProduct() BEGIN productData ← array('productName' ← \$scope.productName, 'productType' ← \$scope.productType, 'productIngredients' ← \$scope.productIngredients, 'productIndication' ← \$scope.productIndication, 'productInstruction' ← \$scope.productInstruction, 'productContraindication' ← \$scope.productContraindication, 'productUses' ← \$scope.productUses, 'productWarning' ← \$scope.productWarning, 'productPrice' ← \$scope.productPrice, 'supervisorId' ← \$scope.supervisorId) manageProductFactory.addProductData(productData).then(function(result){ statusSave ← result.statusSave IF statusSave == TRUE THEN WRITE (layar) "save success" ELSE WRITE (layar) "save failed" END IF }) END PROCEDURE </pre>

```

PROCEDURE updateProduct()
BEGIN
    productData ← array(
        'productId' ← $scope.productId,
        'productName' ← $scope.productName,
        'productType' ← $scope.productType,
        'productIngredients' ← $scope.productIngredients,
        'productIndication' ← $scope.productIndication,
        'productInstruction' ← $scope.productInstruction,
        'productContraindication' ←
            $scope.productContraindication,
        'productUses' ← $scope.productUses,
        'productWarning' ← $scope.productWarning,
        'productPrice' ← $scope.productPrice,
        'supervisorId' ← $scope.supervisorId
    )
    manageProductFactory.updateProductData(productData).then(function(result) {
        statusUpdate ← result.data.statusUpdate
        IF statusUpdate == TRUE THEN
            WRITE (layar) "update success"
        ELSE
            WRITE (layar) "update failed"
        END IF
    })
END PROCEDURE

PROCEDURE deleteProduct()
BEGIN
    productId ← $scope.productId
    manageProductFactory.deleteProduct(productId).then(function(result) {
        statusDelete ← result.data.statusDelete
        IF statusDelete == TRUE THEN
            WRITE (layar) "delete success"
        ELSE
            WRITE (layar) "delete failed"
        END IF
    })
END PROCEDURE

FUNCTION manageProductFactory.getListProduct(supervisorId)
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/manageproduct/list_product?supervisorId=" +
        supervisorId)
END FUNCTION

FUNCTION manageProductFactory.addProductData(productData)
BEGIN
    return $http.post("http://localhost/medical-rep-
        backend/manageproductcontroller/add_product",
        productData)
END PROCEDURE

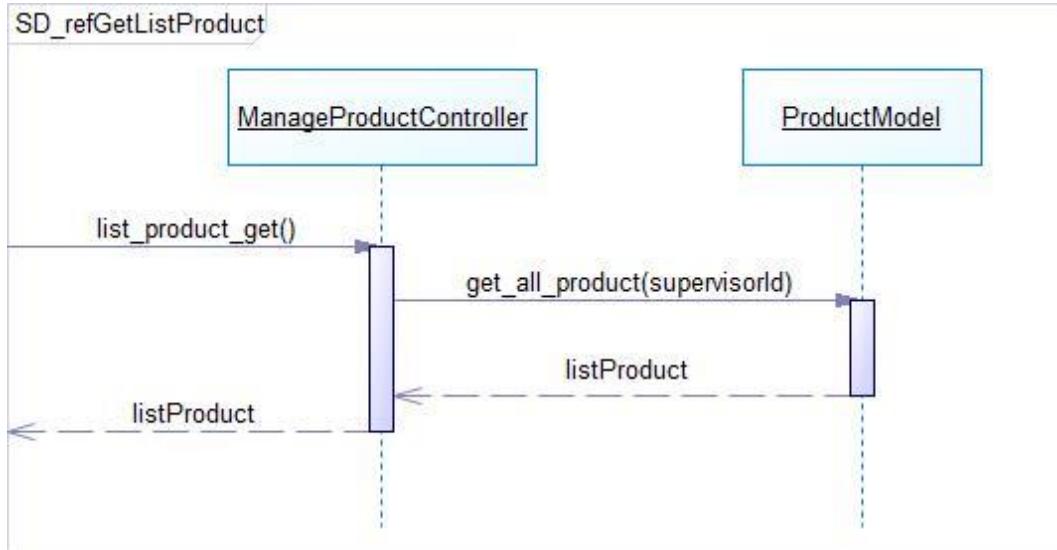
```

```

FUNCTION manageProductFactory.updateProductData (productData)
BEGIN
    return $http.post("http://localhost/medical-rep-
        backend/manageproductcontroller/update_product",
        productData)
END PROCEDURE

FUNCTION manageProductFactory.deleteProduct (productId)
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/manageproductcontroller/delete_product?productI
        d="+ productId)
END PROCEDURE

```

Gambar 97 SD Referensi *Get List Product*

Tabel 136 Keterangan SD_refGetListProduct

ID Sequence diagram	SD-MP-02
Nama Sequence diagram	SD_refGetListProduct
Nama Class Terkait	ManageProductController, ProductModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data produk dari database.
Method yang terkait	list_product_get(), get_all_product(supervisorId)
Sequence diagram yang terkait	-

Algoritma

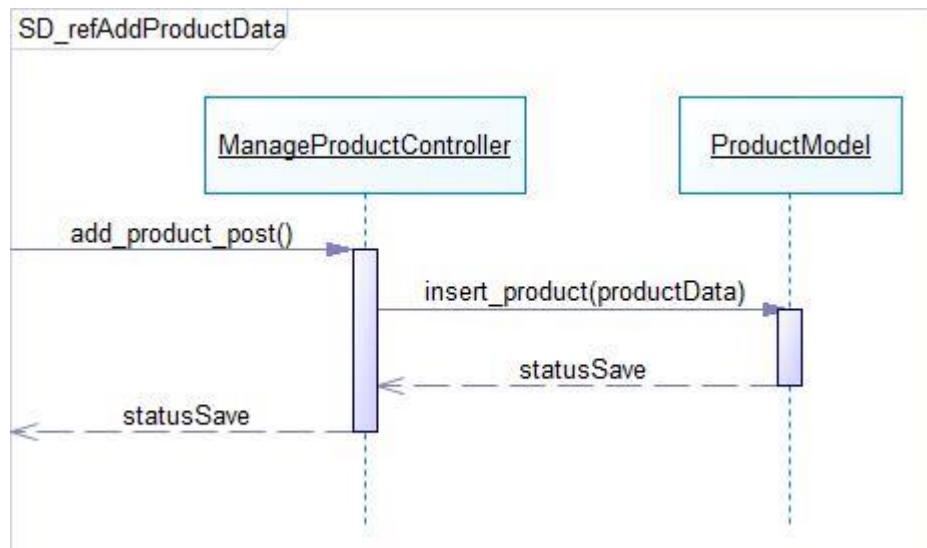
```

Kamus data
    listProduct : variabel array objek

PROCEDURE list_product_get()
BEGIN
    supervisorId ← $this->get('supervisorId')
    listProduct ← $this->productmodel-
        >get_all_product(supervisorId)
END PROCEDURE

FUNCTION get_all_product(supervisorId)
BEGIN
    return $this->db->query("SELECT * FROM product WHERE
        DELETED = 0 AND SUPERVISOR_ID = supervisorId")-
        >result()
END PROCEDURE

```



Gambar 98 SD Referensi Add Product Data

Tabel 137 Keterangan SD_refAddProductData

ID Sequence diagram	SD-MP-03
Nama Sequence diagram	SD_refAddProductData
Nama Class Terkait	ManageProductController, ProductModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan data produk ke database.

<i>Method</i> yang terkait	add_product_post(), insert_product(productData)
<i>Sequence diagram</i> yang terkait	-

Algoritma

Kamus data

```

productData : variabel array objek
statusSave : variabel tunggal bertipe boolean

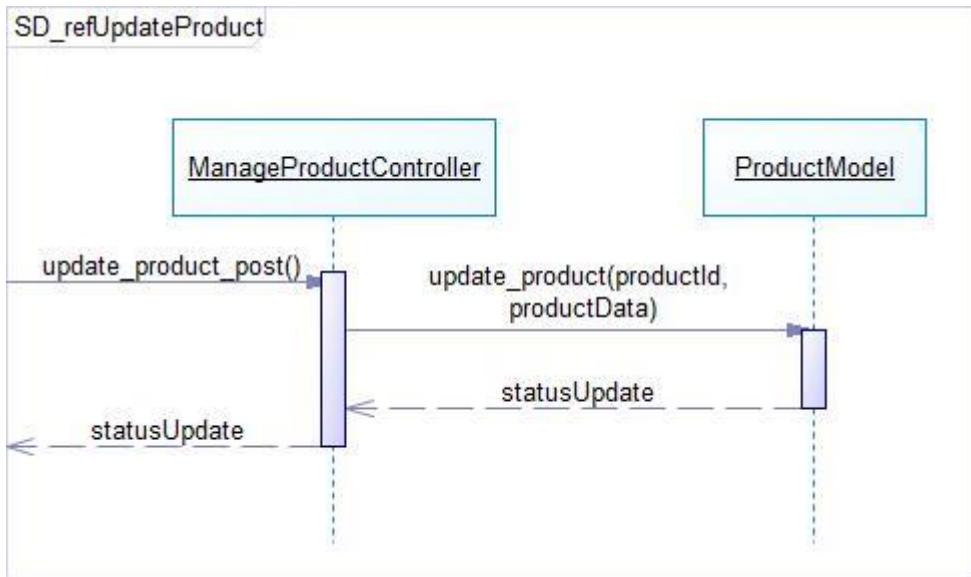
PROCEDURE add_product_post()
BEGIN
    productId ← $this->get_new_id()
    productData ← array(
        'PRODUCT_ID' ← productId,
        'PRODUCT_NAME' ← $this->post('productName'),
        'PRODUCT_TYPE' ← $this->post('productType'),
        'PRODUCT_INGREDIENTS' ← $this-
            >post('productIngredients'),
        'PRODUCT_INDICATION' ← $this-
            >post('productIndication'),
        'PRODUCT_INSTRUCTION' ← $this-
            >post('productInstruction'),
        'PRODUCT CONTRAINDICATION' ← $this-
            >post('productContraindication'),
        'PRODUCTUSES' ← $this->post('productUses'),
        'PRODUCT_WARNING' ← $this->post('productWarning'),
        'PRODUCT_PRICE' ← $this->post('productPrice'),
        'SUPERVISOR_ID' ← $this->post('supervisorId')
    )

    statusSave ← $this->productmodel-
        >save_product(productData)
END PROCEDURE

FUNCTION insert_product(productData) -> boolean
BEGIN
    query ← $this->db->insert('product', productData)

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

```

Gambar 99 SD Referensi *Update Product*

Tabel 138 Keterangan SD_refUpdateProduct

ID Sequence diagram	SD-MP-04
Nama Sequence diagram	SD_refUpdateProduct
Nama Class Terkait	ManageProductController, ProductModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengubah data produk ke database.
Method yang terkait	update_product_post(), update_product(productId, productData)
Sequence diagram yang terkait	-

Algoritma

Kamus data

```

productData : variabel array objek
statusUpdate : variabel tunggal bertipe boolean
  
```

PROCEDURE update_product_post()

BEGIN

```

        productId ← $this->post('productId')
        productData ← array(
            'PRODUCT_ID' ← productId,
            'PRODUCT_NAME' ← $this->post('productName'),
            'PRODUCT_TYPE' ← $this->post('productType'),
            'PRODUCT_INGREDIENTS' ← $this-
                >post('productIngredients'),
            'PRODUCT_INDICATION' ← $this-
                >post('productIndication'),
            'PRODUCT_INSTRUCTION' ← $this-
                >post('productInstruction'),
  
```

```

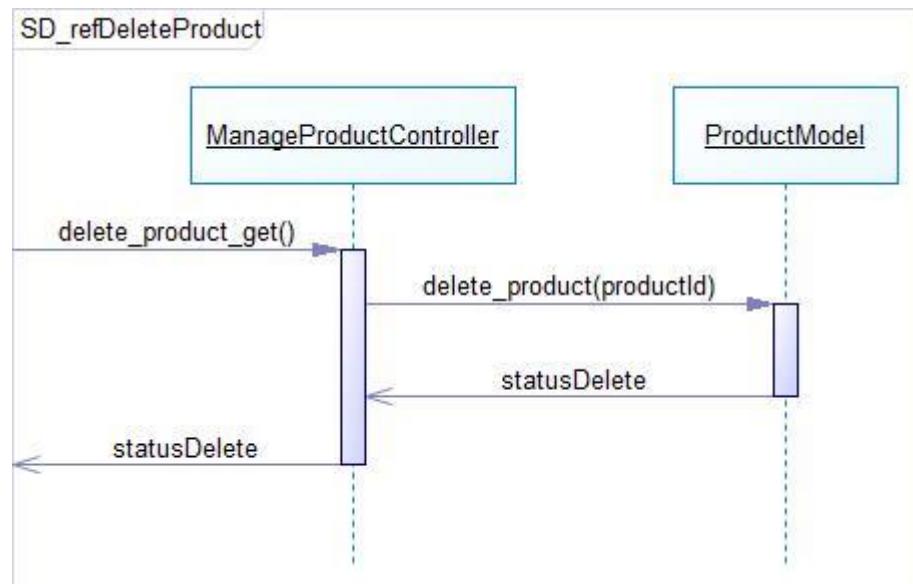
'PRODUCT CONTRAINDICATION' ← $this-
    >post('productContraindication'),
'PRODUCT USES' ← $this->post('productUses'),
'PRODUCT WARNING' ← $this->post('productWarning'),
'PRODUCT PRICE' ← $this->post('productPrice'),
'SUPERVISOR ID' ← $this->post('supervisorId')
)

statusUpdate ← $this->productmodel-
    >update_product(productId, productData)
END PROCEDURE

FUNCTION update_product(productId, productData) -> boolean
BEGIN
    $this->db->where('PRODUCT_ID', productId)
    query ← $this->db->update('product', productData)

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

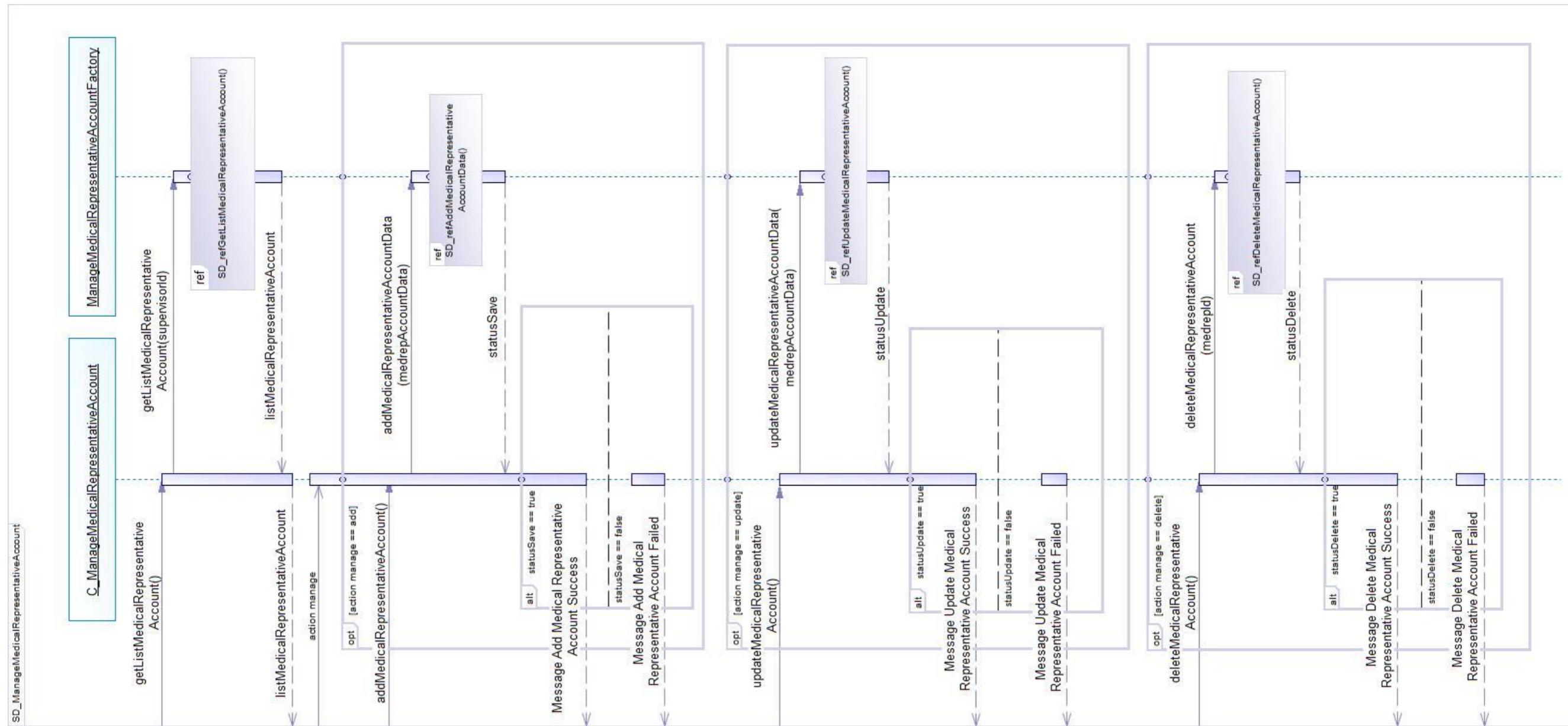
```

Gambar 100 SD Referensi *Delete Product*

Tabel 139 Keterangan SD_refDeleteProduct

ID Sequence diagram	SD-MP-05
Nama Sequence diagram	SD_refDeleteProduct
Nama Class Terkait	ManageProductController, ProductModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menghapus data produk ke database.
Method yang terkait	delete_product_get(), delete_product(productId)
Sequence diagram yang terkait	-
Algoritma	
<p>Kamus data</p> <pre> productId : variabel tunggal bertipe string statusDelete : variabel tunggal bertipe boolean </pre>	
PROCEDURE delete_product_get() <u>BEGIN</u> productId ← \$this->get('productId') statusDelete ← \$this->productmodel- >delete_product(productId) <u>END PROCEDURE</u>	
FUNCTION delete_product(productId) -> boolean <u>BEGIN</u> query ← \$this->db->query("UPDATE product SET DELETED = 1 WHERE PRODUCT_ID = 'productId'") <u>IF</u> query == TRUE <u>THEN</u> <u>return</u> TRUE <u>ELSE</u> <u>return</u> FALSE <u>END IF</u> <u>END FUNCTION</u>	

5. SD – Manage Medical Representative Account



Gambar 101 SD Manage Medical Representative Account

Tabel 140 Keterangan SD_ManageMedicalRepresentativeAccount

ID Sequence diagram	SD-MMR-01
Nama Sequence diagram	SD_ManageMedicalRepresentativeAccount
Nama Class Terkait	C_ManageMedicalRepresentativeAccount, ManageMedicalRepresentativeAccountFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengelola akun medical rep.
Method yang terkait	getListMedicalRepresentativeAccount(), getListMedicalRepresentativeAccount(supervisorId), addMedicalRepresentativeAccount(), addMedicalRepresentativeAccountData(medrepAccountData), updateMedicalRepresentativeAccount(), updateMedicalRepresentativeAccountData(medrepAccountData), deleteMedicalRepresentativeAccount(), deleteMedicalRepresentativeAccount(medrepId).
Sequence diagram yang terkait	SD_refGetListMedicalRepresentativeAccount, SD_refAddMedicalRepresentativeAccountData, SD_refUpdateMedicalRepresentativeAccount, SD_refDeleteMedicalRepresentative
Algoritma	<p>Kamus data</p> <pre> listMedicalRepresentativeAccount : variabel array objek PROCEDURE getListMedicalRepresentativeAccount() BEGIN manageMedicalRepresentativeAccountFactory.getListMedica lRepresentativeAccount(\$scope.supervisorId).then(function(result) { listMedicalRepresentativeAccount ← result.data.listMedicalRepresentativeAccount }) END PROCEDURE PROCEDURE addMedicalRepresentativeAccount() BEGIN medrepAccountData ← array('medrepId' ← \$scope.medrepId, 'medrepName' ← \$scope.medrepName, 'medrepContact' ← \$scope.medrepContact, 'password' ← \$scope.password, 'supervisorId' ← \$scope.supervisorId) manageMedicalRepresentativeAccountFactory.addMedicalRep resentativeAccountData(medrepAccountData).then(function (result) { statusSave ← result.data.statusSave IF statusSave == TRUE THEN WRITE (layar) "save success" ELSE WRITE (layar) "save failed" END IF }) </pre>

```

    }
END PROCEDURE

PROCEDURE updateMedicalRepresentativeAccount()
BEGIN
    medrepAccountData ← array(
        'medrepId' ← $scope.medrepId,
        'medrepName' ← $scope.medrepName,
        'medrepContact' ← $scope.medrepContact,
        'password' ← $scope.password,
        'supervisorId' ← $scope.supervisorId
    )

    manageMedicalRepresentativeAccountFactory.updateMedical
    RepresentativeAccountData(medrepAccountData).then(function(result){
        statusUpdate ← result.data.statusUpdate
        IF statusUpdate == TRUE THEN
            WRITE (layar) "update success"
        ELSE
            WRITE (layar) "update failed"
        END IF
    })
END PROCEDURE

PROCEDURE deleteMedicalRepresentativeAccount()
BEGIN
    medrepId ← $scope.medrepId
    manageMedicalRepresentativeAccountFactory.deleteMedical
    RepresentativeAcccount(medrepId).then(function(result){
        statusDelete ← result.data.statusDelete
        IF statusDelete == TRUE THEN
            WRITE (layar) "delete success"
        ELSE
            WRITE (layar) "delete failed"
        END IF
    })
END PROCEDURE

FUNCTION
manageMedicalRepresentativeAccountFactory.getListMedicalRepresen
tativeAccount(supervisorId)
BEGIN
    return $http.get("http://localhost/medical-rep-
        backend/managemedicalrepresentativeaccount/list_medical
        _representative_account?supervisorId=" + supervisorId)
END FUNCTION

FUNCTION
manageMedicalRepresentativeAccountFactory.addMedicalRepresentati
veAccountData(medrepAccountData)
BEGIN
    return $http.post("http://localhost/medical-rep-
        backend/ managemedicalrepresentativeaccount
        /add_medical_representative_account",
        medrepAccountData)
END PROCEDURE

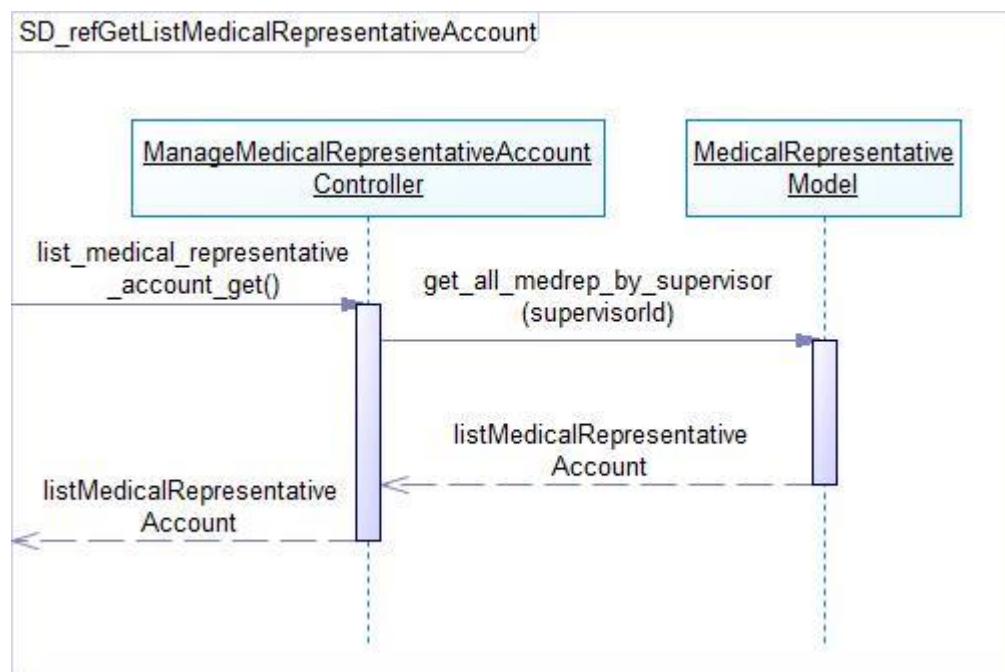
```

```

FUNCTION
manageMedicalRepresentativeAccountFactory.updateMedicalRepresentativeAccountData (medrepAccountData)
BEGIN
    return $http.post("http://localhost/medical-rep-
backend/ managemedicalrepresentativeaccount
/update_medical_representative_account",
medrepAccountData)
END PROCEDURE

FUNCTION
manageMedicalRepresentativeAccountFactory.deleteMedicalRepresentativeAccount (medrepId)
BEGIN
    return $http.get(http://localhost/medical-rep-backend/
managemedicalrepresentativeaccount
/delete_medical_representative_account?medrepId="+
medrepId)
END PROCEDURE

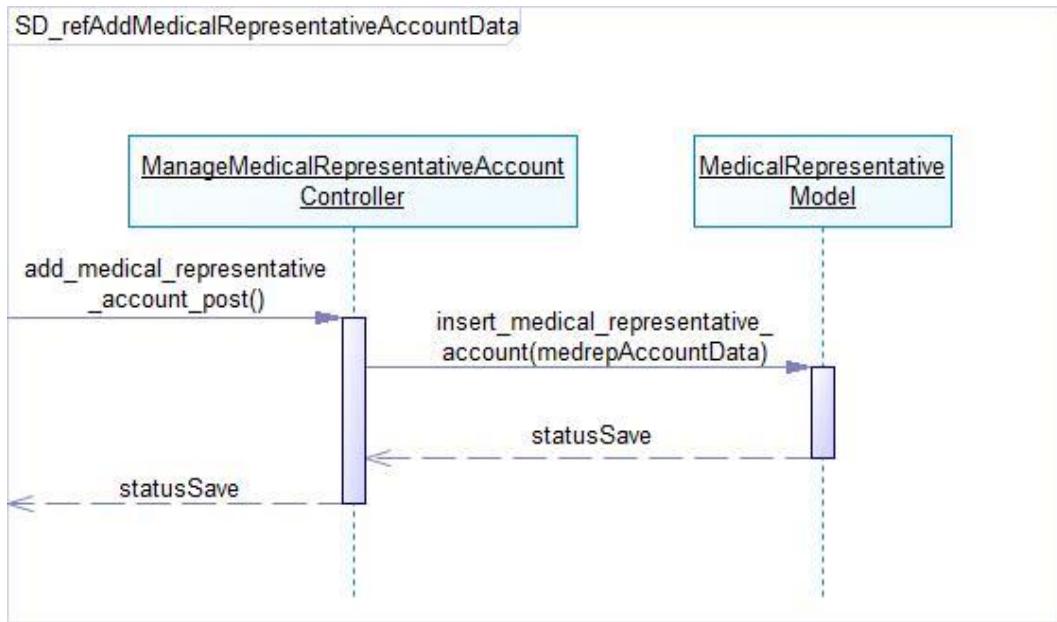
```



Gambar 102 SD Referensi *Get List Medical Representative Account*

Tabel 141 Keterangan SD_refGetListMedicalRepresentativeAccount

ID Sequence diagram	SD-MMR-02
Nama Sequence diagram	SD_refGetListMedicalRepresentativeAccount
Nama Class Terkait	ManageMedicalRepresentativeAccountController, MedicalRepresentativeModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data akun <i>medical rep</i> dari <i>database</i> .
Method yang terkait	list_medicalRepresentativeAccount_get(), get_all_medrep_by_supervisor(supervisorId)
Sequence diagram yang terkait	-
Algoritma	
<pre> Kamus data listMedicalRepresentativeAccount : variabel array objek PROCEDURE list_medicalRepresentativeAccount_get() BEGIN supervisorId ← \$this->get('supervisorId') listMedicalRepresentativeAccount ← \$this- >medicalrepresentativemodel- >get_all_medrep_by_supervisor(supervisorId) END PROCEDURE FUNCTION get_all_medrep_by_supervisor(supervisorId) BEGIN return \$this->db->query("SELECT * FROM medicalrepresentative WHERE DELETED = 0 AND SUPERVISOR_ID = supervisorId")->result() END PROCEDURE </pre>	



Gambar 103 SD Referensi Add Medical Representatice Account Data

Tabel 142 Keterangan SD_refAddMedicalRepresentativeAccountData

ID Sequence diagram	SD-MMR-03
Nama Sequence diagram	SD_refAddMedicalRepresentativeAccountData
Nama Class Terkait	ManageMedicalRepresentativeAccountController, MedicalRepresentativeModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan data akun <i>medical rep</i> ke <i>database</i> .
Method yang terkait	add_medical_representative_account_post(), insert_medical_representative_account(medrepAccountData)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> medrepAccountData : variabel array objek statusSave : variabel tunggal bertipe boolean </pre> <p>PROCEDURE add_medical_representative_account_post()</p> <p><u>BEGIN</u></p> <pre> medrepAccountData ← array('MEDREP_ID' ← \$this->post('medrepId'), 'MEDREP_NAME' ← \$this->post('medrepName'), 'MEDREP_CONTACT' ← \$this->post('medrepContact'), 'PASSWORD' ← \$this->post('password'), 'SUPERVISOR_ID' ← \$this->post('supervisorId')) </pre>

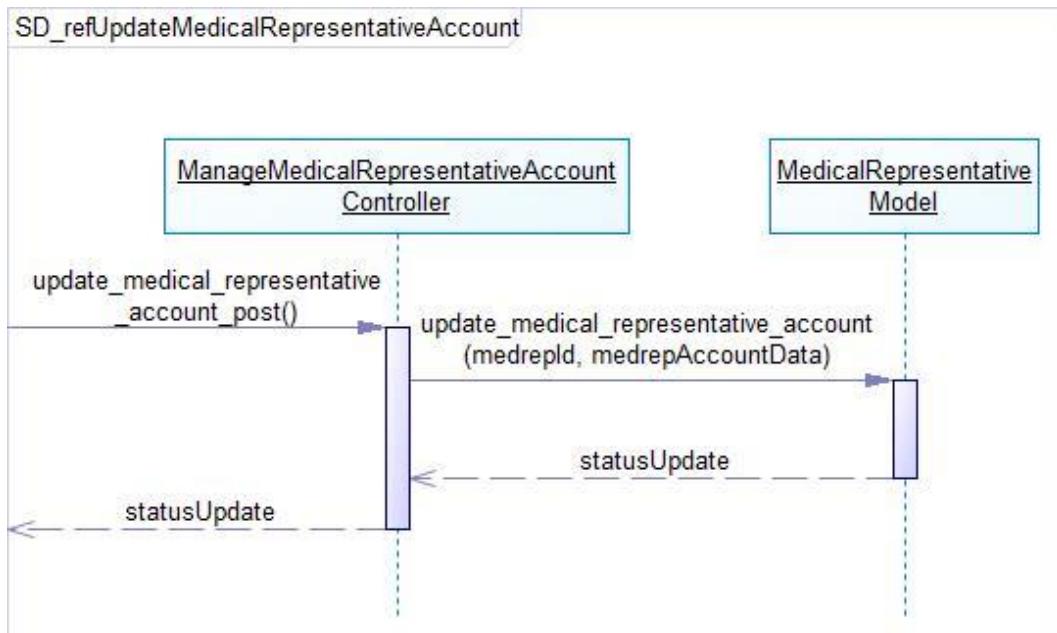
```

statusSave ← $this->medicalrepresentativemodel-
    >save_medicalRepresentative_account(medrepAccountD
ata)
END PROCEDURE

FUNCTION
insert_medicalRepresentative_account(medrepAccountData) ->
boolean
BEGIN
    query ← $this->db->insert('medicalrepresentative,
    medrepAccountData)

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

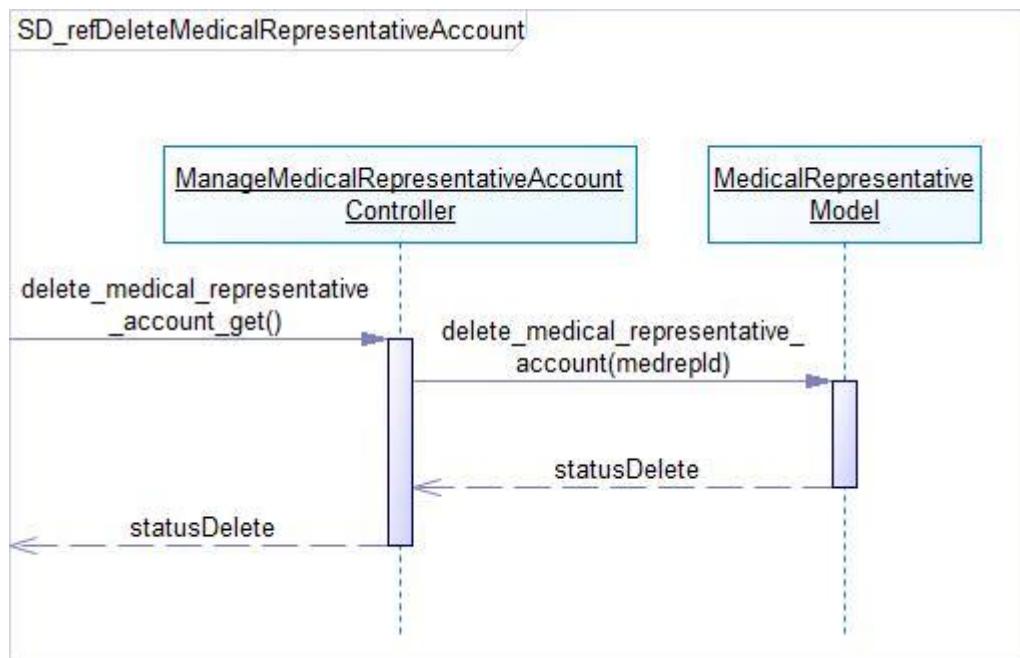
```

Gambar 104 SD Referensi *Update Medical Representative Account*

Tabel 143 Keterangan SD_refUpdateMedicalRepresentativeAccount

ID Sequence diagram	SD-MMR-04
Nama Sequence diagram	SD_refUpdateMedicalRepresentativeAccount
Nama Class Terkait	ManageMedicalRepresentativeAccountController, MedicalRepresentativeModel

Deskripsi	<i>Sequence</i> ini digunakan untuk menjelaskan proses mengubah data akun <i>medical rep</i> ke <i>database</i> .
<i>Method</i> yang terkait	<code>update_medicalRepresentativeAccount_post()</code> , <code>update_medicalRepresentativeAccount(medrepId, medrepAccountData)</code>
<i>Sequence diagram</i> yang terkait	-
Algoritma	
<pre> Kamus data medrepAccountData : variabel array objek statusUpdate : variabel tunggal bertipe boolean PROCEDURE update_medicalRepresentativeAccount_post() BEGIN medrepId ← \$this->post('medrepId') medrepAccountData ← array('MEDREP_NAME' ← \$this->post('medrepName'), 'MEDREP_CONTACT' ← \$this->post('medrepContact'), 'PASSWORD' ← \$this->post('password'), 'SUPERVISOR_ID' ← \$this->post('supervisorId')) statusUpdate ← \$this->medicalRepresentativeModel- >update_medicalRepresentativeAccount(medrepId, medrepAccountData) END PROCEDURE FUNCTION update_medicalRepresentativeAccount(medrepId, medrepAccountData) -> boolean BEGIN \$this->db->where('MEDREP_ID', medrepId) query ← \$this->db->update('medicalRepresentative', medrepAccountData) IF query == TRUE THEN return TRUE ELSE return FALSE END IF END FUNCTION </pre>	

Gambar 105 SD Referensi *Delete Medical Representative Account*

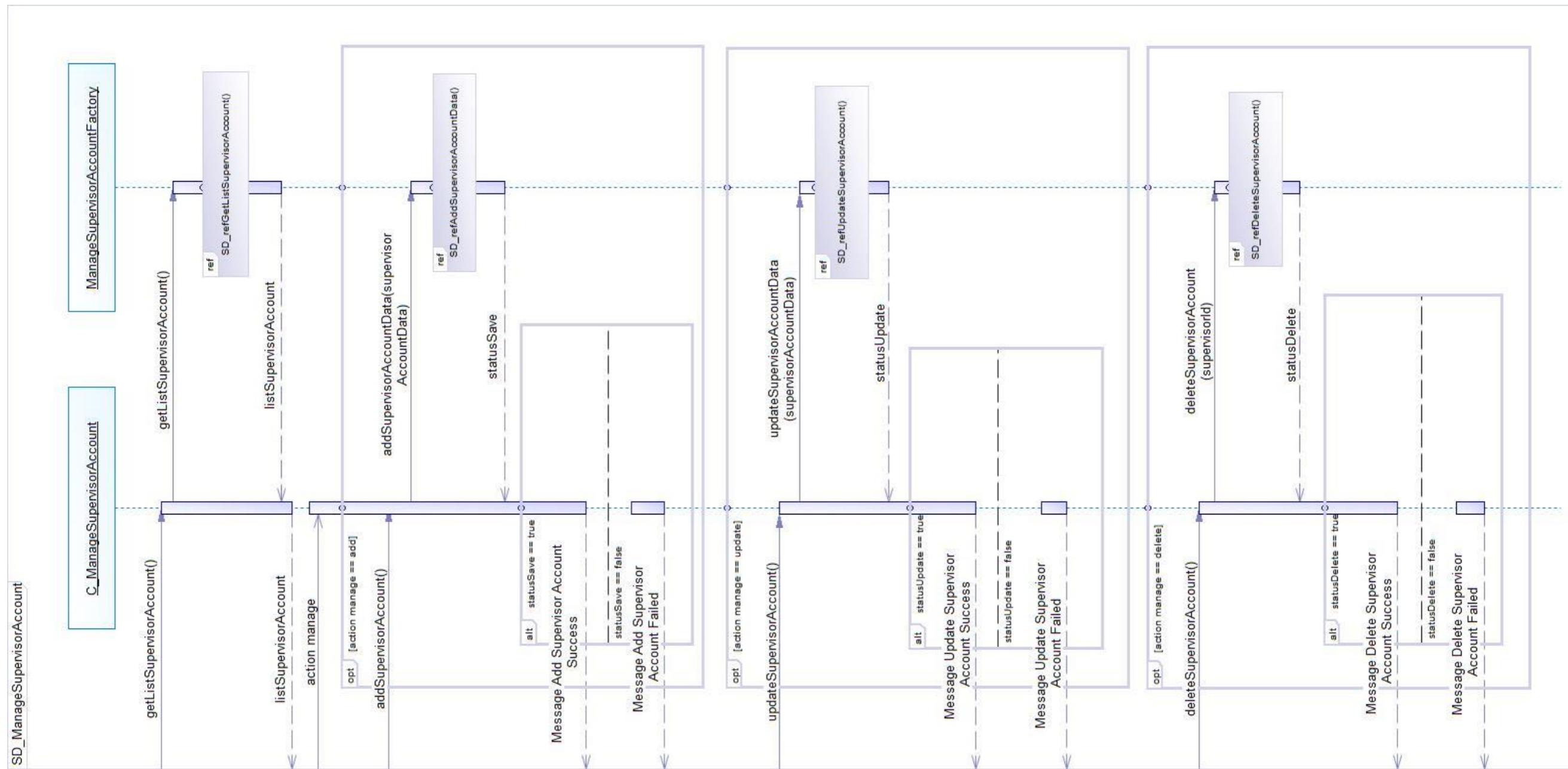
Tabel 144 Keterangan SD_refDeleteMedicalRepresentativeAccount

ID Sequence diagram	SD-MMR-05
Nama Sequence diagram	SD_refDeleteMedicalRepresentativeAccount
Nama Class Terkait	ManageMedicalRepresentativeAccountController, MedicalRepresentativeModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menghapus data akun <i>medical rep</i> ke <i>database</i> .
Method yang terkait	<code>delete_medical_representative_account_get()</code> , <code>delete_medical_representative_account(medrepId)</code>
Sequence diagram yang terkait	-
Algoritma	
Kamus data	<pre> medrepId : variabel tunggal bertipe string statusDelete : variabel tunggal bertipe boolean </pre>
	<pre> PROCEDURE delete_medical_representative_account_get() BEGIN medrepId ← \$this->get('medrepId') statusDelete ← \$this->medicalrepresentativemodel- >delete_medical_representative_account(medrepId) END PROCEDURE </pre>

```
FUNCTION delete_medicalRepresentative_account(medrepId) ->
boolean
BEGIN
    query ← $this->db->query("UPDATE medicalrepresentative
SET DELETED = 1 WHERE MEDREP_ID = 'medrepId'")

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION
```

6. SD – Manage Supervisor Account



Gambar 106 SD Manage Supervisor Account

Tabel 145 Keterangan SD_ManageSupervisorAccount

ID Sequence diagram	SD-MSV-01
Nama Sequence diagram	SD_ManageSupervisorAccount
Nama Class Terkait	C_ManageSupervisorAccount, ManageSupervisorAccountFactory
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengelola akun supervisor.
Method yang terkait	getListSupervisorAccount(), addSupervisorAccount(), addSupervisorAccountData(supervisorAccountData), updateSupervisorAccount(), updateSupervisorAccountData(supervisorAccountData), deleteSupervisorAccount(), deleteSupervisorAccount(supervisorId).
Sequence diagram yang terkait	SD_refGetListSupervisorAccount, SD_refAddSupervisorAccountData, SD_refUpdateSupervisorAccount, SD_refDeleteSupervisor
Algoritma	<pre> Kamus data listSupervisorAccount : variabel array objek PROCEDURE getListSupervisorAccount() BEGIN manageSupervisorAccountFactory.getListSupervisorAccount() .then(function(result) { listSupervisorAccount ← result.data.listSupervisorAccount }) END PROCEDURE PROCEDURE addSupervisorAccount() BEGIN supervisorAccountData ← array('supervisorId' ← \$scope.supervisorId, 'supervisorName' ← \$scope.supervisorName, 'supervisorContact' ← \$scope.supervisorContact, 'password' ← \$scope.password) manageSupervisorAccountFactory.addSupervisorAccountData(supervisorAccountData).then(function(result) { statusSave ← result.data.statusSave IF statusSave == TRUE THEN WRITE (layar) "save success" ELSE WRITE (layar) "save failed" END IF }) END PROCEDURE PROCEDURE updateSupervisorAccount() BEGIN supervisorAccountData ← array('supervisorId' ← \$scope.supervisorId, </pre>

```
'supervisorName' ← $scope.supervisorName,
'supervisorContact' ← $scope.supervisorContact,
'password' ← $scope.password
)

manageSupervisorAccountFactory.updateSupervisorAccountD
ata(supervisorAccountData).then(function(result) {
    statusUpdate ← result.data.statusUpdate
    IF statusUpdate == TRUE THEN
        WRITE (layar) "update success"
    ELSE
        WRITE (layar) "update failed"
    END IF
})
END PROCEDURE

PROCEDURE deleteSupervisorAccount()
BEGIN
    supervisorId ← $scope.supervisorId
    manageSupervisorAccountFactory.deleteSupervisorAcccount
    (supervisorId).then(function(result) {
        statusDelete ← result.data.statusDelete
        IF statusDelete == TRUE THEN
            WRITE (layar) "delete success"
        ELSE
            WRITE (layar) "delete failed"
        END IF
    })
END PROCEDURE

FUNCTION
manageSupervisorAccountFactory.getListSupervisorAccount()
BEGIN
    return $http.get("http://localhost/medical-rep-
    backend/managesupervisoraccount/list_supervisor_account
    ")
END FUNCTION

FUNCTION
manageSupervisorAccountFactory.addSupervisorAccountData(supervis
orAccountData)
BEGIN
    return $http.post("http://localhost/medical-rep-
    backend/managesupervisoraccount
    /add_supervisor_account", supervisorAccountData)
END PROCEDURE

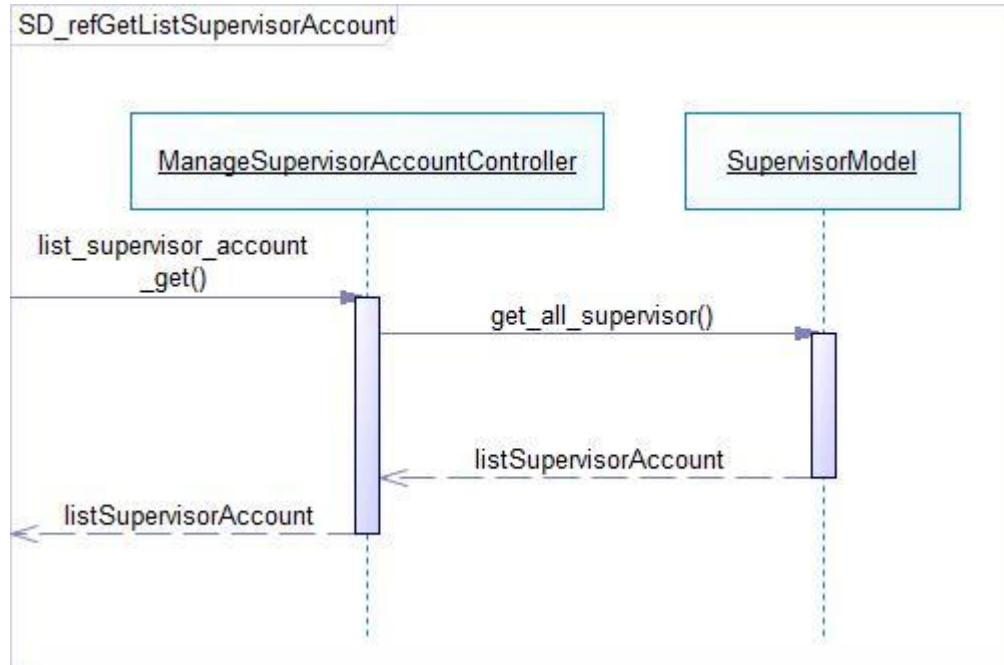
FUNCTION
manageSupervisorAccountFactory.updateSupervisorAccountData(superv
isorAccountData)
BEGIN
    return $http.post("http://localhost/medical-rep-
    backend/managesupervisoraccount
    /update_supervisor_account", supervisorAccountData)
END PROCEDURE

FUNCTION
manageSupervisorAccountFactory.deleteSupervisorAccount(superviso
rId)
```

```

BEGIN
    return $http.get('http://localhost/medical-rep-backend/
managesupervisoraccount
/delete_supervisor_account?supervisorId='+
supervisorId)
END PROCEDURE

```

Gambar 107 SD Referensi *Get List Supervisor Account*

Tabel 146 Keterangan SD_refGetListSupervisorAccount

ID Sequence diagram	SD-MSV-02
Nama Sequence diagram	SD_refGetListSupervisorAccount
Nama Class Terkait	ManageSupervisorAccountController, SupervisorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengambil data akun supervisor dari database.
Method yang terkait	list_supervisor_account_get(), get_all_supervisor()
Sequence diagram yang terkait	-

Algoritma

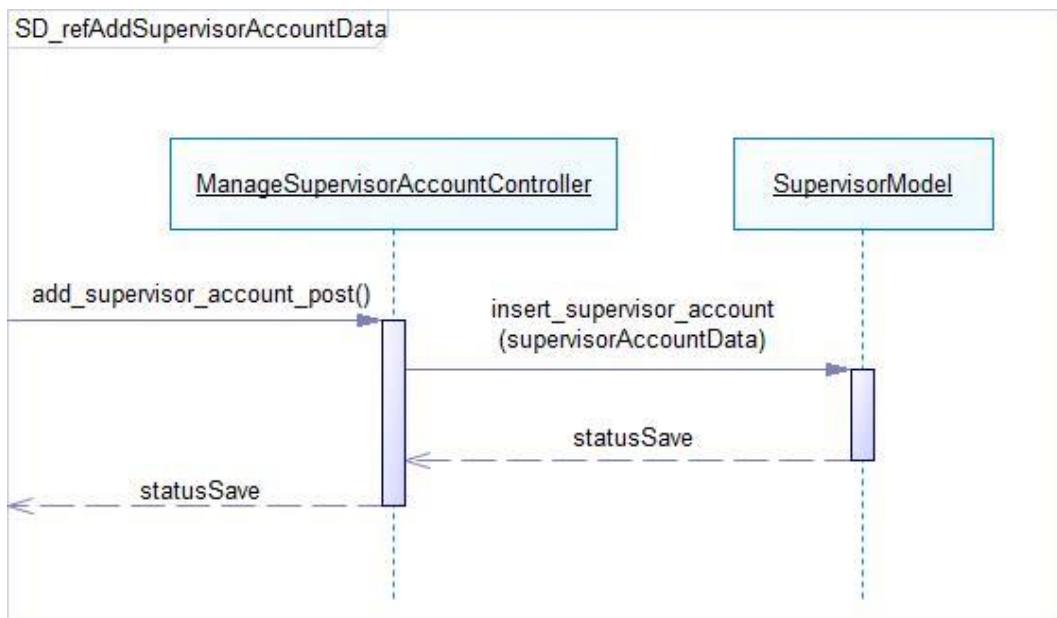
```

Kamus data
    listSupervisorAccount : variabel array objek

PROCEDURE list_supervisor_account_get()
BEGIN
    listSupervisorAccount ← $this->supervisormodel-
        >get_all_supervisor()
END PROCEDURE

FUNCTION get_all_supervisor()
BEGIN
    return $this->db->query("SELECT * FROM supervisor WHERE
        DELETED = 0")->result()
END PROCEDURE

```



Gambar 108 SD Referensi Add Supervisor Account Data

Tabel 147 Keterangan SD_refAddSupervisorAccountData

ID Sequence diagram	SD-MSV-03
Nama Sequence diagram	SD_refAddSupervisorAccountData
Nama Class Terkait	ManageSupervisorAccountController, SupervisorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menambahkan data akun supervisor ke database.

<i>Method</i> yang terkait	add_supervisor_account_post(), insert_supervisor_account(supervisorAccountData)
<i>Sequence diagram</i> yang terkait	-

Algoritma

```

Kamus data
    supervisorAccountData : variabel array objek
    statusSave : variabel tunggal bertipe boolean

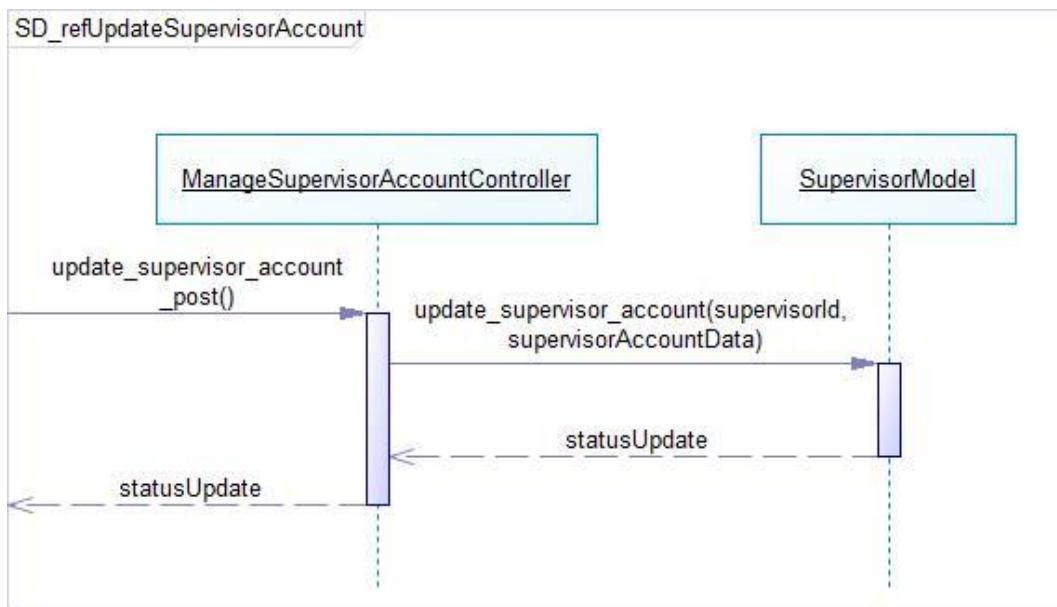
PROCEDURE add_supervisor_account_post()
BEGIN
    supervisorAccountData ← array(
        'SUPERVISOR_ID' ← $this->post('supervisorId'),
        'SUPERVISOR_NAME' ← $this->post('supervisorName'),
        'SUPERVISOR_CONTACT' ← $this-
            >post('supervisorContact'),
        'PASSWORD' ← $this->post('password')
    )

    statusSave ← $this->supervisormodel-
        >save_supervisor_account(supervisorAccountData)
END PROCEDURE

FUNCTION insert_supervisor_account(supervisorAccountData) ->
boolean
BEGIN
    query ← $this->db->insert('supervisor,
    supervisorAccountData)

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

```

Gambar 109 SD Referensi *Update Supervisor Account*

Tabel 148 Keterangan SD_refUpdateSupervisorAccount

ID Sequence diagram	SD-MSV-04
Nama Sequence diagram	SD_refUpdateSupervisorAccount
Nama Class Terkait	ManageSupervisorAccountController, SupervisorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses mengubah data akun supervisor ke database.
Method yang terkait	<code>update_supervisor_account_post()</code> , <code>update_supervisor_account(supervisorId, supervisorAccountData)</code>
Sequence diagram yang terkait	-

Algoritma

Kamus data

```

supervisorAccountData : variabel array objek
statusUpdate : variabel tunggal bertipe boolean

```

PROCEDURE `update_supervisor_account_post()`

BEGIN

```

supervisorId ← $this->post('supervisorId')
supervisorAccountData ← array(
    'SUPERVISOR_NAME' ← $this->post('supervisorName'),
    'SUPERVISOR_CONTACT' ← $this-
        >post('supervisorContact'),
    'PASSWORD' ← $this->post('password')
)

```

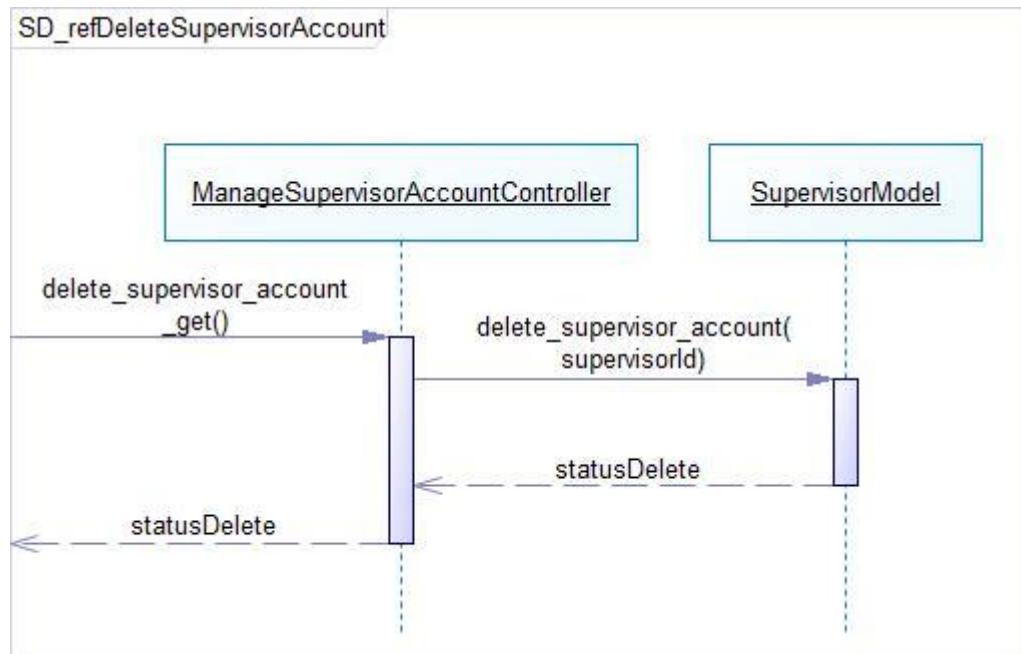
```

statusUpdate ← $this->supervisormodel-
    >update_supervisor_account(supervisorId,
    supervisorAccountData)
END PROCEDURE

FUNCTION update_supervisor_account(supervisorId,
supervisorAccountData) -> boolean
BEGIN
    $this->db->where('SUPERVISOR_ID', supervisorId)
    query ← $this->db->update('supervisor',
    supervisorAccountData)

    IF query == TRUE THEN
        return TRUE
    ELSE
        return FALSE
    END IF
END FUNCTION

```

Gambar 110 SD Referensi *Delete Supervisor Account*

Tabel 149 Keterangan SD_refDeleteSupervisorAccount

ID Sequence diagram	SD-MSV-05
Nama Sequence diagram	SD_refDeleteSupervisorAccount
Nama Class Terkait	ManageSupervisorAccountController, SupervisorModel
Deskripsi	Sequence ini digunakan untuk menjelaskan proses menghapus data akun supervisor ke database.
Method yang terkait	delete_supervisor_account_get(), delete_supervisor_account(supervisorId)
Sequence diagram yang terkait	-
Algoritma	<p>Kamus data</p> <pre> supervisorId : variabel tunggal bertipe string statusDelete : variabel tunggal bertipe boolean </pre> <p>PROCEDURE delete_supervisor_account_get()</p> <p><u>BEGIN</u></p> <pre> supervisorId ← \$this->get('supervisorId') statusDelete ← \$this->supervisormodel- >delete_supervisor_account(supervisorId) </pre> <p><u>END PROCEDURE</u></p> <p>FUNCTION delete_supervisor_account(supervisorId) -> boolean</p> <p><u>BEGIN</u></p> <pre> query ← \$this->db->query("UPDATE supervisor SET DELETED = 1 WHERE SUPERVISOR_ID = 'supervisorId'") IF query == TRUE THEN return TRUE ELSE return FALSE END IF </pre> <p><u>END FUNCTION</u></p>

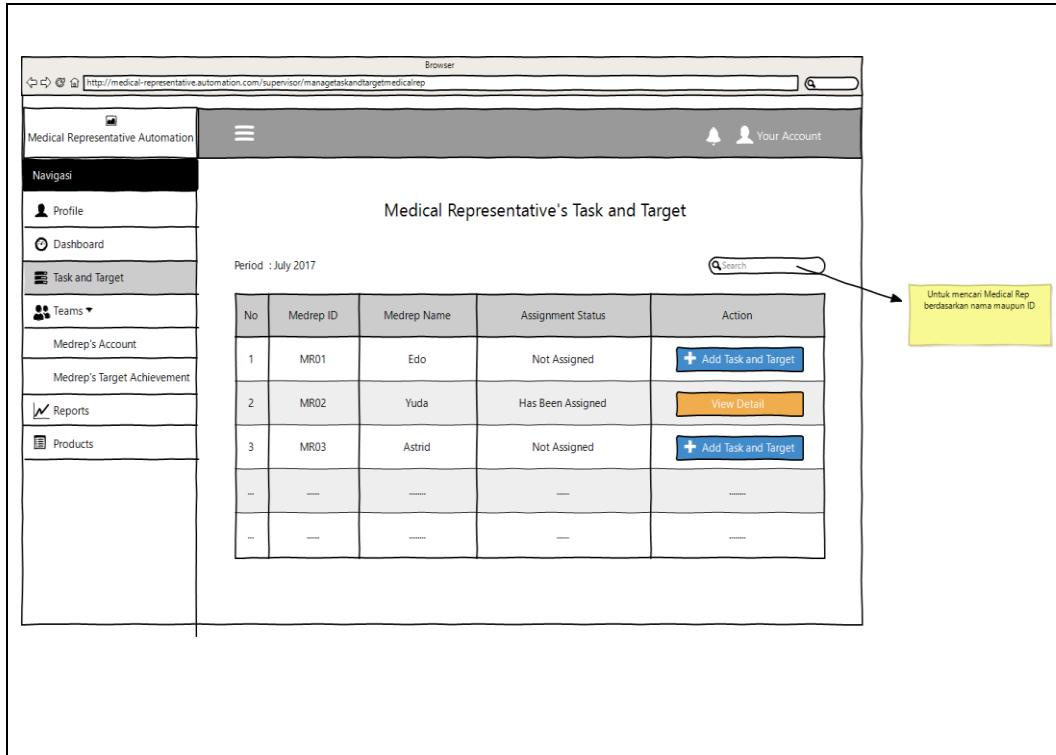
4.2.3 Perancangan *User Interface*

Rancangan *user interface* yang dijelaskan pada sub bab ini dibuat untuk menggambarkan tampilan *web frontend* yang digunakan oleh pengguna untuk berinteraksi dengan aplikasi. Rancangan ini berdasarkan *system sequence diagram* yang telah dibuat.

1. UI – Manage Medical Representative’s Task and Target

Tabel 150 Tampilan Halaman Awal *Task and Target Medical Rep*

No. UI	GUI-MTT-01
Nama UI	Tampilan Halaman Awal <i>Task and Target Medical Rep</i>
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Task and Target</i>.
Deskripsi	<p>Tampilan ini merupakan halaman awal untuk mengelola tugas dan target <i>medical rep</i>. Halaman ini menampilkan data <i>medical rep</i> beserta status pemberian tugasnya (<i>Assignment Status</i>) dalam bentuk tabel. Penejelasan mengenai status pemberian tugas pada tabel yaitu, sebagai berikut:</p> <ul style="list-style-type: none"> - <i>Has Been Assigned</i> : Supervisor sudah memberikan tugas dan target kepada <i>medical rep</i>, kemudian akan tampil tombol <i>View Detail</i> yang dapat digunakan untuk menampilkan <i>detail</i> tugas dan target yang telah diberikan. - <i>Not Assigned</i> : Supervisor belum memberikan tugas dan target kepada <i>medical rep</i>, dan terdapat tombol <i>Add Task and Target</i> pada kolom <i>Action</i> yang akan menuju <i>form</i> untuk mengisi tugas dan target <i>medical rep</i>.
Gambar	



Tabel 151 Tampilan Form Add Medical Representative's Task and Target

No. UI	GUI-MTT-02
Nama UI	Tampilan Form Add Medical Representative's Task and Target
Pengguna	Supervisor
Precondition	Pengguna telah memilih tombol <i>Add Task and Target</i> pada kolom <i>Action</i> di halaman awal <i>Task and Target</i> .
Deskripsi	<p>Tampilan ini merupakan halaman yang menampilkan <i>form</i> untuk pengisian tugas dan target <i>medical rep</i>. <i>Field</i> yang harus diisi, yaitu:</p> <ul style="list-style-type: none"> - Target : jumlah target yang harus dicapai <i>medical rep</i> (dalam rupiah) - Start Date of Task : tanggal awal periode tugas - End Date of Task : tanggal akhir periode tugas - Product Assigned : produk yang harus di-<i>detailing</i> oleh <i>medical rep</i>, berupa <i>checkbox</i>. - Note : catatan mengenai tugas dan target yang diberikan. <p>Sedangkan <i>field</i> Medical Representative's Name: terisi otomatis saat memilih <i>medical rep</i>.</p> <p>Terdapat tombol <i>Submit</i> untuk menyimpan dan mengirimkan data tugas dan target ke <i>medical rep</i>, dan <i>Cancel</i> untuk membatalkan tugas dan target.</p>
Gambar	

Browser
http://medical-representative.automation.com/supervisor/addtaskandtargetmedicalrep-MR01

Medical Representative Automation

Profile

Dashboard

Task and Target

Teams

Medrep's Account

Medrep's Target Achievement

Reports

Products

Add Medical Representative's Task and Target

Period : July 2017

Medical Representative's Name : Edo

Target : input medical representative's target
total target in rupiah

Start Date of Task : DD / Month / YYYY

End Date of Task : DD / Month / YYYY

Products Assigned :

Amoxicillin Ceteme
 Paracetamol Dexametason
 Ambroxol Syrup Salbutamol

Note : Berisi catatan mengenai tugas dan target yang diberikan kepada Medical Rep

Submit Cancel

Informasi Submit Berhasil
Add Medical Representative's Task and Target Has Been Assigned !!

Nama Medical Rep terisi otomatis saat Supervisor memilih Add Task And Target pada nama Medical Rep yang sesuai

Supervisor memilih produk yang harus didetaliling oleh Medical Rep kepada dokter dengan checklist nama produk
*nama produk hanya contoh

Ketika Cancel kembali ke menu awal Task and Target, tampil list medrep

Tabel 152 Tampilan *Detail Medical Representative's Task and Target*

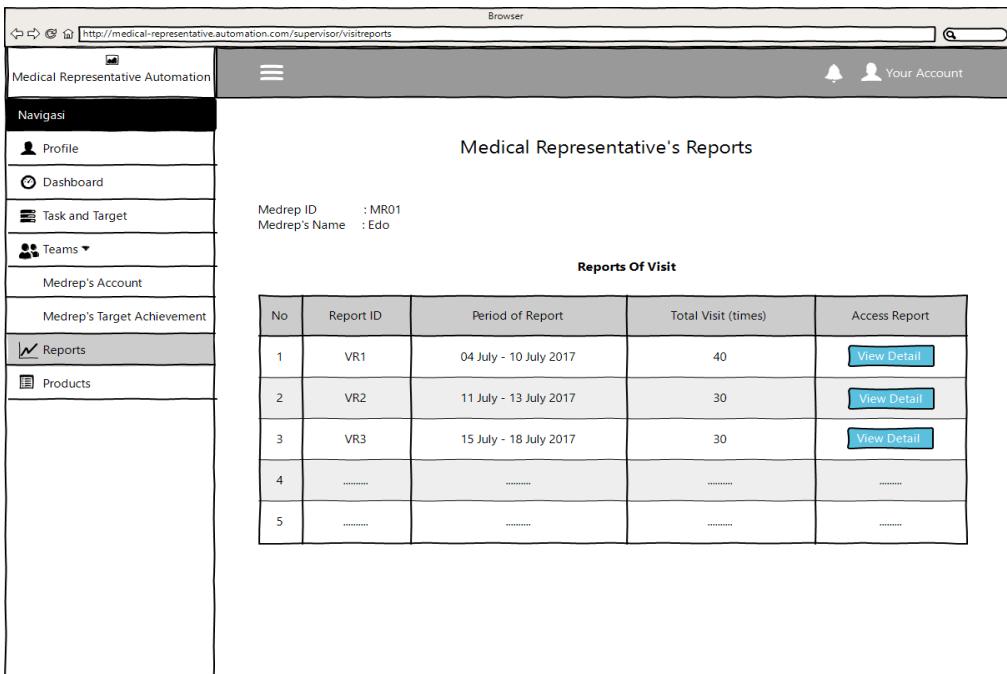
No. UI	GUI-MTT-03
Nama UI	Tampilan <i>Detail Medical Representative's Task and Target</i>
Pengguna	Supervisor
Precondition	Pengguna telah memilih tombol <i>View Detail</i> pada kolom <i>Action</i> di halaman <i>Task and Target Medical Rep.</i>
Deskripsi	Tampilan ini merupakan halaman yang menampilkan rincian atau <i>detail</i> dari tugas dan target yang telah diberikan supervisor kepada <i>medical rep.</i>
Gambar	

2. UI – Show Report

Tabel 153 Tampilan Halaman Awal Show Report

No. UI	GUI-SR-01																																			
Nama UI	Tampilan Halaman Awal Show Report																																			
Pengguna	Supervisor																																			
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Reports</i>. - Pengguna telah memilih menu <i>Medical rep's Report</i> 																																			
Deskripsi	<p>Tampilan ini merupakan halaman awal untuk menampilkan laporan yang dibuat oleh <i>medical rep</i>. Halaman ini menampilkan semua data <i>medical rep</i> yang telah membuat laporan kunjungan (<i>report of visit</i>) maupun laporan transaksi (<i>report of transaction</i>) yang disajikan dalam bentuk tabel. Terdapat dua tombol, yaitu:</p> <ul style="list-style-type: none"> - <i>View Report</i> pada kolom <i>Report Of Visit</i> untuk mengakses laporan kunjungan yang dibuat <i>medical rep</i>. - <i>View Report</i> pada kolom <i>Report Of Transaction</i> untuk mengakses laporan transaksi yang dibuat <i>medical rep</i>. 																																			
Gambar	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Medrep ID</th> <th>Medrep's Name</th> <th>Report Of Visit</th> <th>Report Of Transaction</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>MR01</td> <td>Edo</td> <td>[View Report]</td> <td>[View Report]</td> </tr> <tr> <td>2</td> <td>MR02</td> <td>Yuda</td> <td>[View Report]</td> <td>[View Report]</td> </tr> <tr> <td>3</td> <td>MR03</td> <td>Astrid</td> <td>[View Report]</td> <td>[View Report]</td> </tr> <tr> <td>4</td> <td>-----</td> <td>---</td> <td>---</td> <td>---</td> </tr> <tr> <td>5</td> <td>-----</td> <td>---</td> <td>---</td> <td>---</td> </tr> <tr> <td>6</td> <td>-----</td> <td>---</td> <td>---</td> <td>---</td> </tr> </tbody> </table>	No	Medrep ID	Medrep's Name	Report Of Visit	Report Of Transaction	1	MR01	Edo	[View Report]	[View Report]	2	MR02	Yuda	[View Report]	[View Report]	3	MR03	Astrid	[View Report]	[View Report]	4	-----	---	---	---	5	-----	---	---	---	6	-----	---	---	---
No	Medrep ID	Medrep's Name	Report Of Visit	Report Of Transaction																																
1	MR01	Edo	[View Report]	[View Report]																																
2	MR02	Yuda	[View Report]	[View Report]																																
3	MR03	Astrid	[View Report]	[View Report]																																
4	-----	---	---	---																																
5	-----	---	---	---																																
6	-----	---	---	---																																

Tabel 154 Tampilan list Laporan Kunjungan (*Reports Of Visit*)

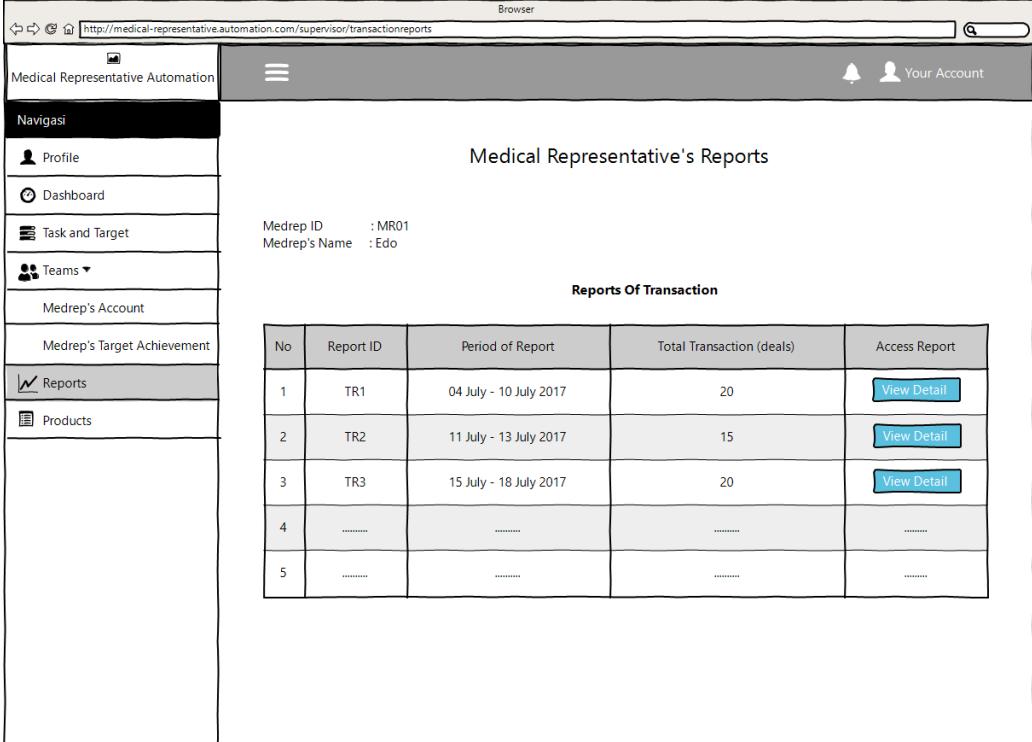
No. UI	GUI-SR-02																														
Nama UI	Tampilan list Laporan Kunjungan (<i>Reports Of Visit</i>)																														
Pengguna	Supervisor																														
Precondition	Pengguna telah memilih tombol <i>View Report</i> pada kolom <i>Report Of Visit</i> pada halaman awal <i>Show Report</i>																														
Deskripsi	Tampilan ini merupakan halaman yang menampilkan daftar laporan kunjungan (<i>reports of visit</i>) yang dibuat oleh <i>medical rep</i> yang telah dipilih pada halaman awal <i>Show Report</i> beserta periode laporannya yang ditampilkan melalui tabel. Terdapat tombol <i>View Detail</i> disetiap baris untuk mengakses detail laporan kunjungan.																														
Gambar	 <p>The screenshot shows a web browser window titled 'Browser' with the URL 'http://medical-representative.automation.com/supervisor/visitreports'. The page has a dark header with a bell icon and 'Your Account'. On the left is a sidebar titled 'Medical Representative Automation' with a 'Navigasi' section containing 'Profile', 'Dashboard', 'Task and Target', 'Teams', 'Medrep's Account', 'Medrep's Target Achievement', 'Reports' (which is highlighted), and 'Products'. The main content area is titled 'Medical Representative's Reports' and displays 'Medrep ID : MR01' and 'Medrep's Name : Edo'. Below this is a table titled 'Reports Of Visit' with columns: No, Report ID, Period of Report, Total Visit (times), and Access Report. The table contains five rows of data:</p> <table border="1"> <thead> <tr> <th>No</th> <th>Report ID</th> <th>Period of Report</th> <th>Total Visit (times)</th> <th>Access Report</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>VR1</td> <td>04 July - 10 July 2017</td> <td>40</td> <td>View Detail</td> </tr> <tr> <td>2</td> <td>VR2</td> <td>11 July - 13 July 2017</td> <td>30</td> <td>View Detail</td> </tr> <tr> <td>3</td> <td>VR3</td> <td>15 July - 18 July 2017</td> <td>30</td> <td>View Detail</td> </tr> <tr> <td>4</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>5</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> </tbody> </table>	No	Report ID	Period of Report	Total Visit (times)	Access Report	1	VR1	04 July - 10 July 2017	40	View Detail	2	VR2	11 July - 13 July 2017	30	View Detail	3	VR3	15 July - 18 July 2017	30	View Detail	4	5
No	Report ID	Period of Report	Total Visit (times)	Access Report																											
1	VR1	04 July - 10 July 2017	40	View Detail																											
2	VR2	11 July - 13 July 2017	30	View Detail																											
3	VR3	15 July - 18 July 2017	30	View Detail																											
4																											
5																											

Tabel 155 Tampilan *Detail Reports Of Visit*

No. UI	GUI-SR-03
Nama UI	Tampilan <i>Detail Reports Of Visit</i>
Pengguna	Supervisor
Precondition	Pengguna telah memilih tombol <i>View Detail</i> pada kolom <i>Show Report</i> di halaman <i>Reports Of Visit</i>
Deskripsi	Tampilan ini merupakan halaman yang menampilkan rincian atau <i>detail</i> dari laporan kunjungan (<i>reports of visit</i>) yang dibuat oleh <i>medical rep</i> yang telah dipilih pada halaman <i>Reports Of Visit</i> . Pada

	<p>halaman ini ditampilkan seluruh rincian laporan berdasarkan periode yang dipilih, tampilan berbentuk <i>collapsed</i>, apabila dipilih salah satu tanggal laporan, maka akan menampilkan rincian laporan dari hari yang dipilih tersebut, jika memilih hari yang lain, maka rincian laporan yang dipilih sebelumnya akan menutup (<i>collapse</i>), dan menampilkan rincian laporan yang baru dipilih.</p>																														
Gambar																															
<table border="1"> <thead> <tr> <th>No</th> <th>Doctor's Name</th> <th>Time of Visit (hour)</th> <th>Location</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Arif Khairuddin</td> <td>10.00</td> <td>RS. Hasan Sadikin</td> <td>dokter menghadapi banyak pasien sehingga detailing sebentar</td> </tr> <tr> <td>2</td> <td>Helmi Fauzi</td> <td>13.00</td> <td>RS. Hasan Sadikin</td> <td>dokter memberikan respon yang bagus tapi belum bisa bekerja sama</td> </tr> <tr> <td>3</td> <td>Inda Nabila</td> <td>15.00</td> <td>RSKIA Hermina</td> <td>dokter memberikan respon yang bagus tapi belum bisa bekerja sama</td> </tr> <tr> <td>4</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>5</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> </tbody> </table> <p>Period of Report : 04 July - 10 July 2017 Medrep ID : MR01 Medrep's Name : Edo</p> <p>04 July 2017 > 05 July 2017 > 06 July 2017 > 07 July 2017 > 08 July 2017 > 09 July 2017 > 10 July 2017 ></p>		No	Doctor's Name	Time of Visit (hour)	Location	Result	1	Arif Khairuddin	10.00	RS. Hasan Sadikin	dokter menghadapi banyak pasien sehingga detailing sebentar	2	Helmi Fauzi	13.00	RS. Hasan Sadikin	dokter memberikan respon yang bagus tapi belum bisa bekerja sama	3	Inda Nabila	15.00	RSKIA Hermina	dokter memberikan respon yang bagus tapi belum bisa bekerja sama	4	5
No	Doctor's Name	Time of Visit (hour)	Location	Result																											
1	Arif Khairuddin	10.00	RS. Hasan Sadikin	dokter menghadapi banyak pasien sehingga detailing sebentar																											
2	Helmi Fauzi	13.00	RS. Hasan Sadikin	dokter memberikan respon yang bagus tapi belum bisa bekerja sama																											
3	Inda Nabila	15.00	RSKIA Hermina	dokter memberikan respon yang bagus tapi belum bisa bekerja sama																											
4																											
5																											

Tabel 156 Tampilan List Laporan Transaksi (*Reports Of Transaction*)

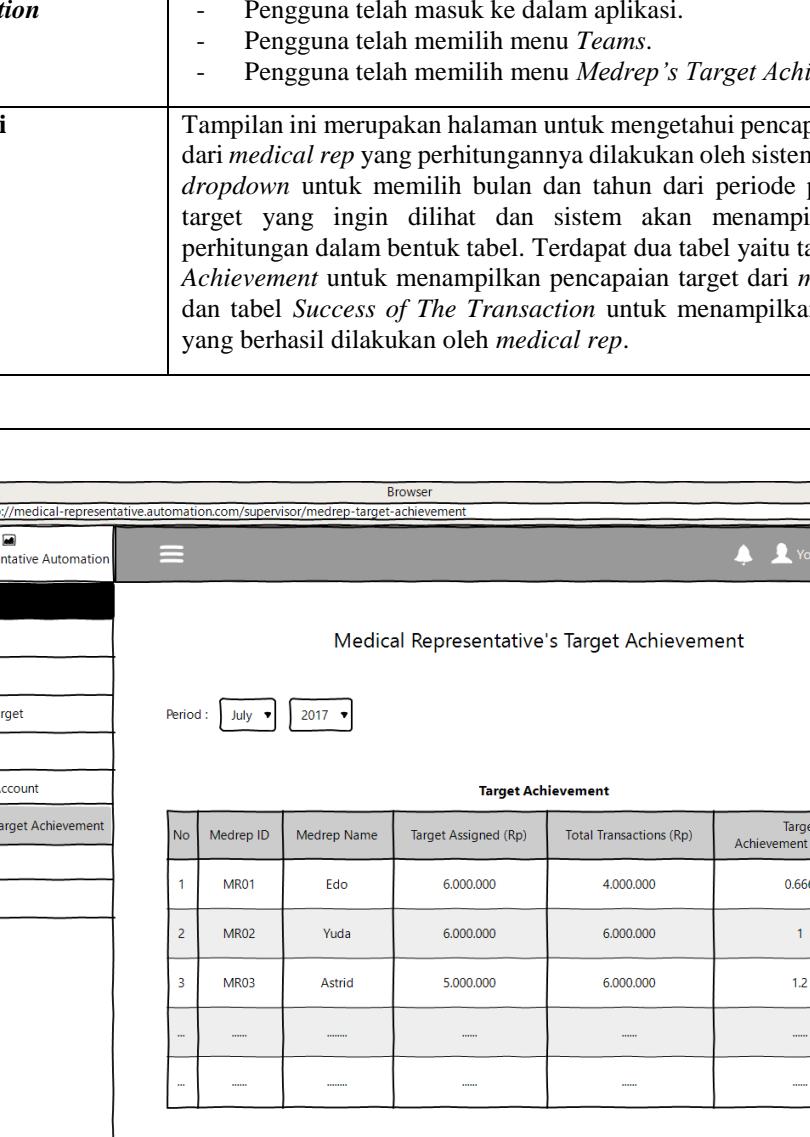
No. UI	GUI-SR-04																														
Nama UI	Tampilan List Laporan Transaksi (<i>Reports Of Transaction</i>)																														
Pengguna	Supervisor																														
Precondition	Pengguna telah memilih tombol <i>View Report</i> pada kolom <i>Report Of Transaction</i> pada halaman awal <i>Show Report</i> .																														
Deskripsi	Tampilan ini merupakan halaman yang menampilkan daftar laporan transaksi (<i>reports of transaction</i>) yang dibuat oleh <i>medical rep</i> yang telah dipilih pada halaman awal <i>Show Report</i> pada kolom <i>Report Of Transaction</i> beserta periode laporannya yang ditampilkan melalui tabel. Terdapat tombol <i>View Detail</i> disetiap baris untuk mengakses detail laporan transaksi.																														
Gambar	 <p>The screenshot shows a web browser window titled 'Browser' with the URL 'http://medical-representative.automation.com/supervisor/transactionreports'. The page is titled 'Medical Representative's Reports' and displays information for a user with ID MR01 and name Edo. A table titled 'Reports Of Transaction' lists five reports (TR1, TR2, TR3, 4, 5) with their respective report IDs, periods (04 July - 10 July 2017, 11 July - 13 July 2017, 15 July - 18 July 2017), total transactions (20, 15, 20), and a 'View Detail' button for each row.</p> <table border="1"> <thead> <tr> <th>No</th> <th>Report ID</th> <th>Period of Report</th> <th>Total Transaction (deals)</th> <th>Access Report</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>TR1</td> <td>04 July - 10 July 2017</td> <td>20</td> <td>View Detail</td> </tr> <tr> <td>2</td> <td>TR2</td> <td>11 July - 13 July 2017</td> <td>15</td> <td>View Detail</td> </tr> <tr> <td>3</td> <td>TR3</td> <td>15 July - 18 July 2017</td> <td>20</td> <td>View Detail</td> </tr> <tr> <td>4</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>5</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> </tbody> </table>	No	Report ID	Period of Report	Total Transaction (deals)	Access Report	1	TR1	04 July - 10 July 2017	20	View Detail	2	TR2	11 July - 13 July 2017	15	View Detail	3	TR3	15 July - 18 July 2017	20	View Detail	4	5
No	Report ID	Period of Report	Total Transaction (deals)	Access Report																											
1	TR1	04 July - 10 July 2017	20	View Detail																											
2	TR2	11 July - 13 July 2017	15	View Detail																											
3	TR3	15 July - 18 July 2017	20	View Detail																											
4																											
5																											

Tabel 157 Tampilan *Detail Reports Of Transaction*

No. UI	GUI-SR-05																														
Nama UI	Tampilan <i>Detail Reports Of Transaction</i>																														
Pengguna	Supervisor																														
Precondition	Pengguna telah memilih tombol <i>View Detail</i> pada kolom <i>Show Report</i> di halaman <i>Reports Of Transaction</i> .																														
Deskripsi	Tampilan ini merupakan halaman yang menampilkan rincian atau <i>detail</i> dari laporan transaksi (<i>reports of transaction</i>) yang dibuat oleh <i>medical rep</i> yang telah dipilih pada halaman <i>Reports Of Transaction</i> . Pada halaman ini ditampilkan seluruh rincian laporan berdasarkan periode yang dipilih, tampilan berbentuk <i>collapsed</i> , apabila dipilih salah satu tanggal laporan, maka akan menampilkan rincian laporan dari hari yang dipilih tersebut, jika memilih hari yang lain, maka rincian laporan yang dipilih sebelumnya akan menutup (<i>collapse</i>), dan menampilkan rincian laporan yang baru dipilih.																														
Gambar	<table border="1"> <thead> <tr> <th>No</th> <th>Doctor's Name</th> <th>Product</th> <th>Prescription Target</th> <th>Period</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Arif Khairuddin</td> <td>Amoxilin Dexametason Paracetamol</td> <td>Rp2.000.000 Rp1.000.000 Rp1.000.000</td> <td>2 Months</td> </tr> <tr> <td>2</td> <td>Helmi Fauzi</td> <td>Paracetamol Dexametason</td> <td>Rp2.000.000 Rp1.000.000</td> <td>3 Months</td> </tr> <tr> <td>3</td> <td>Inda Nabila</td> <td>Dexametason</td> <td>Rp2.000.000</td> <td>4 Months</td> </tr> <tr> <td>4</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>5</td> <td>.....</td> <td>.....</td> <td>.....</td> <td>.....</td> </tr> </tbody> </table> <p>05 July 2017 > 06 July 2017 > 07 July 2017 > 08 July 2017 > 09 July 2017 > 10 July 2017 ></p>	No	Doctor's Name	Product	Prescription Target	Period	1	Arif Khairuddin	Amoxilin Dexametason Paracetamol	Rp2.000.000 Rp1.000.000 Rp1.000.000	2 Months	2	Helmi Fauzi	Paracetamol Dexametason	Rp2.000.000 Rp1.000.000	3 Months	3	Inda Nabila	Dexametason	Rp2.000.000	4 Months	4	5
No	Doctor's Name	Product	Prescription Target	Period																											
1	Arif Khairuddin	Amoxilin Dexametason Paracetamol	Rp2.000.000 Rp1.000.000 Rp1.000.000	2 Months																											
2	Helmi Fauzi	Paracetamol Dexametason	Rp2.000.000 Rp1.000.000	3 Months																											
3	Inda Nabila	Dexametason	Rp2.000.000	4 Months																											
4																											
5																											

3. UI – Calculate Medical Representative's Target Achievement

Tabel 158 Tampilan Daftar Hasil Perhitungan Pencapaian Target *Medical Rep*

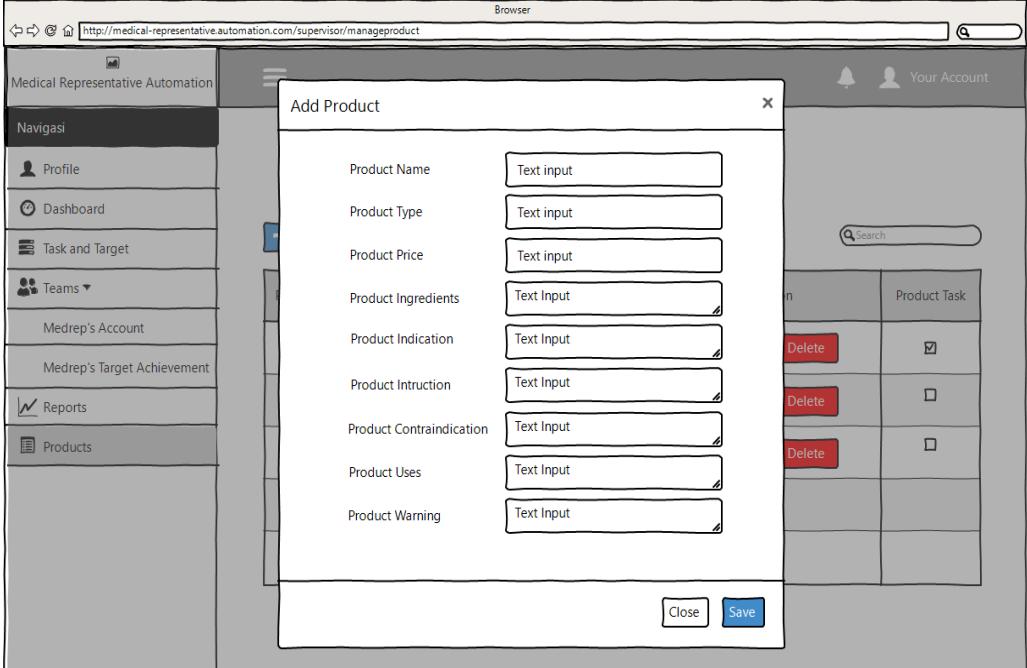
No. UI	GUI-CT-01																																				
Nama UI	Tampilan Daftar Hasil Perhitungan Pencapaian Target <i>Medical Rep</i>																																				
Pengguna	Supervisor																																				
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Teams</i>. - Pengguna telah memilih menu <i>Medrep's Target Achievement</i> 																																				
Deskripsi	Tampilan ini merupakan halaman untuk mengetahui pencapaian target dari <i>medical rep</i> yang perhitungannya dilakukan oleh sistem. Terdapat <i>dropdown</i> untuk memilih bulan dan tahun dari periode pencapaian target yang ingin dilihat dan sistem akan menampilkan hasil perhitungan dalam bentuk tabel. Terdapat dua tabel yaitu tabel <i>Target Achievement</i> untuk menampilkan pencapaian target dari <i>medical rep</i> dan tabel <i>Success of The Transaction</i> untuk menampilkan transaksi yang berhasil dilakukan oleh <i>medical rep</i> .																																				
Gambar	 <table border="1" data-bbox="574 1401 1295 1684"> <thead> <tr> <th>No</th><th>Medrep ID</th><th>Medrep Name</th><th>Target Assigned (Rp)</th><th>Total Transactions (Rp)</th><th>Target Achievement (X Target)</th></tr> </thead> <tbody> <tr> <td>1</td><td>MR01</td><td>Edo</td><td>6.000.000</td><td>4.000.000</td><td>0.666</td></tr> <tr> <td>2</td><td>MR02</td><td>Yuda</td><td>6.000.000</td><td>6.000.000</td><td>1</td></tr> <tr> <td>3</td><td>MR03</td><td>Astrid</td><td>5.000.000</td><td>6.000.000</td><td>1.2</td></tr> <tr> <td>...</td><td>.....</td><td>.....</td><td>.....</td><td>.....</td><td>.....</td></tr> <tr> <td>...</td><td>.....</td><td>.....</td><td>.....</td><td>.....</td><td>.....</td></tr> </tbody> </table>	No	Medrep ID	Medrep Name	Target Assigned (Rp)	Total Transactions (Rp)	Target Achievement (X Target)	1	MR01	Edo	6.000.000	4.000.000	0.666	2	MR02	Yuda	6.000.000	6.000.000	1	3	MR03	Astrid	5.000.000	6.000.000	1.2
No	Medrep ID	Medrep Name	Target Assigned (Rp)	Total Transactions (Rp)	Target Achievement (X Target)																																
1	MR01	Edo	6.000.000	4.000.000	0.666																																
2	MR02	Yuda	6.000.000	6.000.000	1																																
3	MR03	Astrid	5.000.000	6.000.000	1.2																																
...																																
...																																

4. UI – Manage Product

Tabel 159 Tampilan Halaman Awal *Manage Product*

No. UI	GUI-MP-01
Nama UI	Tampilan Halaman Awal <i>Manage Product</i>
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Product</i>.
Deskripsi	<p>Tampilan ini merupakan halaman awal untuk mengelola data produk yang menjadi tugas dari timnya. Halaman ini menampilkan daftar produk dalam tabel yang berisi data produk id, nama produk, tipe produk dan harga produk, serta pilihan aksi, yaitu <i>edit</i> dan <i>delete</i>. Terdapat <i>checkbox</i> untuk memilih produk yang menjadi tugas supervisor pada periode tertentu. Selain itu terdapat tombol untuk menambahkan data produk baru yang posisinya terletak di atas tabel dan <i>textbox</i> untuk melakukan pencarian produk berdasarkan id atau nama produk.</p>
Gambar	

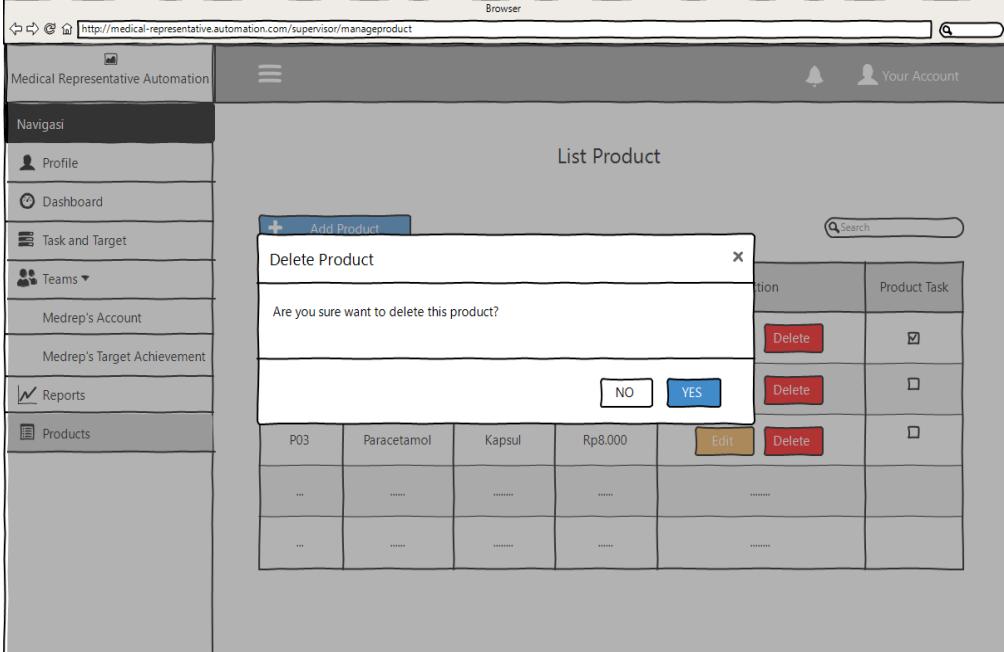
Tabel 160 Tampilan Halaman Add New Product

No. UI	GUI-MP-02
Nama UI	Tampilan Halaman Add New Product
Pengguna	Supervisor
Precondition	- Pengguna telah memilih tombol untuk menambahkan produk atau “Add Product”
Deskripsi	Tampilan ini merupakan untuk melakukan penambahan data produk. Sistem akan menampilkan <i>form input</i> data produk yang berupa <i>modal screen</i> . Tersedia label yang menunjukkan data yang harus diisikan pada <i>text input</i> . Kemudian terdapat tombol <i>Close</i> apabila pengguna ingin meng-cancel peng-input-an data, dan <i>Save</i> untuk menyimpan data produk.
Gambar	

Tabel 161 Tampilan Halaman *Edit Product Data*

No. UI	GUI-MP-03
Nama UI	Tampilan Halaman <i>Edit Product Data</i>
Pengguna	Supervisor
Precondition	- Pengguna telah memilih tombol untuk mengubah data produk atau tombol “Edit” pada tabel.
Deskripsi	Tampilan ini untuk melakukan perubahan data produk. Sistem akan menampilkan <i>form edit</i> data produk yang berupa <i>modal screen</i> . Sistem akan menampilkan terlebih dahulu data produk sebelumnya yang tersimpan di <i>database</i> . Tersedia label yang menunjukkan data yang harus diisi pada <i>text input</i> . Kemudian terdapat tombol <i>Close</i> apabila pengguna ingin <i>cancel input</i> data, dan <i>Save</i> untuk menyimpan perubahan data produk.
Gambar	

Tabel 162 Tampilan Halaman *Delete Product Data*

No. UI	GUI-MP-04
Nama UI	Tampilan Halaman <i>Delete Product Data</i>
Pengguna	Supervisor
Precondition	- Pengguna telah memilih tombol untuk menghapus data produk atau tombol “Delete” pada tabel.
Deskripsi	Terdapat tombol <i>Delete</i> untuk menghapus data produk pada tabel, apabila pengguna memilih <i>delete</i> maka sistem akan menampilkan kotak dialog apakah data ingin dihapus atau tidak, jika iya maka memilih tombol “Yes”, jika tidak maka memilih tombol “No” dan akan kembali ke halaman awal <i>manage product</i> .
Gambar	

5. UI – Manage Medical Representative Account

Tabel 163 Tampilan Daftar Data Akun *Medical Representative*

No. UI	GUI-MMR.01
Nama UI	Tampilan Daftar Data Akun <i>Medical Representative</i> .
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Medical Representative Account</i>.
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan semua data akun <i>medical representative</i> . Pada tampilan ini terdapat tabel yang menunjukkan daftar <i>medical representative</i> , serta terdapat button untuk menambah akun, edit akun, dan hapus akun <i>medical representative</i> .
Gambar	

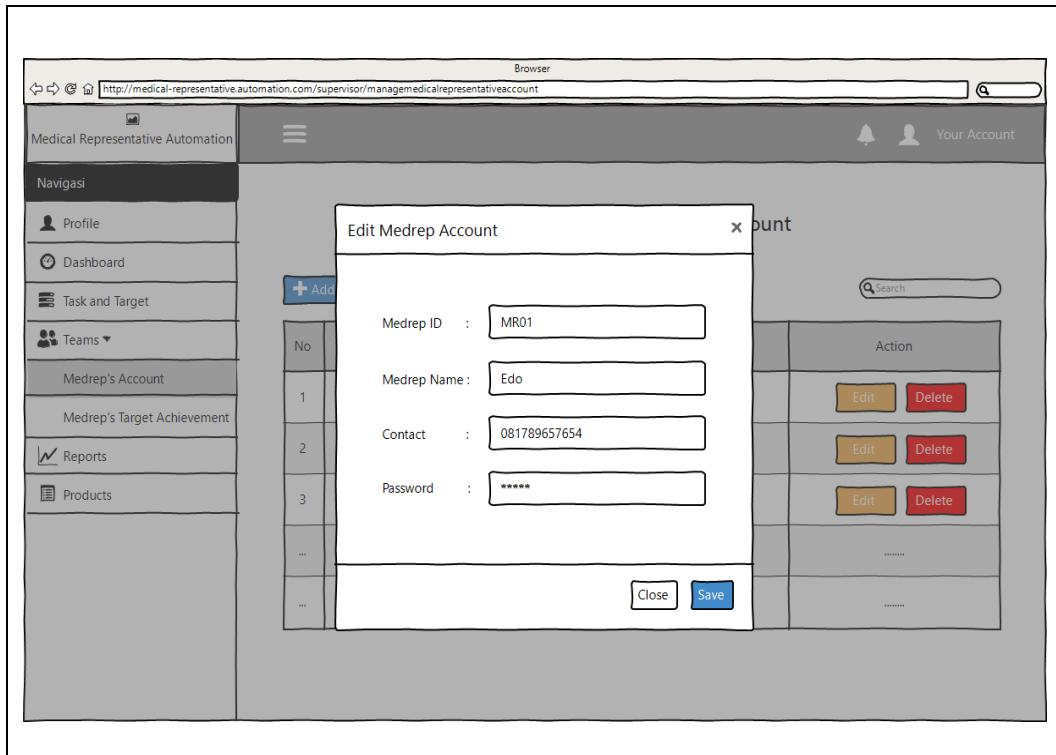
Tabel 164 Tampilan Form Add *Medical Representative Account*

No. UI	GUI-MMR.02
Nama UI	Tampilan Form Add <i>Medical Representative Account</i> .
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Medical Representative Account</i>. - Pengguna telah memilih aksi “Add <i>Medical Representative</i>”

Deskripsi	Tampilan ini merupakan <i>form</i> untuk menambah data akun <i>medical representative</i> yang terdiri dari komponen <i>modal, button, input field</i> sesuai data yang dibutuhkan untuk akun <i>medical representative</i> .
Gambar	

Tabel 165 Tampilan *Form Edit Medical Representative Account*

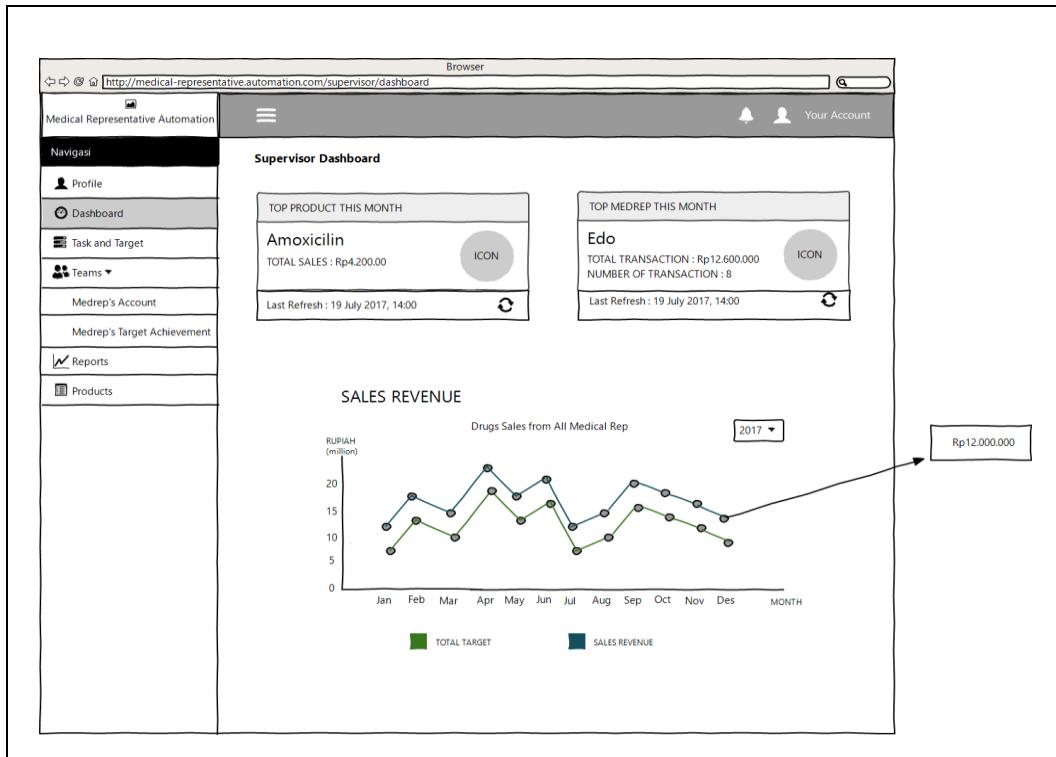
No. UI	GUI-MMR.03
Nama UI	Tampilan <i>Form Edit Medical Representative Account</i> .
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Medical Representative Account</i>. - Pengguna telah memilih aksi “<i>Edit Medical Representative</i>”
Deskripsi	Tampilan ini merupakan <i>form</i> untuk mengedit data akun <i>medical representative</i> yang terdiri dari komponen <i>modal, button, input field</i> sesuai data yang dibutuhkan untuk akun <i>medical representative</i> .
Gambar	



6. UI – Supervisor Dashboard

Tabel 166 Tampilan *Supervisor Dashboard*

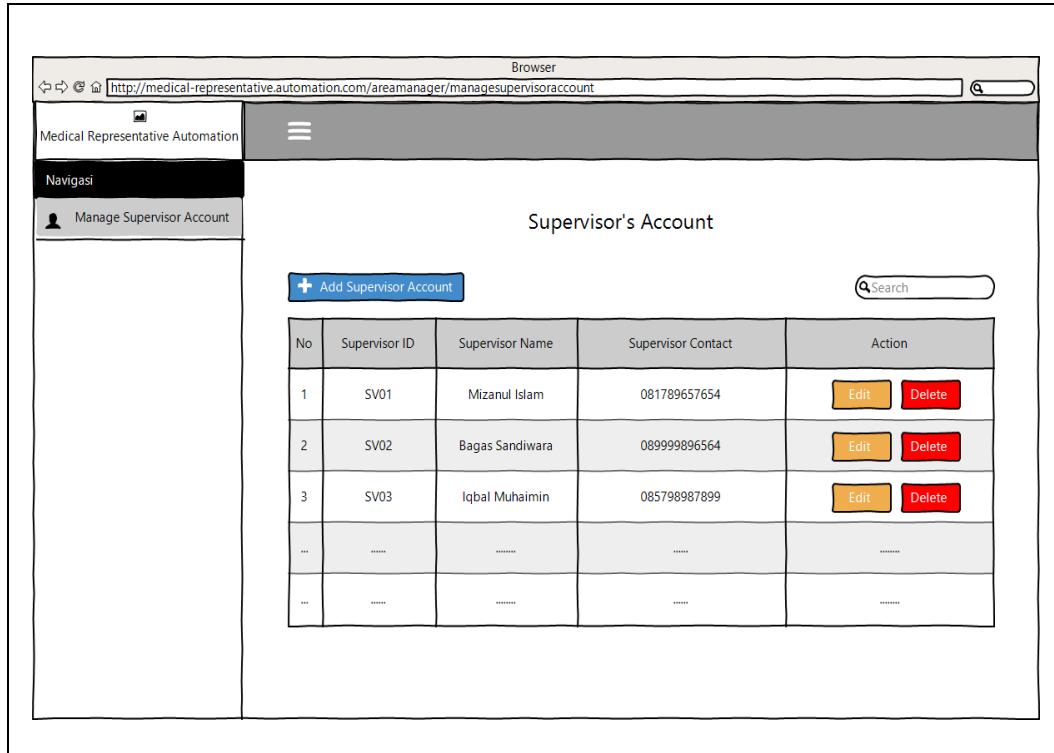
No. UI	GUI-DB.01
Nama UI	Tampilan <i>Supervisor Dashboard</i>
Pengguna	Supervisor
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Dashboard</i>.
Deskripsi	<p>Tampilan ini merupakan halaman <i>dashboard</i> yang menampilkan informasi:</p> <ol style="list-style-type: none"> 1. <i>Top product</i> : berdasarkan jumlah transaksi <i>product</i> selama satu periode terakhir, dengan menampilkan salah satu produk yang jumlah transaksinya paling besar. Dilengkapi tombol <i>refresh</i>, setiap kali supervisor ingin mengetahui <i>top product</i> terbaru. 2. <i>Top medical rep</i> : berdasarkan rasio pencapaian target selama satu periode terakhir, dengan menampilkan salah satu <i>medical rep</i> yang rasio pencapaian targetnya paling besar. Dilengkapi tombol <i>refresh</i>, setiap kali supervisor ingin mengetahui <i>top medical rep</i> terbaru. 3. Grafik penjualan obat dari seluruh <i>medical rep</i> : menampilkan jumlah penjualan dari seluruh <i>medical rep</i> setiap bulan dan disajikan dalam bentuk grafik selama setahun. Terdapat <i>dropdown</i> untuk melihat data penjualan dari tahun yang dipilih, maximal lima tahun yang lalu. Apabila di-hover pada setiap titik akan muncul informasi rupiah yang dicapai.
Gambar	



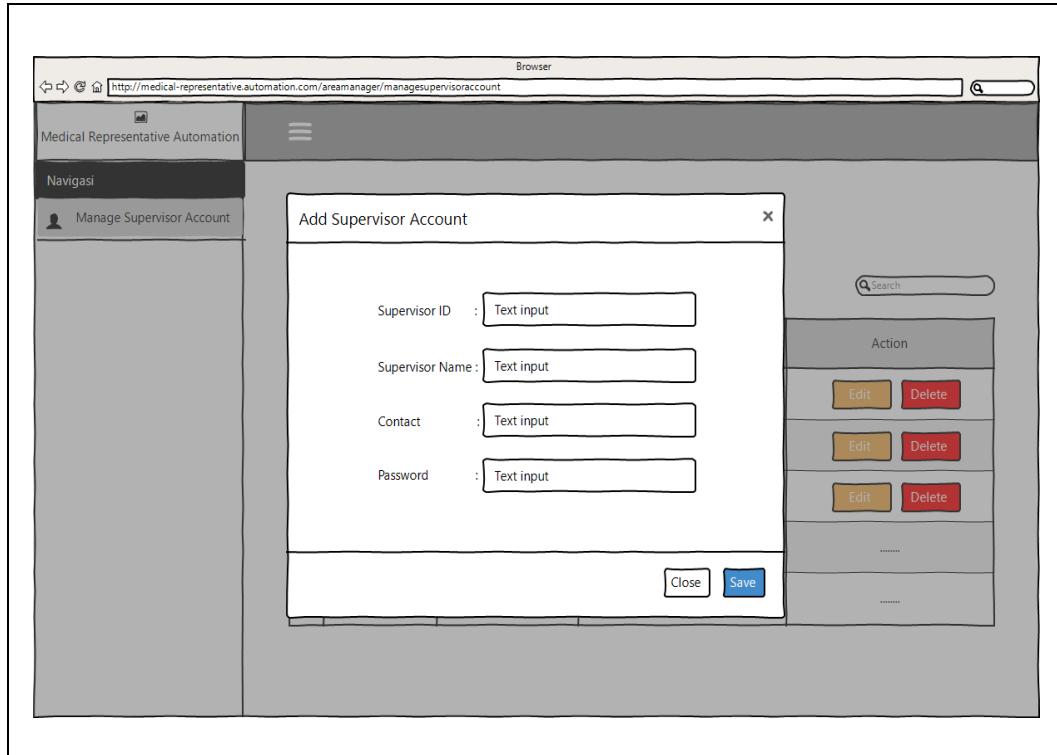
7. UI – Manage Supervisor Account

Tabel 167 Tampilan Daftar Data Akun Supervisor

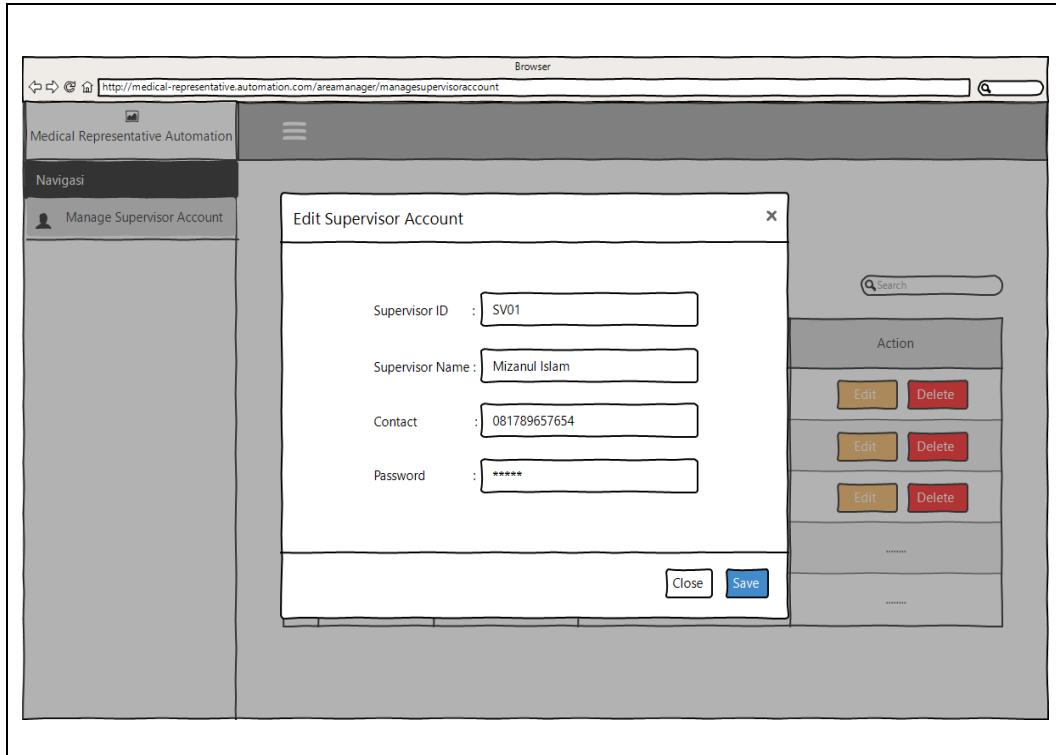
No. UI	GUI-MSV.01
Nama UI	Tampilan Daftar Data Akun Supervisor.
Pengguna	Area Sales Manager
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Supervisor Account</i>.
Deskripsi	Tampilan ini merupakan halaman untuk menampilkan semua data akun supervisor. Pada tampilan ini terdapat tabel yang menunjukkan daftar supervisor, serta terdapat <i>button</i> untuk menambah akun, edit akun, dan hapus akun supervisor.
Gambar	

Tabel 168 Tampilan *Form Add Supervisor Account*

No. UI	GUI-MSV.02
Nama UI	Tampilan <i>Form Add Supervisor Account</i> .
Pengguna	<i>Area Sales Manager</i>
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Supervisor Account</i>. - Pengguna telah memilih aksi “Add Supervisor”
Deskripsi	Tampilan ini merupakan <i>form</i> untuk menambah data akun supervisor yang terdiri dari komponen <i>modal, button, input field</i> sesuai data yang dibutuhkan untuk akun <i>medical representative</i> .
Gambar	

Tabel 169 Tampilan *Form Edit Supervisor Account*

No. UI	GUI-MSV.03
Nama UI	Tampilan <i>Form Edit Supervisor Account</i> .
Pengguna	Area Sales Manager
Precondition	<ul style="list-style-type: none"> - Pengguna telah masuk ke dalam aplikasi. - Pengguna telah memilih menu <i>Manage Supervisor Account</i>. - Pengguna telah memilih aksi “Edit Supervisor”
Deskripsi	Tampilan ini merupakan <i>form</i> untuk mengedit data akun supervisor yang terdiri dari komponen <i>modal, button, input field</i> sesuai data yang dibutuhkan untuk akun supervisor.
Gambar	

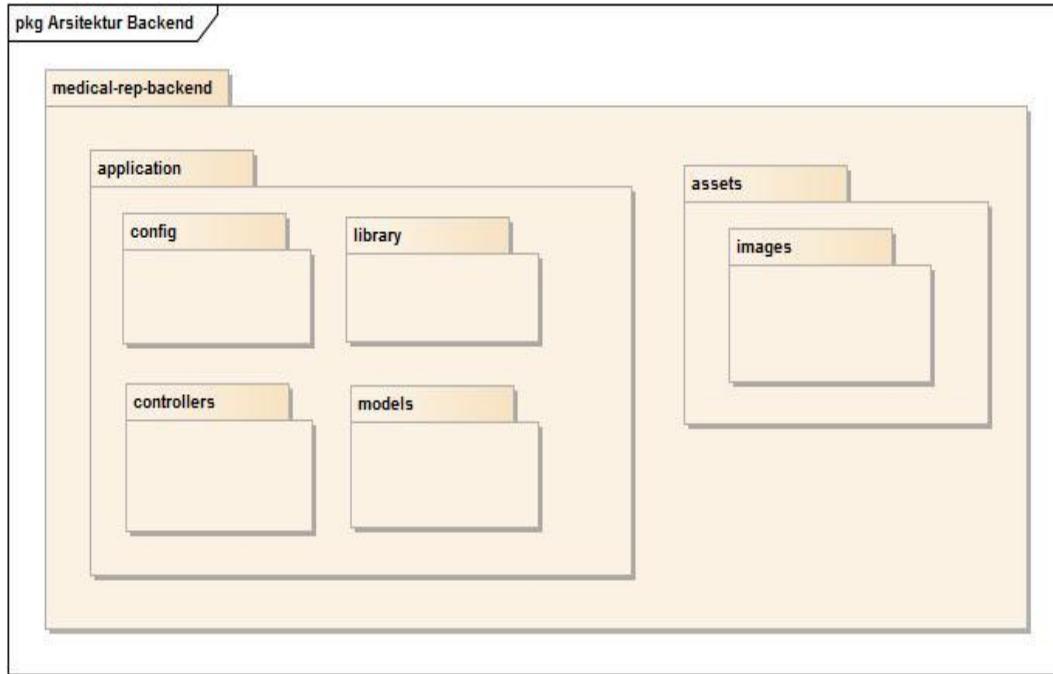


4.3 Perancangan Backend

Sub bab ini dijelaskan mengenai perancangan *backend* yang akan dibangun meliputi: perancangan arsitektur, *class*, interaksi antar objek yang digambarkan dengan *sequence diagram* dan *user interface*.

4.3.1 Perancangan Arsitektur

Perancangan ini dilakukan untuk memberikan gambaran mengenai arsitektur *backend* pada pengembangan aplikasi CRM yang digambarkan pada Gambar 111.

Gambar 111 Arsitektur *Backend*

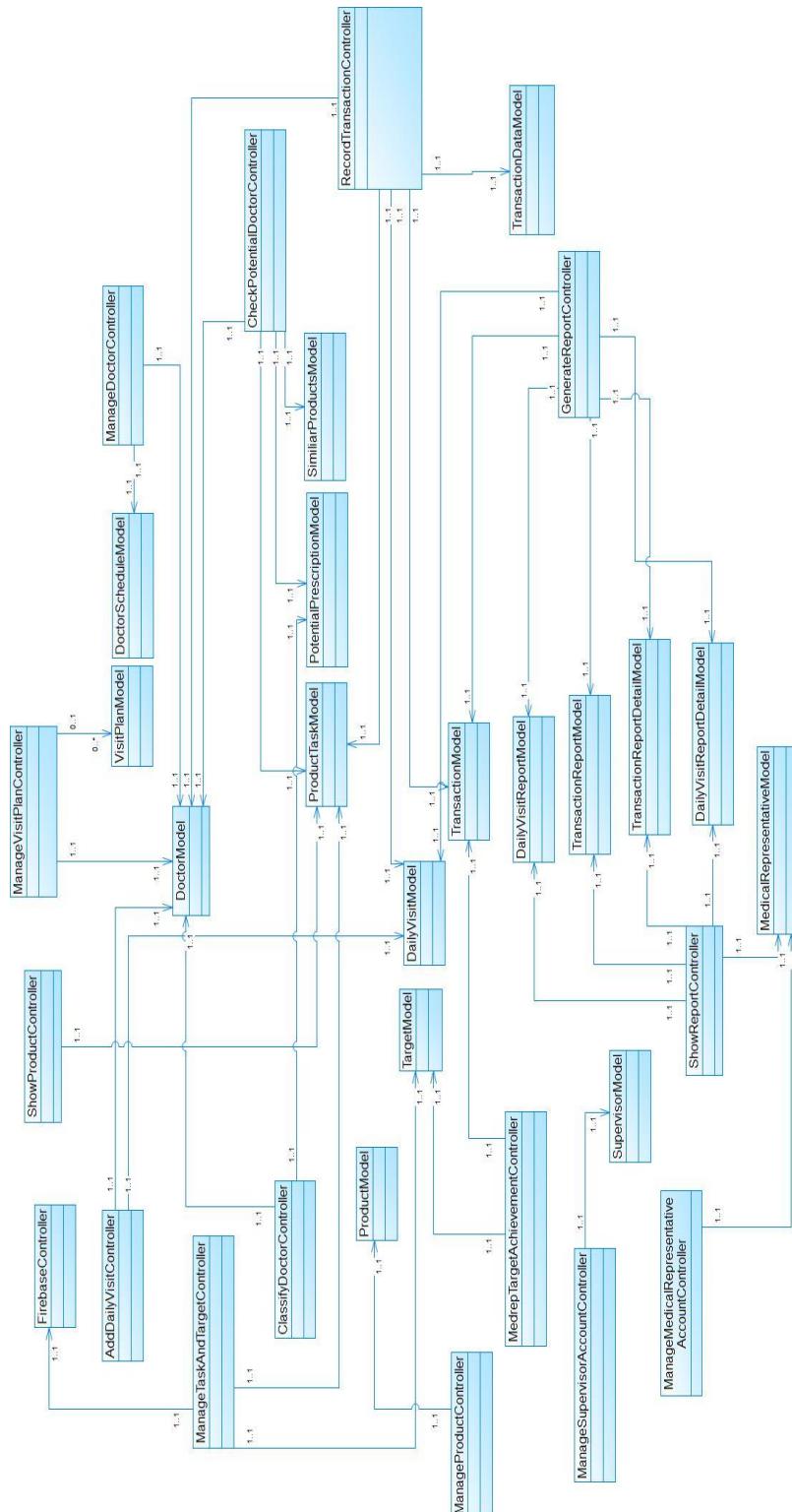
Keterangan *package diagram* pada *backend* akan dijelaskan pada Tabel 170

Tabel 170 Penjelasan Arsitektur *Backend*

Nama Package	Deskripsi
assets	Berisi file-file pendukung seperti images
images	Berisi file-file images untuk kebutuhan aplikasi
application	Berisi file-file buatan yang merupakan bagian inti pada aplikasi
config	Berisi file-file untuk konfigurasi database dan library yang akan digunakan
library	Berisi file-file library pendukung aplikasi
controllers	Berisi file-file controller untuk mengatur alur kerja sistem
models	Berisi file-file model untuk mengolah data dari database

4.3.2 Perancangan Class

Berikut perancangan *class* untuk pengembangan aplikasi CRM pada Gambar 112 untuk bagian *backend*.



Gambar 112 Class Diagram *Backend*

Nama Class	AddDailyVisitController		
Deskripsi	<i>Class AddDailyVisitController berfungsi untuk mengelola data kunjungan harian Medical Representative.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorModel	Private	DoctorModel	-
dailyVisitModel	Private	DailyVisitModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
doctor_for_visit_by_medrep_get()	Public	Void	Method untuk mengambil data dokter sesuai dengan id <i>medical rep.</i>
add_daily_visit_post()	Public	Void	Method untuk menambah data kunjungan harian.

Nama Class	CheckPotentialDoctorController		
Deskripsi	<i>Class CheckPotentialDoctorController berfungsi untuk mengelola data potensi peresepan dokter.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productTaskModel	Private	ProductTaskModel	-
potentialPrescriptionModel	Private	PotentialPrescriptionModel	-
similiarProductsModel	Private	SimiliarProductsModel	-
doctorModel	Private	DoctorModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
product_task_medrep_get()	Public	Void	Method untuk mengambil data produk sesuai tugas <i>medical rep.</i>
add_potential_prescription_post()	Public	Void	Method untuk menambahkan data potensi peresepan dokter.
get_new_potential_prescription_id()	Public	String	Method untuk membuat id potensi peresepan dokter yang baru.

Nama Class	GenerateReportController		
Deskripsi	<i>Class GenerateReportController berfungsi untuk mengelola laporan medical representative.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
dailyVisitModel	Private	DailyVisitModel	-
transactionModel	Private	TransactionModel	-
dailyVisitReportModel	Private	DailyVisitReportModel	-
transactionReportModel	Private	TransactionReportModel	-
dailyVisitReportDetailModel	Private	DailyVisitReportDetailModel	-
transactionReportDetailModel	Private	TransactionReportDetailModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
generate_report_post()	Public	Void	Method untuk membuat laporan <i>medical rep</i> .
get_new_daily_visit_report_id()	Public	String	Method untuk membuat id laporan kunjungan harian <i>medical rep</i> yang baru.
get_new_transaction_report_id()	Public	String	Method untuk membuat id transaksi <i>medical rep</i> yang baru.

Nama Class	ManageDoctorController		
Deskripsi	<i>Class ManageDoctorController berfungsi untuk mengelola data dokter.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorModel	Private	DoctorModel	-
doctorScheduleModel	Private	DoctorScheduleModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
doctor_by_medrep_and_status_get()	Public	Void	Method untuk mengambil data dokter berdasarkan id <i>medical rep</i> dan status dokter
set_doctor_with_schedule(listDoctorData)	Public	Array	Method untuk menambahka

			n jadwal dokter pada data daftar dokter.
add_lead_doctor_post()	Public	Void	Method untuk menambahkan data dokter dengan status “Lead”.
get_new_id()	Public	String	Method untuk membuat id doctor yang baru.
update_doctor_data_post()	Public	Void	Method untuk mengubah data dokter.
delete_doctor_id()	Public	Void	Method untuk menghapus data dokter.

Nama Class	ManageTaskAndTargetController		
Deskripsi	<i>Class ManageTaskAndTargetController berfungsi untuk mengelola data tugas dan target Medical Representative.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
targetModel	Private	TargetModel	-
productTaskModel	Private	ProductTaskModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
add_task_and_target_post()	Public	Void	Method untuk menambahkan tugas dan target <i>medical rep.</i>

Nama Class	ManageVisitPlanController		
Deskripsi	<i>Class ManageVisitPlanController berfungsi untuk mengelola data rencana kunjungan Medical Representative.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
visitPlanModel	Private	VisitPlanModel	-
doctorModel	Private	DoctorModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
visit_plan_by_medrep_get()	Public	Void	Method untuk mengambil data

			<i>visit plan</i> atau rencana kunjungan.
add_visit_plan_post()	Public	Void	Method untuk menambahkan data rencana kunjungan.
all_doctor_by_medrep_and_practice_day_get()	Public	Void	Method untuk mengambil data dokter berdasarkan id <i>medical rep</i> dan hari praktek dokter.
update_visit_plan_post()	Public	Void	Method untuk mengubah data rencana kunjungan.

Nama Class	MedrepTargetAchievementController		
Deskripsi	<i>Class MedrepTargetAchievementController</i> berfungsi untuk mengelola data pencapaian target <i>Medical Representative</i> .		
Attribut			
Nama	Visibility	Type Data	Deskripsi
targetModel	Private	TargetModel	-
transactionModel	Private	TransactionModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
medrep_target_achievement_get()	Public	Void	Method untuk mengambil data pencapaian target <i>medical rep</i> .
calculate_medrep_target_achievement(Array listMedrepAndTarget, Array listMedrepAndTransaction)	Public	Array	Method untuk menghitung pencapaian target <i>medical rep</i> .

Nama Class	ShowProductController		
Deskripsi	<i>Class ShowProductController</i> berfungsi untuk mengelola data produk yang ditugaskan kepada <i>Medical Representative</i> .		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productTaskModel	Private	ProductTaskModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
product_task_medrep_get()	Public	Void	Method untuk mengambil data produk yang ditugaskan kepada <i>medical rep</i> .

Nama Class	ShowReportController		
Deskripsi	<i>Class ShowReportController berfungsi untuk mengelola data laporan yang dibuat oleh <i>Medical Representative</i>.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
dailyVisitReportModel	Private	DailyVisitReportModel	-
dailyVisitReportDetailModel	Private	DailyVisitReportDetailModel	-
transactionReportModel	Private	TransactionReportModel	-
transactionReportDetailModel	Private	TransactionReportDetailModel	-
medicalRepresentativeModel	Private	MedicalRepresentativeModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
all_report_by_type_get()	Public	Void	Method untuk mengambil data laporan berdasarkan jenis laporan.
report_by_id_get()	Public	Void	Method untuk mengambil data laporan sesuai dengan id laporan.

Nama Class	ClassifyDoctorController		
Deskripsi	<i>Class ClassifyDoctorController berfungsi untuk mengelola klasifikasi dokter.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
doctorModel	Private	DoctorModel	-
prescriptionTargetModel	Private	PrescriptionTargetModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
classify_doctor()	Public	Void	Method untuk mengklasifikasikan dokte.

Nama Class	RecordTransactionController		
Deskripsi	<i>Class RecordTransactionController berfungsi untuk mengelola data transaksi.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
transactionModel	Private	TransactionModel	-

transactionDataModel	Private	TransactionDataModel	-
doctorModel	Private	DoctorModel	-
productTaskModel	Private	ProductTaskModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
add_transaction_post()	Public	Void	Method untuk menambahkan transaksi
get_new_transaction_id()	Public	String	Method untuk membuat id transaksi yang baru
product_task_medrep_get()	Public	Void	Method untuk mengambil data produk yang ditugaskan kepada <i>medical rep</i>

Nama Class	ManageMedicalRepresentativeAccountController		
Deskripsi	<i>Class ini berfungsi untuk mengelola data akun medical representative.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
medicalRepresentativeModel	Private	Medical RepresentativeModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
list_medical_representative_get()	Public	Void	Method untuk mengambil seluruh data akun <i>medical representative</i> .
add_medical_representative_account_post()	Public	Void	Method untuk menambahkan data akun <i>medical representative</i> .
update_medical_representative_account_post()	Public	Void	Method untuk mengubah data akun <i>medical representative</i> .
delete_medical_representative_account_post()	Public	Void	Method untuk menghapus data akun <i>medical representative</i> .

Nama Class	ManageSupervisorAccountController		
Deskripsi	<i>Class ini berfungsi untuk mengelola data akun supervisor.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
supervisorModel	Private	SupervisorModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
list_supervisor_get()	Public	Void	Method untuk mengambil seluruh data akun supervisor.
add_supervisor_account_post()	Public	Void	Method untuk menambahkan data akun supervisor.
update_supervisor_account_post()	Public	Void	Method untuk mengubah data akun supervisor.
delete_supervisor_account_post()	Public	Void	Method untuk menghapus data akun supervisor.

Nama Class	ManageProductController		
Deskripsi	<i>Class ini berfungsi untuk mengelola data product.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productModel	Private	ProductModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
list_product_get()	Public	Void	Method untuk mengambil seluruh data product.
add_product_post()	Public	Void	Method untuk menambahkan data product.
update_product_post()	Public	Void	Method untuk mengubah data product.
delete_product_account_post()	Public	Void	Method untuk menghapus data product.

Nama Class	FirebaseController		
Deskripsi	<i>Class ini berfungsi untuk mengelola notifikasi yang berhubungan dengan Firebase Cloud Messaging.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
-	-	-	-
Method			
Nama	Visibility	Return Value	Deskripsi
send_notification_post()	Public	Void	Method untuk menyimpan data yang diperlukan untuk <i>push notification</i> dan menampilkan informasi status pengiriman <i>notification</i> .
save_token_post()	Public	Void	Method untuk menyimpan data token <i>medical representative</i> ke <i>database</i> .
sendNotificationToFcm(String token, String title, String message)	Public	JSON	Method untuk mengirimkan data ke sistem <i>Firebase Cloud Messaging</i> untuk keperluan <i>push notification</i> .

Nama Class	SupervisorModel		
Deskripsi	<i>Class ini berfungsi untuk mengelola data supervisor dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productModel	Private	ProductModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
get_all_supervisor()	Public	Array	Method untuk mengambil seluruh data supervisor dari <i>database</i> .
insert_supervisor_account(Array supervisorAccountData)	Public	Boolean	Method untuk menambahkan data supervisor dari ke <i>database</i> .
update_supervisor_account(String supervisorId, Array supervisorAccountData)	Public	Boolean	Method untuk mengubah data

			supervisor pada database.
delete_supervisor_account(String supervisorId)	Public	Boolean	Method untuk menghapus data supervisor pada database.

Nama Class	ProductModel		
Deskripsi	<i>Class ini berfungsi untuk mengelola data product dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
productModel	Private	ProductModel	-
Method			
Nama	Visibility	Return Value	Deskripsi
get_all_product(String supervisorId)	Public	Array	Method untuk mengambil seluruh data product di database.
insert_product(Array productData)	Public	Boolean	Method untuk menambahkan data product ke database.
update_product(String productId, Array productData)	Public	Boolean	Method untuk mengubah data product pada database.
delete_product(String productId)	Public	Boolean	Method untuk menghapus data product pada database.

Nama Class	DailyVisitModel		
Deskripsi	<i>Class DailyVisitModel berfungsi untuk mengelola data kunjungan harian dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_daily_visit(String medrepId, String doctorId, String visitResult)	Public	Boolean	Method untuk menambahkan data kunjungan harian ke database.
get_visit_result(String startDate, String endDate, String medrepId)	Public	Array	Method untuk mengambil daftar hasil kunjungan sesuai periode awal dan akhir yang sudah ditentukan dari database.

get_detail_daily_visit_data(String reportDetailId)	Public	Array	Method untuk mengambil data kunjungan untuk keperluan laporan dari <i>database</i> .
--	--------	-------	---

Nama Class	TargetModel		
Deskripsi	<i>Class TargetModel berfungsi untuk mengelola data target Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
get_medrep_target_by_period(String medrepId, int monthOfPeriod)	Public	Array	Method untuk mengambil target <i>medical rep</i> berdasarkan bulan dari <i>database</i> .
insert_target_medrep(String medrepId, int medrepTarget, String note, Date startDateTargetPeriod, Date endDateTargetPeriod)	Public	Array	Method untuk menambahkan target <i>medical rep</i> ke <i>database</i> .
get_medrep_target(String medrepId, String targetId)	Public	Array	Method untuk mengambil target medreo berdasarkan id target dari <i>database</i>
check_target_assigned(String medrepId, Date startDateTargetPeriod, Date endDateTargetPeriod)	Public	Boolean	Method untuk pengecekan ke <i>database</i> apakah target telah diberikan kepada <i>medical rep</i> .

Nama Class	ProductTaskModel		
Deskripsi	<i>Class ProductTaskModel berfungsi untuk mengelola data product yang merupakan tugas Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
get_product_task_medrep (String medrepId, String targetId)	Public	Array	Method untuk mengambil data produk yang ditugaskan kepada <i>medical rep</i> dari <i>database</i> .
insert_product_task(String medrepId, String productId)	Public	Void	Method untuk menambahkan produk

			yang merupakan tugas <i>medical rep</i> ke <i>database</i> .
--	--	--	---

Nama Class	DoctorModel		
Deskripsi	<i>Class DoctorModel berfungsi untuk mengelola data dokter dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
get_doctor_for_visit_by_medrep(String medrepId)	Public	Array	Method untuk mengambil data dokter untuk kunjungan harian berdasarkan id <i>medical rep</i> dari <i>database</i> .
get_doctor_by_medrep_and_status(String medrep_id, String doctor_status)	Public	Array	Method untuk mengambil data dokter berdasarkan id <i>medical rep</i> dan status dokter dari <i>database</i> .
insert_lead_doctor(String doctorId, String doctorSpecialist, String doctorStatus, int deleted, String medrepId)	Public	Boolean	Method untuk menyimpan data <i>lead</i> dokter ke <i>database</i> .
update_doctor_data(String doctorId, String doctorName, String doctorSpecialist, String doctorStatus, String medrepId)	Public	Boolean	Method untuk mengubah data dokter di <i>database</i> .
delete_doctor_data(String doctorId)	Public	Boolean	Method untuk menghapus data dokter di <i>database</i> .
get_doctor_by_medrep_and_practice_day(String medrepId, String practiceDay)	Public	Array	Method untuk mengambil data dokter berdasarkan jadwal praktek dari <i>database</i> .
update_doctor_status(doctorId, String status)	Public	Void	Method untuk mengubah status doctor di <i>database</i> .
get_doctor_by_id(String doctorId)	Public	Array	Method untuk mengambil data dokter berdasarkan id dokter dari <i>database</i> .
get_practice_time(String doctorId, String practiceDay)	Public	Array	Method untuk mengambil data waktu praktek dokter dari <i>database</i> .

Nama Class	DoctorScheduleModel		
Deskripsi	<i>Class DoctorScheduleModel berfungsi untuk mengelola data jadwal dokter dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi

Method			
Nama	Visibility	Return Value	Deskripsi
insert_doctor_schedule(Array doctorSchedule)	Public	Void	Method untuk menambahkan data jadwal dokter ke <i>database</i> .
delete_doctor_schedule(String doctorId)	Public	Void	Method untuk menghapus data jadwal dokter berdasarkan id dokter di <i>database</i> .

Nama Class	PotentialPrescriptionModel		
Deskripsi	<i>Class</i> PotentialPrescriptionModel berfungsi untuk mengelola data potensi peresepan dokter dari/ ke <i>database</i> .		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_potential_prescription (String potentialPrescriptionId, String doctorId, int numberOfPractices, int numberOfPatients, int totalPotentialPr)	Public	Boolean	Method untuk menyimpan data potensi peresepan dokter ke <i>database</i> .
get_total_prescription(String doctorId)	Public	List potential prescription	Method untuk mengambil data potensi peresepan dokter

Nama Class	SimiliarProductsModel		
Deskripsi	<i>Class</i> SimiliarProductsModel berfungsi untuk mengelola data produk sejenis pada potensi peresepan dokter dari/ke <i>database</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_similiar_products(String potentialPrescriptionId, String productId, int quantity, int productPrice, String competitor)	Public	Void	Method untuk menambahkan produk sejenis yang dokter resepkan pada potensi peresepan ke <i>database</i> .

Nama Class	TransactionModel		
Deskripsi	<i>Class TransactionModel berfungsi untuk mengelola data transaksi dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_transaction (String transactionId, String medrepId, String doctorId, int period)	Public	Boolean	Method untuk menambahkan data transaksi ke <i>database</i> .
get_medrep_transaction (String medrepId, int monthOfPeriod)	Public	Array	Method untuk mengambil data transaksi berdasarkan bulan dari <i>database</i> .
get_transaction_data(String startDate, String endDate, String medrepId)	Public	Array	Method untuk mengambil daftar transaksi <i>medical rep</i> dari <i>database</i> , berdasarkan periode awal dan akhir yang sudah ditentukan.
get_detail_transaction_data(String reportDetailId)	Public	Array	Method untuk mengambil data transaksi untuk keperluan laporan dari <i>database</i> .

Nama Class	TransactionDataModel		
Deskripsi	<i>Class TransactionDataModel berfungsi untuk mengelola data detail transaksi dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_transaction_data (String transactionId, String productId, int prescriptionTarget)	Public	Void	Method untuk menambahkan data detail transaksi ke <i>database</i>

Nama Class	VisitPlanModel		
Deskripsi	<i>Class ini berfungsi untuk mengelola data rencana kunjungan Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
get_visit_plan_by_medrep (String medrepId)	Public	Array	Method untuk mengambil data rencana kunjungan <i>medical rep</i> dari <i>database</i> .
save_visit_plan(String doctorId, String medrepId, String visitDate, String reminderVisirTime, String note)	Public	Boolean	Method untuk menyimpan rencana kunjungan <i>medical rep</i> ke <i>database</i> .
update_visit_plan(String visitPlanId, String doctorId, String medrepId, String visitDate, String reminderVisirTime, String note)	Public	Boolean	Method untuk mengubah rencana kunjungan <i>medical rep</i> di <i>database</i> .

Nama Class	MedicalRepresentativeModel		
Deskripsi	<i>Class ini berfungsi untuk mengelola data Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
get_medrep_task_and_target()	Public	Array	Method untuk mengambil daftar <i>medical rep</i> dari <i>database</i> .
get_list_medrep_for_show_report()	Public	Array	Method untuk mengambil daftar <i>medical rep</i> untuk keperluan <i>show report</i> dari <i>database</i> .
get_all_medrep_by_supervisor(String supervisorId)	Public	Array	Method untuk mengambil seluruh data <i>medical rep</i> dari <i>database</i> .
insert_medicalRepresentative_account(Array medrepAccountData)	Public	Boolean	Method untuk menambahkan data <i>medical rep</i> dari ke <i>database</i> .
update_medicalRepresentative_account(String medrepId, Array medrepAccountData)	Public	Boolean	Method untuk mengubah data <i>medical rep</i> pada <i>database</i> .

<code>delete_medicalRepresentative _account(String medrepId)</code>	Public	Boolean	Method untuk menghapus data <i>medical representative</i> pada <i>database</i> .
---	--------	---------	--

Nama Class			
Deskripsi			
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
<code>insert_daily_visit_report(String dailyVisitReportId, String startDate, String endDate, String medrepId)</code>	Public	Void	Method untuk menambahkan laporan kunjungan harian <i>medical rep</i> ke <i>database</i> .
<code>get_all_daily_visit_report_ by_medrep(String medrepId)</code>	Public	Void	Method untuk mengambil seluruh laporan kunjungan dari <i>database</i> berdasarkan <i>medrepId</i>

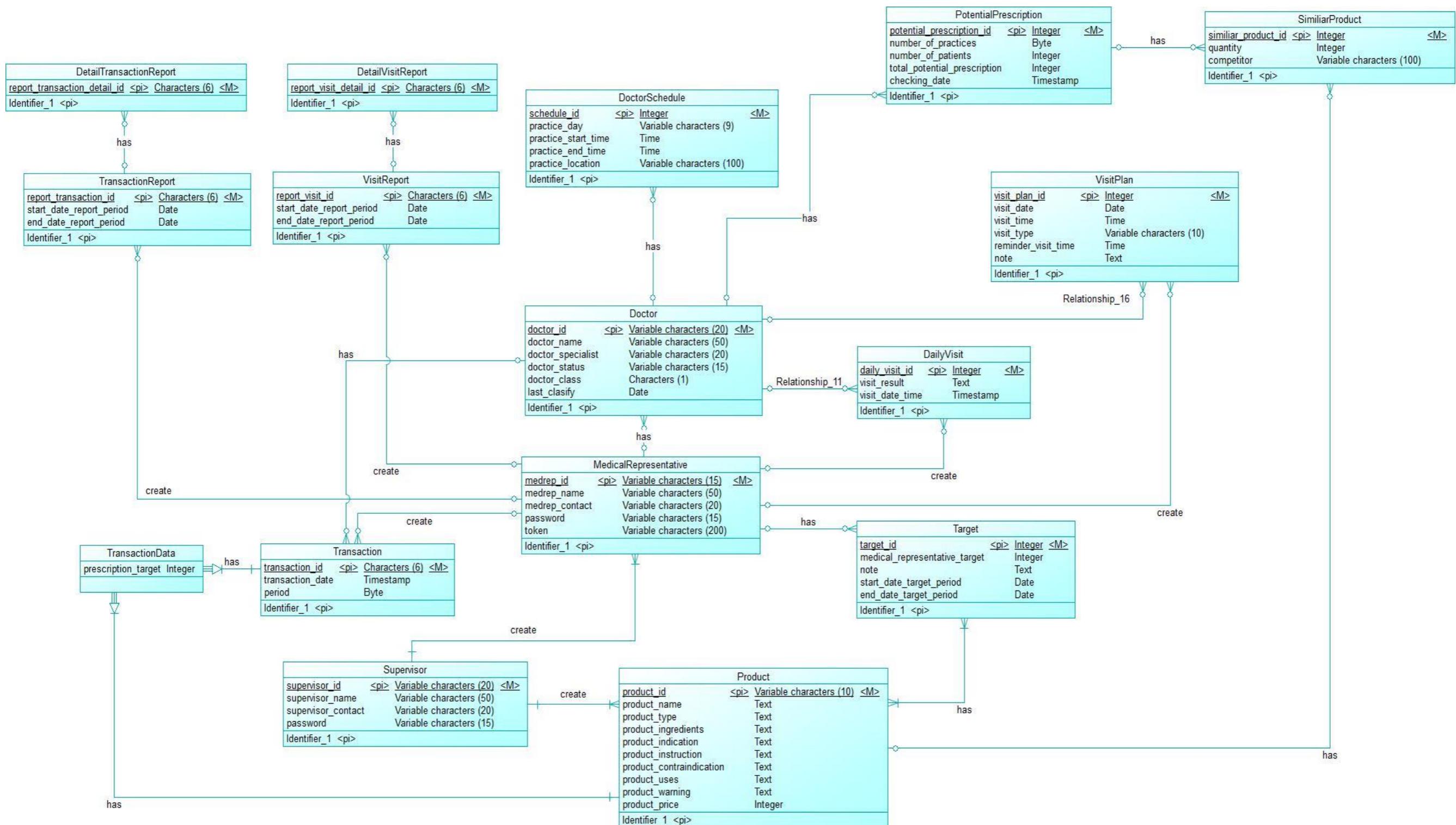
Nama Class			
Deskripsi			
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
<code>insert_daily_visit_report_ detail(String dailyVisitReportId, String visitResult)</code>	Public	Void	Method untuk menambahkan detail dari laporan kunjungan <i>medical rep</i> ke <i>database</i> .
<code>count_visit_result_by_report _id(String reportId)</code>	Public	Void	Method untuk menghitung jumlah kunjungan pada laporan kunjungan.
<code>get_visit_report_detail_by_ report_id(String reportId)</code>	Public	Array	Method untuk mengambil seluruh detail laporan kunjungan <i>medical rep</i> dari <i>database</i> , berdasarkan id laporan.

Nama Class	TransactionReportModel		
Deskripsi	<i>Class TransactionReportModel berfungsi untuk mengelola laporan transaksi Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_transaction_report(String transactionReportId, String startDate, String endDate, String medrepId)	Public	Void	Method untuk menambahkan laporan transaksi <i>medical rep</i> ke <i>database</i> .
get_all_transaction_report_ by_medrep(String medrepId)	Public	Void	Method untuk mengambil seluruh laporan transaksi dari <i>database</i> berdasarkan medrepId

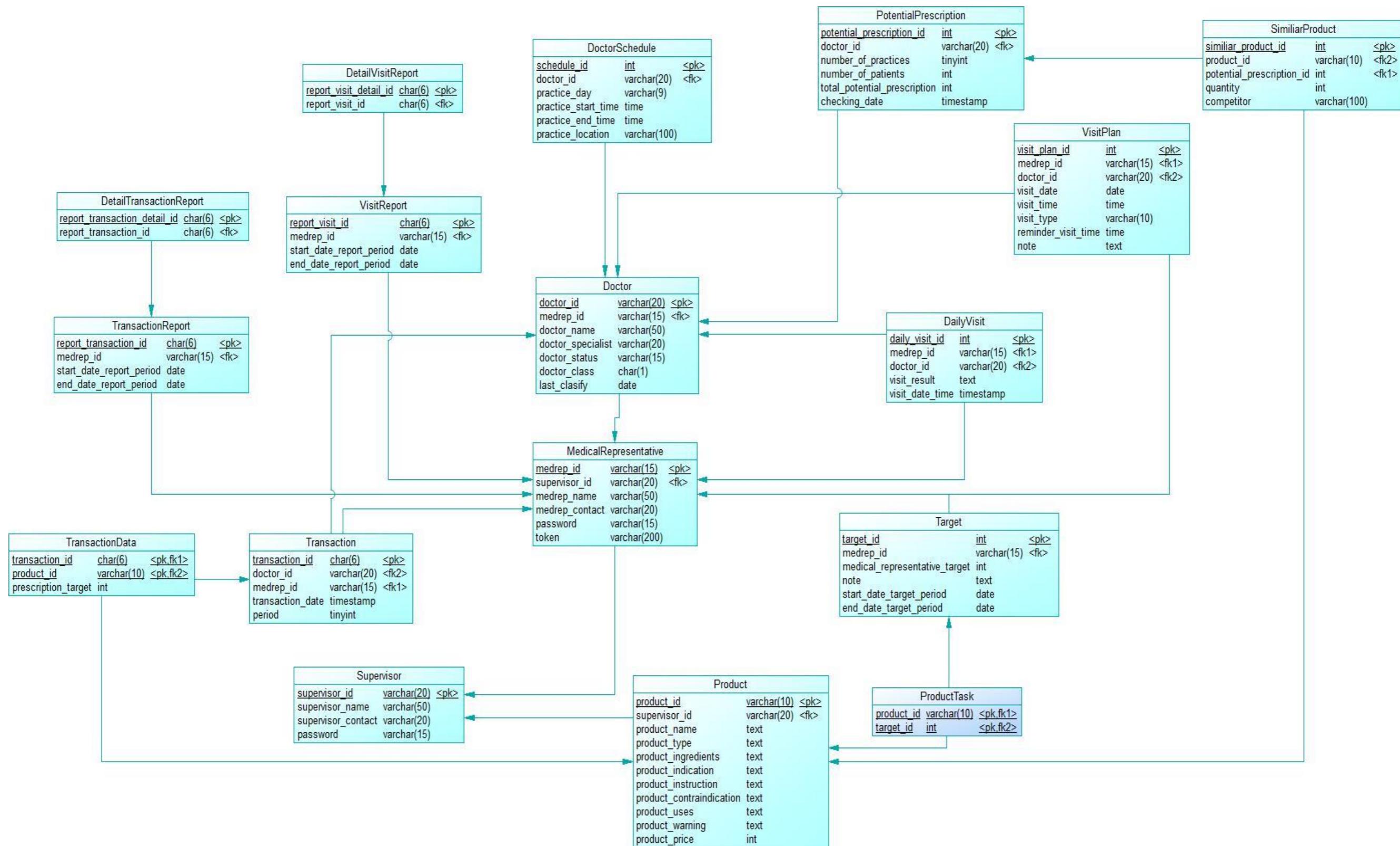
Nama Class	TransactionReportDetailModel		
Deskripsi	<i>Class TransactionReportDetailModel berfungsi untuk mengelola detail laporan transaksi Medical Representative dari/ ke database.</i>		
Attribut			
Nama	Visibility	Type Data	Deskripsi
Method			
Nama	Visibility	Return Value	Deskripsi
insert_transaction_report_ detail(String transactionReportId, String transactionData)	Public	Void	Method untuk menambahkan detail dari laporan transaksi <i>medical rep</i> ke <i>database</i> .
count_transaction_by_report _id(String reportId)	Public	Void	Method untuk menghitung jumlah transaksi pada laporan transaksi.
get_transaction_report_ detail_by_report_id(String reportId)	Public	Void	Method untuk mengambil seluruh detail laporan transaksi <i>medical rep</i> dari <i>database</i> , berdasarkan id laporan.

4.4 Perancangan *Database*

Perancangan *database* pada pengembangan aplikasi CRM bertujuan untuk memenuhi kebutuhan informasi dari penggunaan aplikasi dan menyediakan struktur data untuk mendukung kebutuhan proses. Gambar 113 dan Gambar 114 menggambarkan perancangan *database*.



Gambar 113 Conceptual Data Model Pengembangan Aplikasi CRM



Gambar 114 Physical Data Model Pengembangan Apikasi CRM

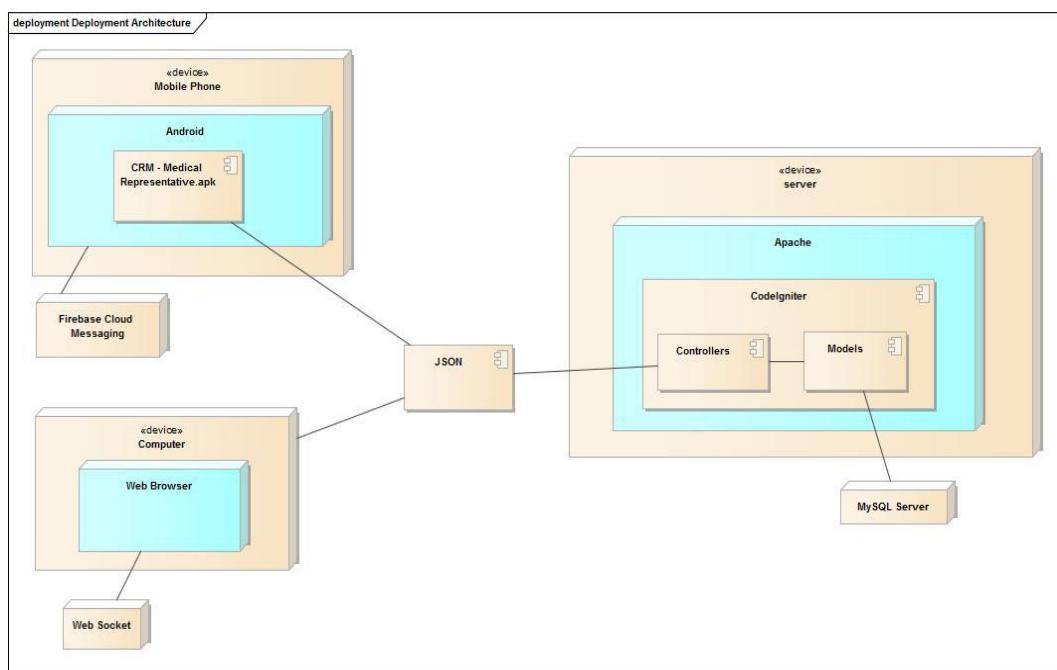
BAB V

IMPLEMENTASI

Bab ini menjelaskan tentang pembuatan aplikasi yang merupakan implementasi dari hasil analisis dan perancangan yang telah dibuat.

5.1 Model Deployment

Model *deployment* digambarkan dengan *deployment diagram* untuk mendeskripsikan fisik dari sistem, seperti komputer, perangkat keras/*device* dan hubungan dari hal-hal tersebut. Berikut gambaran mengenai model *deployment* pada pengembangan aplikasi CRM yang digambarkan dengan Gambar 115.



Gambar 115 Model *Deployment* Pengembangan Aplikasi CRM

Berdasarkan gambar 115, penjelasan untuk setiap node pada *deployment diagram* dijelaskan pada tabel 171.

Tabel 171 Penjelasan *Node*

Nama	Deskripsi
Apache	Merupakan <i>node</i> yang menyediakan service untuk menjalankan fungsi-fungsi PHP yang pada kali ini digunakan sebagai server <i>backend</i> aplikasi
MySQL Server	Merupakan <i>node</i> yang merepresentasikan <i>database management system</i> yang digunakan
Firebase Cloud Messaging	Merupakan <i>node</i> yang memberikan service untuk menampilkan notifikasi secara <i>online</i> pada aplikasi android
Web Socket	Merupakan <i>node</i> yang memberikan service untuk menampilkan notifikasi secara <i>online</i> pada aplikasi website
Android	Merupakan <i>node</i> yang merepresentasikan <i>mobile phone</i> yang akan digunakan.

Component diagram yang digambarkan pada *deployment diagram*, dijelaskan pada tabel 172.

Tabel 172 Penjelasan Komponen

Nama Komponen	Deskripsi
JSON	Merupakan komponen yang merepresentasikan bentuk data yang digunakan untuk saling bertukar informasi
Controllers	Merupakan komponen yang merepresentasikan pengatur alur data dari <i>backend</i> ke <i>frontend</i> android ataupun <i>frontend</i> website.
Models	Merupakan komponen yang merepresentasikan penyedia service untuk mengambil data dari <i>database</i>

5.2 Rancangan yang Diimplementasi

Berikut tabel yang menampilkan daftar *requirements* yang berhasil diimplementasikan berdasarkan *requirement* yang didapatkan pada tahap analisis.

Tabel 173 Rancangan *Frontend* yang Diimplementasi

Nomor <i>Requirement</i>	<i>Nama Functional Requirement</i>	Interaksi	Mockup	Hasil Implementasi	Status
[REQ-F-01]	Aplikasi menyediakan fitur untuk mengelola tugas dan target <i>medical representative</i>	SD-MTT-01	GUI-MTT-01 GUI-MTT-02 GUI-MTT-03	IM-MTT-01	Selesai
[REQ-F-02]	Aplikasi menyediakan fitur untuk menampilkan informasi pencapaian target <i>medical representative</i>	SD-CT-01	GUI-CT-01	IM-CT-01	Selesai
[REQ-F-03]	Aplikasi menyediakan fitur untuk menampilkan laporan yang dikirimkan oleh <i>medical representative</i>	SD-SR-01	GUI-SR-01 GUI-SR-02 GUI-SR-03 GUI-SR-04 GUI-SR-05	IM-SR-01 IM-SR-02 IM-SR-03 IM-SR-04 IM-SR-05	Selesai
[REQ-F-04]	Aplikasi menyediakan fitur untuk mengklasifikasikan dokter	SD-CD-01 SD-CD-02	GUI-CD.01	IM-CD-01	Selesai
[REQ-F-05]	Aplikasi menyediakan fitur untuk mengelola data dokter	SD-MD-02 SD-MD-03 SD-MD-04 SD-MD-05	GUI-MD.01 GUI-MD.02 GUI-MD.03 GUI-MD.04	IM-MDD-01 IM-MDD-02 IM-MDD-03 IM-MDD-04 IM-MDD-05 IM-MDD-06	Selesai
[REQ-F-06]	Aplikasi menyediakan fitur untuk mengkualifikasi dokter	SD-CPD-01	GUI-CP.01	IM-CP-01	Selesai
[REQ-F-07]	Aplikasi menyediakan fitur untuk mengelola rencana kunjungan <i>medical representative</i>	SD-MVP-01 SD-MVP-03 SD-MVP-06	GUI-VP.01 GUI-VP.02 GUI-VP.03	IM-MVP-01 IM-MVP-02 IM-MVP-03 IM-MVP-04	Selesai
[REQ-F-08]	Aplikasi menyediakan fitur untuk memasukkan informasi hasil kunjungan harian dengan dokter	SD-DV-01	GUI-DV.01 GUI-DV.02	IM-DV-01 IM-DV-02 IM-DV-03	Selesai
[REQ-F-09]	Aplikasi menyediakan fitur untuk memasukkan informasi hasil	SD-RT-01	GUI-RT.01	IM-RT-01	Selesai

	transaksi(kesepakatan) dengan dokter				
[REQ-F-10]	Aplikasi menyediakan fitur untuk meng- <i>generate</i> laporan <i>medical representative</i>	SD-GR-01	GUI-GR.01	IM-GR-01	Selesai
[REQ-F-11]	Aplikasi menyediakan fitur untuk menampilkan informasi product perusahaan.	SD-SP-01 SD-SP-02	GUI-SP.01 GUI-SP.02	IM-SP-01 IM-SP-02	Selesai
[REQ-F-12]	Aplikasi menyediakan fitur untuk mengelola produk perusahaan	SD-MP-01 SD-MP-02 SD-MP-03 SD-MP-04 SD-MP-05	GUI-MP.01 GUI-MP.02 GUI-MP.03 GUI-MP.04	IM-MP-01 IM-MP-02 IM-MP-03	Selesai
[REQ-F-13]	Aplikasi menyediakan fitur untuk mengelola akun <i>medical representative</i>	SD-MMR-01 SD-MMR-02 SD-MMR-03 SD-MMR-04 SD-MMR-05	GUI-MMR.01 GUI-MMR.02 GUI-MMR.03	IM-MMR-01 IM-MMR-02 IM-MMR-03	Selesai
[REQ-F-14]	Aplikasi menyediakan fitur untuk mengelola akun supervisor	SD-MSV-01 SD-MSV-02 SD-MSV-03 SD-MSV-04 SD-MSV-05	GUI-MSV.01 GUI-MSV.02 GUI-MSV.03	IM-MSV-01 IM-MSV-02 IM-MSV-03	Selesai

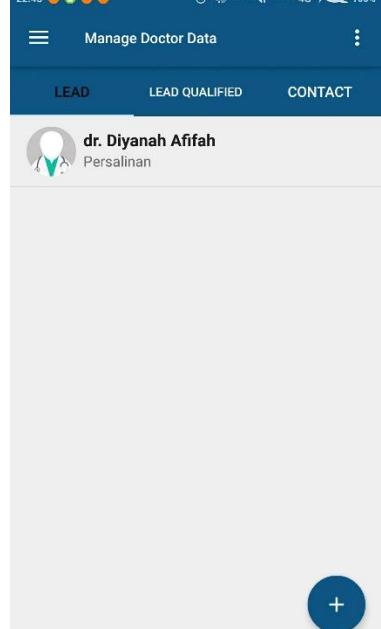
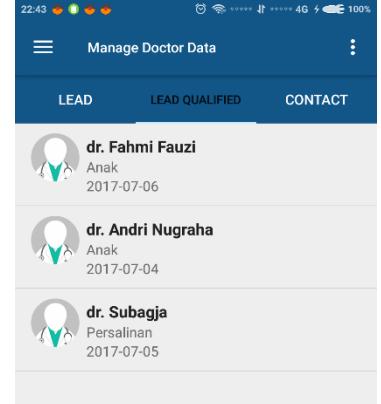
5.3 Hasil Implementasi

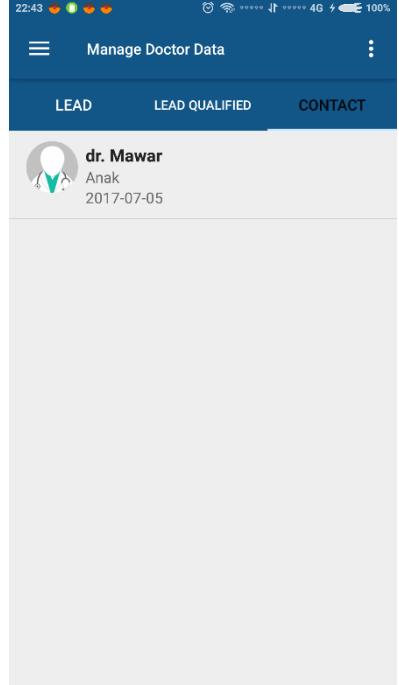
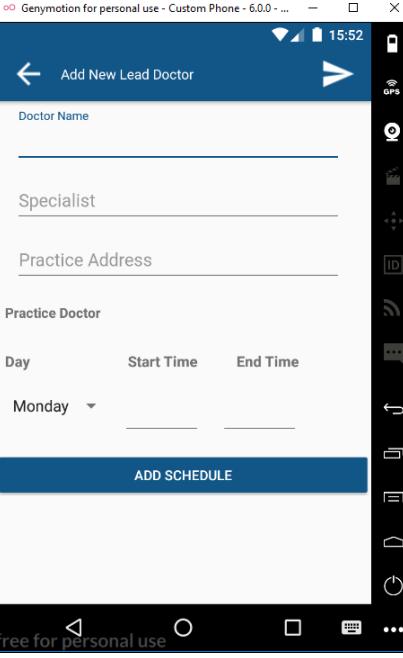
Berikut gambar hasil *screenshot* dari *requirement* yang telah diimplementasikan.

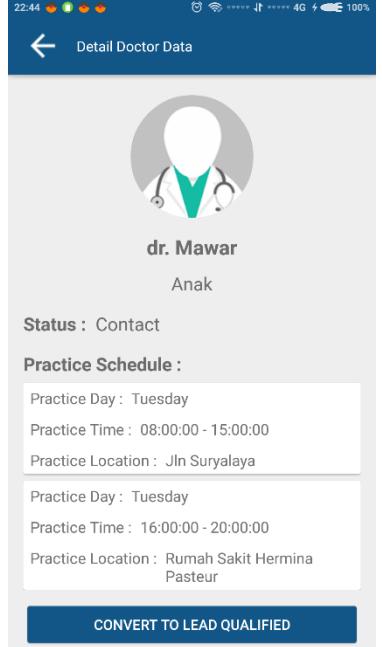
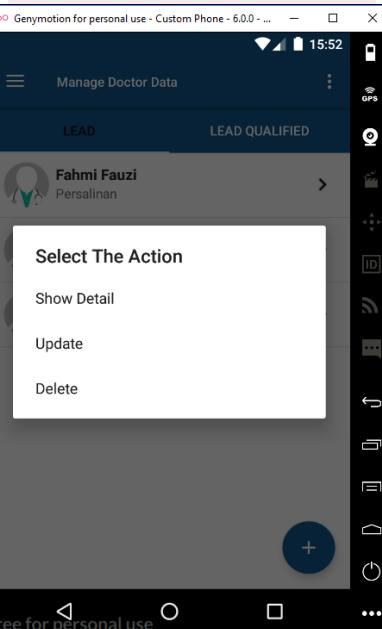
A. Halaman Utama Aplikasi *mobile medical rep*

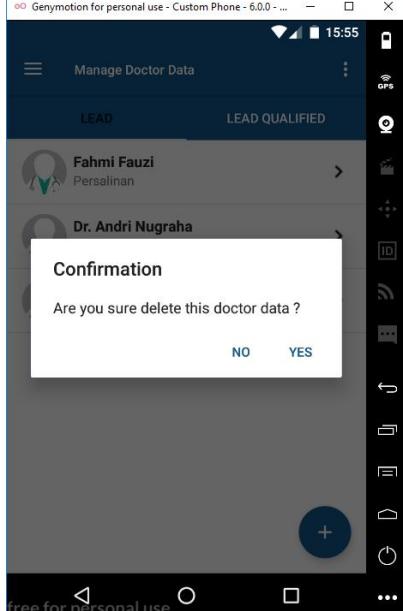
No	Nama UI	Hasil Tampilan
1	Show Dashboard (IM-DD-01)	

B. Manage Doctor Data

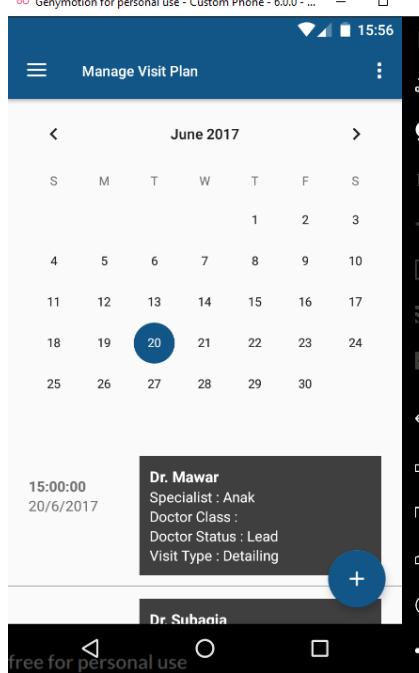
No	Nama UI	Hasil Tampilan
1	Show Lead Doctor (IM-MDD-01)	 <p>22:43 22:43 100% Manage Doctor Data ...</p> <p>LEAD LEAD QUALIFIED CONTACT</p> <p> dr. Diyanah Afifah Persalinan</p> <p>+ (blue button)</p>
2	Show Lead Qualified Doctor (IM-MDD-02)	 <p>22:43 22:43 100% Manage Doctor Data ...</p> <p>LEAD LEAD QUALIFIED CONTACT</p> <p> dr. Fahmi Fauzi Anak 2017-07-06</p> <p> dr. Andri Nugraha Anak 2017-07-04</p> <p> dr. Subagja Persalinan 2017-07-05</p>

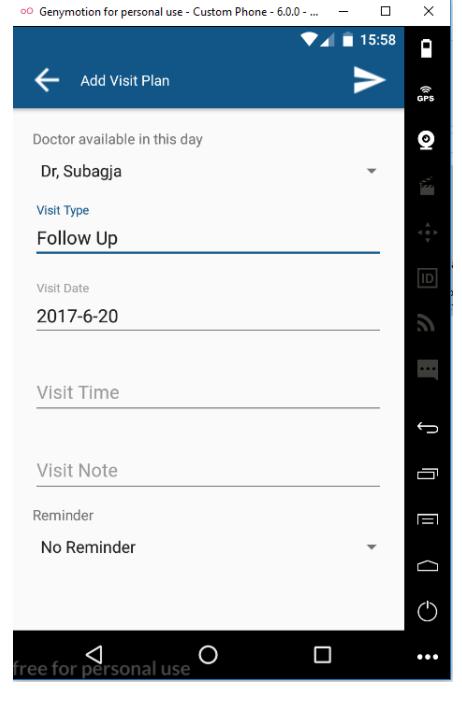
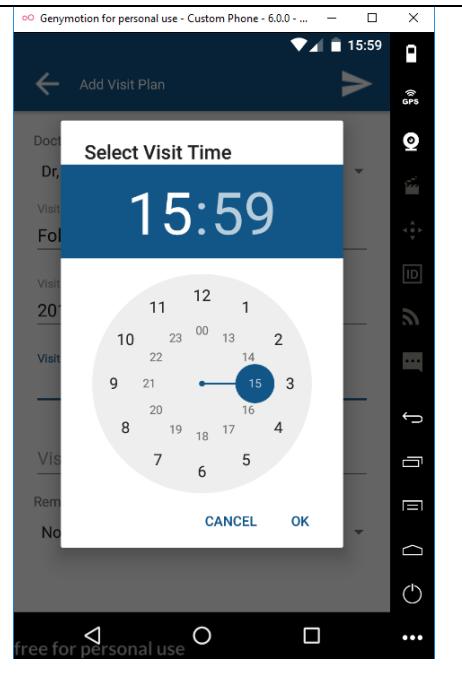
3	<i>Show Doctor Contact (IM-MDD-03)</i>	 A screenshot of a mobile application titled "Manage Doctor Data". At the top, there are tabs for "LEAD", "LEAD QUALIFIED", and "CONTACT". The "CONTACT" tab is selected. Below the tabs, there is a profile card for "dr. Mawar" with the subtitle "Anak" and the date "2017-07-05". The background is light gray.
4	<i>Add New Lead Doctor (IM-MDD-04)</i>	 A screenshot of a mobile application screen titled "Add New Lead Doctor". The screen has fields for "Doctor Name", "Specialist", "Practice Address", and "Practice Doctor". It includes a "Day" dropdown set to "Monday", and "Start Time" and "End Time" fields. A large blue button at the bottom right is labeled "ADD SCHEDULE". The background shows a navigation bar with icons for back, forward, and other functions.

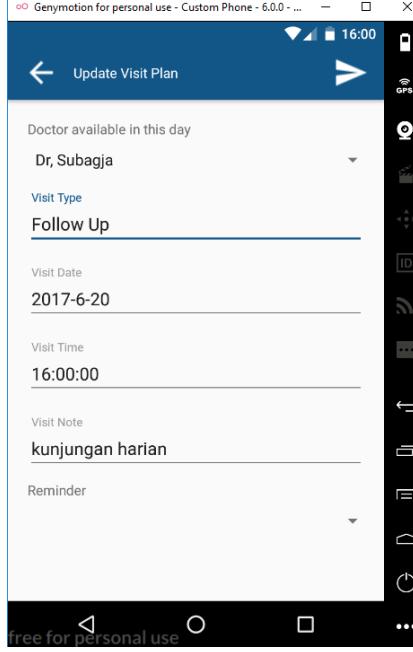
5	<i>Show Detail Doctor (IM-MDD-05)</i>	 <p>22:44 4G 100% ← Detail Doctor Data</p> <p>dr. Mawar Anak</p> <p>Status : Contact</p> <p>Practice Schedule :</p> <table border="1"><tr><td>Practice Day : Tuesday</td></tr><tr><td>Practice Time : 08:00:00 - 15:00:00</td></tr><tr><td>Practice Location : Jln Suryalaya</td></tr></table> <table border="1"><tr><td>Practice Day : Tuesday</td></tr><tr><td>Practice Time : 16:00:00 - 20:00:00</td></tr><tr><td>Practice Location : Rumah Sakit Hermina Pasteur</td></tr></table> <p>CONVERT TO LEAD QUALIFIED</p>	Practice Day : Tuesday	Practice Time : 08:00:00 - 15:00:00	Practice Location : Jln Suryalaya	Practice Day : Tuesday	Practice Time : 16:00:00 - 20:00:00	Practice Location : Rumah Sakit Hermina Pasteur
Practice Day : Tuesday								
Practice Time : 08:00:00 - 15:00:00								
Practice Location : Jln Suryalaya								
Practice Day : Tuesday								
Practice Time : 16:00:00 - 20:00:00								
Practice Location : Rumah Sakit Hermina Pasteur								
6	<i>Show popup action (IM-MDD-06)</i>	 <p>15:52 Genymotion for personal use - Custom Phone - 6.0.0 - ...</p> <p>Manage Doctor Data</p> <p>LEAD LEAD QUALIFIED</p> <p>Fahmi Fauzi Persalinan</p> <p>Select The Action</p> <ul style="list-style-type: none">Show DetailUpdateDelete						

7	<i>Show popup delete (IM-MDD-07)</i>	
---	--------------------------------------	--

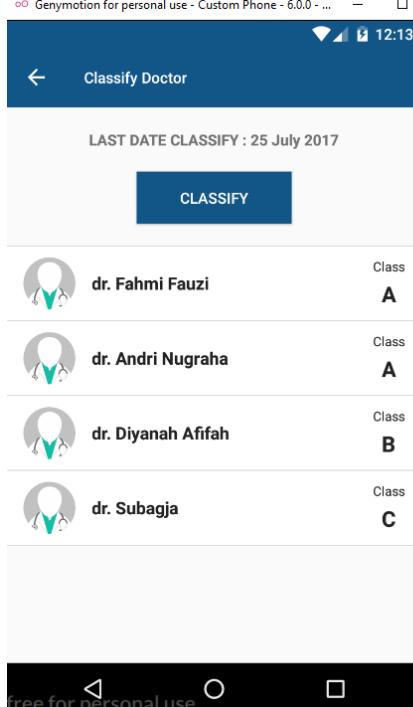
C. Manage Visit Plan

No	Nama UI	Hasil Tampilan
1	<i>Show Calendar (IM-MVP-01)</i>	

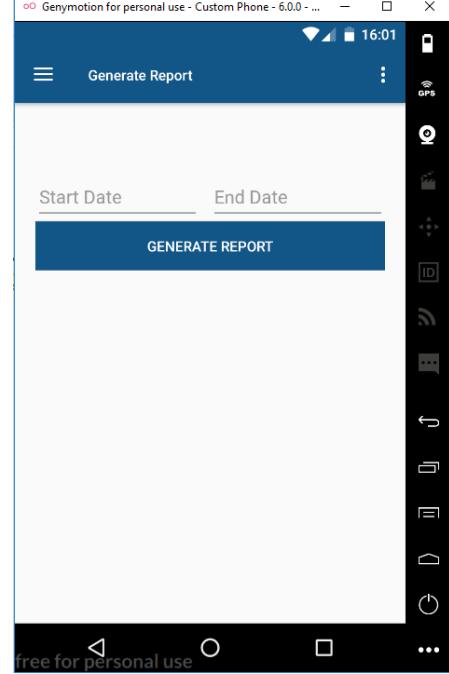
2	Add Visit Plan (IM-MVP-02)	
3	Select Time picket visit time (IM-MVP-03)	

4	Update visit plan (IM-MVP-04)	
---	-------------------------------	--

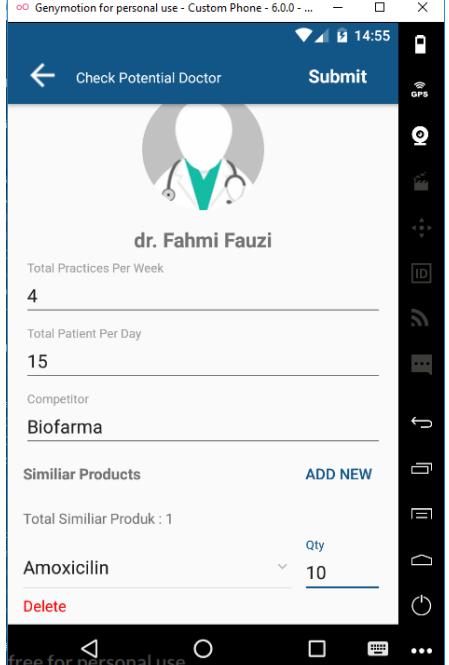
D. Classify Doctor

No	Nama UI	Hasil Tampilan								
1	<i>Form Classify Doctor (IM-CD-01)</i>	 <table border="1" style="margin-top: 10px; width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: center; padding: 5px;">dr. Fahmi Fauzi</td> <td style="width: 10%; text-align: center; padding: 5px;">Class A</td> </tr> <tr> <td style="text-align: center; padding: 5px;">dr. Andri Nugraha</td> <td style="text-align: center; padding: 5px;">Class A</td> </tr> <tr> <td style="text-align: center; padding: 5px;">dr. Diyanah Afifah</td> <td style="text-align: center; padding: 5px;">Class B</td> </tr> <tr> <td style="text-align: center; padding: 5px;">dr. Subagja</td> <td style="text-align: center; padding: 5px;">Class C</td> </tr> </table>	dr. Fahmi Fauzi	Class A	dr. Andri Nugraha	Class A	dr. Diyanah Afifah	Class B	dr. Subagja	Class C
dr. Fahmi Fauzi	Class A									
dr. Andri Nugraha	Class A									
dr. Diyanah Afifah	Class B									
dr. Subagja	Class C									

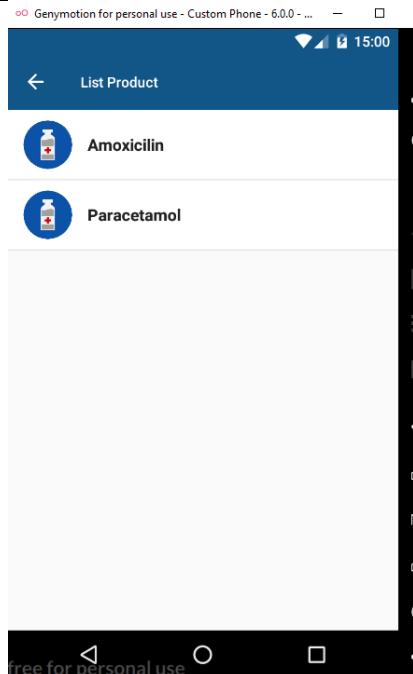
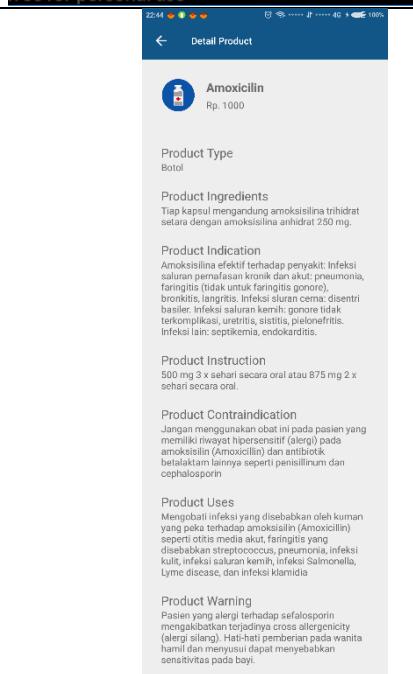
E. Generate Report

No	Nama UI	Hasil Tampilan
1	Form Generate Report (IM-GR-01)	 <p>The screenshot shows a mobile application interface titled "Generate Report". It features two input fields labeled "Start Date" and "End Date". Below these fields is a large, prominent blue button with the text "GENERATE REPORT" in white. The overall design is clean with a white background and a dark blue header bar.</p>

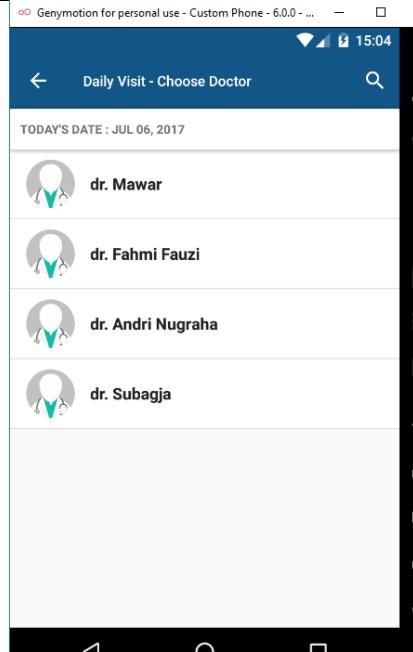
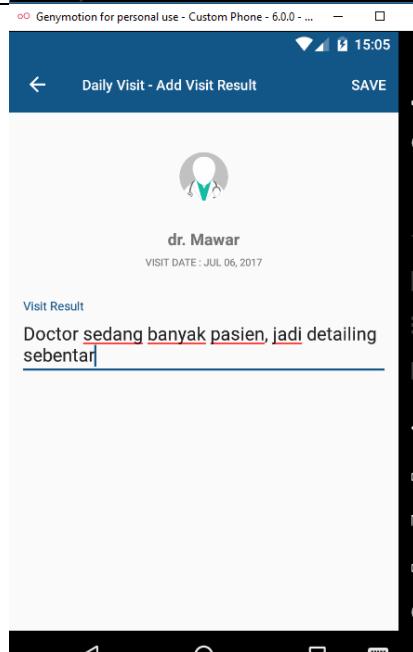
F. Check Potential Doctor

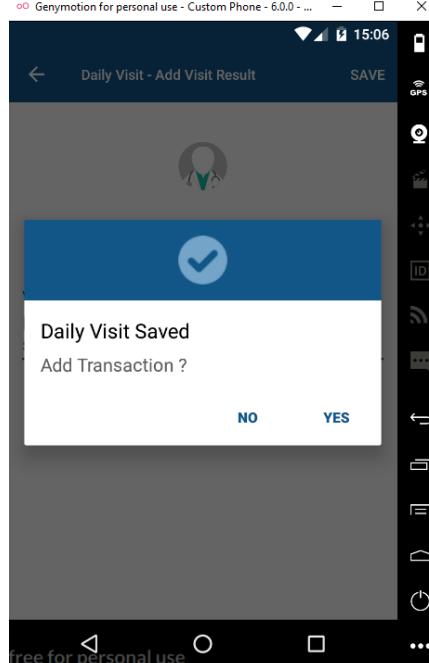
No	Nama UI	Hasil Tampilan
1	Form Check Potential Doctor (IM-CP-01)	 <p>The screenshot shows a mobile application interface titled "Check Potential Doctor". At the top, there is a profile icon of a doctor wearing a stethoscope and the name "dr. Fahmi Fauzi". Below this, there are several data entries: "Total Practices Per Week" with the value "4", "Total Patient Per Day" with the value "15", and "Competitor" with the value "Biofarma". There is also a section for "Similiar Products" which lists "Amoxicilin" with a quantity of "10". A red "Delete" button is visible next to the product entry. The bottom of the screen includes a standard Android navigation bar with icons for back, home, and recent apps.</p>

G. Show Product

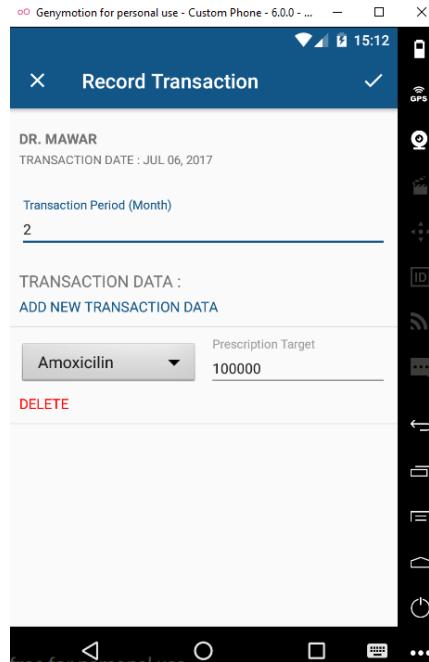
No	Nama UI	Hasil Tampilan
1	Show List Product (IM-SP-01)	 <p>The screenshot shows a mobile application interface titled "List Product". It displays two product entries: "Amoxicillin" and "Paracetamol". Each entry includes a small icon of a medicine bottle.</p>
2	Show Detail Product (IM-SP-02)	 <p>The screenshot shows a detailed view of the "Amoxicillin" product. It includes the following information:</p> <ul style="list-style-type: none"> Product Type: Botol Product Ingredients: Tiap kapsul mengandung amoksikillina trihidrat setara dengan amoksikillina anhidrat 250 mg. Product Indication: Amoksikillina efektif terhadap penyakit: Infeksi saluran pernafasan kronik dan akut; pneumonia, faringitis (tidak untuk faringitis gonore), bronkitis, bronjitis. Infeksi saluran cerna disentri basah. Infeksi saluran kencing: infeksi tidak terkompilasi, uretritis, sistitis, pielonefritis. Infeksi lain: septikemia, endokarditis. Product Instruction: 500 mg 3 x sehari secara oral atau 875 mg 2 x sehari secara oral. Product Contraindication: Jangan menggunakan obat ini pada pasien yang memiliki riwayat hipersensitif (alergi) pada amoksikillin (Amoxicillin) dan antibiotik beta-laktam lainnya seperti penicilllinum dan cephalosporin. Product Uses: Mengobati infeksi yang disebabkan oleh kuman yang peka terhadap amoksikillin (Amoxicillin) seperti infeksi pada akut, faringitis yang disebabkan streptococcus, pneumonia, infeksi kulit, infeksi saluran kencing, infeksi Salmonella, Lyme disease, dan infeksi kloridio. Product Warning: Pasien yang alergi terhadap sefalosporin mengalami ketidaknyamanan cross allergenicity (alergi silang). Hati-hati pemberian pada wanita hamil dan menyusui dapat menyebabkan sensivitas pada bayi.

H. Add Daily Visit

No	Nama UI	Hasil Tampilan
1	Show List Doctor To Add Daily Visit (IM-DV-01)	
2	Form Add Daily Visit (IM-DV-02)	

3	<i>Dialog Add Transaction (IM-DV-03)</i>	 <p>The screenshot shows a mobile application interface titled "Daily Visit - Add Visit Result". At the top right is a "SAVE" button. In the center, there is a blue rectangular overlay with a white checkmark icon. Below the icon, the text "Daily Visit Saved" is displayed, followed by "Add Transaction ?". At the bottom of the overlay are two buttons: "NO" on the left and "YES" on the right. The background of the app shows a dark grey header with a logo and some icons on the right side.</p>
---	--	---

I. Record Transaction

No	Nama UI	Hasil Tampilan
1	<i>Show Form Record Transaction (IM-RT-01)</i>	 <p>The screenshot shows a mobile application interface titled "Record Transaction". At the top right is a checkmark icon. The background displays the text "DR. MAWAR" and "TRANSACTION DATE : JUL 06, 2017". Below this, there is a section labeled "Transaction Period (Month)" with the value "2" entered. Further down, it says "TRANSACTION DATA :" and "ADD NEW TRANSACTION DATA". A table row is shown with "Amoxicillin" in the dropdown and "Prescription Target" set to "100000". A red "DELETE" link is visible below the table. The bottom of the screen features a standard Android navigation bar.</p>

J. Manage Task and Target Medical Representative

No	Nama UI
1	Tampilan Halaman Awal Task and Target Medical Rep (GUI-MTT-01)
Hasil Tampilan	

No	Nama UI
2	Tampilan Halaman form add task and target (IM-MTT-02)
Hasil Tampilan	

No	Nama UI
3	Tampilan Halaman <i>detail add task and target</i> (IM-MTT-03)

Hasil Tampilan

CRM - MRA

Medical Representative's Task and Target

Add Medical Representative's Task and Target

Period of Task : July 2017

Medrep Id : MR001

Medical Representative's Name : Arif Khairuddin

Target : Rp 10.000.000,00

Start Date of Task : 2017-07-01

End Date of Task : 2017-07-31

Product Assigned : 1. Amoxicillin
2. Paracetamol

Note : Semangat untuk menempuh target nya yaaa

Back

CRM - Medical Representative Version 1.0.0

K. Medical rep Target Achievement

No	Nama UI
1	Tampilan Halaman <i>Show Medical Rep Target Achievement</i> (IM-MT-01)

Hasil Tampilan

CRM - MRA

Medical Representative's Target Achievement

Medical Representative's Target Achievement

Period : --Month-- Year--

Show 5 entries

No	Medrep ID	Medrep's Name	Target Assigned (Rp)	Total Transaction (Rp)	Target Achievement (X Target)
1	MR001	Arif Khairuddin	1000000	300000	0.3
2	MR002	Inda Nabilah Maulida	5000000	1750000	0.35
3	MR003	Helmi Fauzi	0	0	0
4	MR004	Basuki Cahya	0	0	0
5	MR005	Bayu Arafii	0	0	0

Showing 1 to 5 of 8 entries

Previous 1 2 Next

CRM - Medical Representative Version 1.0.0

L. Show Report

No	Nama UI
1	Tampilan Halaman <i>Show Report</i> (IM-SR-01)

Hasil Tampilan

No	Nama UI
2	Tampilan Halaman <i>Show Report Visit</i> (IM-SR-02)

Hasil Tampilan

No	Nama UI
3	Tampilan Halaman <i>Show Report Visit Detail</i> (IM-SR-03)

Hasil Tampilan

Medical Representative's Report

Detail Report of Visit

Medrep ID : MR002
Medrep Name : Inda Nabila Maulida

04 July 2017

Period of Report : 01 July 2017 - 31 July 2017

Show 5 entries

No	Doctor's Name	Time of Visit	Location	Result
1	Fahmi Fauzi	17:00		Dokter sedang sibuk, jadi detailing sebentar
2	Andri Nugraha	17:00		Dokter sedang banyak pasien, jadi detailing sebentar

Showing 1 to 2 of 2 entries

Previous 1 Next

CRM - Medical Representative Version 1.0.0

No	Nama UI
4	Tampilan Halaman <i>Show Report Transaction</i> (IM-SR-04)

Hasil Tampilan

Medical Representative's Report

Medical Representative's Report of Transaction

Medrep ID : MR002
Medrep Name : Inda Nabila Maulida

Show 10 entries

No	Report ID	Period of Report	Total Transaction	Access Report
1	DTR001	01 July 2017 - 31 July 2017	1	View Detail

Showing 1 to 1 of 1 entries

Previous 1 Next

CRM - Medical Representative Version 1.0.0

No	Nama UI
5	Tampilan Halaman <i>Show Report Transaction Detail</i> (IM-SR-05)

Hasil Tampilan

The screenshot shows the CRM - MRA application interface. On the left is a dark sidebar with user profile (Mizanul Islam, Supervisor), navigation links (Dashboard, Task and Target, Teams, Reports, Products), and a search bar. The main content area has a blue header 'Medical Representative's Report'. Below it, a sub-header 'Detail Report of Transaction' and specific details ('Medrep ID : MR002', 'Medrep Name : Indra Nabila Maulida'). A date filter '04 July 2017' is applied. A table displays the results with columns: No, Doctor's Name, Product, Prescription Target, and Period. Two entries are shown: Subagia (Amoxicillin 500000) and Paracetamol (250000). At the bottom, there are buttons for 'Previous', 'Next', and a search bar.

M. Manage Product

No	Nama UI
1	Tampilan Halaman <i>Manage Product</i> (IM-MP-01)

Hasil Tampilan

The screenshot shows the CRM - MRA application interface. On the left is a dark sidebar with user profile (Mizanul Islam, Supervisor), navigation links (Dashboard, Task and Target, Teams, Reports, Products), and a search bar. The main content area has a blue header 'Manage Product'. Below it, a sub-header 'List Product Data' and a button '+Add Product'. A table displays the products with columns: Product Id, Product Name, Product Type, Product Price, and Action. Two entries are shown: P001 (Amoxicillin, Botol, 1000) and P005 (Paracetamol, Kapsul, 5000). Each product row has 'Edit' and 'Delete' buttons. At the bottom, there are buttons for 'Previous', 'Next', and a search bar.

No	Nama UI
2	Tampilan Halaman Add Product (IM-MP-02)

Hasil Tampilan

No	Nama UI
3	Tampilan Halaman Update Product (IM-MP-02)

Hasil Tampilan

N. Manage Medical Representative Account

No	Nama UI
1	Tampilan Halaman <i>Manage Medical Representative</i> (IM-MMR-01)

Hasil Tampilan

The screenshot shows a CRM interface titled 'CRM - MRA'. On the left is a sidebar with user info (Mizanul Islam, Supervisor) and navigation links (Dashboard, Task and Target, Teams, Medrep's Account, Medrep's Target Achievement, Reports, Products). The main area is titled 'Manage Medical Representative Account' and shows a table of data:

No	Medrep ID	Medrep Name	Medrep Contact	Action
1	MR001	Arif Khairuddin	085624097151	<button>Edit</button> <button>Delete</button>
2	MR002	Inda Nabilah Maulida	08988334411	<button>Edit</button> <button>Delete</button>
3	MR003	Helmi Fauzi	0811224451	<button>Edit</button> <button>Delete</button>
4	MR004	Basuki Cahya	0876655221	<button>Edit</button> <button>Delete</button>
5	MR005	Bayu Arifli	082348891023	<button>Edit</button> <button>Delete</button>

At the bottom, it says 'Showing 1 to 5 of 8 entries' and has 'Previous' and 'Next' buttons. The footer says 'CRM - Medical Representative' and 'Version 1.0.0'.

No	Nama UI
2	Tampilan Halaman <i>Add Medical Representative</i> (IM-MMR-02)

Hasil Tampilan

The screenshot shows a 'Manage Medical Representative Account' page with a modal dialog titled 'Add Medrep Account' overlaid. The modal contains input fields for 'Medrep ID', 'Medrep Name', 'Contact', and 'Password'. Below the modal is a table of existing data:

No	Medrep ID	Medrep Name	Contact	Action
1	MR001	Arif Khairuddin	085624097151	<button>Edit</button> <button>Delete</button>
2	MR002	Inda Nabilah Maulida	08988334411	<button>Edit</button> <button>Delete</button>
3	MR003	Helmi Fauzi	0811224451	<button>Edit</button> <button>Delete</button>
4	MR004	Basuki Cahya	0876655221	<button>Edit</button> <button>Delete</button>
5	MR005	Bayu Arifli	082348891023	<button>Edit</button> <button>Delete</button>

At the bottom, it says 'Showing 1 to 5 of 8 entries' and has 'Previous' and 'Next' buttons. The footer says 'CRM - Medical Representative' and 'Version 1.0.0'.

No	Nama UI
3	Tampilan Halaman <i>Update Medical Representative</i> (IM-MMR-03)

Hasil Tampilan

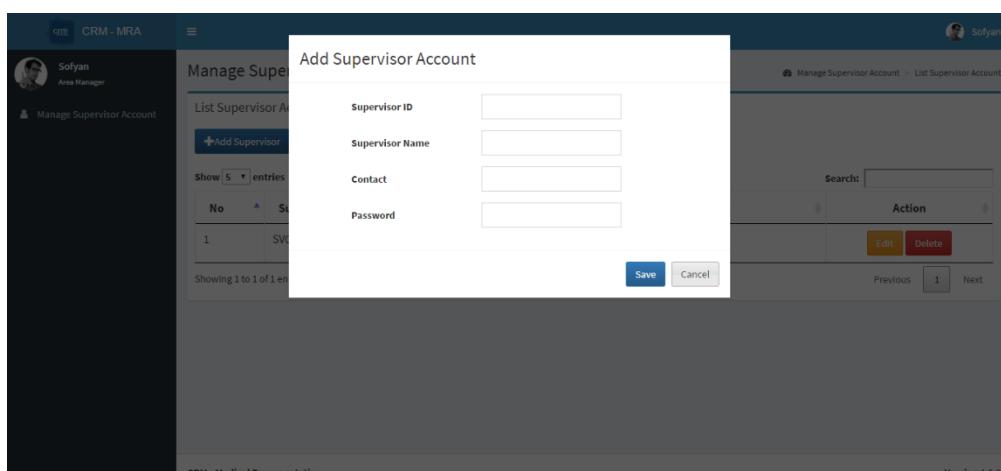
O. Manage Supervisor Account

No	Nama UI
1	Tampilan Halaman <i>Manage Supervisor Account</i> (IM-MSV-01)

Hasil Tampilan

No	Nama UI
2	Tampilan Halaman <i>Add Supervisor Account</i> (IM-MSV-02)

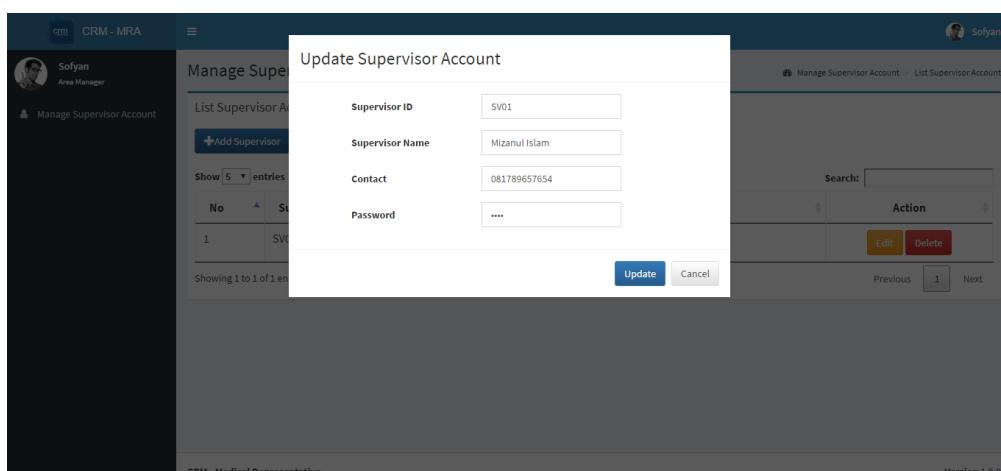
Hasil Tampilan



The screenshot shows the 'Add Supervisor Account' dialog box. It contains four input fields: 'Supervisor ID' (SV01), 'Supervisor Name' (Mizanul Islam), 'Contact' (081789657654), and 'Password' (****). Below the fields are 'Save' and 'Cancel' buttons.

No	Nama UI
3	Tampilan Halaman <i>Update Supervisor Account</i> (IM-MSV-02)

Hasil Tampilan



The screenshot shows the 'Update Supervisor Account' dialog box. It contains four input fields: 'Supervisor ID' (SV01), 'Supervisor Name' (Mizanul Islam), 'Contact' (081789657654), and 'Password' (****). Below the fields are 'Update' and 'Cancel' buttons.

BAB VI

PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan dan saran dari penggerjaan tugas akhir Pengembangan Aplikasi *Customer Relationship Management* Studi Kasus *Medical Representative*.

6.1 Kesimpulan

Berdasarkan tujuan yang telah didefinisikan di awal laporan serta hasil analisis dan perancangan yang dilakukan, aplikasi yang dikembangkan telah berhasil mengimplementasikan semua kebutuhan-kebutuhan sesuai dengan fungsinya sebagai berikut:

1. Aplikasi menyediakan suatu model data yang terpusat yang dapat mempersingkat proses perekapan data kunjungan maupun transaksi yang dilakukan oleh *medical rep*. Dengan adanya model data terpusat aplikasi mampu mencatat dan mengelola perencanaan kunjungan yang dilakukan *medical rep* dan mencatat hasil kunjungan maupun transaksi.
2. Aplikasi dapat men-generate hasil kunjungan maupun transaksi yang dicatat oleh *medical rep* pada aplikasi menjadi laporan kunjungan dan laporan transaksi, serta menampilkan laporan tersebut.
3. Aplikasi dapat mengintegrasikan penyerahan laporan kunjungan dan laporan transaksi dari *medical rep* ke supervisor.
4. Aplikasi dapat mengelola data dokter dan mengklasifikasikan dokter yang akan dikunjungi *medical rep* berdasarkan besar potensi peresepan dokter, sehingga dapat membantu *medical rep* dalam memprioritaskan kunjungannya.
5. Aplikasi dapat mencatat serta menampilkan tugas dan target dari *medical rep* yang diberikan oleh supervisor.
6. Aplikasi dapat menampilkan notifikasi tugas dan target pada aplikasi *mobile medical rep*.

7. Aplikasi dapat menampilkan hasil perhitungan pencapaian target dari *medical rep* pada aplikasi *website*.
8. Aplikasi dapat melakukan pengelolaan produk oleh supervisor dan menampilkan informasi produk perusahaan yang ditugaskan kepada *medical rep*.
9. Aplikasi dapat melakukan pengelolaan akun supervisor yang dilakukan oleh *area sales manager* dan pengelolaan akun *medical rep* dilakukan oleh supervisor.

6.2 Saran

Saran yang dapat diberikan untuk mengembangkan aplikasi *customer relationship management* studi kasus *medical representative* adalah:

1. Aplikasi diharapkan dapat menangani pembagian area kerja dari *medical rep* dan pemberian tugas dan target yang dilakukan supervisor dapat memperhitungkan jumlah dokter dan produk yang banyak dipakai di area kerja *medical rep*.
2. Untuk menambah kemampuan aplikasi dalam membantu *medical rep* memprioritaskan kunjungan, aplikasi sebaiknya menampilkan urutan dokter berdasarkan hasil perhitungan potensi peresepan dokter dan memperhatikan jarak lokasi dari setiap dokter yang akan dikunjungi.
3. Pada penilaian pencapaian target *medical rep* dapat dikembangkan menjadi penilaian kinerja *medical rep* dengan melihat data pencapaian target dengan mempertimbangkan banyaknya dokter yang berhasil bekerja sama terhadap banyaknya dokter yang harus dikunjungi atau banyaknya kunjungan yang dilakukan dan tambahan data pendukung lainnya, misalnya data kualitatif yang dapat dikuantitatifkan seperti: sikap, kedisiplinan, dan sebagainya.

DAFTAR PUSTAKA

- [1] [KBBI Daring] Kamus Besar Bahasa Indonesia. KBBI online kemendikbud. <https://kbbi.kemdikbud.go.id/>. Diakses, Maret 2017.
- [2] Peraturan Menteri Kesehatan Republik Indonesia Huk Nomor 1799/MENKES/PER/XII/2010 tentang Industri Famasi.
- [3] Ministry of Health Republic of Indonesian. “2013 Indonesian Pharmaceutical Industry Directory”. Directorate General of Pharmaceutical and Medical Devices : Ministry of Health Republic of Indonesia. <http://binfar.kemkes.go.id/2014/10/indonesian-pharmaceutical-industry-directory-2013/#.WML5DDt97Dc>. Diakses, Maret 2017.
- [4] Buttle, F. (2009). “Customer Relationship Management Concepts and Technologies” : 2nd. Oxford : Elsevier Ltd.
- [5] Ebert, J. dan Griffin, R,. (2013). “Business Essentials” : 9th. USA : Pearson Prentice Hall.
- [6] [Oxford University Press] Oxford English Dictionary. Online Dictionary. <https://www.oxforddictionaries.com/>. Diakses, Agustus 2017.
- [7] Wardana, F. C. (2010). “Jenjang Karier *Medical representative* Menuju Kursi Direktur”. Jakarta : PT Elex Media Komputindo.
- [8] Kode Etik IPMG tentang Praktik Pemasaran Produk Farmasi Revisi Juli 2013.
- [9] [Rational] The Software Development Company “Rational Unified Process Best Practices for Software Development Teams” : Rational Software White Paper, TP026B, Rev 11/01.

- [10] [AMA] American Marketing Association. Online Dictionary. <https://www.ama.org/resources/Pages/Dictionary.aspx?>. Diakses, Maret 2017.
- [11] Kotler, P. dan Keller K. L., (2012). “Marketing Management” : 14th. USA : Pearson Prentice Hall.

- [12] Yadav, S. K. S. (2016). “Customer Relationship Management is The Need Today” : International Journal of Humanities, Arts, Medicine, and Sciences, 4, pp 107-116.
- [13] Heizer, J. dan Render, B. (2010). “Operations Management” : 10th. Texas Lutheran University : Pearson Prentice Hall.
- [14] Wardana, F. C. (2015). “Kiat Sukses Pengawasan Repetisi Medis”. Yogyakarta : Aswaja Pressindo.
- [15] Jobber, D., Lancaster, G,. (2009). “Selling and Sales Management” : 8th. UK : Pearson Prentice Hall.
- [16] Rahardi, D. (2010). “Manajemen Kinerja Sumber Daya Manusia”. Malang : Tunggal Mandiri Publishing.
- [17] EllisLab. “Code Igniter”. <https://www.codeigniter.com/>. Diakses, Maret 2017.
- [18] Android Developer. “Android, The World's Most Popular Mobile Platform”. <https://developer.android.com/about/android.html>. Diakses, Maret 2017.
- [19] Google “What Is AngularJS”. <https://docs.angularjs.org/guide/introduction>. Diakses, 11 Maret 2017
- [20] Google. “Firebase Cloud Messaging”. <https://firebase.google.com/docs/cloud-messaging/>. Diakses, 23 Maret 2017
- [21] Sargent, M., Linthicum, D,. “REST (Representational State Transfer)”. <http://searchsoa.techtarget.com/definition/REST>. Diakses, Maret 2017.
- [22] F, Adikara., B, Sihotang., and B, Hendradjaya,. “Goal-Oriented Requirements Engineering : State of the Art And Beyond,” in the 2nd International Conference on Information Technology and Business Application, 2013.

- [23] University of Toronto. “GRL – Goal-oriented Requirement Language”.
<http://www.cs.toronto.edu/km/GRL/>. Diakses, 13 April 2017

LAMPIRAN A
DOKUMEN

SOFTWARE REQUIREMENTS SPECIFICATION