



TUGAS AKHIR - KI141502

RANCANG BANGUN *INTERACTIVE VOICE RESPONSE (IVR)* SEBAGAI PENGINGAT MEDIKASI PASIEN LANSIA BERBASIS WEB

**FARIZ AULIA PRADIPTA
NRP 5112100021**

**Dosen Pembimbing I
Fajar Baskoro, S.Kom., M.T.**

Dosen Pembimbing II

--

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**RANCANG BANGUN *INTERACTIVE VOICE RESPONSE*
(IVR) SEBAGAI PENGINGAT MEDIKASI PASIEN LANSIA
BERBASIS WEB**

**FARIZ AULIA PRADIPTA
NRP 5112100021**

**Dosen Pembimbing I
Fajar Baskoro, S.Kom., M.T.**

Dosen Pembimbing II

--

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

DESIGN OF A WEB-BASED INTERACTIVE VOICE RESPONSE (IVR) AS MEDICATION REMINDER FOR ELDERS

FARIZ AULIA PRADIPTA
NRP 5112100021

Supervisor I
Fajar Baskoro, S.Kom., M.T.

Supervisor II

--

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN *INTERACTIVE VOICE RESPONSE (IVR) SEBAGAI PENGINGAT MEDIKASI PASIEN LANSIA BERBASIS WEB*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh
FARIZ AULIA PRADIPTA
NRP : 5112 100 021

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Fajar Baskoro, S.Kom., M.T
NIP: 197404031999031002
.....
(Pembimbing 1)
2. --
NIP: --
.....
(Penimbimbing 2)

SURABAYA
JANUARI, 2018

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN INTERACTIVE VOICE RESPONSE (IVR) SEBAGAI PENGINGAT MEDIKASI PASIEN LANSIA BERBASIS WEB

Nama Mahasiswa : FARIZ AULIA PRADIPTA
NRP : 5112100021
Jurusan : Teknik Informatika FTIK-ITS
Dosen Pembimbing 1 : Fajar Baskoro, S.Kom., M.T.
Dosen Pembimbing 2 : --

Abstrak

Medikasi merupakan terapi pasien dengan menggunakan obat. Obat biasa diberikan kepada pasien sebagai terapi mandiri dan sebagai terapi lanjutan untuk meringankan atau menyembuhkan keluhan penyakit yang dimiliki pasien. Medikasi terlihat mudah, namun tidak boleh dipandang sepele karena obat memiliki zat-zat aktif yang bereaksi dalam tubuh. Jika penggunaannya kurang efektif atau kurang tepat, maka dapat membahayakan pasien.

Hal ini sering terjadi pada lansia, dimana fungsi organ tubuh sudah mulai terbatas. Timbul penyakit-penyakit yang membutuhkan perlakuan khusus, terutama pada obat-obatan yang dikonsumsi oleh lansia. Penyakit-penyakit ini melibatkan obat-obatan yang dimana pemakaiannya akan lebih intensif, berhati-hati, berkala. Dengan keterbatasan memori lansia dan fungsi indera, pasien lansia seringkali mengalami kendala saat melakukan medikasi yang diberikan oleh dokter.

Oleh karena itu, sistem ini dibuat untuk meminimalisir kesalahan atau kelalaian pasien lansia terhadap penggunaan obat beserta dengan jadwalnya. Sistem akan mengakomodir jadwal minum obat, data obat yang sedang diberikan untuk masa terapi, serta sebagai komunikasi antara pasien lansia dengan dokternya.

Kata kunci : Interactive Voice Response, Web, Speech Recognition, Speech Synthesis, Medikasi, Pengingat, Perpesan

DESIGN OF A WEB-BASED INTERACTIVE VOICE RESPONSE (IVR) AS MEDICATION REMINDER FOR ELDERS

Student's Name	: FARIZ AULIA PRADIPTA
Student's ID	: 5112100021
Department	: Teknik Informatika FTIK-ITS
Supervisor I	: Fajar Baskoro, S.Kom., M.T.
Supervisor II	: --

Abstract

Medication is a process of curing or preventing illness with the usage of drugs. Drugs are prescribed by doctor as the continuation treatment for patient. Medication can be described as self-service treatment for patient in curing their illness. Unfortunately, medication process is often considered trivial, whereas the drugs consist active substances that react inside our body. If drugs are not properly used, it might harm patients instead of curing their illnesses.

Such schemes might be very dangerous for elders, as elders have special illnesses and health issues. Elders have tendency to take on more medications than others. This could prove challenging for elders to keep on their medications on-track to prevent undesirable side effects.

This paper proposes a design of a medication reminder and feedback system using Interactive Voice Response (IVR) that could encourage and remind the elders to take their scheduled medications. The design also accommodates the detailed prescribed drugs given, diagnosis histories, and messaging system to provide two-way communication between the doctor and the patient.

Keywords: *Interactive Voice Response, Web, Speech Recognition, Speech synthesis, Medication, Reminder, Messaging*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“RANCANG BANGUN *INTERACTIVE VOICE RESPONSE (IVR)* SEBAGAI PENGINGAT MEDIKASI PASIEN LANSIA BERBASIS WEB”.

Bagi penulis, pengerjaan Tugas Akhir ini merupakan sebuah pengalaman yang berharga. Selama pengerjaan Tugas Akhir, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menjalani perkuliahan di Teknik Informatika ITS dan Tugas Akhir ini adalah implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ayah tercinta, yang selalu memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Keluarga tersayang, dimulai dari Mama dan Papa yang sudah meninggalkan kami terlebih dahulu, yang selalu memberikan motivasi bagi penulis untuk tetap semangat. Ibu, yang selalu khawatir namun tetap sayang dalam keadaan baik susah maupun senang. Kakak dan adik yang terkadang menjengkelkan namun tetap memberikan tawa dan canda bagi penulis untuk tetap terhibur dalam penger-

- jaan tugas akhir ini. Bulik dan pakde yang memberikan penulis dorongan agar tetap bertahan.
4. Bapak Fajar Baskoro, S.Kom., M.T. selaku pembimbing I yang telah membimbing dan membantu penulis serta memberikan motivasi dalam menyelesaikan Tugas Akhir ini dengan sabar.
 5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya serta staf karyawan Jurusan Teknik Informatika ITS yang telah memberikan bantuan demi kelancaran admisnistrasi penulis selama kuliah.
 6. Ibu Dr. Ir. Siti Rochimah, M.T. selaku dosen wali penulis yang telah memberikan pesan moral beserta kemudahan dalam menjalani masa akademik.
 7. "*Geng Kapuk*" – Mandak, Wahyu, Larisi, Ariani yang selalu menemani disaat penulis sedih, memberi motivasi, menjadi rekan berbagi dan teman bermain bagi penulis.
 8. Ahem, Bebek, Samidi, Abu, Ibor yang telah memberi dukungan dan menjadi teman bermain.
 9. Bapak Azis Arijaya yang selalu memberikan penulis semangat serta berbagai bantuan kepada penulis.
 10. Rekan-rekan HMTC khususnya departemen MEDFO yang telah memberi penulis banyak kenangan indah selama perkuliahan dan dukungan dalam penyelesaian tugas akhir.
 11. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga, penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depannya.

Surabaya, Januari 2018

DAFTAR ISI

LEMBAR PENGESAHAN ... Error! Bookmark not defined.	
Abstrak	viii
Abstract	x
KATA PENGANTAR.....	xii
DAFTAR ISI.....	xiv
DAFTAR GAMBAR	xx
DAFTAR TABEL.....	xxiv
DAFTAR KODE SUMBER.....	xxvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah	3
1.6 Metodologi	4
1.7 Sistematika Penulisan Laporan Tugas Akhir	6
BAB II TINJAUAN PUSTAKA	9
2.1 Medikasi	9
2.2 <i>Interactive Voice Response</i>	15
2.3 <i>Speech Recognition</i>	17
2.4 <i>Speech Synthesis</i>	17
2.5 Web Speech API.....	18
BAB III PERANCANGAN SISTEM.....	19
3.1 Analisis Perancangan Sistem	19
3.1.1 Analisis Permasalahan.....	19
3.1.2 Deskripsi Umum Sistem.....	20
3.1.3 Proses Bisnis	22
3.1.3.1 Detil Proses Bisnis Pengiriman Pesan	23
3.1.3.2 Detil Proses Memberikan Konfirmasi Medikasi .	24
3.1.3.3 Detil Proses Pengiriman Pesan Pemberitahuan...	26
3.1.4 Arsitektur Sistem.....	26

3.1.5 Kebutuhan Fungsional.....	27
3.1.5.1 Deskripsi Kasus Kebutuhan UC-001.....	29
3.1.5.2 Deskripsi Kasus Kebutuhan UC-002.....	32
3.1.5.3 Deskripsi Kasus Kebutuhan UC-003.....	33
3.1.5.4 Deskripsi Kasus Kebutuhan UC-004.....	35
3.1.5.5 Deskripsi Kasus Kebutuhan UC-005.....	36
3.1.5.6 Deskripsi Kasus Kebutuhan UC-006.....	37
3.1.5.7 Deskripsi Kasus Kebutuhan UC-007.....	39
3.1.5.8 Deskripsi Kasus Kebutuhan UC-008.....	40
3.2 Perancangan Sistem.....	41
3.2.1 Perancangan Basis Data	42
3.2.1.1 Rancangan Tabel Pasien.....	43
3.2.1.2 Rancangan Tabel Dokter	44
3.2.1.3 Rancangan Tabel Diagnosa	44
3.2.1.4 Rancangan Tabel Preskripsi	45
3.2.1.5 Rancangan Tabel Notifikasi	45
3.2.1.6 Rancangan Tabel Perpesanan	46
3.2.1.7 Rancangan Tabel Pemberitahuan	47
3.2.1.8 Rancangan Tabel Laporan.....	47
3.2.2 Penggunaan <i>Third Party</i>	48
3.2.2.1 OneSignal Web Push Notification	48
3.2.2.2 Artyom.js.....	50
3.2.3 Perancangan Antarmuka Dokter.....	50
3.2.3.1 Antarmuka Mengelola Perpesanan	50
3.2.3.2 Antarmuka Membuat Diagnosa Pasien.....	51
3.2.3.3 Antarmuka Membuat Preskripsi Pasien	52
3.2.3.4 Antarmuka Mengirim Pesan Pemberitahuan	52
3.2.3.5 Antarmuka Melihat Perkembangan Pasien	53
3.2.4 Perancangan Antarmuka Pasien	54
3.2.4.1 Antarmuka Mengelola Perpesanan	54
3.2.4.2 Antarmuka Melihat Pesan Pemberitahuan	54
3.2.4.3 Antarmuka Mengirim Pesan Konfirmasi	55
3.2.4.4 Antarmuka Mengubah Waktu Minum	56
BAB IV IMPLEMENTASI.....	57
4.1 Lingkungan Implementasi	57

4.1.1. Lingkungan Implementasi Perangkat Keras	57
4.1.2. Lingkungan Implementasi Perangkat Lunak	58
4.2 Implementasi Basis Data	58
4.2.1 Implementasi Tabel Pasien	58
4.2.2 Implementasi Tabel Dokter	59
4.2.3 Implementasi Tabel Diagnosa	59
4.2.4 Implementasi Tabel Preskripsi	60
4.2.5 Implementasi Tabel Notifikasi	61
4.2.6 Implementasi Tabel Perpesanan	61
4.2.7 Implementasi Tabel Pemberitahuan.....	62
4.2.8 Implementasi Tabel Laporan	63
4.3 Implementasi <i>Third Party</i>	63
4.3.1 OneSignal Web Push Notification.....	63
4.3.1.1 <i>Subscription prompt</i>	64
4.3.1.2 Create Notification	65
4.3.2 Artyom.js.....	67
4.3.2.1 Inisialisasi Artyom.js.....	67
4.3.2.2 Inisialisasi Speech Recognition dan Bahasa	68
4.3.2.3 Menambahkan Perintah	68
4.3.2.4 Menambahkan <i>Prompt</i>	69
4.4 Implementasi Antarmuka Aplikasi <i>Web</i>	70
4.4.1 Implementasi Antarmuka Dokter	70
4.4.1.1 Antarmuka Mengelola Perpesanan	70
4.4.1.2 Antarmuka Membuat Diagnosa Pasien.....	71
4.4.1.3 Antarmuka Membuat Preskripsi Pasien	72
4.4.1.4 Antarmuka Mengirim Pesan Pemberitahuan	72
4.4.1.5 Antarmuka Melihat Perkembangan Pasien	73
4.4.2 Implementasi Antarmuka Pasien	74
4.4.2.1 Antarmuka Mengelola Perpesanan	74
4.4.2.2 Antarmuka Melihat Pesan Pemberitahuan	74
4.4.2.3 Antarmuka Mengirim Pesan Konfirmasi	75
4.4.2.4 Antarmuka Mengubah Waktu Minum	76
4.5 Implementasi Kebutuhan Fungsional.....	76
4.5.1 Implementasi Mengelola Perpesanan	77
4.5.2 Implementasi Membuat Diagnosa Pasien	81

4.5.3 Implementasi Membuat Preskripsi Medikasi	
Pasien	83
4.5.4 Implementasi Mengirim Pesan Pemberitahuan	84
4.5.5 Implementasi Melihat Pesan Pemberitahuan	87
4.5.6 Implementasi Melihat Perkembangan Pasien	89
4.5.7 Implementasi Mengirim Pesan Konfirmasi	93
4.5.8 Implementasi Mengubah Waktu Minum	94
4.6 Implementasi Sistem IVR.....	102
BAB V PENGUJIAN DAN EVALUASI.....	105
5.1 Lingkungan Pengujian.....	105
5.2 Skenario Pengujian.....	106
5.3 Skenario dan Hasil Uji Coba	106
5.3.1 Pengujian Fungsionalitas Mengelola Perpesanan ..	106
5.3.2 Pengujian Fungsionalitas Membuat Diagnosa	
Pasien	109
5.3.3 Pengujian Fungsionalitas Membuat Preskripsi	
Medikasi Pasien	111
5.3.4 Pengujian Fungsionalitas Mengirim Pesan	
Pemberitahuan.....	113
5.3.5 Pengujian Fungsionalitas Melihat Pesan	
Pemberitahuan.....	115
5.3.6 Pengujian Fungsionalitas Melihat Perkembangan	
Pasien	116
5.3.7 Pengujian Fungsionalitas Mengirim Pesan	
Konfirmasi	117
5.3.8 Pengujian Fungsionalitas Mengubah Waktu	
Minum.....	119
5.3.9 Pengujian Sistem IVR	126
5.3.10 Pengujian OneSignal Web Push Notification	129
5.4 Evaluasi Pengujian	133
5.4.1 Evaluasi Pengujian Kebutuhan Fungsional	133
5.4.2 Evaluasi Pengujian Fitur	134
5.4.3 Evaluasi Kelebihan dan Kekurangan Sistem	134
BAB VI KESIMPULAN DAN SARAN	137
6.1 Kesimpulan	137

6.2 Saran.....	138
DAFTAR PUSTAKA	139
BIODATA PENULIS	143

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2. 1. Grafik kadar obat dalam darah terhadap waktu: menjaga efektivitas obat dengan minum tepat waktu	12
Gambar 2. 2. Grafik reaksi obat ketika dosis berikutnya terlambat atau dilupakan	13
Gambar 2. 3. Grafik reaksi obat ddl (<i>Demeclocycline</i>) terhadap aturan makan	14
Gambar 2. 4. Grafik reaksi obat <i>Efavirenz</i> saat perut kosong dan saat setelah makan	15
Gambar 2. 5. Contoh skema implementasi IVR	16
Gambar 3. 1. Desain rancangan sistem IVR	22
Gambar 3. 2. Gambaran rangkaian proses bisnis secara keseluruhan	23
Gambar 3. 5. Cara kerja sistem pengingat medikasi pasien ...	25
Gambar 3. 6. Proses bisnis pengiriman pesan pemberitahuan	26
Gambar 3. 7. Arsitektur sistem	27
Gambar 3. 8. Diagram kasus penggunaan	27
Gambar 3. 8. Diagram aktivitas mengelola perpesanan untuk pengguna pasien	30
Gambar 3. 9. Diagram aktivitas mengelola perpesanan untuk pengguna dokter	31
Gambar 3. 10. Diagram aktivitas membuat diagnosa baru	33
Gambar 3. 11. Diagram aktivitas membuat preskripsi medikasi pasien	34
Gambar 3. 12. Diagram aktivitas mengirim pesan pemberitahuan	36
Gambar 3. 13. Diagram aktivitas melihat pesan pemberitahuan	37
Gambar 3. 14. Diagram aktivitas melihat perkembangan pasien	38
Gambar 3. 15. Diagram aktivitas melihat perkembangan pasien	40
Gambar 3. 16. Diagram aktivitas mengubah waktu minum... <td>41</td>	41

Gambar 3. 17. Conceptual Data Model	42
Gambar 3. 18. Physical Data Model.....	43
Gambar 3. 19. Diagram alir OneSignal	49
Gambar 3. 20. Rancangan antarmuka mengelola perpesanan	51
Gambar 3. 21. Rancangan antarmuka membuat diagnosa pasien	51
Gambar 3. 22. Rancangan antarmuka membuat preskripsi pasien	52
Gambar 3. 23. Rancangan antarmuka mengirim pesan pemberitahuan.....	53
Gambar 3. 24. Rancangan antarmuka melihat perkembangan pasien	53
Gambar 3. 25. Rancangan antarmuka mengelola perpesanan	54
Gambar 3.26. Rancangan antarmuka melihat pesan pemberitahuan.....	55
Gambar 3. 27. Rancangan antarmuka mengirim pesan konfirmasi	55
Gambar 3. 28. Rancangan antarmuka mengubah waktu minum	56
Gambar 4. 1. Implementasi antarmuka mengelola perpesanan (dokter)	70
Gambar 4. 2. Implementasi antarmuka membuat diagnosa pasien	71
Gambar 4. 3. Implementasi antarmuka membuat diagnosa pasien	72
Gambar 4. 4. Implementasi antarmuka mengirim pesan pemberitahuan.....	73
Gambar 4. 5. Implementasi antarmuka melihat perkembangan pasien	73
Gambar 4. 6. Implementasi antarmuka mengelola perpesanan (pasien)	74
Gambar 4. 7. Implementasi antarmuka melihat pesan pemberitahuan.....	75
Gambar 4. 8. Implementasi antarmuka mengirim pesan konfirmasi	75

Gambar 4. 9. Implementasi antarmuka mengubah waktu minum	76
Gambar 5. 1. Pengujian pengiriman pesan	108
Gambar 5. 2. Pengujian penerimaan pesan	109
Gambar 5. 3. Pengujian pembuatan data pasien baru	110
Gambar 5. 4. Pengujian pengisian data diagnosa.....	111
Gambar 5. 5. Jadwal minum obat yang tersimpan pada <i>server OneSignal</i>	112
Gambar 5. 6. Hasil pengujian membuat preskripsi medikasi pasien	113
Gambar 5. 7. Hasil pengujian mengirim pesan pemberitahuan	114
Gambar 5. 8. Menu utama pesan pemberitahuan.....	115
Gambar 5. 9. Isi pesan pemberitahuan dari dokter yang ditampilkan dalam bentuk percakapan	116
Gambar 5. 10. Hasil pengujian melihat perkembangan pasien	117
Gambar 5. 11. Hasil pengujian mengirim pesan konfirmasi	119
Gambar 5. 12. Hasil pengujian memilih menu ingatkan nanti	119
Gambar 5. 13. Kondisi awal pengujian mengubah jadwal minum obat satu kali sehari.....	121
Gambar 5. 14. Pengujian menyimpan jadwal minum obat satu kali sehari	121
Gambar 5. 15. Hasil akhir pengujian mengubah jadwal minum obat satu kali sehari	122
Gambar 5. 16. Hasil akhir pengujian mengubah jadwal minum obat satu kali sehari pada OneSignal	122
Gambar 5. 17. Kondisi awal pengujian mengubah jadwal minum obat dua kali sehari	122
Gambar 5. 18. Pengujian menyimpan jadwal minum obat dua kali sehari	123
Gambar 5. 19. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari.....	123

Gambar 5. 20. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari pada OneSignal	124
Gambar 5. 21. Kondisi awal pengujian mengubah jadwal minum obat tiga kali sehari	124
Gambar 5. 22. Pengujian menyimpan jadwal minum obat tiga kali sehari	125
Gambar 5. 23. Hasil akhir pengujian mengubah jadwal minum obat tiga kali sehari	125
Gambar 5. 24. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari pada OneSignal	126
Gambar 5. 25. Hasil pengujian mengubah waktu konfirmasi minum obat	126
Gambar 5. 26. Pengujian sistem IVR skenario pertama, menggunakan perintah suara	128
Gambar 5. 27. Pengujian sistem IVR skenario kedua, memilih perintah dari <i>command box</i>	129
Gambar 5. 28. Pengujian sistem IVR skenario ketiga, menulis perintah pada <i>textbox</i>	129
Gambar 5. 29. Pengujian skenario pertama, jendela <i>subscription</i> pada <i>desktop</i>	131
Gambar 5. 30. Pengujian skenario pertama, jendela <i>subscription</i> pada perangkat bergerak	131
Gambar 5. 31. Hasil pengujian skenario pertama, kedua perangkat berhasil terdaftar pada OneSignal	132
Gambar 5. 32. Penerimaan notifikasi pada <i>desktop</i>	132
Gambar 5. 33. Penerimaan notifikasi pada perangkat bergerak	132
Gambar 5. 34. Koneksi aman dibutuhkan untuk mengakses mikrofon	136

DAFTAR TABEL

Tabel 2. 1. Penggolongan obat	10
Tabel 2. 2. Penggunaan obat yang umum dituliskan pada resep dan obat pasien [4]	11
Tabel 3. 1. Karakteristik pengguna sistem.....	20
Tabel 3. 1. Deskripsi kasus penggunaan.....	28
Tabel 3. 2. Rincian alur kasus penggunaan UC-001	29
Tabel 3. 3. Rincian alur kasus penggunaan UC-002	32
Tabel 3. 4. Rincian alur kasus penggunaan UC-003.....	34
Tabel 3. 5. Rincian alur kasus penggunaan UC-004.....	35
Tabel 3. 6. Rincian alur kasus penggunaan UC-005	36
Tabel 3. 7. Rincian alur kasus penggunaan UC-006.....	38
Tabel 3. 8. Rincian alur kasus penggunaan UC-007	39
Tabel 3. 9. Rincian alur kasus penggunaan UC-008.....	40
Tabel 3. 10. Atribut tabel Pasien	43
Tabel 3. 11. Atribut tabel Dokter.....	44
Tabel 3. 12. Atribut tabel Diagnosa.....	44
Tabel 3. 13. Atribut tabel Preskripsi	45
Tabel 3. 14. Atribut tabel Notifikasi	46
Tabel 3. 15. Atribut tabel Perpesanan.....	46
Tabel 3. 15. Atribut tabel Pemberitahuan	47
Tabel 3. 16. Atribut tabel Laporan	48
Tabel 4. 1. Spesifikasi lingkungan pengembangan.....	57
Tabel 4. 2. Parameter utama yang dibutuhkan untuk pengiriman pesan notifikasi.....	67
Tabel 5. 2. Spesifikasi lingkungan uji coba	105
Tabel 5. 2. Pengujian fungsionalitas mengelola perpesanan	106
Tabel 5. 3. Pengujian fungsionalitas mengelola perpesanan	109
Tabel 5. 4. Pengujian fungsionalitas membuat preskripsi medikasi pasien	112
Tabel 5. 5. Pengujian fungsionalitas mengirim pesan pemberitahuan	113

Tabel 5. 6. Pengujian fungsionalitas melihat pesan pemberitahuan	115
Tabel 5. 7. Pengujian fungsionalitas melihat perkembangan pasien	116
Tabel 5. 8. Pengujian fungsionalitas mengirim pesan konfirmasi	118
Tabel 5. 9. Pengujian fungsionalitas mengubah waktu minum	120
Tabel 5. 10. Pengujian sistem IVR	127
Tabel 5. 11. Pengujian OneSignal web push notification	130
Tabel 5. 12. Rangkuman hasil pengujian kebutuhan fungsional	133
Tabel 5. 13. Rangkuman hasil pengujian fitur	134
Tabel 5. 14. Perbandingan sistem dengan aplikasi Android	135

DAFTAR KODE SUMBER

Kode Sumber 4. 1. Implementasi struktur Tabel Pasien	59
Kode Sumber 4. 2. Implementasi struktur Tabel Dokter	59
Kode Sumber 4. 3. Implementasi struktur Tabel Diagnosa ...	60
Kode Sumber 4. 4. Implementasi struktur Tabel Preskripsi...	60
Kode Sumber 4. 5. Implementasi struktur Tabel Notifikasi...	61
Kode Sumber 4. 6. Implementasi struktur Tabel Perpesanan	62
Kode Sumber 4. 7. Implementasi struktur Tabel Perpesanan	62
Kode Sumber 4. 8. Implementasi struktur Tabel Perpesanan	63
Kode Sumber 4. 8. Implementasi <i>subscription prompt</i>	64
Kode Sumber 4. 9. Fungsi <i>subscribe</i> pengguna pasien.....	65
Kode Sumber 4. 10. Potongan kode mengirim pesan notifikasi	66
Kode Sumber 4. 11. Potongan kode inisialisasi Artyom.js pada kode PHP	67
Kode Sumber 4. 12. Inisialisasi <i>speech recognition</i> dan bahasa	68
Kode Sumber 4. 13. Menambahkan perintah sapaan	68
Kode Sumber 4. 14. Contoh potongan kode <i>prompt</i>	69
Kode Sumber 4. 14. Controller Perpesanan.....	78
Kode Sumber 4. 15. Model Perpesanan.....	81
Kode Sumber 4. 16. Controller Diagnosa.....	82
Kode Sumber 4. 17. Model Diagnosa.....	83
Kode Sumber 4. 18. Controller Preskripsi	84
Kode Sumber 4. 19. Model Preskripsi	84
Kode Sumber 4. 20. Controller Pemberitahuan	85
Kode Sumber 4. 21. Model Pemberitahuan	87
Kode Sumber 4. 22. Controller Pemberitahuan	87
Kode Sumber 4. 23. Model Pemberitahuan	89
Kode Sumber 4. 24. Controller Preskripsi	90
Kode Sumber 4. 25. Model Preskripsi.....	93
Kode Sumber 4. 26. Controller Konfirmasi.....	94
Kode Sumber 4. 27. Model Konfirmasi.....	94

Kode Sumber 4. 28. Fungsi mengubah jadwal minum pada controller preskripsi	95
Kode Sumber 4. 29. Fungsi model mengubah jadwal minum	100
Kode Sumber 4. 30. Fungsi mengubah konfirmasi waktu minum pada controller preskripsi	101
Kode Sumber 4. 31. Fungsi model mengubah konfirmasi waktu minum	101

BAB I

PENDAHULUAN

1.1 Latar Belakang

Umur mempengaruhi perubahan fungsi fisiologis yang menyebabkan penurunan fungsi tubuh, terutama pada lansia. Perubahan juga terjadi dalam beberapa proses seperti absorpsi, distribusi, metabolisme, ekskresi, serta organ tubuh lainnya. Demikian menyebabkan fungsi-fungsi organ tubuh tidak bekerja dengan baik sehingga menimbulkan berbagai penyakit bagi lansia.

Perawatan pada lansia melibatkan diagnosa yang berkelanjutan serta penanganan yang harus dilakukan secara intensif. Dimulai dari pemberian diagnosa, pra-medikasi (bila diperlukan), obat-obatan, jadwal kontrol, dan lain sebagainya. Salah satu faktor utama dalam perawatan adalah medikasi atau pemberian obat-obatan dalam jumlah yang lebih banyak dan berkala kepada lansia agar tetap menjaga fungsi organ tetap sehat atau sebagai langkah preventif agar penyakit kronis tidak menjangkiti pasien.

Penelitian menunjukkan bahwa seringkali terjadi ketidakpatuhan minum obat pada lansia dikarenakan lupa [1]. Hal ini menyebabkan perawatan melalui medikasi kurang efektif sehingga dapat menimbulkan beberapa masalah seperti peningkatan lamanya perawatan, biaya kesehatan, dan lain sebagainya. Untuk menghindari masalah tersebut, diperlukan pengingat agar tetap menjaga ‘kepatuhan’ lansia terhadap minum obat sesuai anjuran dokter.

Seiring berkembangnya teknologi informasi, pertukaran informasi dapat dilakukan dimana saja secara *online*. Berbagai *platform* juga tersedia untuk menunjang penggunaan teknologi tersebut, salah satunya ada *web* yang terhubung dengan *server* utama. Pengguna dapat berkomunikasi lewat PC, perangkat bergerak, dan perangkat lain yang mendukung layanan internet

dan *browser*. Dengan demikian, pengingat dapat dibuat secara nirkabel untuk tetap mengontrol medikasi lansia yang didampingi oleh pengasuh (keluarga atau perawat).

Pembuatan sistem pengingat harus dibuat *user-friendly* bagi para lansia. Sehingga, berbagai macam teks dan pilihan menjadikan sistem kurang efektif. Untuk memudahkan, sistem suara dapat diimplementasikan sebagai interaksi utama dalam sistem. IVR (*Interactive Voice Response*) menggunakan *voice command* yang diutarakan oleh sistem kepada pengguna sebagai ‘asisten’ pengguna dalam mengingatkan pengguna untuk mengambil medikasi serta memberi opsi pengguna untuk memberikan *feedback* kepada dokter. Sistem menggunakan *speech recognition* untuk menangkap inputan pengguna melalui suara sebagai feedback medikasi atau pesan yang mungkin ingin disampaikan ketika pengguna menemui masalah dalam medikasi.

Dengan perancangan sistem pengingat dan pengiriman *feedback* pada medikasi lansia beserta dengan komunikasi dua arah, diharapkan dapat memicu kedisiplinan lansia dalam meminum obat serta membantu mereka ketika permasalahan muncul dalam masa medikasi.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat sebuah sistem yang dapat mengontrol medikasi pasien lansia melalui *web*?
2. Bagaimana membuat sistem pengingat yang memudahkan pasien lansia untuk tetap melakukan medikasi sesuai anjuran dokter?
3. Bagaimana membuat sistem yang dapat menangkap serta menyalurkan masalah atau pertanyaan pasien lansia kepada dokter tentang medikasi yang telah diberikan?

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah untuk merancang sistem yang berguna sebagai kontrol medikasi terhadap pasien lansia melalui pesan pengingat dan *feedback* kepada dokter menggunakan *Interactive Voice Response* dan berbasis *web (online)*.

1.4 Manfaat

Manfaat yang diharapkan dari pengerjaan tugas akhir ini antara lain, sebagai berikut:

1. Bagi dokter
 - a. Memudahkan kontrol pasien melalui informasi medikasi pasien.
 - b. Menyediakan komunikasi dengan pasien diluar jadwal tatap muka.
 - c. Meningkatkan efektivitas penanganan diagnosa.
2. Bagi Pasien Lansia
 - a. Meningkatkan efektivitas proses medikasi pasien.
 - b. Menyediakan komunikasi dengan dokter diluar jadwal tatap muka.

1.5 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Sistem berbasis *web* dibangun menggunakan bahasa pemrograman PHP dengan framework CodeIgniter yang memanfaatkan arsitektur *Model-View-Controller* (MVC).
2. Arsitektur Sistem berupa *web server*.
3. Manajemen rancangan *database* relasional menggunakan MySQL.
4. Sistem *Interactive Voice Response* (IVR) diimplementasikan menggunakan Web Speech API yang mendukung *browser* Mozilla Firefox, Google Chrome versi terkini.

5. Sistem IVR hanya melibatkan *client PC* dan perangkat bergerak yang terhubung dengan koneksi internet saja dan menggunakan *browser* yang mendukung saja. Yakni Google Chrome dan Mozilla Firefox versi terkini.
6. Pasien lansia adalah pasien yang berumur 60 tahun keatas dan diasumsikan didampingi oleh pendamping pasien (*caregiver*) yang termasuk dalam pengguna sistem.
7. Sistem IVR berguna sebagai penyalur infomasi pengingat dan *feedback* dalam ranah medikasi saja. Bukan sebagai sistem pakar yang menyediakan solusi terhadap masalah.
8. Sistem memanfaatkan *microphone* sebagai komunikasi suara.
9. Sistem *push notification* yang digunakan adalah layanan OneSignal Web Push Notification pada *browser* yang mendukung Web Notification API.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam penggerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.
Tahap awal untuk memulai penggerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Pada proposal tersebut dijelaskan secara garis besar tentang cara kerja sistem dan kerangka sistem.
2. Studi literatur
Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk penggerjaan Tugas Akhir sekaligus mempelajarinya. Beberapa literatur yang perlu dipelajari lebih dalam lagi untuk proses pengambilan suara (*speech recognition*) dan *speech synthesis* sebagai pembangun system IVR. *Web push notification* sebagai pengirim pengingat medikasi, *messaging system* sebagai komu-

nikasi dua arah antara pasien dan dokter menggunakan pesan teks.

3. Analisis dan desain perangkat lunak

Pada tahap ini akan dilakukan analisis dan design perancangan aplikasi sesuai dengan tujuan yang dijabarkan. Desain sistem dibantu oleh Bootstrap4 membantu pengembang *web* dalam mengembangkan *website* yang mengutamakan perangkat bergerak (*mobile first*). Fitur yang akan dikembangkan pada sistem adalah sebagai berikut:

- a. Pengiriman pengingat minum obat
- b. Pengelolaan perpesanan
- c. Konfirmasi pesan pengingat medikasi
- d. Pengelolaan perkembangan pasien
- e. Pengelolaan diagnosa pasien
- f. Pengelolaan medikasi pasien

4. Implementasi perangkat lunak

Perangkat lunak ini akan dibangun dengan menggunakan bahasa pemrograman PHP yang menggunakan CodeIgniter sebagai kerangka kerja dan MySQL sebagai *database*. Kakas bantu yang digunakan adalah Atom Text Editor sebagai *text editor*. Program pihak ketiga (*third-party*) yang digunakan untuk mendukung pengembangan adalah OneSignal Web Push Notification, Artyom.js, dan Bootstrap4.

5. Pengujian dan evaluasi

Pengujian dari aplikasi ini akan dilakukan dengan pengujian *blackbox*. Pengujian *blackbox* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, pengujian dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program.

6. Penyusunan buku Tugas Akhir.

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

Bab I Pendahuluan

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi penggeraan dan sistematika penulisan laporan Tugas Akhir juga terdapat di dalamnya.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Uji Coba Dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*blackbox testing*) untuk mengetahui apakah implementasi perangkat lunak sudah sesuai dengan perancangan serta pengujian-pengujian data dibalik pengguna sudah mendukung kebutuhan fungsional.

Bab VI Kesimpulan Dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Daftar Pustaka Merupakan daftar referensi yang digunakan untuk mengembangkan penelitian ini.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Dalam bab ini akan dijelaskan mengenai teori-teori yang merupakan dasar dari pembangunan sistem. Selain itu terdapat penjelasan yang menunjang pengerjaan Tugas Akhir ini sehingga dapat memberikan gambaran secara umum sistem yang akan dibangun.

2.1 Medikasi

Medikasi merupakan proses terapi dengan pemberian zat (obat) untuk keperluan diagnosis, penyembuhan, terapi, penurun (peredra), pencegahan penyakit. Obat adalah bahan atau paduan bahan-bahan yang digunakan untuk mempengaruhi atau menelidiki sistem fisiologi atau keadaan patologi dalam rangka penetapan diagnosis, pencegahan, penyembuhan, pemulihan, peningkatan kesehatan, dan kontrasepsi termasuk produk biologi. [2].

Terdapat tiga jenis golongan obat yaitu obat bebas, obat bebas terbatas dan obat keras. Obat bebas adalah obat yang dijual bebas di pasaran dan dapat dibeli tanpa resep dokter. Tanda khusus untuk obat bebas adalah berupa lingkaran berwarna hijau dengan garis tepi berwarna hitam. Obat bebas terbatas adalah obat yang dijual bebas dan dapat dibeli tanpa dengan resep dokter, tetapi disertai dengan tanda peringatan. Tanda khusus untuk obat ini adalah lingkaran berwarna biru dengan garis tepi hitam. Khusus untuk obat bebas terbatas, selain terdapat tanda khusus lingkaran biru, diberi pula tanda peringatan untuk aturan pakai obat, karena hanya dengan takaran dan kemasan tertentu, obat ini aman dipergunakan untuk pengobatan sendiri. Tanda peringatan berupa empat persegi panjang dengan huruf putih pada dasar hitam. Obat Keras adalah obat yang hanya dapat diperoleh dengan resep dokter. Ciri-cirinya adalah bertanda lingkaran bulat merah dengan garis tepi berwarna hitam, huruf K di tengah yang

menyentuh garis tepi. Obat ini hanya boleh dijual di apotek dan harus dengan resep dokter pada saat membeli [3].

Tabel 2. 1. Pengolongan obat

Simbol	Keterangan
	Obat bebas. Dijual bebas di pasaran dan dapat dibeli tanpa resep dokter.
	Obat bebas terbatas. Dijual bebas dan dapat dibeli tanpa dengan resep dokter. Disertai dengan tanda peringatan
P. No.1 Awas! Obat Keras. Baca aturan pakainya P. No. 2 Awas! Obat Keras Hanya untuk kumur jangan ditelan P. No.3 Awas! Obat keras Hanya untuk bagian luar badan P. No. 4 Awas! Obat keras Hanya untuk dibakar P.No.5 Awas! Obat Keras Tidak boleh ditelan P. No.6 Awas! Obat keras Obat wasir, jangan ditelan.	Tanda peringatan obat. Tanda peringatan hanya dapat ditemui pada obat bebas terbatas dan obat keras, disertai dengan kode dan peringatan.
	Obat Keras. Obat yang hanya dapat diperoleh dengan resep dokter. Tanda peringatan di-

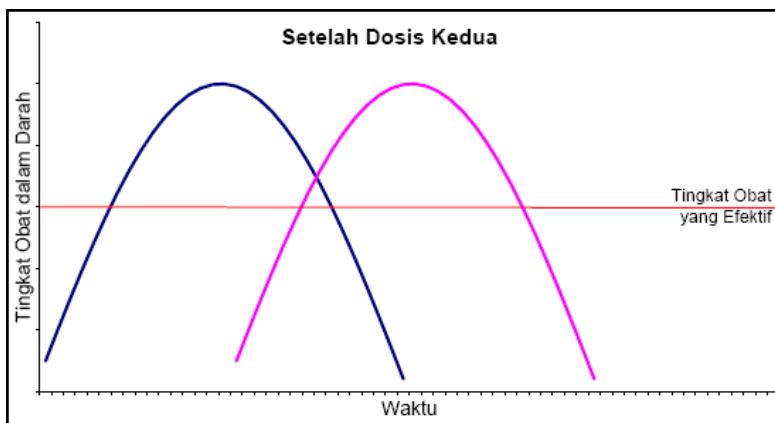
	cantumkan
--	-----------

Obat memiliki reaksi kimia yang terjadi di dalam tubuh. Reaksi obat dapat bermacam-macam untuk menyembuhkan atau mencegah penyakit. Sebaliknya, obat juga dapat membahayakan tubuh apabila digunakan dengan kurang tepat. Ada efek samping tertentu yang mempengaruhi kinerja tubuh saat obat bereaksi. Oleh karena itu penggunaan obat harus dengan aturan yang tepat dan sesuai dengan petunjuk brosur atau resep dokter. Aturan minum obat yang umum digunakan seperti yang dijelaskan pada tabel berikut.

Tabel 2. 2. Penggunaan obat yang umum dituliskan pada resep dan obat pasien [4]

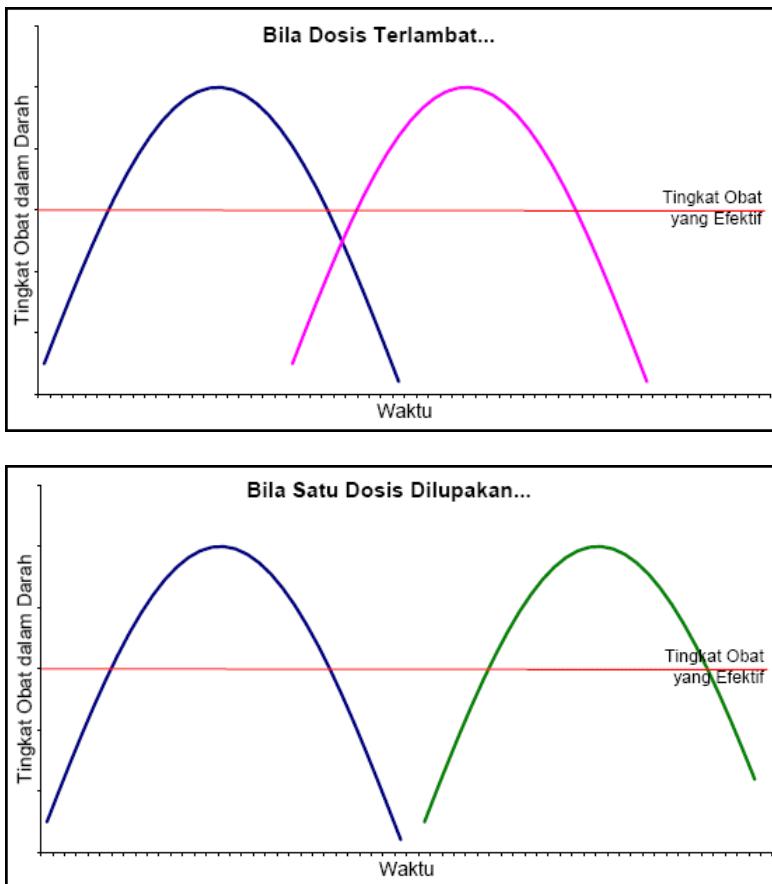
Singkatan	Kepanjangan (dlm. Bhs. Latin)	Arti
Aturan Makan		
a.c.	<i>ante coenam</i>	Sebelum makan
d.c.	<i>durante coenam</i>	Pada waktu makan
p.c.	<i>post coenam</i>	Setelah makan
a.p.	<i>ante prandium</i>	Sebelum sarapan pagi
Waktu Penggunaan		
a.h.	<i>alternis horis</i>	Selang satu jam
h.v.	<i>hora vespertina</i>	Malam hari
h.s.	<i>hora somni</i>	Waktu tidur (sebelum tidur)
h.m.	<i>hora matutina</i>	Pagi hari
s.d.d.	<i>semel de die</i>	Sekali sehari
b.d.d.	<i>bis de die</i>	Dua kali sehari
t.d.d.	<i>ter de dir</i>	Tiga kali sehari
q.d.d	<i>quarter de dir</i>	Empat kali sehari
Kondisional		
s.n.s	<i>si necesse sit</i>	Bila perlu
abs.febr	<i>absente febre</i>	Bila tidak demam
i.m.m	<i>In manus medici</i>	Berikan kepada dokter (khusus yang memerlukan aplikasi dokter)

Salah satu faktor yang penting adalah penggunaan obat yang sesuai dengan jadwal. Reaksi obat yang umum akan mencapai tingkat yang tinggi dalam darah, kemudian tingkat akan mulai turun sebagaimana obat diuraikan, umumnya waktu darah disaring oleh hati atau ginjal. Obat yang dikonsumsi selalu diserap lebih cepat daripada tubuh menguraikannya. Jadi kepekatan tertinggi tercapai dalam waktu yang relatif singkat, dan kemudian membutuhkan waktu yang lebih lama untuk keluar dari tubuh [5].



Gambar 2. 1. Grafik kadar obat dalam darah terhadap waktu: menjaga efektivitas obat dengan minum tepat waktu

Karena penguraian obat berlangsung lebih lama dan tingkat efektivitas yang semakin berkurang, maka dibutuhkan dosis berikutnya untuk menjaga efektivitas obat. Bila tingkat obat dalam darah di bawah tingkat efektivitas, virus masih dapat bereplikasi, tetapi virus yang resisten terhadap obat yang dipakai lebih mungkin unggul [5]. Penjadwalan dosis obat merupakan salah satu faktor utama untuk menjaga keefektifan tingkat obat.

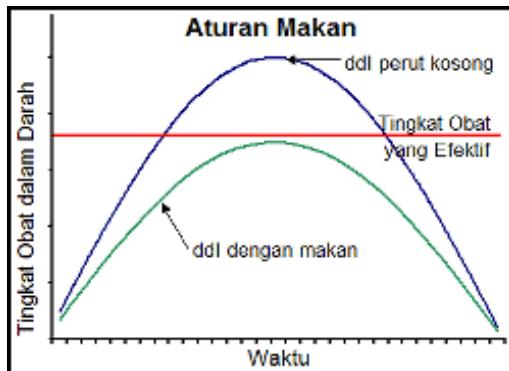


Gambar 2. 2. Grafik reaksi obat ketika dosis berikutnya terlambat atau dilupakan

Ketika jadwal obat terabaikan, baik terlambat maupun dilupakan, akan terjadi waktu renggang dimana tubuh tidak bisa merasakan efektivitas obat dalam jangka waktu tertentu. Waktu renggang tersebut, dapat dimanfaatkan oleh virus yang tersisa di dalam tubuh untuk membentuk antigen atau mereplikasi ulang.

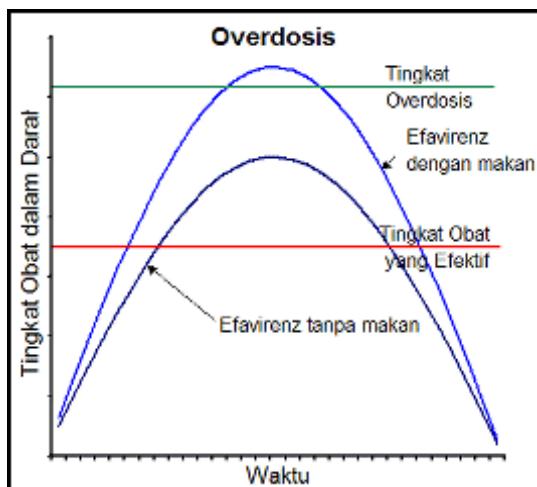
Yang lebih berbahaya apabila obat antibiotik yang berperan membunuh sel-sel penyakit dosisnya terlambat atau dilupakan, virus akan membentuk antigen untuk meningkatkan resistensi kadar obat antibiotik disaat tingkat efektivitas menurun. Hasilnya, obat antibiotik tidak dapat memerangi virus yang memiliki *resistensi* lebih terhadap obat.

Kepatuhan medikasi tidak hanya sebatas jadwal meminum yang tepat, namun juga kepatuhan dalam aturan penggunaan. Faktor lain yang mempengaruhi tingkat efektivitas obat adalah aturan makan: sesudah makan, sebelum makan, dan saat makan.



Gambar 2. 3. Grafik reaksi obat ddl (*Demeclocycline*) terhadap aturan makan

Obat ddl (*Demeclocycline*) bekerja dengan baik bila diminum pada saat perut kosong setidaknya 1 jam sebelum atau 2 jam setelah makan [6]. Ketika obat diminum saat makan atau perut dalam keadaan terisi, efektivitas obat akan berkurang. Pada gambar 2.3., grafik menunjukkan reaksi obat tidak mencapai tingkat obat yang efektif sehingga medikasi berjalan kurang maksimal.



Gambar 2. 4. Grafik reaksi obat *Efavirenz* saat perut kosong dan saat setelah makan

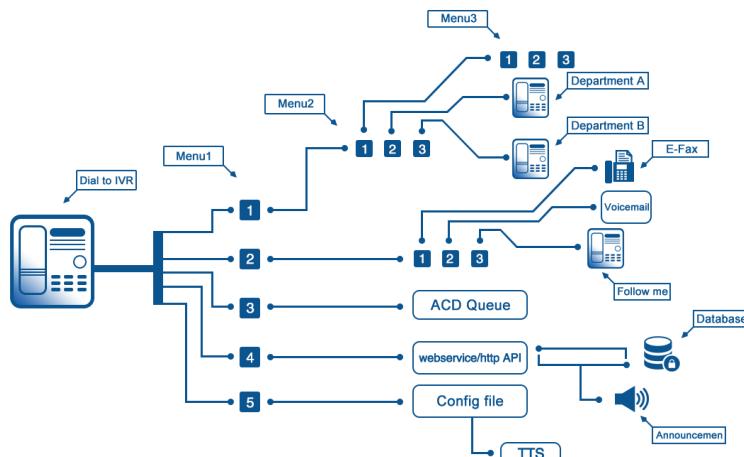
Penyalahgunaan aturan makan dapat berakibat efek samping yang membahayakan. Salah satu contoh obat *Efavirenz* yang disarankan dikonsumsi saat perut kosong. Obat ini lebih efektif ketika dikonsumsi pada malam hari saat ingin tidur [7]. Ketika *Efavirenz* dikonsumsi saat perut terisi atau setelah makan, tingkat efektivitasnya akan menjulang tinggi sehingga mencapai batas tingkat overdosis. Efek samping dari obat ini ketika mencapai tingkat overdosis adalah halusinasi dan menurunnya tingkat kesadaran [7].

2.2 *Interactive Voice Response*

Interactive Voice Response merupakan teknologi layanan berbasis suara yang mengatur interaksi komputer melalui suara terhadap inputan pengguna. Layanan ini biasa dijumpai pada pusat layanan telepon (*call center*) yang menggunakan sistem jawab secara otomatis tanpa harus berinteraksi dengan petugas.

Tujuan IVR adalah memberikan informasi, menangkap inputan pengguna melalui nada sentuh pada tombol telepon, serta memberikan balasan atas inputan yang diterima. IVR memanfaatkan rekaman suara atau nada suara untuk berkomunikasi dengan pengguna. Secara umum, IVR digunakan oleh perusahaan, bank, atau instansi pemerintahan untuk mem-berikan layanan kepada masyarakat umum dalam hal konsultasi, menampung keluhan, maupun pemberian informasi mengenai suatu hal tertentu [8].

Sistem IVR menggunakan inputan nada sentuh pengguna sebagai input. Sistem akan memproses inputan dengan aturan (*rules*) yang sudah ditetapkan oleh pengembang. Setelah diproses, sistem akan memberikan *output* suara pada pengguna beserta aksi yang dapat dilakukan sistem sesuai permintaan.



Gambar 2. 5. Contoh skema implementasi IVR

Sistem IVR memanfaatkan sebuah perangkat *web server* sebagai perangkat utama pemrosesan. Pengguna menghubungi nomor telepon yang telah didefinisikan oleh operator yang terhubung dengan jaringan telepon tetap atau biasa disebut PSTN (*public switched telephone network*). Operator jaringan telepon kemudian terhubung dengan *server* IVR menggunakan PABX

(*Private Automatic Branch Exchange*) untuk mengatur masuk dan keluaranya jaringan telepon dengan nomor tujuan tertentu, biasa digunakan pada hotel dan perkantoran [9]. Setelah terhubung, IVR akan memberikan nada pesan beserta menu yang dapat dipilih oleh pengguna. Pengguna dapat memilih menu dan IVR mem-berikan respon beserta aksi yang dapat dilakukan oleh *server IVR* yang terhubung dengan layanan lainnya.

2.3 Speech Recognition

Speech Recognition (pengenalan wicara) merupakan metode atau teknik yang diterapkan oleh komputer untuk mengenali wicara atau ucapan manusia dan mengubahnya dengan bentuk teks. Komputer menangkap sinyal gelombang berupa suara yang kemudian diubah menjadi suatu data yang dimengerti oleh komputer. Data tersebut kemudian digunakan untuk memproses suatu perintah yang telah ditetapkan. *Speech recognition* juga dikenal dengan *automatic speech recognition* (ASR) atau *speech-to-text* (STT).

Speech recognition menjembatani interaksi manusia dengan komputer melalui interaksi suara yang diucapkan oleh pengguna. Komputer menangkap perintah suara kemudian menerjemahkan ucapan dan menjalankan perintah sesuai dengan ucapan pengguna.

2.4 Speech Synthesis

Speech Synthesis atau mesin pengucap merupakan proses menghasilkan ucapan lisan dengan mesin berdasarkan inputan tertulis atau teks. *Speech Synthesis* juga dikenal dengan *text-to-speech* (TTS) dimana komputer menerjemahkan teks ke dalam ucapan lisan [10]. Inputan teks dimasukkan oleh pengguna yang kemudian diucapkan lisan oleh komputer. *Speech synthesis* biasa digunakan pada aplikasi berbasis suara untuk memberikan *feedback* kepada inputan pengguna yang telah ditetapkan oleh

pengembang atau sebagai bantuan suara dari teks untuk memudahkan pengguna (misal: *Captcha*).

2.5 Web Speech API

Web Speech API merupakan antarmuka pemrograman aplikasi (*application programming interface*) berbasis *JavaScript*. API ini memungkinkan aplikasi *web* untuk menggunakan teknologi suara seperti pengenalan wicara (*speech recognition*) oleh pengguna dan mesin pengucap (*speech synthesis*) oleh *browser*. Web Speech API dikembangkan dan dispesifikasikan oleh W3C (*World Wide Web Consortium*) [11].

Teknologi *speech-to-text* (STT) dan *text-to-speech* (TTS) sudah lama dikenal pada perangkat bergerak. Namun, pada situs *web*, implementasi tersebut hanya terbatas pada produk berbayar dan *closed source* (kode sumber tertutup). Untuk mengembangkan *speech recognition* dan *speech synthesis* akan sangat sulit pada *server-side*. Dengan adanya Web Speech API yang didefinisikan, terintegrasi dengan browser dan teknologi *JavaScript*, pengembang dapat membangun aplikasi suara dengan *offline* (luar jaringan). Namun untuk menggunakan berbagai macam bahasa, aplikasi harus terhubung internet untuk mengambil sampel suara sistem dan pengenalan wicara dalam bahasa lain selain Bahasa Inggris.

Saat ini Web Speech API didukung dan dapat digunakan oleh browser Google Chrome versi 33 keatas, Mozilla Firefox versi 49 keatas, dan Microsoft Edge (hanya *speech synthesis*). Web Speech API juga mendukung multi bahasa termasuk Bahasa Indonesia dan menggunakan Microsoft Speech Synthesis dan Google Speech Synthesis sebagai mesin pengucap [12].

BAB III

PERANCANGAN SISTEM

Perancangan merupakan bagian perencanaan-perencanaan secara teknis perangkat lunak yang dibuat. Bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses, alur, dan implementasinya.

3.1 Analisis Perancangan Sistem

Pada subbab berikut akan dijelaskan analisa pembuatan aplikasi perangkat bergerak dan *web*. Analisis yang dilakukan meliputi analisa permasalahan, kebutuhan umum perangkat lunak, deskripsi umum sistem, arsitektur dan kebutuhan fungsional sistem.

3.1.1 Analisis Permasalahan

Medikasi adalah proses terapi secara mandiri menggunakan obat. Karenanya, penggunaan obat yang tepat serta kepatuhan dalam meminum obat harus diperhatikan agar menghasilkan efek penyembuhan atau pencegahan yang efektif. Kepatuhan yang harus diperhatikan oleh pasien meliputi jadwal minum obat, aturan pakai, dan aturan makan.

Namun, menjaga kepatuhan obat akan sangat sulit bagi lansia karena adanya beberapa faktor, seperti fungsi indera dan fisik yang mulai menurun, banyaknya masalah kesehatan yang bermunculan sehingga obat yang diberikan bisa dalam jumlah lebih banyak dari usia muda, dan masalah kesehatan yang spesial membutuhkan penggunaan obat yang lebih spesifik.

Penelitian menunjukkan bahwa, 73% pasien lansia tidak meminum obat dengan alasan lupa [1]. Banyaknya obat yang digunakan juga merupakan salah satu faktor risiko terjadinya ketidakpatuhan terhadap minum obat sehingga pasien lansia yang

terkadang menjadi lupa karena banyaknya obat yang diberikan walaupun cara penggunaan sudah tertulis dengan detil pada obat.

Perancangan sistem didasarkan pada kepatuhan minum obat dengan penjadwalan minum obat disertai pengingat. Adapun detil obat diimplementasikan agar memudahkan pasien dalam menggunakan obat. Serta sistem perpesanan sebagai komunikasi pasien dengan dokter, bila diperlukan.

3.1.2 Deskripsi Umum Sistem

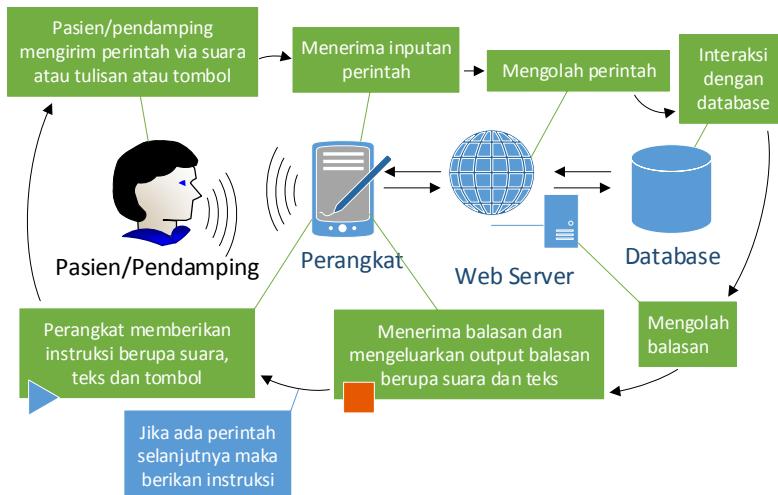
Aplikasi yang dibangun pada Tugas Akhir melibatkan dua entitas, yaitu pasien dan dokter. Dokter memiliki akses terhadap diagnosa pasien, pemberian obat (preskripsi), riwayat diagnosa pasien, serta perpesanan sebagai komunikasi. Pasien memiliki akses asisten sistem menggunakan IVR yang berbasis *web*, melihat daftar obat pada masa terapi beserta dengan data diagnosa, menerima dan konfirmasi pengingat obat, dan perpesanan sebagai komunikasi dengan dokter.

Tabel 3. 1. Karakteristik pengguna sistem

No	Kategori Pengguna	Tugas	Hak Akses ke sistem	Kemampuan yang harus dimiliki
1.	Pasien / pendamping pasien	<ul style="list-style-type: none"> - Mengirimkan pertanyaan mengenai medikasi atau diagnosa bila perlu. - Konfirmasi minum obat setelah pengingat dikirim. 	<ul style="list-style-type: none"> - Mengirim pesan konfirmasi medikasi. - Perpesanan sebagai komunikasi. 	<ul style="list-style-type: none"> - Kemampuan untuk mengoperasikan perangkat.

No	Kategori Pengguna	Tugas	Hak Akses ke sistem	Kemampuan yang harus dimiliki
2.	Dokter	<ul style="list-style-type: none"> - Memaparkan data diagnosa. - Memaparkan preskripsi pasien. - Membantu pasien melalui tanya jawab apabila ada masalah tentang medikasi. 	<ul style="list-style-type: none"> - Memasukkan keterangan diagnosa pasien. - Memasukkan penggunaan obat. - Memasukkan pesan tambahan kepada pasien bila perlu. - Membalas bantuan berupa jawaban dari pertanyaan pasien. 	<ul style="list-style-type: none"> - Kemampuan untuk mengoperasikan perangkat.

Pasien lansia dapat didampingi oleh seorang perawat atau pihak keluarga dalam penggunaan sistem. Sistem IVR yang digunakan pasien menggunakan teknologi penangkapan suara atau mengisi formulir perintah jika mikrofon tidak tersedia. Perintah-perintah yang digunakan adalah perintah umum yang dapat dimanfaatkan pasien untuk mengambil data tanpa harus menggunakan navigasi halaman *web* tertentu, seperti data obat, jadwal minum berikutnya. Selain itu, IVR dapat menampilkan pengingat berupa pesan teks dan suara apabila sudah waktunya minum obat. Pasien dapat membalas pesan sebagai pesan konfirmasi minum melalui IVR ataupun secara manual lewat navigasi halaman data obat.



Gambar 3. 1. Desain rancangan sistem IVR

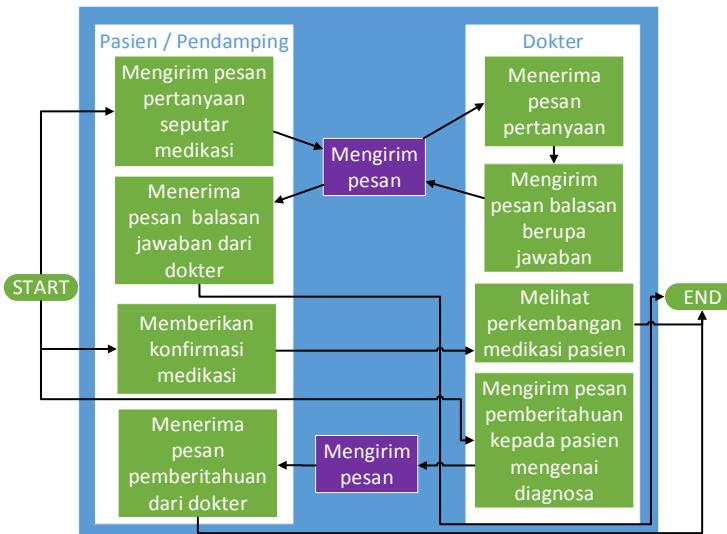
Dokter dapat memantau perkembangan pasien lewat riwayat diagnosa, preskripsi, serta kepatuhan pasien dalam meminum obat.

3.1.3 Proses Bisnis

Sesuai dengan deskripsi umum sistem pada subbab 3.1.2., proses bisnis atau alur kerja sistem dapat dikembangkan dapat dijabarkan sebagai berikut:

1. Pengiriman pesan dua arah seputar medikasi atau diagnosa.
2. Memberikan konfirmasi medikasi.
3. Mengirimkan pesan pemberitahuan kepada pasien.

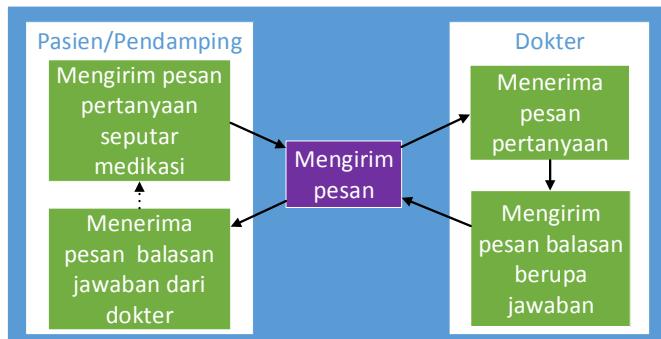
Proses bisnis pengiriman pesan tanya jawab bersifat dua arah, yakni pendamping pasien dan dokter. Sedangkan, proses bisnis pesan pemberitahuan dan konfirmasi bersifat satu arah.



Gambar 3. 2. Gambaran rangkaian proses bisnis secara keseluruhan

3.1.3.1 Detil Proses Bisnis Pengiriman Pesan

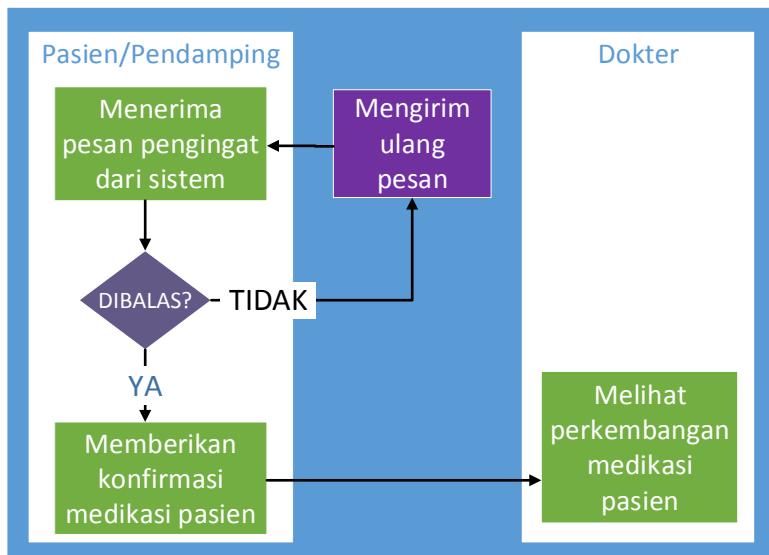
Pengiriman pesan bersifat dua arah. Awalan dari proses bisnis ini bisa berawal dari pasien maupun dokter.



Gambar 3. 3. Proses bisnis pengiriman pesan pertanyaan

Sistem perpesanan tidak hanya mengakomodasi pertanyaan, namun saran serta informasi lain yang mungkin dibutuhkan oleh kedua entitas. Seperti pertanyaan tentang obat, perkembangan medikasi pasien, umur, dan berat badan. Perpesanan merupakan penghubung komunikasi antar kedua pihak tanpa memberikan informasi pribadi selain untuk kepentingan terapi.

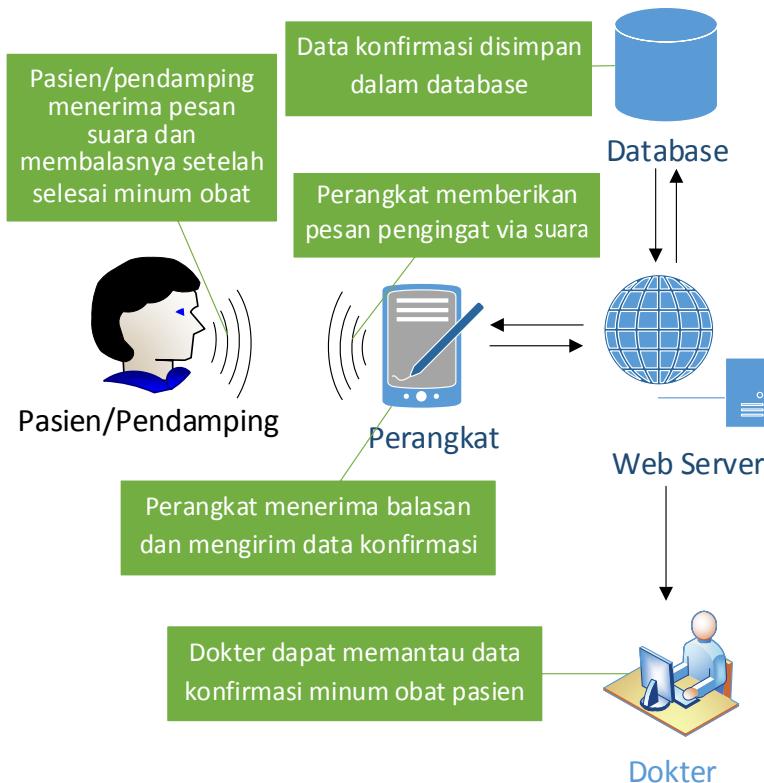
3.1.3.2 Detil Proses Memberikan Konfirmasi Medikasi



Gambar 3. 4. Proses bisnis memberikan konfirmasi medikasi

Pada gambar 3.4 ditunjukkan detil proses konfirmasi medikasi pasien oleh pasien atau pendamping. Sebelumnya, sistem mengirimkan sebuah pesan pengingat berupa suara kepada pasien atau pendamping agar pasien melakukan medikasi sesuai yang telah dijadwalkan. Setelah menerima pesan pengingat, sistem akan menunggu balasan dari pendamping pasien selama batas waktu yang ditentukan. Apabila pasien atau pendamping tidak

membalas atau tidak memberikan konfirmasi, maka sistem akan mengirimkan ulang pesan pengingat. Jika pasien atau pendamping telah memberikan konfirmasi, sistem akan menyimpan pesan konfirmasi bahwa medikasi telah dilakukan yang nantinya dapat dilihat oleh dokter.

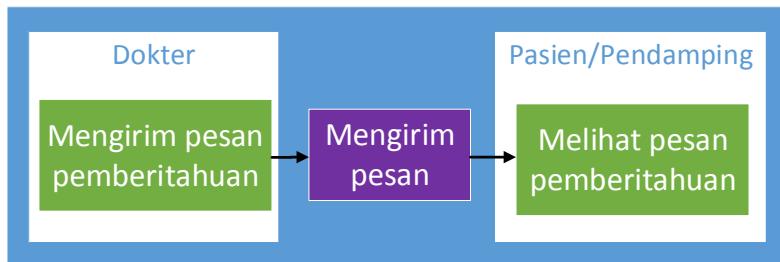


Gambar 3.5. Cara kerja sistem pengingat medikasi pasien

Gambar 3.5. mengilustrasikan cara kerja sistem pengingat medikasi pasien. Ketika waktu minum obat telah tiba, perangkat akan memberikan pengingat kepada pasien melalui suara. Pasien diharapkan membalas dan memberikan konfirmasi melalui suara.

Setelah memberikan konfirmasi, perangkat menyalurkan data ke *web server* dan data konfirmasi disimpan pada *database*. Dokter dapat memantau data tersebut ketika dibutuhkan.

3.1.3.3 Detil Proses Pengiriman Pesan Pemberitahuan

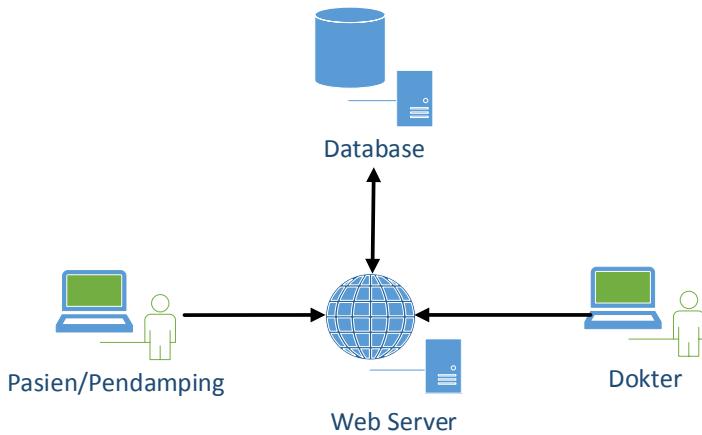


Gambar 3.6. Proses bisnis pengiriman pesan pemberitahuan

Gambar 3.6. memperlihatkan alur proses pengiriman pesan pemberitahuan kepada pasien oleh dokter. Pesan ini bersifat satu arah dengan tujuan dokter dapat mengirim pesan kepada pasien yang tidak membutuhkan balasan. Seperti, pesan untuk kontrol, pesan memo, dan lain sebagainya.

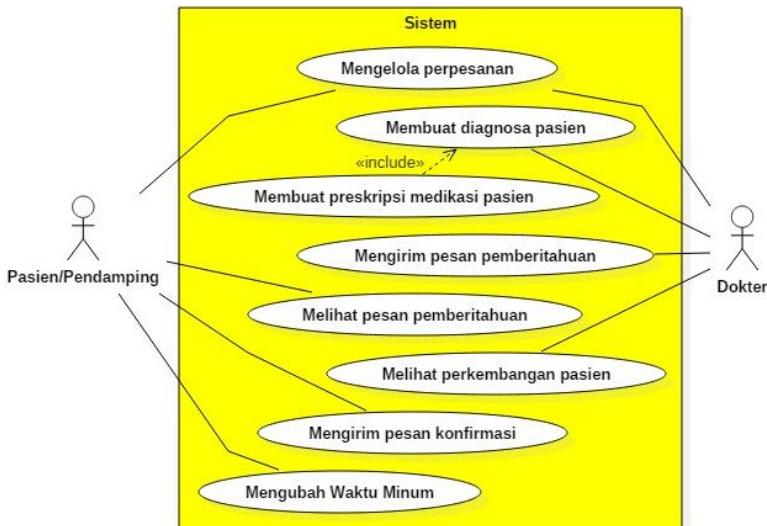
3.1.4 Arsitektur Sistem

Arsitektur sistem menggunakan arsitektur *client-server two tier*. *Client* dapat terhubung ke *server* menggunakan perangkat bergerak atau *desktop* dengan *browser*. Disarankan menggunakan *browser* seperti Mozilla Firefox atau Google Chrome versi 39 keatas agar dapat menggunakan fitur pengenalan wicara (*speech recognition*) dan pengucapan (*speech synthesis*). *Web server* yang digunakan adalah APACHE yang ter-hubung dengan *database* MySQL. Arsitektur sistem dari aplikasi ditunjukkan pada Gambar 3.7.



Gambar 3. 7. Arsitektur sistem

3.1.5 Kebutuhan Fungsional



Gambar 3. 8. Diagram kasus penggunaan

Kebutuhan fungsionalitas sistem dapat digambarkan pada gambar 3.8 dan dijelaskan pada tabel 3.1.

Tabel 3. 1. Deskripsi kasus penggunaan

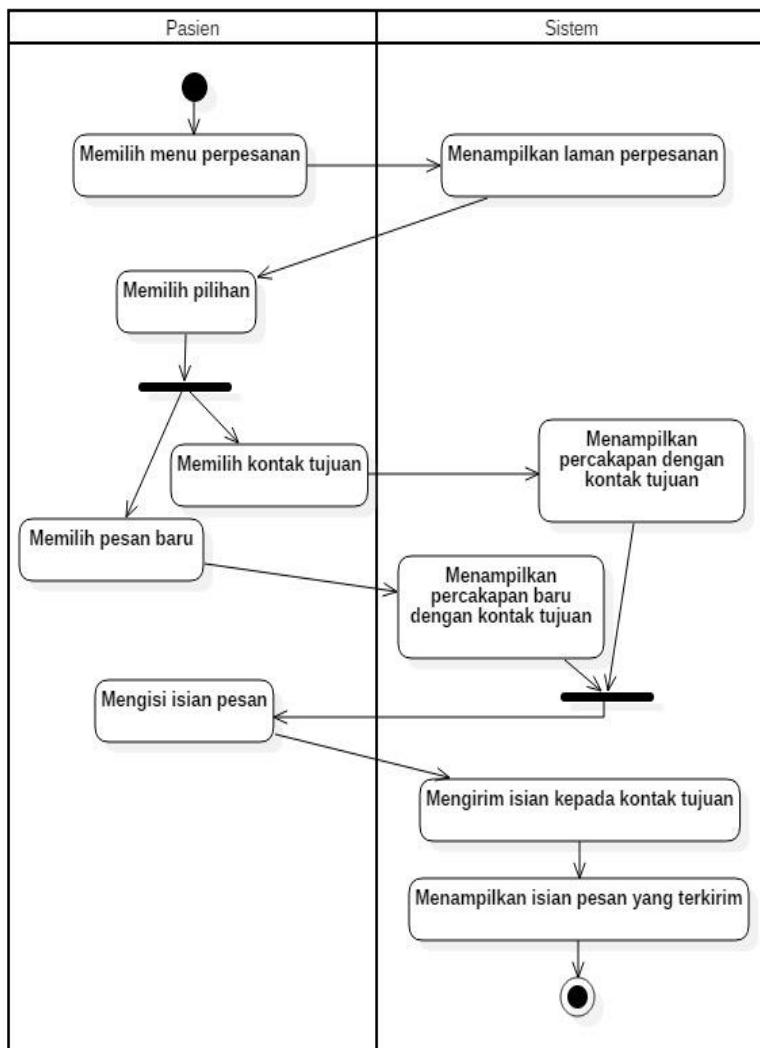
No.	Kode	Nama	Keterangan
1	UC-001	Mengelola Perpesanan	Fitur dimana pasien dan dokter melakukan komunikasi dua arah via pesan teks
2	UC-002	Membuat diagnosa pasien	Fitur dimana dokter membuat diagnosa atas keluhan pasien
3	UC-003	Membuat preskripsi medikasi pasien	Fitur dimana dokter membuat preskripsi untuk pasien setelah diagnosa dibuat
4	UC-004	Mengirim pesan pemberitahuan	Fitur dimana dokter mengirimkan pesan pemberitahuan yang tidak memerlukan balasan pasien
5	UC-005	Melihat pesan pemberitahuan	Fitur dimana pasien dapat membaca pesan pemberitahuan yang dibuat oleh dokter
6	UC-006	Melihat perkembangan pasien	Fitur dimana dokter dapat memantau medikasi pasien
7	UC-007	Mengirim pesan konfirmasi	Fitur dimana pasien memberitahukan sistem bahwa pasien telah minum obat setelah pesan pengingat diterima oleh pasien
8	UC-008	Mengubah Waktu Minum	Fitur dimana pasien atau pendamping mengubah jadwal minum atau konfirmasi waktu minum obat

3.1.5.1 Deskripsi Kasus Kebutuhan UC-001

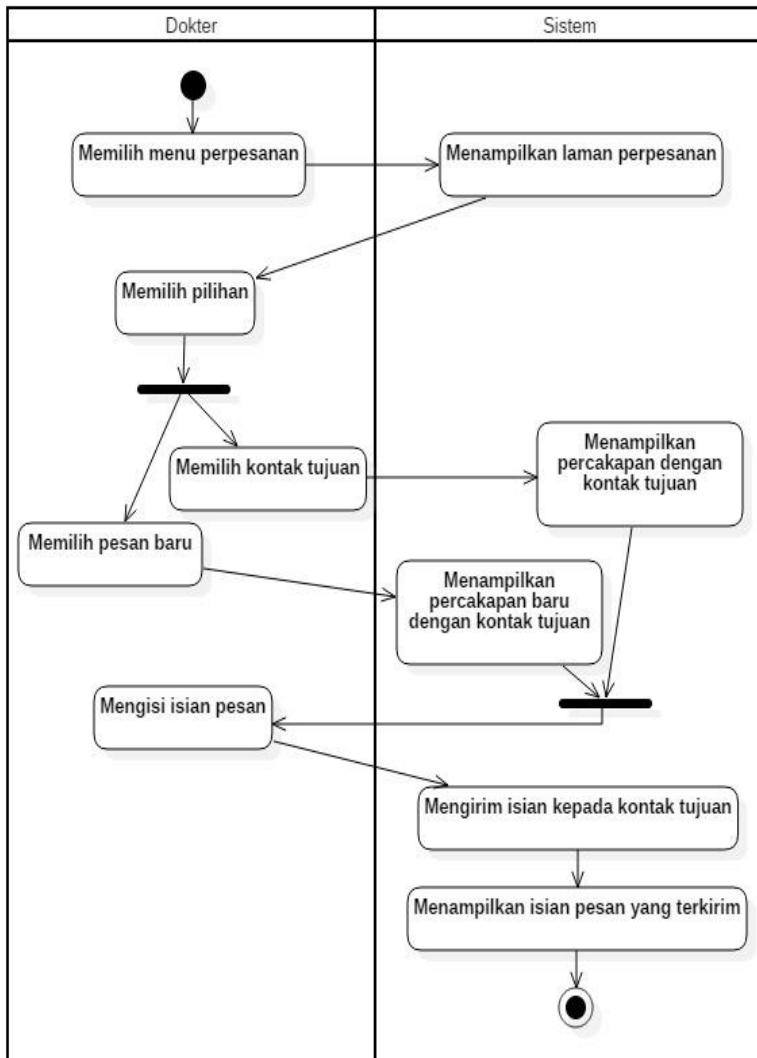
Kasus kebutuhan kode UC-001 adalah kasus kebutuhan mengelola perpesanan. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.2 dan diagram aktifitas kasus dijelaskan pada Gambar 3.8 untuk pasien, dan Gambar 3.9 untuk dokter.

Tabel 3. 2. Rincian alur kasus penggunaan UC-001

Nama use case	Mengelola perpesanan
Kode	UC-001
Aktor	Dokter, Pasien
Deskripsi	Pasien dan dokter melakukan komunikasi dua arah via pesan teks
Relasi	-
Kondisi Awal	Perpesanan belum dibuka
Kondisi Akhir	Pesan berhasil dikirim
Alur Normal	<ol style="list-style-type: none"> 3. Pengguna memilih menu perpesanan. 4. Sistem menampilkan laman perpesanan. 5. Pengguna memilih kontak tujuan. <ol style="list-style-type: none"> A1. Pengguna memilih menu pesan baru. 6. Sistem menampilkan percakapan dengan kontak. 7. Pengguna mengisi isian pesan. 8. Sistem mengirim isian pesan kepada kontak tujuan. 9. Sistem menampilkan isian pesan yang telah terkirim.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih menu pesan baru. <ol style="list-style-type: none"> 1. Sistem menampilkan percakapan baru dengan kontak tujuan. 2. Berlanjut ke alur normal 5.



Gambar 3. 8. Diagram aktivitas mengelola perpesanan untuk pengguna pasien



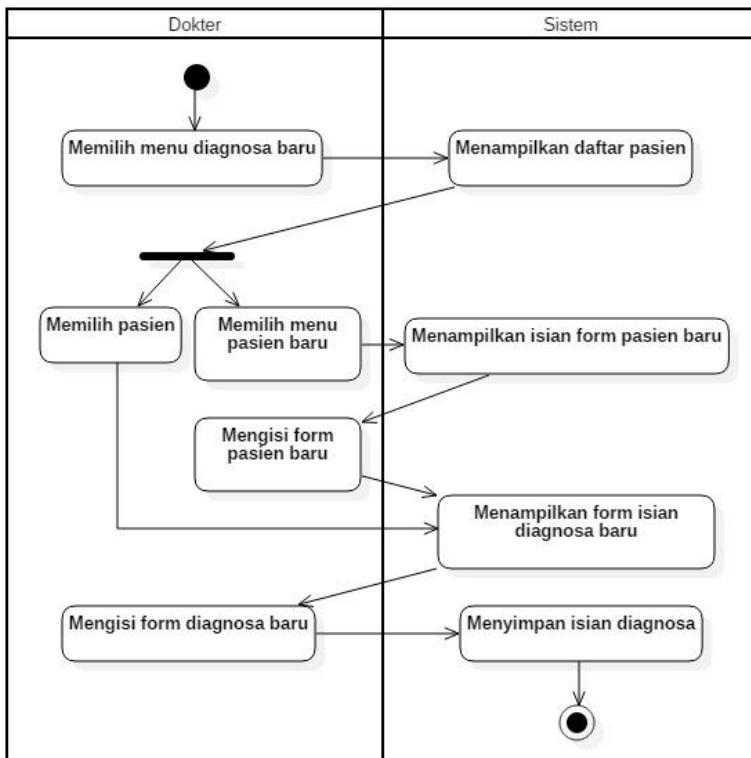
Gambar 3. 9. Diagram aktivitas mengelola perpesanan untuk pengguna dokter

3.1.5.2 Deskripsi Kasus Kebutuhan UC-002

Kasus kebutuhan kode UC-002 merupakan kasus kebutuhan membuat diagnosa pasien. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.3 dan diagram aktifitas kasus dijelaskan pada Gambar 3.10.

Tabel 3. 3. Rincian alur kasus penggunaan UC-002

Nama Use Case	Membuat diagnosa pasien
Kode	UC-002
Aktor	Dokter
Deskripsi	Dokter membuat diagnosa atas keluhan pasien
Relasi	-
Kondisi Awal	Diagnosa belum dibuat
Kondisi Akhir	Diagnosa berhasil disimpan
Alur Normal	<ol style="list-style-type: none"> Pengguna memilih menu diagnosa baru. Sistem menampilkan daftar pasien. Pengguna memilih pasien. <ol style="list-style-type: none"> Pengguna memilih menu pasien baru. Sistem menampilkan form isian diagnosa baru. Pengguna mengisi form diagnosa baru. Sistem menyimpan isian diagnosa.
Alur Alternatif	<ol style="list-style-type: none"> Pengguna memilih menu pasien baru. <ol style="list-style-type: none"> Sistem menampilkan isian form pasien baru. Pengguna mengisi form pasien baru. Berlanjut ke alur normal 4



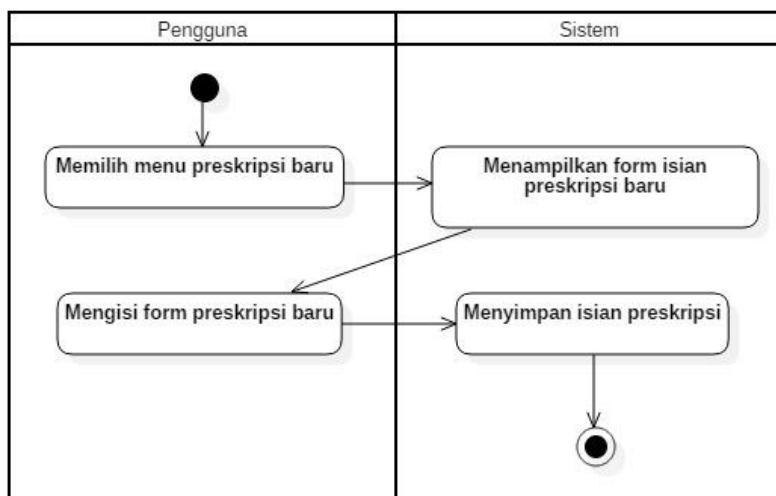
Gambar 3. 10. Diagram aktivitas membuat diagnosa baru

3.1.5.3 Deskripsi Kasus Kebutuhan UC-003

Kasus kebutuhan kode UC-003 merupakan kasus kebutuhan membuat preskripsi medikasi pasien. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.4 dan diagram aktifitas kasus dijelaskan pada Gambar 3.11.

Tabel 3. 4. Rincian alur kasus penggunaan UC-003

Nama Use Case	Membuat preskripsi medikasi pasien
Kode	UC-003
Aktor	Dokter
Deskripsi	Dokter membuat preskripsi untuk pasien setelah diagnosa dibuat
Relasi	Include
Kondisi Awal	Diagnosa telah dibuat atau dibuka
Kondisi Akhir	Preskripsi berhasil disimpan
Alur Normal	<ol style="list-style-type: none"> Pengguna memilih menu preskripsi baru. Sistem menampilkan form isian preskripsi baru. Pengguna mengisi form preskripsi baru. Sistem menyimpan isian preskripsi.

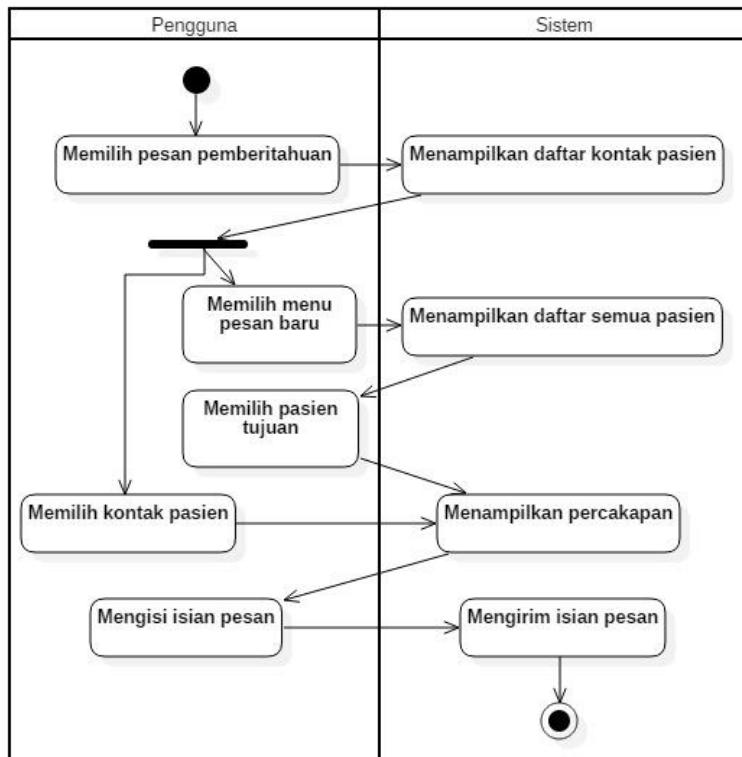
**Gambar 3. 11. Diagram aktivitas membuat preskripsi medikasi pasien**

3.1.5.4 Deskripsi Kasus Kebutuhan UC-004

Kasus kebutuhan kode UC-004 merupakan kasus kebutuhan mengirim pesan pemberitahuan. Rincian alur kasus menge-lola perpesanan dijelaskan pada Tabel 3.5 dan diagram aktifitas kasus dijelaskan pada Gambar 3.12.

Tabel 3. 5. Rincian alur kasus penggunaan UC-004

Nama Use Case	Mengirim pesan pemberitahuan
Kode	UC-004
Aktor	Dokter
Deskripsi	Dokter mengirimkan pesan pemberitahuan yang tidak memerlukan balasan kepada pasien
Relasi	-
Kondisi Awal	Pesan pemberitahuan belum dibuat
Kondisi Akhir	Pesan pemberitahuan telah dikirim
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pesan pemberitahuan. 2. Sistem menampilkan daftar kontak pasien. 3. Pengguna memilih kontak pasien. <ol style="list-style-type: none"> A1. Pengguna memilih menu pesan baru. 4. Sistem menampilkan percakapan. 5. Pengguna mengisi isian pesan. 6. Sistem mengirim isian pesan.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih menu pesan baru. <ol style="list-style-type: none"> 1. Sistem menampilkan daftar semua pasien. 2. Pengguna memilih pasien tujuan. 3. Berlanjut ke alur normal 4



Gambar 3. 12. Diagram aktivitas mengirim pesan pemberitahuan

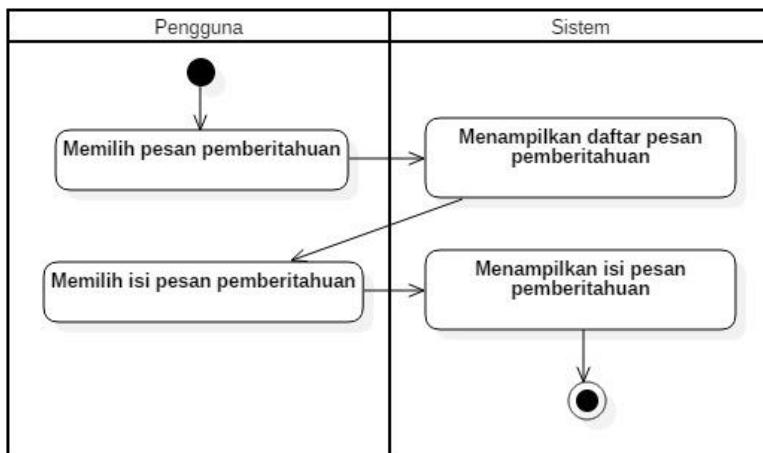
3.1.5.5 Deskripsi Kasus Kebutuhan UC-005

Kasus kebutuhan kode UC-005 adalah kebutuhan melihat pesan pemberitahuan. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.6 dan diagram aktifitas kasus dijelaskan pada Gambar 3.13.

Tabel 3. 6. Rincian alur kasus penggunaan UC-005

Nama Use Case	Melihat pesan pemberitahuan
---------------	-----------------------------

Kode	UC-005
Aktor	Pasien
Deskripsi	Pasien dapat membaca pesan pemberitahuan yang dibuat oleh dokter
Relasi	-
Kondisi Awal	Pesan pemberitahuan belum dibaca
Kondisi Akhir	Pesan pemberitahuan dibuka
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih pesan pemberitahuan. 2. Sistem menampilkan daftar pesan pemberitahuan. 3. Pengguna memilih isi pesan pemberitahuan. 4. Sistem menampilkan isi pesan pemberitahuan.



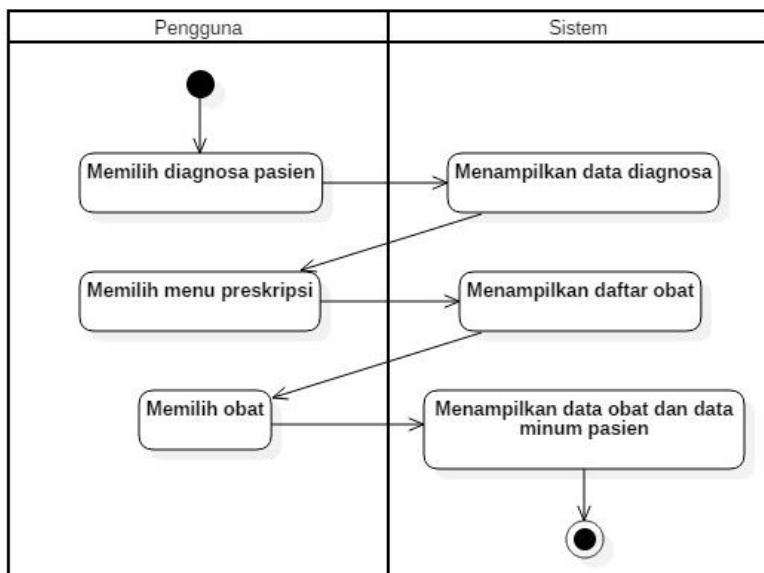
Gambar 3. 13. Diagram aktivitas melihat pesan pemberitahuan

3.1.5.6 Deskripsi Kasus Kebutuhan UC-006

Kasus kebutuhan kode UC-006 merupakan kasus kebutuhan melihat perkembangan pasien. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.7 dan diagram aktifitas kasus dijelaskan pada Gambar 3.14.

Tabel 3. 7. Rincian alur kasus penggunaan UC-006

Nama Use Case	Melihat perkembangan pasien
Kode	UC-006
Deskripsi	Dokter dapat memantau perkembangan medikasi pasien lewat preskripsi yang telah diberikan
Relasi	-
Aktor	Dokter
Kondisi Awal	Data diagnosa pasien belum dibuka
Kondisi Akhir	Data obat dan histori minum obat dibuka
Alur Normal	<ol style="list-style-type: none"> Pengguna memilih diagnosa pasien. Sistem menampilkan data diagnosa. Pengguna memilih menu preskripsi. Sistem menampilkan daftar obat. Pengguna memilih obat. Sistem menampilkan data obat dan data minum pasien.

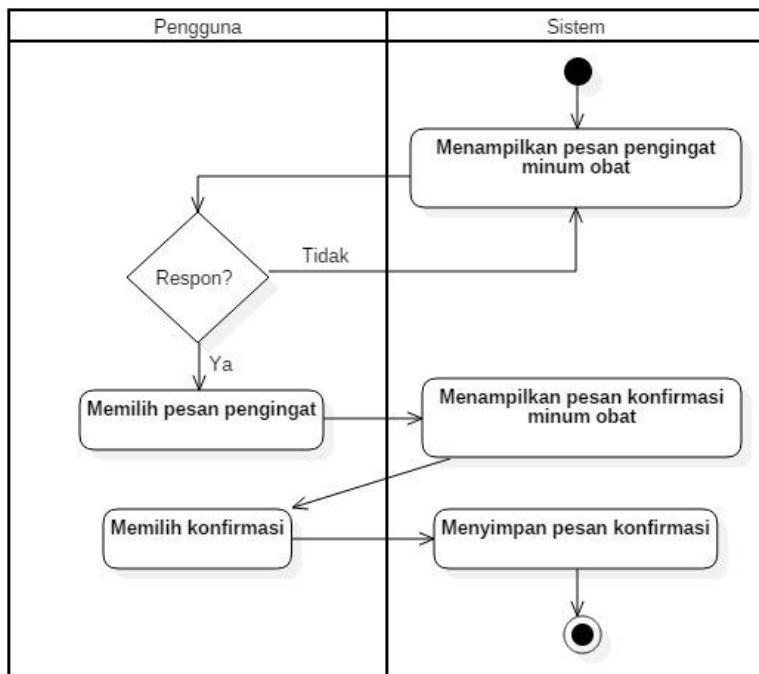
**Gambar 3. 14. Diagram aktivitas melihat perkembangan pasien**

3.1.5.7 Deskripsi Kasus Kebutuhan UC-007

Kasus kebutuhan kode UC-007 merupakan kasus kebutuhan mengirim pesan konfirmasi. Rincian alur kasus mengelola perpesanan dijelaskan pada Tabel 3.8 dan diagram aktifitas kasus dijelaskan pada Gambar 3.15.

Tabel 3. 8. Rincian alur kasus penggunaan UC-007

Nama Use Case	Mengirim pesan konfirmasi
Kode	UC-007
Aktor	Pasien
Deskripsi	Pasien memberitahukan sistem bahwa pasien telah minum obat setelah pesan pengingat diterima oleh pasien
Relasi	-
Kondisi Awal	Jadwal minum obat telah tiba
Kondisi Akhir	Pesan konfirmasi telah disimpan
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pesan pengingat minum obat. 2. Pengguna memilih pesan pengingat. A1. Pengguna tidak merespon pada pesan pengingat. 3. Sistem menampilkan pesan konfirmasi minum obat. 4. Pengguna memilih konfirmasi. 5. Sistem menyimpan pesan konfirmasi.
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna tidak merespon pada pesan pengingat. <ol style="list-style-type: none"> 1. Kembali ke alur normal 1.



Gambar 3. 15. Diagram aktivitas melihat perkembangan pasien

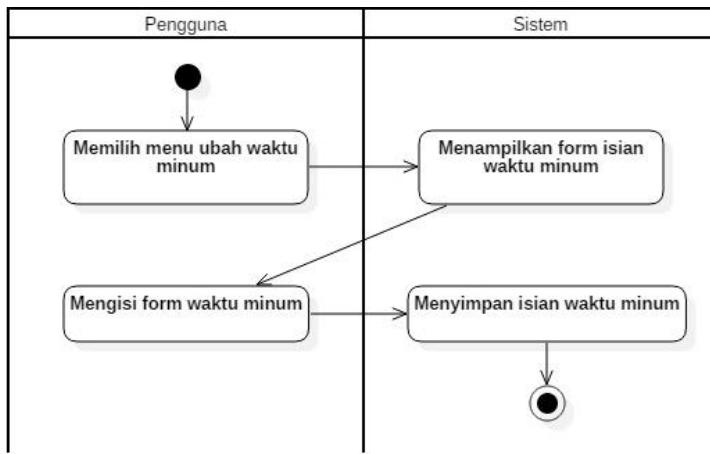
3.1.5.8 Deskripsi Kasus Kebutuhan UC-008

Kasus kebutuhan kode UC-008 merupakan kasus kebutuhan mengubah waktu minum. Rincian alur kasus mengubah waktu minum dijelaskan pada Tabel 3.9 dan diagram aktifitas kasus dijelaskan pada Gambar 3.16.

Tabel 3. 9. Rincian alur kasus penggunaan UC-008

Nama Use Case	Mengubah waktu minum
Kode	UC-008
Aktor	Pasien

Deskripsi	Pasien dapat mengubah jadwal minum atau waktu konfirmasi minum obat
Relasi	-
Kondisi Awal	Pengguna berada pada halaman data obat
Kondisi Akhir	Waktu minum telah disimpan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih opsi ubah waktu minum. 2. Sistem menampilkan formulir waktu minum obat. 3. Pengguna mengisi formulir isian. 4. Sistem menyimpan isian formulir.



Gambar 3. 16. Diagram aktivitas mengubah waktu minum

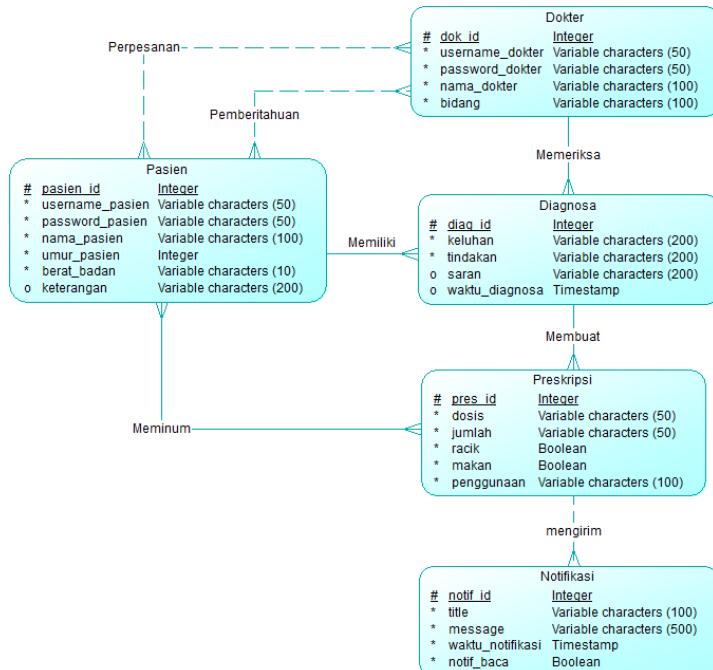
3.2 Perancangan Sistem

Subbab ini membahas tentang perancangan sistem. Topik yang akan dibahas meliputi perancangan basis data dan perancangan antarmuka. Pembahasan lebih detil akan dibahas berikut ini.

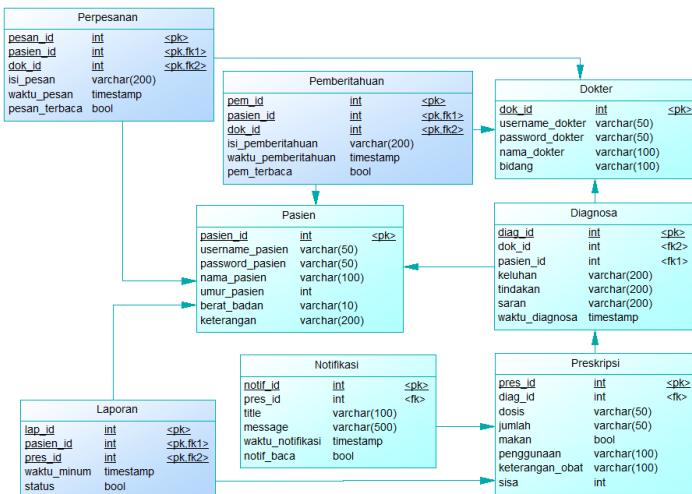
3.2.1 Perancangan Basis Data

Pada subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada sistem. Basis data pada sistem yang ini menggunakan sistem manajemen basis data relasional MySQL. Basis data memiliki lima entitas pada *Conceptual Data Model* (CDM). Setiap entitas memiliki atribut-atribut isi tabel dan variabel yang membangun tabel basis data. Entitas memiliki relasi dengan entitas lainnya untuk membentuk kolaborasi data. Adanya relasi *many-to-many* memunculkan beberapa tabel transaksi baru yang dapat dijabarkan pada *Physical Data Model* (PDM).

Conceptual Data Model dan *Physical Data Model* sebagai dari perancangan basis data sistem ini dijabarkan pada Gambar 3.17 dan Gambar 3.18.



Gambar 3. 17. Conceptual Data Model



Gambar 3. 18. Physical Data Model

3.2.1.1 Rancangan Tabel Pasien

Tabel Pasien digunakan untuk menyimpan data pasien. Tabel Pasien berisi data-data pribadi yang akan digunakan sebagai identitas dan kepentingan diagnosa. Detil tabel Pasien dijelaskan pada Tabel 3.10.

Tabel 3. 10. Atribut tabel Pasien

Nama Kolom	Tipe Data	Keterangan
pasien_id	integer	<i>Primary Key</i> dari tabel pasien
username_pasien	varchar(50)	<i>Username</i> untuk pasien
password_pasien	varchar(50)	<i>Password</i> untuk pasien
nama_pasien	varchar(100)	Nama pasien
umur_pasien	integer	Umur pasien
berat_badan	varchar(10)	Berat badan pasien
keterangan	varchar(200)	Keterangan pasien. Dicantumkan apabila ada beberapa keterangan tambahan yang berguna untuk diagnosa pasien akan datang

3.2.1.2 Rancangan Tabel Dokter

Tabel Dokter digunakan untuk menyimpan data dokter. Tabel Dokter berisi data dokter sebagai identitas yang dibutuhkan dokter lain dan pasien. Detil tabel Dokter dijelaskan pada Tabel 3.11.

Tabel 3. 11. Atribut tabel Dokter

Nama Kolom	Tipe Data	Keterangan
dok_id	integer	<i>Primary Key</i> dari tabel dokter
username_dokter	varchar(50)	<i>Username</i> untuk dokter
password_dokter	varchar(50)	<i>Password</i> untuk dokter
nama_dokter	varchar(100)	Nama dokter
bidang	varchar(100)	Keahlian dokter

3.2.1.3 Rancangan Tabel Diagnosa

Tabel Diagnosa digunakan untuk menyimpan data diagnosa pasien. Tabel Diagnosa diisi oleh Dokter yang berisi data keluhan pasien, komplikasi, dan tindakan yang diambil. Detil tabel Diagnosa dijelaskan pada Tabel 3.12.

Tabel 3. 12. Atribut tabel Diagnosa

Nama Kolom	Tipe Data	Keterangan
diag_id	integer	<i>Primary Key</i> dari tabel diagnosa
dok_id	integer	<i>Foreign Key</i> dari tabel dokter
pasien_id	integer	<i>Foreign Key</i> dari tabel pasien
keluhan	varchar(200)	Keluhan pasien
tindakan	varchar(200)	Tindakan diagnosa
saran	varchar(200)	Saran untuk pasien. Berisi tindakan yang bisa atau harus pasien lakukan selama masa terapi
waktu_diagnosa	timestamp	<i>Timestamp</i> untuk waktu diagnosa

3.2.1.4 Rancangan Tabel Preskripsi

Tabel Preskripsi digunakan untuk menyimpan data obat. Tabel Preskripsi diisi oleh dokter setelah mengisi data diagnosa. Data obat berisi dosis, jumlah, jenis obat, dan aturan pakai atau penggunaan obat. Detil tabel Preskripsi dijelaskan pada Tabel 3.13.

Tabel 3. 13. Atribut tabel Preskripsi

Nama Kolom	Tipe Data	Keterangan
pres_id	integer	<i>Primary Key</i> dari tabel preskripsi
diag_id	integer	<i>Foreign Key</i> dari tabel diagnosa
dosis	varchar(50)	Dosis obat dalam satu kali minum
jumlah	varchar(50)	Jumlah obat yang diberikan
makan	bool	Aturan makan obat
penggunaan	varchar(100)	Aturan penggunaan obat
keterangan_obat	varchar(100)	Keterangan obat. Diisi apabila ada penggunaan khusus atau peringatan obat.
sisa	integer	Sisa obat yang belum diminum (diambil dari jumlah - diminum)

3.2.1.5 Rancangan Tabel Notifikasi

Tabel Notifikasi digunakan untuk menyimpan data pesan pengingat. Tabel Notifikasi diisi oleh sistem setelah dokter mengisi data preskripsi. Notifikasi berisi waktu notifikasi yang sesuai dengan aturan penggunaan obat dari tabel Preskripsi beserta atribut notif_baca sebagai penanda apabila pasien telah membuka pesan pengingat notifikasi atau tidak. Isi pesan dalam tabel juga berfungsi sebagai pengiriman pesan pengingat yang terhubung dengan OneSignal. Detil tabel Notifikasi dijelaskan pada Tabel 3.14.

Tabel 3. 14. Atribut tabel Notifikasi

Nama Kolom	Tipe Data	Keterangan
notif_id	integer	<i>Primary Key</i> dari tabel notifikasi
pres_id	integer	<i>Foreign Key</i> dari tabel preskripsi
title	varchar(100)	Judul notifikasi (disesuaikan dengan spesifikasi API OneSignal)
message	varchar(500)	Isi pesan pengingat (disesuaikan dengan spesifikasi API OneSignal)
waktu_notifikasi	timestamp	Aturan makan obat
notif_baca	bool	Aturan penggunaan obat
onesignal_id	varchar(50)	Kode unik notifikasi OneSignal (disesuaikan dengan spesifikasi API OneSignal)

3.2.1.6 Rancangan Tabel Perpesanan

Tabel Perpesanan merupakan tabel transaksi antara tabel Dokter dan tabel Pasien. Relasi antar tabel adalah *many-to-many*, menghasilkan tabel Perpesanan dengan beberapa atribut tambahan pada tabel. Tabel Perpesanan esensial untuk menyimpan pesan yang dikirimkan oleh dokter dan pasien. Tabel Perpesanan mengakomodasi fitur komunikasi dua arah. Detil tabel Notifikasi akan dijelaskan pada Tabel 3.15.

Tabel 3. 15. Atribut tabel Perpesanan

Nama Kolom	Tipe Data	Keterangan
pesan_id	integer	<i>Primary Key</i> dari tabel pasien
pasien_id	integer	<i>Foreign Key</i> dari tabel pasien
dok_id	integer	<i>Foreign Key</i> dari tabel dokter
isi_pesan	varchar(200)	Isi pesan
waktu_pesan	timestamp	<i>Timestamp</i> untuk waktu pesan
sender	varchar(10)	Sebagai penanda pengirim pesan
pesan_terbaca	bool	Sebagai penanda untuk pesan

		terbaca atau belum terbaca
--	--	----------------------------

3.2.1.7 Rancangan Tabel Pemberitahuan

Tabel Pemberitahuan merupakan tabel transaksi antara tabel Dokter dan tabel Pasien. Relasi antar tabel adalah many-to-many, menghasilkan tabel Pemberitahuan dengan beberapa atribut tambahan pada tabel. Tabel Pemberitahuan berguna untuk menyimpan pesan pemberitahuan dari dokter kepada pasien, dimana pesan tidak membutuhkan balasan pasien (satu arah). Detil tabel Pemberitahuan dijelaskan pada Tabel 3.15.

Tabel 3. 15. Atribut tabel Pemberitahuan

Nama Kolom	Tipe Data	Keterangan
pem_id	integer	<i>Primary Key</i> dari tabel pemberitahuan
pasien_id	integer	<i>Foreign Key</i> dari tabel pasien
dok_id	integer	<i>Foreign Key</i> dari tabel dokter
isi_pemberitahuan	varchar(200)	Isi pesan pemberitahuan
waktu_pemberitahuan	timestamp	<i>Timestamp</i> untuk waktu pesan pemberitahuan
pem_terbaca	bool	Sebagai penanda untuk pesan terbaca atau belum terbaca

3.2.1.8 Rancangan Tabel Laporan

Tabel Laporan merupakan tabel transaksi antara tabel Pasien dan tabel Preskripsi. Relasi antar tabel adalah *many-to-many*, menghasilkan tabel Laporan dengan beberapa atribut tambahan pada tabel. Tabel Laporan menyimpan laporan konfirmasi minum obat pasien. Detil tabel Laporan dijelaskan pada Tabel 3.16.

Tabel 3. 16. Atribut tabel Laporan

Nama Kolom	Tipe Data	Keterangan
lap_id	integer	<i>Primary Key</i> dari tabel laporan
pasien_id	integer	<i>Foreign Key</i> dari tabel pasien
pres_id	integer	<i>Foreign Key</i> dari tabel preskripsi
waktu_minum	timestamp	<i>Timestamp</i> untuk waktu pesan minum obat
status	bool	Sebagai penanda apabila waktu minum obat dilaksanakan atau tidak

3.2.2 Penggunaan *Third Party*

Third party adalah penggunaan program diluar sistem yang terlibat dalam pembuatan sistem atau berjalananya sistem. Sebagai pendukung dalam pengembangan, penulis menggunakan beberapa *third party*. *Third Party* juga mendukung proses bisnis serta sebagai pendukung kebutuhan fungsional sistem. Detil *third party* yang digunakan akan dijelaskan pada subbab berikutnya.

3.2.2.1 OneSignal Web Push Notification

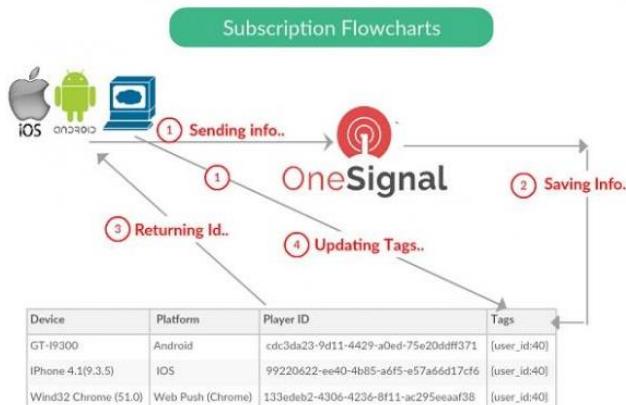
OneSignal merupakan layanan *push notification* untuk berbagai macam *platform*. *Platform* yang didukung saat ini adalah Apple IOS, Google Android, Windows Phone 8.0/8.1, Amazon Fire, Mac OS X, Google chrome Apps & Extension, dan *Website Push Notifications*. Sistem akan menggunakan layanan ini untuk platform *Website Push Notification* pada Google Chrome dan Mozilla Firefox [13].

Keunggulan OneSignal Web Push Notification adalah kemampuan pengiriman notifikasi saat *browser* perangkat tidak berada di tab situs atau ketika *browser* sedang ditutup namun berjalan pada *background* (latar belakang). Hal ini penulis gunakan karena adanya keterbatasan Web Notification API dalam

pengiriman notifikasi, yaitu *browser* harus dalam keadaan terbuka dan dalam keadaan tab situs terbuka.

OneSignal menggunakan sistem *subscriptions* (langganan). Pengguna harus mendaftarkan atau berlangganan untuk menerima *push notification* dari OneSignal. Setiap pengguna yang terdaftar memiliki kode unik *Player_ID* untuk menandakan *platform* yang digunakan dan *tags* sebagai *identifier* yang ditentukan oleh pengembang. Nantinya, *tags* dibutuhkan untuk pengiriman notifikasi kepada tujuan tertentu sesuai yang telah didefinisikan oleh pengembang [14].

OneSignal Flow



Gambar 3. 19. Diagram alir OneSignal

3.2.2.2 Artyom.js

Artyom.js merupakan *wrapper* dari Web Speech API yang ditulis dalam bahasa JavaScript. Artyom.js memanfaatkan library *speechSynthesis* dan *webkitSpeechRecognition* dari Web Speech API yang memungkinkan pengembang untuk mengembangkan aplikasi berbasis *web* dengan teknologi wicara [15]. *Library speech synthesis* digunakan untuk menerjemahkan teks ke dalam suara (*Text-to-Speech*) dan *speech recognition* untuk menerjemahkan suara ke dalam teks (*Speech-to-Text*).

Artyom.js juga dilengkapi dengan berbagai contoh dan fungsi seperti perintah. Layaknya Google Voice, *wrapper* ini dapat membuat aplikasi dua arah antara pengguna dengan sistem. Karena berbasis Web Speech API, *wrapper* ini mendukung berbagai bahasa seperti Bahasa Indonesia. Sayangnya, seperti yang dijelaskan pada subbab 2.4, Web Speech API hanya mendukung beberapa *browser* saja seperti Google Chrome dan Mozilla Firefox. *Wrapper* tersedia gratis dan dapat langsung digunakan tanpa pemasangan program lain.

3.2.3 Perancangan Antarmuka Dokter

Pada subbab ini akan dibahas secara mendetil dari perancangan antarmuka pengguna dokter sesuai dengan kebutuhan fungsional yang dijabarkan pada tabel 3.1.

3.2.3.1 Antarmuka Mengelola Perpesan

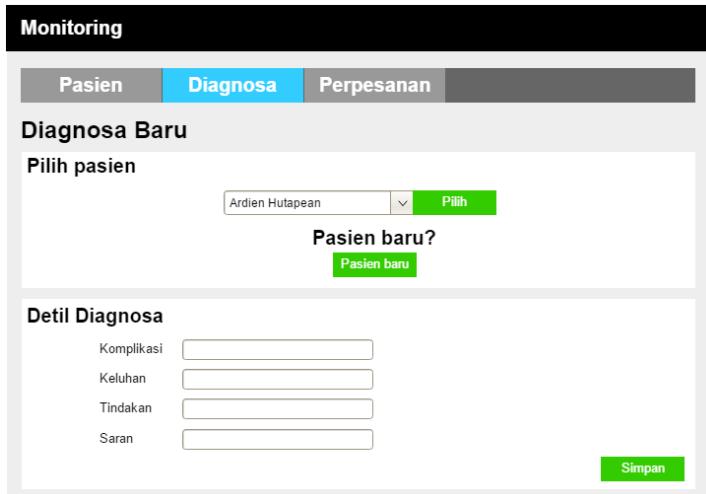
Perancangan antarmuka mengelola perpesanan ditunjukkan pada Gambar 3.20. Antarmuka mengakomodasi daftar kontak dan percakapan.



Gambar 3. 20. Rancangan antarmuka mengelola perpesanan

3.2.3.2 Antarmuka Membuat Diagnosa Pasien

Perancangan antarmuka membuat diagnosa pasien ditunjukkan pada Gambar 3.21. Dokter memilih pasien terlebih dahulu kemudian mengisi isian detil diagnosa.



Gambar 3. 21. Rancangan antarmuka membuat diagnosa pasien

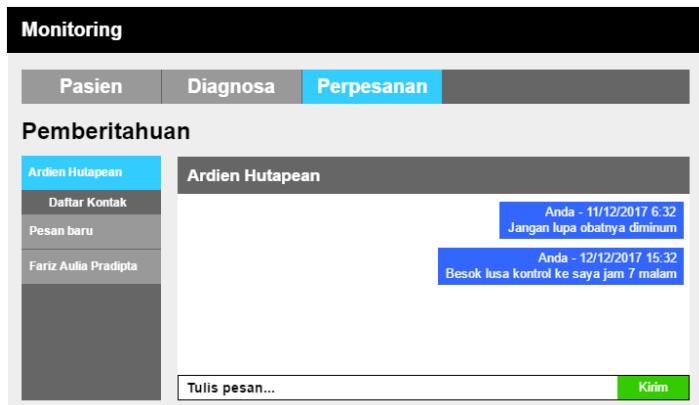
3.2.3.3 Antarmuka Membuat Preskripsi Pasien

Perancangan antarmuka membuat preskripsi pasien ditunjukkan pada Gambar 3.22. Pembuatan preskripsi obat dilakukan oleh dokter. Data preskripsi pasien dibuat setelah data diagnosa dibuat. Kemudian data preskripsi sebagai data obat bagi pasien dan digunakan untuk melihat perkembangan minum obat dari pasien.

Gambar 3. 22. Rancangan antarmuka membuat preskripsi pasien

3.2.3.4 Antarmuka Mengirim Pesan Pemberitahuan

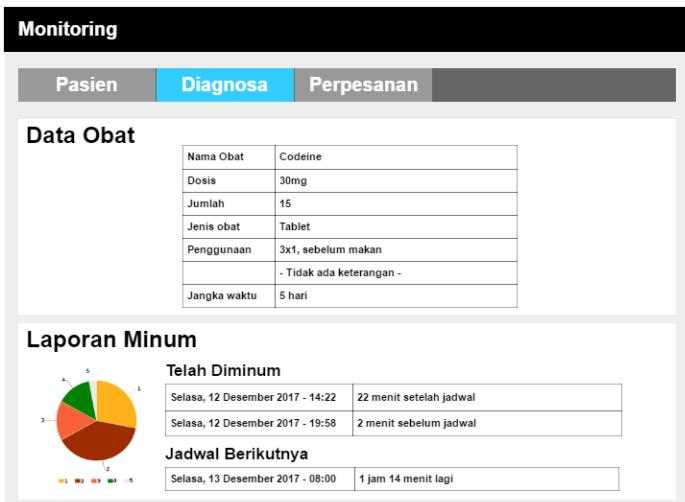
Perancangan antarmuka mengirim pesan pemberitahuan ditunjukkan pada Gambar 3.23. Pesan pemberitahuan dibuat oleh dokter kepada pasien. Pesan pemberitahuan merupakan pesan satu arah yang tidak membutuhkan balasan pasien.



Gambar 3. 23. Rancangan antarmuka mengirim pesan pemberitahuan

3.2.3.5 Antarmuka Melihat Perkembangan Pasien

Perancangan antarmuka melihat perkembangan pasien ditunjukkan pada Gambar 3.24.



Gambar 3. 24. Rancangan antarmuka melihat perkembangan pasien

3.2.4 Perancangan Antarmuka Pasien

Pada subbab ini akan dibahas secara mendetil dari perancangan antarmuka pengguna pasien sesuai dengan kebutuhan fungsional.

3.2.4.1 Antarmuka Mengelola Perpesanan

Perancangan antarmuka mengelola perpesanan ditunjukkan pada Gambar 3.25. Antarmuka mengakomodasi daftar kontak dan percakapan.



Gambar 3. 25. Rancangan antarmuka mengelola perpesanan

3.2.4.2 Antarmuka Melihat Pesan Pemberitahuan

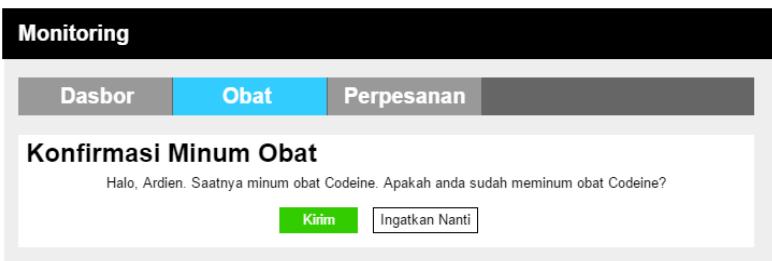
Perancangan antarmuka melihat pesan pemberitahuan ditunjukkan pada Gambar 3.26. Pasien hanya dapat membuka dan membaca pesan pemberitahuan, namun tidak dapat membalas pesan.



Gambar 3.26. Rancangan antarmuka melihat pesan pemberitahuan

3.2.4.3 Antarmuka Mengirim Pesan Konfirmasi

Perancangan antarmuka mengirim pesan konfirmasi ditunjukkan pada Gambar 3.27. Pasien melakukan konfirmasi pesan pengingat setelah sistem mengirim pesan kepada pasien.



Gambar 3. 27. Rancangan antarmuka mengirim pesan konfirmasi

3.2.4.4 Antarmuka Mengubah Waktu Minum

Perancangan antarmuka mengubah waktu minum ditunjukkan pada Gambar 3.28. Pasien atau pendamping dapat menentukan jadwal minum atau mengubah waktu konfirmasi minum obat.

Monitoring

Dasbor	Obat	Perpesanan														
Data Obat																
<table border="1"><tr><td>Nama Obat</td><td>Codeine</td></tr><tr><td>Dosis</td><td>30mg</td></tr><tr><td>Jumlah</td><td>15</td></tr><tr><td>Jenis obat</td><td>Tablet</td></tr><tr><td>Penggunaan</td><td>3x1, sebelum makan</td></tr><tr><td colspan="2">- Tidak ada keterangan -</td></tr><tr><td>Jangka waktu</td><td>5 hari</td></tr></table>			Nama Obat	Codeine	Dosis	30mg	Jumlah	15	Jenis obat	Tablet	Penggunaan	3x1, sebelum makan	- Tidak ada keterangan -		Jangka waktu	5 hari
Nama Obat	Codeine															
Dosis	30mg															
Jumlah	15															
Jenis obat	Tablet															
Penggunaan	3x1, sebelum makan															
- Tidak ada keterangan -																
Jangka waktu	5 hari															

Ubah Waktu Minum		
Waktu mulai		
Simpan		

Gambar 3. 28. Rancangan antarmuka mengubah waktu minum

BAB IV

IMPLEMENTASI

Bab ini akan menjelaskan implementasi dari perancangan sistem dari bab III. Dalam bab ini akan dibahas mengenai kode sumber program, lingkungan pembangunan sistem, implementasi antarmuka pengguna, dan implementasi kasus penggunaan pada sistem. Implementasi sistem mengacu pada perancangan yang ditulis pada bab III. Namun, tidak menutup kemungkinan adanya perubahan-perubahan dari rancangan tersebut apabila perubahan memang diperlukan.

4.1 Lingkungan Implementasi

Dalam merancang dan mengimplementasikan sistem ini, digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1. Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan sistem ini adalah sebagai berikut:

Tabel 4. 1. Spesifikasi lingkungan pengembangan

Perangkat	Spesifikasi	
Perangkat Pengembangan	Perangkat	: Laptop Toshiba Satellite L-50
	Prosesor	: Intel Core i5-4200U CPU @ 1.60 GHz
	Memori	: 4.00 GB
	Mikrofon	: Conexant SmartAudio HD
	OS	: Microsoft Windows 10 64-bit
Web Server	<i>Browser</i>	: Google Chrome 63
	<i>Web Server</i>	: Apache 2.4.12
	OS	: CloudLinux 6.8
	Prosesor	: Intel Xeon
	<i>Database</i>	: MySQL 5.5.52

4.1.2. Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada lingkungan pengembangan sistem adalah sebagai berikut:

- ❖ Windows 10 64-bit sebagai sistem operasi
- ❖ Atom Text Editor sebagai *text editor* untuk membangun aplikasi web.
- ❖ MySQL sebagai *database server*
- ❖ XAMPP sebagai *server* lokal untuk pengujian dan implementasi
- ❖ Google Chrome 63 sebagai *browser* dalam tahap pengujian dan pengembangan.
- ❖ PowerDesigner untuk perancangan *Conceptual Data Model* (CDM), *Physical Data Model* (PDM), dan perancangan basis data.
- ❖ StarUML untuk pembuatan perancangan sistem dan diagram saat implementasi perangkat lunak.

4.2 Implementasi Basis Data

Pada subbab ini akan dibahas implementasi dari rancangan basis data yang telah dibahas pada subbab 3.2. Pada subbab ini akan dibahas implementasi struktur basis data. Implementasi basis data menggunakan sintaks SQL.

4.2.1 Implementasi Tabel Pasien

Tabel pasien digunakan untuk menyimpan data pasien. Data digunakan untuk *login* masuk ke sistem dan digunakan sebagai data pribadi untuk diagnosa. Implementasi tabel pasien dapat dilihat pada Kode Sumber 4.1.

```
1. create table PASIEN
2. (
3.     PASIEN_ID          int not null AUTO_INCREMENT,
4.     USERNAME_PASIEN    varchar(50) not null,
5.     PASSWORD_PASIEN    varchar(50) not null,
```

```

6.      NAMA_PASIEN      varchar(100) not null,
7.      UMUR_PASIEN       int not null,
8.      BERAT_BADAN       varchar(10),
9.      KETERANGAN        varchar(200),
10.     primary key (PASIEN_ID)
11. );

```

Kode Sumber 4. 1. Implementasi struktur Tabel Pasien

4.2.2 Implementasi Tabel Dokter

Tabel Dokter digunakan untuk menyimpan data dokter. Data digunakan sebagai identitas pada sistem. Implementasi tabel dokter dapat dilihat pada Kode Sumber 4.2.

```

1.  create table DOKTER
2.  (
3.      DOK_ID           int not null AUTO_INCREMENT,
4.      USERNAME_DOKTER varchar(50) not null,
5.      PASSWORD_DOKTER varchar(50) not null,
6.      NAMA_DOKTER      varchar(100) not null,
7.      BIDANG           varchar(100) not null,
8.      primary key (DOK_ID)
9. );

```

Kode Sumber 4. 2. Implementasi struktur Tabel Dokter

4.2.3 Implementasi Tabel Diagnosa

Tabel Diagnosa digunakan untuk menyimpan data diagnosa. Sebagai tabel transaksi, tabel diagnosa memiliki dua *foreign key* dari tabel pasien dan tabel dokter. Implementasi tabel diagnosa dapat dilihat pada Kode Sumber 4.3.

```

1.  create table DIAGNOSA
2.  (
3.      DIAG_ID          int not null AUTO_INCREMENT,
4.      DOK_ID           int not null,
5.      PASIEN_ID        int not null,
6.      KELUHAN          varchar(200) not null,
7.      TINDAKAN         varchar(200) not null,

```

```

8.      SARAN           varchar(200),
9.      WAKTU_DIAGNOSA   timestamp,
10.     primary key (DIAG_ID)
11. );
12. alter table DIAGNOSA add constraint FK_MEMERIKSA foreign key
    (DOK_ID)
13.     references DOKTER (DOK_ID) on delete cascade on update c
    ascade;
14. alter table DIAGNOSA add constraint FK_MEMILIKI foreign key (
    PASIEN_ID)
15.     references PASIEN (PASIEN_ID) on delete cascade on updat
    e cascade;

```

Kode Sumber 4. 3. Implementasi struktur Tabel Diagnosa

4.2.4 Implementasi Tabel Preskripsi

Tabel Preskripsi digunakan untuk menyimpan data preskripsi. Tabel preskripsi memiliki *foreign key* dari tabel diagnosa. Implementasi tabel dapat dilihat pada Kode Sumber 4.4.

```

1. create table PRESKRIPSI
2. (
3.     PRES_ID          int not null AUTO_INCREMENT,
4.     DIAG_ID          int not null,
5.     DOSIS            varchar(50) not null,
6.     JUMLAH           varchar(50) not null,
7.     MAKAN            bool not null,
8.     PENGUNAAN        varchar(100) not null,
9.     KETERANGAN_OBAT  varchar(100),
10.    SISA              int,
11.    primary key (PRES_ID)
12. );
13. alter table PERPESANAN add constraint FK_PERPESANAN2 foreign
    key (DOK_ID)
14.     references DOKTER (DOK_ID) on delete cascade on update c
    ascade;
15. alter table PRESKRIPSI add constraint FK_MEMBUAT foreign key
    (DIAG_ID)
16.     references DIAGNOSA (DIAG_ID) on delete cascade on upda
    te cascade;

```

Kode Sumber 4. 4. Implementasi struktur Tabel Preskripsi

4.2.5 Implementasi Tabel Notifikasi

Tabel Notifikasi digunakan untuk menyimpan data pesan notifikasi atau pesan pengingat. Tabel notifikasi memiliki sebuah *foreign key* dari tabel preskripsi. Implementasi tabel notifikasi dapat dilihat pada Kode Sumber 4.5.

```

1. create table NOTIFIKASI
2. (
3.     NOTIF_ID          int not null AUTO_INCREMENT,
4.     PRES_ID           int,
5.     TITLE             varchar(100) not null,
6.     MESSAGE            varchar(500) not null,
7.     WAKTU_NOTIFIKASI   timestamp not null,
8.     NOTIF_BACA         bool not null,
9.     ONE SIGNAL_ID      varchar(50) not null,
10.    primary key (NOTIF_ID)
11. );
12. alter table NOTIFIKASI add constraint FK_MENGIRIM foreign key
13.                  (PRES_ID)
14.                  references PRESKRIPSI (PRES_ID) on delete cascade on up
15.                  date cascade;
```

Kode Sumber 4. 5. Implementasi struktur Tabel Notifikasi

4.2.6 Implementasi Tabel Perpesanan

Tabel Perpesanan digunakan untuk mengolah data perpesanan. Tabel perpesanan merupakan tabel transaksi, memiliki dua *foreign key* dari tabel pasien dan tabel dokter. Implementasi tabel perpesanan dapat dilihat pada Kode Sumber 4.6.

```

1. create table PERPESANAN
2. (
3.     PESAN_ID          int not null AUTO_INCREMENT,
4.     PASIEN_ID          int not null,
5.     DOK_ID             int not null,
6.     ISI_PESAN          varchar(200),
7.     WAKTU_PESAN        timestamp,
8.     SENDER              varchar(10),
```

```

9.      PESAN_TERBACA      bool,
10.     primary key (PESAN_ID, PASIEN_ID, DOK_ID)
11.   );
12.   alter table PERPESANAN add constraint FK_PERPESANAN foreign k
    ey (PASIEN_ID)
13.     references PASIEN (PASIEN_ID) on delete cascade on upda
    te cascade;
14.   alter table PERPESANAN add constraint FK_PERPESANAN2 foreign
    key (DOK_ID)
15.     references DOKTER (DOK_ID) on delete cascade on update
    restrict;
```

Kode Sumber 4. 6. Implementasi struktur Tabel Perpesanan

4.2.7 Implementasi Tabel Pemberitahuan

Tabel Pemberitahuan digunakan untuk menyimpan pesan perpesanan. Tabel pemberitahuan merupakan tabel transaksi, memiliki dua *foreign key* dari tabel pasien dan tabel dokter. Implementasi tabel pemberitahuan dapat dilihat pada Kode Sumber 4.7.

```

1. create table PEMBERITAHUAN
2. (
3.     PEM_ID           int not null AUTO_INCREMENT,
4.     PASIEN_ID        int not null,
5.     DOK_ID           int not null,
6.     ISI_PEMBERITAHUAN varchar(200),
7.     WAKTU_PEMBERITAHUAN timestamp,
8.     PEM_TERBACA      bool,
9.     primary key (PEM_ID, PASIEN_ID, DOK_ID)
10.   );
11.   alter table PEMBERITAHUAN add constraint FK_PEMBERITAHUAN for
    eign key (PASIEN_ID)
12.     references PASIEN (PASIEN_ID) on delete cascade on upda
    te cascade;
13.   alter table PEMBERITAHUAN add constraint FK_PEMBERITAHUAN2 fo
    reign key (DOK_ID)
14.     references DOKTER (DOK_ID) on delete cascade on update
    cascade;
```

Kode Sumber 4. 7. Implementasi struktur Tabel Perpesanan

4.2.8 Implementasi Tabel Laporan

Tabel Laporan digunakan untuk menyimpan data konfirmasi minum obat pasien. Tabel laporan merupakan tabel transaksi, memiliki dua *foreign key* dari tabel pasien dan tabel presripsi. Implementasi tabel laporan dapat dilihat pada Kode Sumber 4.8.

```

1.  create table LAPORAN
2.  (
3.    LAP_ID          int not null AUTO_INCREMENT,
4.    PASIEN_ID       int not null,
5.    PRES_ID         int not null,
6.    WAKTU_MINUM     timestamp,
7.    STATUS          bool,
8.    primary key (LAP_ID, PASIEN_ID, PRES_ID)
9.  );
10. alter table LAPORAN add constraint FK_MEMINUM foreign key (PA
11.   SIEN_ID)
12.   references PASIEN (PASIEN_ID) on delete cascade on upda
13.   te cascade;
14. alter table LAPORAN add constraint FK_MEMINUM2 foreign key (P
15.   RES_ID)
16.   references PRESKRIPSI (PRES_ID) on delete cascade on up
17.   date cascade;
```

Kode Sumber 4. 8. Implementasi struktur Tabel Perpesanan

4.3 Implementasi *Third Party*

Pada tahap ini dijelaskan implementasi pengaya atau program *third party* (pihak ketiga) yang digunakan untuk mendukung pembangunan sistem. Penjabaran dijelaskan pada subbab berikut.

4.3.1 OneSignal Web Push Notification

OneSignal menggunakan beberapa fungsi yang dapat digunakan oleh penulis. Pada sistem ini, penulis menggunakan

beberapa fungsi yang digunakan untuk implementasi pengiriman pesan pengingat kepada pasien.

4.3.1.1 *Subscription prompt*

OneSignal menggunakan metode berlangganan untuk menerima notifikasi. Pengguna harus mendaftarkan diri untuk masuk ke dalam daftar berlangganan. Untuk menunjukkan jendela berlangganan, digunakan Kode Sumber 4.8.

```

1. var OneSignal = window.OneSignal || [];
2. OneSignal.push(["init", {
3.   appId: "8b91f202-aa43-4cc6-b2ce-7da23a0bfcf8",
4.   subdomainName: 'monitoring',
5.   autoRegister: true,
6.   promptOptions: {
7.     actionMessage: "Halo, untuk menerima notifikasi minum obat dan pesan lainnya, silahkan klik 'aktifkan'.",
8.     acceptButtonText: "Aktifkan",
9.     cancelButtonText: "Jangan sekarang"
10.   },
11.   welcomeNotification: {
12.     "title": "Notifikasi telah diaktifkan",
13.     "message": "Anda akan menerima rangkaian notifikasi saat minum obat dan menerima pesan"
14.   }
15. }]);

```

Kode Sumber 4. 8. Implementasi *subscription prompt*

Setelah jendela berlangganan ditampilkan dan pengguna memilih “Aktifkan”, maka sistem akan mendaftarkan pengguna dengan *tag*: *user_id*, dimana *user_id* adalah kode unik *user*. Pada sistem menyediakan dua kode unik, yaitu *pasien_id* dan *dokter_id*. Maka *tag* yang dikirimkan ke OneSignal adalah *pasien_id:pasien_id* jika pengguna masuk sebagai pasien atau *dokter_id:dokter_id* jika pengguna masuk sebagai dokter.

```

1. function subscribe(){
2.   OneSignal.push(["registerForPushNotifications"]);
3.   event.preventDefault();
4. }
5.
6. var OneSignal = OneSignal || [];
7. OneSignal.push(function() {
8.   OneSignal.on('subscriptionChange', function (isSubscribed)
9.   {
10.     console.log("The user's subscription state is now:", isSu
bscribed);
11.     OneSignal.sendTag("pasien_id","pasien_id", function(tagsS
ent)
12.     {
13.       console.log("Tags have finished sending!");
14.     });
15. });

```

Kode Sumber 4.9. Fungsi *subscribe* pengguna pasien

Setelah terdaftar, pasien dapat menerima pesan pengingat minum obat berupa notifikasi dari OneSignal.

4.3.1.2 Create Notification

Untuk mengirim pesan pengingat, sistem harus mengirim pesan notifikasi kepada OneSignal yang kemudian dikirimkan kepada pengguna dengan tag tertentu. Tujuan diatur dengan kode unik *tag pasien_id*. Pengiriman menggunakan *REST API* yang mendukung kode program PHP. Setelah notifikasi dibuat dan dijadwalkan pada jam tertentu, API memberikan *return value* (nilai balik) berupa *identifier* atau kode unik notifikasi OneSignal. Berikut adalah potongan kode program membuat notifikasi baru.

```

1. public function sendNotif($pasien_id)
2. {
3.   $headings = array(
4.     "en" => "Codein"
5.   );

```

```

6.
7.     $content = array(
8.         "en" => "Waktunya minum obat Codein sebelum makan"
9.     );
10.    $fields = array(
11.        'app_id' => "8b91f202-aa43-4cc6-b2ce-7da23a0bfcf8",
12.        'url' => "" . base_url('konfirmasi/obat/8'),
13.        'filters' => array(array("field" => "tag", "key" => "pasien_id", "relation" => "=", "value" => "$pasien_id")),
14.        'headings' => $headings,
15.        'contents' => $content
16.    );
17.
18.    $fields = json_encode($fields);
19.
20.    $ch = curl_init();
21.    curl_setopt($ch, CURLOPT_URL, "https://onesignal.com/api/v1
22. /notifications");
22.    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-
23. Type: application/json; charset=utf-8',
23.         'Authorization: Basic YzE4YjcyYWtYWIxNS00Zju1LTlizDgtZ
23. DdjMzJjYzJjOTJj'));
24.    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
25.    curl_setopt($ch, CURLOPT_HEADER, FALSE);
26.    curl_setopt($ch, CURLOPT_POST, TRUE);
27.    curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
28.    curl_setopt($ch, CURLOPT_SSL_VERIFYPeer, FALSE);
29.
30.    $response = curl_exec($ch);
31.    curl_close($ch);
32.
33.    $response = json_decode($response, true);
34.    $OneSignal_id = $response['id'];
35. }

```

Kode Sumber 4. 10. Potongan kode mengirim pesan notifikasi

Beberapa parameter dibutuhkan untuk mengirimkan pesan. Parameter dimasukkan kedalam bentuk data *json* dan dikirimkan menggunakan fungsi *curl* PHP yang membutuhkan OneSignal API *key* dan REST API *key* (pada potongan kode 11 dan 21 pada kode sumber 4.10).

Tabel 4. 2. Parameter utama yang dibutuhkan untuk pengiriman pesan notifikasi

Parameter	Tipe Data	Keterangan
app_id	string	API Key OneSignal
headings	array	Judul pesan notifikasi
content	array	Isi pesan notifikasi
url	string	Alamat tujuan ketika notifikasi dipilih
filters	array	Saring data tags untuk pengiriman pesan kepada pengguna dengan tag yang dituju.

4.3.2 Artyom.js

Wrapper Artyom.js menggunakan metode *speechSynthesis* dan *webkitSpeechRecognition* yang keduanya adalah bagian dari Web Speech API. Web Speech API memungkinkan *browser* melakukan pengucapan dari bentuk tulisan ke dalam bentuk suara dan menerjemahkan pengenalan wicara melalui tangkapan dalam bentuk suara ke dalam bentuk tulisan. Berikut adalah fitur yang digunakan pada *wrapper Artyom.js*.

4.3.2.1 Inisialisasi Artyom.js

Pengembang menyematkan berkas *JavaScript artyom.js* dan *artyom.window.js*. Kemudian melakukan inisialisasi obyek, yang nantinya dapat digunakan dimana saja oleh sistem.

```

1. <script src="=base_url('assets/pages/scripts/artyom.js');?&gt;
   " type="text/javascript"&gt;&lt;/script&gt;
2. &lt;script src="<?=base_url('assets/pages/scripts/artyom.window.
js');?&gt;" type="text/javascript"&gt;&lt;/script&gt;
3. &lt;script&gt;
4.   window.artyom = new Artyom();
5. &lt;/script&gt;</pre

```

Kode Sumber 4. 11. Potongan kode inisialisasi Artyom.js pada kode PHP

4.3.2.2 Inisialisasi Speech Recognition dan Bahasa

Setelah berkas JavaScript `artyom.js` dan object Artyom telah didefinisikan, pengembang dapat memulai pengenalan wicara atau *speech recognition* dan bahasa yang digunakan. Pada spesifikasi Web Speech API oleh W3C, `webkitSpeechRecognition` dan `speechSynthesis` mendukung beberapa bahasa, termasuk Bahasa Indonesia. Sehingga Artyom.js dapat menangkap input suara dan output suara menggunakan Bahasa Indonesia. Kode sumber 4.12 adalah contoh untuk inisialisasi dalam Bahasa Indonesia.

```

1. function startArtyom()
2. {
3.   artyom.initialize({
4.     lang: "id-ID", //Bahasa Indonesia
5.     continuous: true,
6.     debug: true,
7.     listen: true
8.   });
9. }
```

Kode Sumber 4. 12. Inisialisasi *speech recognition* dan bahasa

4.3.2.3 Menambahkan Perintah

Layaknya Google Voice, Artyom.js memiliki fitur perintah dimana perintah dapat disuarakan oleh pengguna dan balasan perintah dengan suara. Berikut adalah potongan kode untuk menambah perintah. Katakan “halo” atau “hai” untuk menyapa!

```

1. const artyom = new Artyom();
2.
3. var perintahSapaan = {
4.   indexes:["halo","hai"], //user berkata halo/hai
5.   action:function(){
6.     artyom.say("Halo, apa kabar?"); //balasan sistem
7.   }
8. };
9. artyom.addCommands(perintahSapaan);
```

Kode Sumber 4. 13. Menambahkan perintah sapaan

4.3.2.4 Menambahkan *Prompt*

Prompt merupakan perintah spesifik beserta dengan pertanyaan (disuarakan oleh sistem) yang membutuhkan jawaban spesifik dari pengguna. Dengan demikian, perintah selain *prompt* yang didefinisikan tidak dapat dieksekusi. Pengguna diharapkan menggunakan pilihan perintah yang ditetapkan pada *prompt*.

```

1. artyom.newPrompt({
2.     question: "",
3.     options:["ya","satu","tidak","dua"],
4.     onMatch: (i) => {
5.         let action;
6.         if(i == 0 || i == 1){
7.             action = () => {
8.                 response = 'Membuka detil obat pada jendela baru.';
9.                 window.open('/monitoring/presripsi/detil/' + pres_id
10.                , '_blank');
11.                aksi = "perintah-after";
12.            }
13.        } else if(i == 2 || i == 3){
14.            action = () => {
15.                response = 'Baiklah, apakah ada yang bisa saya bant
16.                u lagi?';
17.                aksi = "perintah-after";
18.            }
19.        setTimeout(function(){
20.            artyom.shutUp();
21.            artyom.say(response);
22.            textResponse(response, aksi);
23.        }, 1000);
24.        return action;
25.    }
26. });

```

Kode Sumber 4. 14. Contoh potongan kode *prompt*

4.4 Implementasi Antarmuka Aplikasi Web

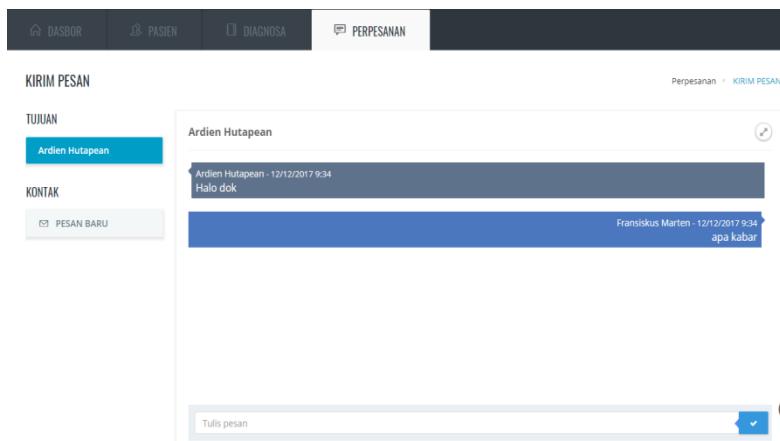
Subbab ini akan membahas implementasi antarmuka aplikasi *web* yang disesuaikan dengan rancangan yang dijelaskan pada subbab 3.2.3 dan 3.2.4.

4.4.1 Implementasi Antarmuka Dokter

Subbab ini akan membahas implementasi dari rancangan antarmuka pengguna dokter yang dijelaskan pada subbab 3.2.3 dan mengacu pada kebutuhan fungsional yang dijabarkan pada subbab 3.1.5.

4.4.1.1 Antarmuka Mengelola Perpesanan

Implementasi antarmuka perpesanan disesuaikan dengan perancangan. Dengan adanya daftar kontak di sebelah kiri dan percakapan di sebelah kanan sebagai komunikasi dokter dengan pasien.



Gambar 4. 1. Implementasi antarmuka mengelola perpesanan (dokter)

4.4.1.2 Antarmuka Membuat Diagnosa Pasien

Dokter memilih pasien terlebih dahulu. Setelah memilih, data pasien akan ditunjukkan dan formulir isian detil diagnosa muncul. Kemudian dokter dapat mengisi detil diagnosa dan menyimpannya.

Gambar 4. 2. Implementasi antarmuka membuat diagnosa pasien

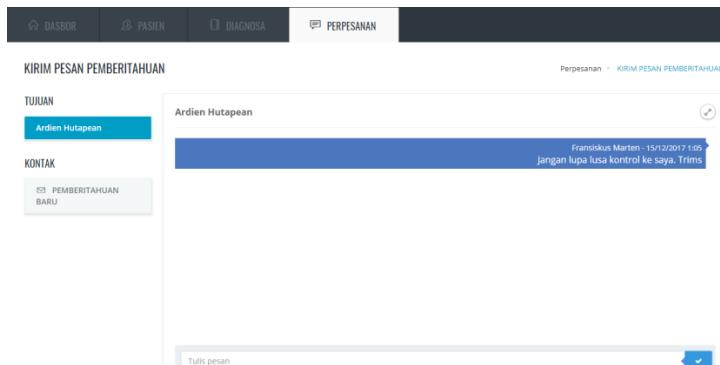
4.4.1.3 Antarmuka Membuat Preskripsi Pasien

Setelah membuat diagnosa pasien, dokter kemudian membuat preskripsi obat yang akan diminum pasien. Sistem menyiakan formulir isian untuk dokter. Dokter kemudian mengisi formulir preskripsi dengan benar yang berisi obat-obatan atau terapi medikasi yang akan dijalani oleh pasien.

Gambar 4. 3. Implementasi antarmuka membuat diagnosa pasien

4.4.1.4 Antarmuka Mengirim Pesan Pemberitahuan

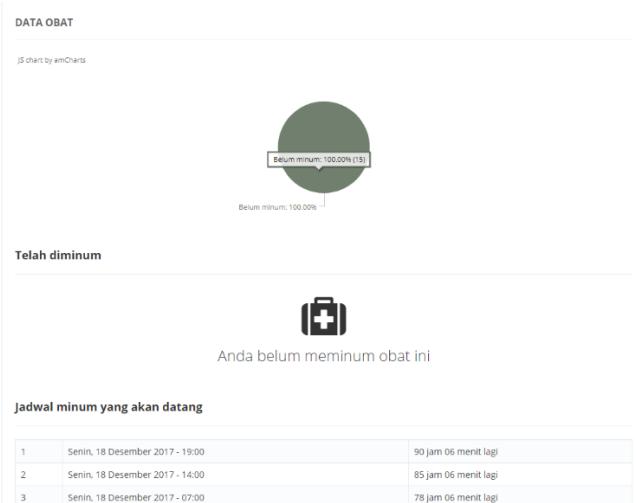
Dokter mengirimkan pesan pemberitahuan kepada pasien. Memiliki antarmuka yang sama dengan mengelola perpesanan dengan tidak adanya balasan dari pasien.



Gambar 4. 4. Implementasi antarmuka mengirim pesan pemberitahuan

4.4.1.5 Antarmuka Melihat Perkembangan Pasien

Dokter dapat memantau jadwal obat yang telah diminum dan jadwal minum yang akan datang. Diagram *pie* juga tersedia sebagai visualisasi data.



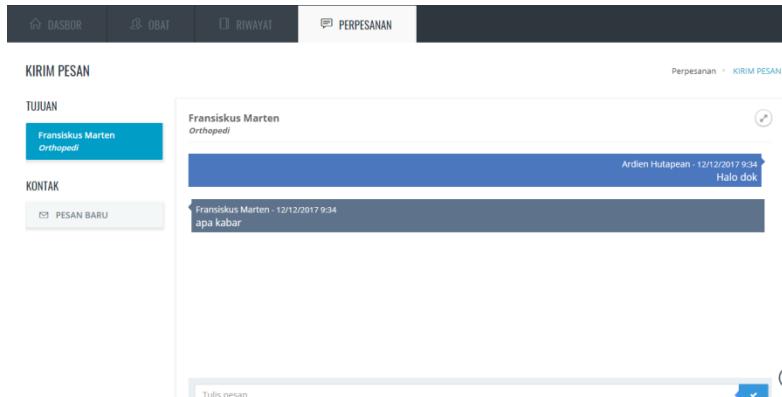
Gambar 4. 5. Implementasi antarmuka melihat perkembangan pasien

4.4.2 Implementasi Antarmuka Pasien

Subbab ini akan membahas implementasi dari rancangan antarmuka pengguna pasien yang dijelaskan pada subbab 3.2.4 dan mengacu pada kebutuhan fungsional yang dijabarkan pada tabel 3.1.

4.4.2.1 Antarmuka Mengelola Perpesanan

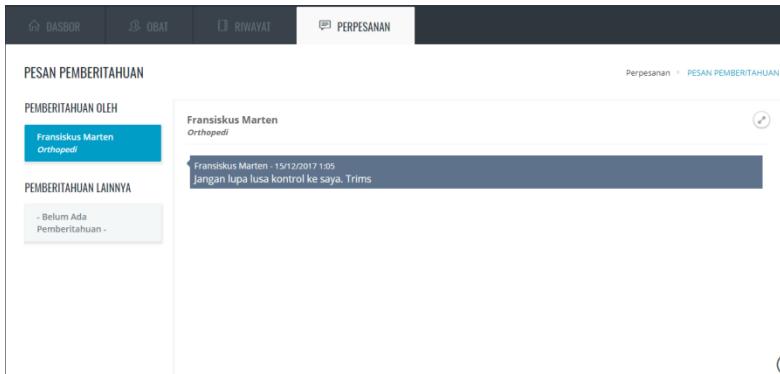
Implementasi antarmuka perpesanan sama dengan dokter. Dengan adanya daftar kontak di sebelah kiri dan percakapan di sebelah kanan sebagai komunikasi pasien dengan dokter.



Gambar 4. 6. Implementasi antarmuka mengelola perpesanan (pasien)

4.4.2.2 Antarmuka Melihat Pesan Pemberitahuan

Antarmuka yang mirip dengan mengelola perpesanan. Hanya saja pasien tidak bisa membalas pesan pemberitahuan sehingga pesan hanyalah sebagai informasi saja.



Gambar 4. 7. Implementasi antarmuka melihat pesan pemberitahuan

4.4.2.3 Antarmuka Mengirim Pesan Konfirmasi

Antarmuka dapat diakses ketika pasien telah menerima pesan pengingat notifikasi. Setelah notifikasi dibuka, antarmuka konfirmasi muncul. Pasien dapat memilih konfirmasi apabila sudah meminum obat atau memilih ‘ingatkan nanti’ untuk menampilkan pesan pengingat 30 menit kedepan. Namun, pasien tetap dapat melakukan konfirmasi sebelum pengingat mengirim pesan pengingat kembali.



Gambar 4. 8. Implementasi antarmuka mengirim pesan konfirmasi

4.4.2.4 Antarmuka Mengubah Waktu Minum

Pasien dapat memilih mengubah waktu minum pada jadwal minum obat atau konfirmasi minum obat yang sudah dilakukan. Tujuannya untuk mengatur waktu minum obat yang sesuai pada kebutuhan pasien atau melakukan kroscek konfirmasi waktu minum.

Gambar 4. 9. Implementasi antarmuka mengubah waktu minum

4.5 Implementasi Kebutuhan Fungsional

Pada bagian ini akan dijelaskan mengenai implementasi pada kebutuhan fungsional yang telah dijabarkan pada subbab 3.1.5. sesuai dengan kerangka kerja CodeIgniter yang menggunakan arsitektur *Model-View-Controller*, potongan kode akan menampilkan kode pada *controller* dan *model*. *Controller* adalah fungsi utama untuk memanggil *model* dan *view*. *Model* adalah fungsi-fungsi yang mengambil atau mengirim data ke *database*. *View* adalah antarmuka yang telah dijelaskan pada subbab 4.4.

4.5.1 Implementasi Mengelola Perpesanan

Pada kebutuhan fungsional mengelola perpesanan, kedua aktor dapat mengakses kebutuhan fungsional. Keduanya memiliki alur normal yang sama, dan cara pengiriman yang sama. Namun, memiliki perbedaan pada implementasi atribut basis data ‘*sender*’ pada tabel perpesanan (tabel basis data dapat dilihat pada tabel 3.14).

```
1. public function percakapan($id_tujuan)
2. {
3.     if($this->session->userdata('akses') == 'Pasien')
4.     {
5.         $pasien_id = $this->session->userdata('id');
6.         $dok_id = $id_tujuan;
7.         $this->load->model('M_pesan');
8.         $this->load->model('M_dokter');
9.
10.        $data['dataDokter'] = $this->M_dokter-
>getDataDokter($dok_id);
11.        $data['listDokter'] = $this->M_dokter-
>getListDokterInConversation($pasien_id);
12.        $data['listConversation'] = $this->M_pesan-
>getConversation($pasien_id, $dok_id);
13.
14.        $this->load->view('header/header_pasien');
15.        $this->load-
>view('perpesanan/percakapan_pasien', $data);
16.        $this->load->view('footer/footer');
17.    }
18.    else if($this->session->userdata('akses') == 'Dokter')
19.    {
20.        $dok_id = $this->session->userdata('id');
21.        $pasien_id = $id_tujuan;
22.        $this->load->model('M_pesan');
23.        $this->load->model('M_pasien');
24.
25.        $data['dataPasien'] = $this->M_pasien-
>getDataPasien($pasien_id);
26.        $data['listPasien'] = $this->M_pasien-
>getListPasienInConversation($dok_id);
```

```

27.      $data['listConversation'] = $this->M_pesanan-
>getConversation($pasien_id, $dok_id);
28.
29.      $this->load->view('header/header_dokter');
30.      $this->load-
>view('perpesanan/percakapan_dokter', $data);
31.      $this->load->view('footer/footer');
32.  }
33. }
```

Kode Sumber 4. 14. Controller Perpesanan

```

1.  public function getDataPasien($pasien_id)
2.  {
3.    $this->db->select("*");
4.    $this->db->from("pasien");
5.    $this->db->where("pasien_id = '$pasien_id'");
6.
7.    $query = $this->db->get();
8.
9.    if ($query->num_rows() > 0)
10.   {
11.     foreach ($query->result() as $row)
12.     {
13.       $data[] = $row;
14.     }
15.     return $data;
16.   }
17. else
18. {
19.   return false;
20. }
21. }
22.
23. public function getListPasienInConversation($dok_id)
24. {
25.   $this->db->distinct();
26.   $this->db->select("pas.PASIEN_ID, pas.NAMA_PASIEN");
27.   $this->db-
>from("diagnosa as diag, pasien as pas, dokter as dok, perpesanan as pes");
28.   $this->db-
>where("diag.DOK_ID = dok.DOK_ID AND diag.PASIEN_ID = pas.PASIEN_ID");
}
```

```
1EN_ID AND pes.PASIEN_ID = pas.PASIEN_ID AND pes.DOK_ID = dok
.DOK_ID AND dok.DOK_ID = '$dok_id'");
29. $this->db->order_by("pas.NAMA_PASIEN asc");
30.
31. $query = $this->db->get();
32. if ($query->num_rows() > 0)
33. {
34.     foreach ($query->result() as $row)
35.     {
36.         $data[] = $row;
37.     }
38.     return $data;
39. }
40. else
41. {
42.     return false;
43. }
44. }
45.
46. public function getDataDokter($dok_id)
47. {
48.     $this->db->select("*");
49.     $this->db->from("dokter");
50.     $this->db->where("dok_id = '$dok_id'");
51.
52.     $query = $this->db->get();
53.
54.     if ($query->num_rows() > 0)
55.     {
56.         foreach ($query->result() as $row)
57.         {
58.             $data[] = $row;
59.         }
60.         return $data;
61.     }
62.     else
63.     {
64.         return false;
65.     }
66. }
67.
68. public function getListDokterInConversation($pasien_id)
69. {
```

```
70.     $this->db->distinct();
71.     $this->db->select("dok.NAMA_DOKTER, dok.BIDANG");
72.     $this->db->from("perpesanan as pes, dokter as dok");
73.     $this->db-
    >where("pes.DOK_ID = dok.DOK_ID AND pes.PASIEN_ID = '$pasien_
    id'");
74.     $this->db->order_by("pes.WAKTU_PESAN DESC");
75.
76.     $query = $this->db->get();
77.
78.     if ($query->num_rows() > 0)
79.     {
80.         foreach ($query->result() as $row)
81.         {
82.             $data[] = $row;
83.         }
84.         return $data;
85.     }
86.     else
87.     {
88.         return false;
89.     }
90. }
91.
92. public function getConversation($pasien_id, $dok_id)
93. {
94.     $this->db-
    >select("pas.PASIEN_ID, pas.NAMA_PASIEN, dok.DOK_ID, dok.NAMA_
    _DOKTER, pes.PESAN_ID, pes.ISI_PESAN, pes.SENDER, DATE_FORMAT(pes.WAKTU_PESAN, '%e/%c/%Y %k:%i') as WAKTU_PESAN");
95.     $this->db-
    >from("perpesanan as pes, dokter as dok, pasien as pas");
96.     $this->db-
    >where("pes.DOK_ID = dok.DOK_ID AND pes.PASIEN_ID = pas.PASIE_
    N_ID AND pes.PASIEN_ID = '$pasien_id' AND pes.DOK_ID = '$dok_
    id'");
97.     $this->db->order_by("pes.WAKTU_PESAN");
98.
99.     $query = $this->db->get();
100.
101.    if ($query->num_rows() > 0)
102.    {
103.        foreach ($query->result() as $row)
```

```

104.    {
105.        $data[] = $row;
106.    }
107.    return $data;
108. }
109. else
110. {
111.    return false;
112. }
113.}

```

Kode Sumber 4. 15. Model Perpesanan

4.5.2 Implementasi Membuat Diagnosa Pasien

Kode program menggunakan *controller* Diagnosa, *model* M_pasien untuk menampilkan daftar pasien dan M_diagnosa untuk menyimpan data diagnosa ke *database*. Fungsi baru() dipanggil sebagai *view* dan fungsi add() dipanggil untuk menambahkan data diagnosa yang telah diisikan.

```

1. public function baru()
2. {
3.     $this->load->model('M_pasien');
4.     $data['listPasien'] = $this->M_pasien-
>getListPasienTerdaftar();
5.
6.     $this->load->view('header/header_dokter');
7.     $this->load->view('diagnosa/diagnosa_tambah', $data);
8.     $this->load->view('footer/footer');
9. }
10.
11. public function add()
12. {
13.     $dok_id = $this->input->post('dok_id');
14.     $pasien_id = $this->input->post('pasien_id');
15.     $komplikasi = $this->input->post('komplikasi');
16.     $keluhan = $this->input->post('keluhan');
17.     $tindakan = $this->input->post('tindakan');
18.     $saran = $this->input->post('saran');
19.     $data = array(

```

```

1.      'dok_id' => $dok_id,
2.      'pasien_id' => $pasien_id,
3.      'komplikasi' => $komplikasi,
4.      'keluhan' => $keluhan,
5.      'tindakan' => $tindakan,
6.      'saran' => $saran
7.  );
8.  $this->load->model('M_diagnosa');
9.  $diag_id = $this->M_diagnosa->addDiagnosa($data);
10. echo '<script language="javascript">';
11. echo 'alert("Data diagnosa baru berhasil dimasukkan");';
12. echo 'window.location.href = "' . site_url('diagnosa/detil/
13. '. $diag_id.'/'.$pasien_id.'') . '";';
14. echo '</script>';
15. }

```

Kode Sumber 4. 16. Controller Diagnosa

```

1.  public function getListPasienTerdaftar()
2.  {
3.      $this->db-
4.          >select("pas.pasien_id, pas.nama_pasien, pas.umur_pasien, pas.
5. alamat, pas.keterangan");
6.      $this->db->from("pasien as pas");
7.      $this->db->order_by("nama_pasien ASC");
8.  }
9.  {
10.     $query = $this->db->get();
11.     if ($query->num_rows() > 0)
12.     {
13.         foreach ($query->result() as $row)
14.         {
15.             $data[] = $row;
16.         }
17.         return $data;
18.     }
19.     else
20.     {
21.         return false;
22.     }
23. }

```

```

24.     $this->db->insert('diagnosa', $data);
25.     return $this->db->insert_id();
26. }
```

Kode Sumber 4.17. Model Diagnosa

4.5.3 Implementasi Membuat Preskripsi Medikasi Pasien

Implementasi ini melibatkan *controller* preskripsi dan *model* preskripsi. Fungsi baru() dipanggil sebagai *view* dan fungsi add() dipanggil untuk menambahkan data preskripsi.

```

1. public function baru($diag_id, $pasien_id)
2. {
3.     $data['diag_id'] = $diag_id;
4.     $data['pasien_id'] = $pasien_id;
5.     $this->load->view('header/header_dokter');
6.     $this->load->view('preskripsi/preskripsi_baru', $data);
7.     $this->load->view('footer/footer');
8. }
9.
10. public function add($diag_id, $pasien_id)
11. {
12.     $jumlah_form = $this->input->post('jumlah_form');
13.     $this->load->model('M_preskripsi');
14.     for ($i=0; $i < $jumlah_form; $i++)
15.     {
16.         $data = array(
17.             'diag_id' => $diag_id,
18.             'nama_obat' => $this->input-
>post('preskripsi['.$i.'][nama_obat']),
19.             'dosis' => $this->input-
>post('preskripsi['.$i.'][dosis']),
20.             'jumlah' => $this->input-
>post('preskripsi['.$i.'][jumlah']),
21.             'jenis_obat' => $this->input-
>post('preskripsi['.$i.'][jenis_obat']),
22.             'makan' => $this->input-
>post('preskripsi['.$i.'][makan']),
23.             'penggunaan' => $this->input-
>post('preskripsi['.$i.'][penggunaan]),
```

```

24.      'keterangan_obat' => $this->input-
    >post('presripsi['.$i.'][keterangan])',
25.      'reminder_hari' => $this->input-
    >post('presripsi['.$i.'][jangka])',
26.      'reminder_sisa' => $this->input-
    >post('presripsi['.$i.'][jangka])'
27.      );
28.      $pres_id = $this->M_presripsi->addPresripsi($data);
29.  }
30.
31. echo '<script language="javascript">';
32. echo 'alert("Data presripsi baru berhasil dimasukkan");'

33. echo 'window.location.href = "' . site_url('diagnosa/detil/
    '. $diag_id.'/'.$pasien_id.'') . '"';
34. echo '</script>';
35. }

```

Kode Sumber 4. 18. Controller Preskripsi

```

1. public function addPresripsi($data)
2. {
3.     $this->db->insert('presripsi', $data);
4.     return $this->db->insert_id();
5. }

```

Kode Sumber 4. 19. Model Preskripsi

4.5.4 Implementasi Mengirim Pesan Pemberitahuan

Implementasi kebutuhan Mengirim Pesan Pemberitahuan mengambil beberapa fungsi dari *controller* perpesanan. Fungsi *pemberitahuan()* untuk menampilkan *view* pemberitahuan dan *model* pemberitahuan.

```

1. public function pemberitahuan($id_tujuan)
2. {
3.     $this->load->model('M_pesan');
4.     $this->load->model('M_pasien');
5.
6.     $data['dataPasien'] = $this->M_pasien-
    >getDataPasien($pasien_id);

```

```

7.      $data['listPasien'] = $this->M_pasien-
8.          >getListPasienInConversation($dok_id);
9.      $data['listPemberitahuan'] = $this->M_pesan-
10.         >getPemberitahuan($pasien_id, $dok_id);
11.     $this->load->view('header/header_dokter');
12.     $this->load-
13.         >view('perpesanan/pemberitahuan_dokter', $data);
14.     $this->load->view('footer/footer');
15. }

```

Kode Sumber 4. 20. Controller Pemberitahuan

```

1. public function getDataPasien($pasien_id)
2. {
3.     $this->db->select("*");
4.     $this->db->from("pasien");
5.     $this->db->where("pasien_id = '$pasien_id'");
6.
7.     $query = $this->db->get();
8.
9.     if ($query->num_rows() > 0)
10.    {
11.        foreach ($query->result() as $row)
12.        {
13.            $data[] = $row;
14.        }
15.        return $data;
16.    }
17.    else
18.    {
19.        return false;
20.    }
21. }
22.
23. public function getListPasienInConversation($dok_id)
24. {
25.     $this->db->distinct();
26.     $this->db->select("pas.PASIEN_ID, pas.NAMA_PASIEN");
27.     $this->db-
         >from("diagnosa as diag, pasien as pas, dokter as dok, perpes-
             anan as pes");

```

```
28.    $this->db-
>where("diag.DOK_ID = dok.DOK_ID AND diag.PASIEN_ID = pas.PASIE-
N_ID AND pes.PASIEN_ID = pas.PASIEN_ID AND pes.DOK_ID = dok-
.DOK_ID AND dok.DOK_ID = '$dok_id'");
29.    $this->db->order_by("pas.NAMA_PASIEN asc");
30.
31.    $query = $this->db->get();
32.    if ($query->num_rows() > 0)
33.    {
34.        foreach ($query->result() as $row)
35.        {
36.            $data[] = $row;
37.        }
38.        return $data;
39.    }
40.    else
41.    {
42.        return false;
43.    }
44. }
45.
46. public function getPemberitahuan($pasien_id, $dok_id)
47. {
48.    $this->db-
>select("pas.PASIEN_ID, pas.NAMA_PASIEN, dok.DOK_ID, dok.NAMA-
_DOKTER, pem.PEM_ID, pem.ISI_PEMBERITAHUAN, DATE_FORMAT(pes-
.WAKTU_PESAN, '%e/%c/%Y %k:%i') as WAKTU_PEMBERITAHUAN");
49.    $this->db-
>from("pemberitahuan as pem, dokter as dok, pasien as pas");

50.    $this->db-
>where("pem.DOK_ID = dok.DOK_ID AND pem.PASIEN_ID = pas.PASIE-
N_ID AND pem.PASIEN_ID = '$pasien_id' AND pem.DOK_ID = '$dok-
_id'");
51.    $this->db->order_by("pem.WAKTU_PEMBERITAHUAN");
52.
53.    $query = $this->db->get();
54.
55.    if ($query->num_rows() > 0)
56.    {
57.        foreach ($query->result() as $row)
58.        {
59.            $data[] = $row;
```

```

60.      }
61.      return $data;
62.    }
63.    else
64.    {
65.      return false;
66.    }
67.  }

```

Kode Sumber 4. 21. Model Pemberitahuan

4.5.5 Implementasi Melihat Pesan Pemberitahuan

Controller pemberitahuan memanggil fungsi yang berbeda, yaitu fungsi `pemberitahuan_pasien()` untuk menampilkan kontak dokter yang pernah mengirimkan pesan pemberitahuan dan isi pesan pemberitahuan yang dipilih oleh pasien.

```

1.  public function pemberitahuan_pasien($id_tujuan)
2.  {
3.    $pasien_id = $this->session->userdata('id');
4.    $dok_id = $id_tujuan;
5.    $this->load->model('M_pesanan');
6.    $this->load->model('M_dokter');
7.
8.    $data['dataDokter'] = $this->M_dokter-
>getDataDokter($dok_id);
9.    $data['listDokter'] = $this->M_dokter-
>getListDokterInConversation($pasien_id);
10.   $data['listPemberitahuan'] = $this->M_pesanan-
>getPemberitahuan($pasien_id, $dok_id);
11.
12.   $this->load->view('header/header_pasien');
13.   $this->load-
>view('perpesanan/pemberitahuan_pasien', $data);
14.   $this->load->view('footer/footer');
15. }

```

Kode Sumber 4. 22. Controller Pemberitahuan

```

1.  public function getDataDokter($dok_id)
2.  {

```

```
3.     $this->db->select("*");
4.     $this->db->from("dokter");
5.     $this->db->where("dok_id = '$dok_id'");
6.
7.     $query = $this->db->get();
8.
9.     if ($query->num_rows() > 0)
10.    {
11.        foreach ($query->result() as $row)
12.        {
13.            $data[] = $row;
14.        }
15.        return $data;
16.    }
17. else
18. {
19.     return false;
20. }
21. }
22.
23. public function getListDokterInConversation($pasien_id)
24. {
25.     $this->db->distinct();
26.     $this->db->select("dok.NAMA_DOKTER, dok.BIDANG");
27.     $this->db->from("perpesanan as pes, dokter as dok");
28.     $this->db-
>where("pes.DOK_ID = dok.DOK_ID AND pes.PASIEN_ID = '$pasien_
id'");
29.     $this->db->order_by("pes.WAKTU_PESAN DESC");
30.
31.     $query = $this->db->get();
32.
33.     if ($query->num_rows() > 0)
34.    {
35.        foreach ($query->result() as $row)
36.        {
37.            $data[] = $row;
38.        }
39.        return $data;
40.    }
41. else
42. {
43.     return false;
```

```

44.    }
45. }
46.
47. public function getPemberitahuan($pasien_id, $dok_id)
48. {
49.     $this->db-
50.         >select("pas.PASIEN_ID, pas.NAMA_PASIEN, dok.DOK_ID, dok.NAMA_
51.             _DOKTER, pem.PEM_ID, pem.ISI_PEMBERITAHUAN,  DATE_FORMAT(pes.
52.                 WAKTU_PESAN, '%e/%c/%Y %k:%i') as WAKTU_PEMBERITAHUAN");
53.     $this->db-
54.         >from("pemberitahuan as pem, dokter as dok, pasien as pas");
55.
56.     $this->db-
57.         >where("pem.DOK_ID = dok.DOK_ID AND pem.PASIEN_ID = pas.PASIE_
58.             N_ID AND pem.PASIEN_ID = '$pasien_id' AND pem.DOK_ID = '$dok_
59.                 id'");
60.     $this->db->order_by("pem.WAKTU_PEMBERITAHUAN");
61.
62.     $query = $this->db->get();
63.
64.     if ($query->num_rows() > 0)
65.     {
66.         foreach ($query->result() as $row)
67.         {
68.             $data[] = $row;
69.         }
70.         return $data;
71.     }
72.     else
73.     {
74.         return false;
75.     }
76. }

```

Kode Sumber 4. 23. Model Pemberitahuan

4.5.6 Implementasi Melihat Perkembangan Pasien

Kebutuhan fungsional ini dapat diakses oleh pasien maupun dokter. Dokter dapat melihat jadwal serta waktu minum obat yang telah diminum oleh pasien.

```

1.  public function detil($pres_id)
2.  {
3.      $this->load->model('M_presripsi');
4.      $this->load->model('M_laporan');
5.      $this->load->model('M_notifikasi');
6.      $data['dataObat'] = $this->M_presripsi-
>getDataObat($pres_id);
7.      $data['dataLaporan'] = $this->M_laporan-
>getLaporanObat($pres_id);
8.      $data['dataJadwal'] = $this->M_notifikasi-
>getJadwalObat($pres_id);
9.      $data['countJadwal'] = $this->M_notifikasi-
>getJumlahJadwalObat($pres_id);
10.
11.     if($this->session->userdata('akses') == 'Pasien')
12.     {
13.         $this->load->view('header/header_pasien');
14.         $this->load-
>view('presripsi/presripsi_detil_pasien', $data);
15.     }
16.     else
17.     {
18.         $this->load->view('header/header_dokter');
19.         $this->load-
>view('presripsi/presripsi_detil_dokter', $data);
20.     }
21.     $this->load->view('footer/footer');
22. }
```

Kode Sumber 4. 24. Controller Presripsi

```

1.  public function getDataObat($pres_id)
2.  {
3.      $this->db->select("*");
4.      $this->db->from("presripsi");
5.      $this->db->where("pres_id = '$pres_id'");
6.
7.      $query = $this->db->get();
8.
9.      if ($query->num_rows() > 0)
10.     {
11.         foreach ($query->result() as $row)
12.         {
```

```
13.         $data[] = $row;
14.     }
15.     return $data;
16. }
17. else
18. {
19.     return false;
20. }
21. }
22.
23. public function getLaporanObat($pres_id)
24. {
25.     $query = $this->db->query("SET lc_time_names = 'id_ID'");
26.
27.     $this->db->select("DATE_FORMAT(WAKTU_MINUM, '%W, %e %M %Y -
28.                         %H:%i') AS WAKTU_MINUM, TIME_FORMAT(SEC_TO_TIME(TIMESTAMPDI-
29.                         FF(SECOND,WAKTU_MINUM,CURRENT_TIMESTAMP())),'%H jam %i menit
30.                         yang lalu') as WAKTU_BERIKUTNYA");
31.     $this->db->from("laporan");
32.     $this->db->where("PRES_ID = $pres_id AND STATUS = 1");
33.     $this->db->order_by("WAKTU_MINUM DESC");
34.
35.     $query = $this->db->get();
36.
37.     if ($query->num_rows() > 0)
38.     {
39.         foreach ($query->result() as $row)
40.         {
41.             $data[] = $row;
42.         }
43.     }
44.     return $data;
45. }
46. }
47.
48. public function getJadwalObat($pres_id)
49. {
50.     $query = $this->db->query("SET lc_time_names = 'id_ID'");
51.
```

```
52.     $this->db-
53.         >select("DATE_FORMAT(WAKTU_NOTIFIKASI, '%W, %e %M %Y -
54.             %H:%i') AS WAKTU_NOTIFIKASI, TIME_FORMAT(SEC_TO_TIME(TIMESTAMPDIFF(SECOND,CURRENT_TIMESTAMP(),WAKTU_NOTIFIKASI)),'%H jam
55.             %i menit lagi') as WAKTU_BERIKUTNYA");
56.
57.     $this->db->from("notifikasi");
58.     $this->db-
59.         >where("PRES_ID = '$pres_id' AND NOTIF_BACA = 0");
60.     $this->db->order_by("WAKTU_NOTIFIKASI DESC");
61.
62.     $query = $this->db->get();
63.
64.     if ($query->num_rows() > 0)
65.     {
66.         foreach ($query->result() as $row)
67.         {
68.             $data[] = $row;
69.         }
70.         return $data;
71.     }
72.
73.     public function getJumlahJadwalObat($pres_id)
74.     {
75.         $this->db->select("COUNT(*) as count");
76.         $this->db->from("notifikasi");
77.         $this->db-
78.             >where("PRES_ID = '$pres_id' AND NOTIF_BACA = 0");
79.         $query = $this->db->get();
80.
81.         if ($query->num_rows() > 0)
82.         {
83.             foreach ($query->result() as $row)
84.             {
85.                 $data[] = $row;
86.             }
87.             return $data;
88.         }
```

```

89.     else
90.     {
91.         return false;
92.     }
93. }
```

Kode Sumber 4. 25. Model Preskripsi

4.5.7 Implementasi Mengirim Pesan Konfirmasi

Controller konfirmasi memanggil fungsi `obat()` sebagai *view* dan fungsi `konfirmMinum()` sebagai pengirim pesan konfirmasi. Fungsi juga memanggil `M_preskripsi` untuk mengambil data obat dari *database*.

```

1. public function obat($pres_id)
2. {
3.     $this->load->model('M_preskripsi');
4.     $data['dataObat'] = $this->M_preskripsi-
>getDataObat($pres_id);
5.
6.     $data['pres_id'] = $pres_id;
7.     $this->load->view('header/header_pasien');
8.     $this->load->view('konfirmasi/konfirmasi_obat', $data);
9.     $this->load->view('footer/footer');
10. }
11.
12. public function konfirmMinum($pasien_id, $pres_id)
13. {
14.     $data = array(
15.         'pasien_id' => $pasien_id,
16.         'pres_id' => $pres_id,
17.         'status' => 1
18.     );
19.
20.     $this->load->model('M_laporan');
21.     $this->load->model('M_notifikasi');
22.     $diag_id = $this->M_laporan->addLaporan($data);
23.     $waktu = $this->M_notifikasi->getNextWaktu($pres_id);
24.     $this->M_notifikasi->updateNotif($pres_id, $waktu);
25. }
```

```

26. echo '<script language="javascript">';
27. echo 'window.location.href = "' . site_url('dasbor') . '";'
28. echo '</script>';
29. }

```

Kode Sumber 4. 26. Controller Konfirmasi

```

1. public function getDataObat($pres_id)
2. {
3.     $this->db->select("*");
4.     $this->db->from("presripsi");
5.     $this->db->where("pres_id = '$pres_id'");
6.
7.     $query = $this->db->get();
8.
9.     if ($query->num_rows() > 0)
10.    {
11.        foreach ($query->result() as $row)
12.        {
13.            $data[] = $row;
14.        }
15.        return $data;
16.    }
17. else
18. {
19.    return false;
20. }
21. }

```

Kode Sumber 4. 27. Model Konfirmasi

4.5.8 Implementasi Mengubah Waktu Minum

Pengubahan waktu minum melibatkan dua fungsi yang memiliki tujuan yang berbeda, yaitu mengubah jadwal waktu minum dan mengubah konfirmasi waktu minum.

Untuk pengubahan jadwal waktu minum, *Controller* yang digunakan adalah *controller* presripsi dengan memanggil fungsi *ubahJadwal()* sebagai *view* dan fungsi *updateJadwal()* untuk mengatur proses pengambilan data melalui *model*.

```
1. public function ubahJadwal($pres_id)
2. {
3.     $this->load->model('M_presripsi');
4.     $this->load->model('M_notifikasi');
5.     $data['dataObat'] = $this->M_presripsi-
>getDataObat($pres_id);
6.     $data['dataTanggal'] = $this->M_notifikasi-
>getDataTanggalPertamaObatbyID($pres_id);
7.     $data['dataJadwal'] = $this->M_notifikasi-
>getDataJadwalObatbyID($pres_id);
8.
9.     if($this->session->userdata('akses') == 'Pasien')
10.    {
11.        $this->load->view('header/header_pasien');
12.        $this->load-
>view('presripsi/presripsi_ubah_jadwal_pasien', $data);
13.    }
14.    else
15.    {
16.        $this->load->view('header/header_dokter');
17.        $this->load-
>view('presripsi/presripsi_ubah_jadwal_dokter', $data);
18.    }
19.    $this->load->view('footer/footer');
20.}
21.
22. public function updateJadwal($pres_id)
23.{
24.    $tanggal = $this->input->post('tanggalPertama');
25.    $waktu = $this->input->post('waktu');
26.    $penggunaan = $this->input->post('penggunaan');
27.    $pasien_id = $this->input->post('pasien_id');
28.    $nama_obat = $this->input->post('nama_obat');
29.    $makan = $this->input->post('makan');
30.    $tanggalobj = new DateTime($tanggal);
31.    $tanggal = $tanggalobj->format('Y-m-d');
32.    $this->load->model('M_presripsi');
33.    $this->load->model('M_notifikasi');
34.    $allNotifID = $this->M_notifikasi-
>getAllNotifID($pres_id);
35.    $banyakTanggal = $this->M_notifikasi-
>getBanyakTanggal($pres_id, $tanggal);
36.
```

```
37. //range Jadwal
38. $jadwal = new DateTime($waktu);
39. if($penggunaan == 1)
40. {
41.     $range = '+24 hours';
42. }
43. else if($penggunaan == 2)
44. {
45.     $range = '+12 hours';
46.
47.     $jadwal1 = $jadwal->format('H:i');
48.     $jadwal->modify($range);
49.     $jadwal2 = $jadwal->format('H:i');
50.
51.     if($banyakTanggal == 2)
52.     {
53.         $waktu = $jadwal1;
54.     }
55.     else if($banyakTanggal == 1)
56.     {
57.         $waktu = $jadwal2;
58.     }
59. }
60. else if($penggunaan == 3)
61. {
62.     $range = '+8 hours';
63.
64.     $jadwal1 = $jadwal->format('H:i');
65.     $jadwal->modify($range);
66.     $jadwal2 = $jadwal->format('H:i');
67.     $jadwal->modify($range);
68.     $jadwal3 = $jadwal->format('H:i');
69.
70.     if($banyakTanggal == 3)
71.     {
72.         $waktu = $jadwal1;
73.     }
74.     else if($banyakTanggal == 2)
75.     {
76.         $waktu = $jadwal2;
77.     }
78.     else if($banyakTanggal == 1)
79.     {
```

```
80.      $waktu = $jadwal3;
81.    }
82.  }
83. //utk database
84. $waktu = $tanggal.' '.$waktu.':00';
85. $waktu = new DateTime($waktu);
86.
87. if($allNotifID != NULL)
88. {
89.   foreach($allNotifID as $row)
90.   {
91.     if($row->NOTIF_BACA == 0)
92.     {
93.       //utk onesignal
94.       $waktu_notifikasi_onesignal = $waktu->format('Y-m-
d H:i:s').' GMT+0700';
95.
96.       //message
97.       if($makan == 0)
98.       {
99.         $message = "Waktunya minum obat ".$nama_obat." sebelum
makan";
100.        }
101.      else
102.      {
103.        $message = "Waktunya minum obat ".$nama_obat." sesu-
dah makan";
104.      }
105.
106.      // cancel notification onesignal //
107.      $ch = curl_init();
108.      $httpHeader = array(
109.        'Authorization: Basic ZDIyNzI2YTEtMGYzM00Nz
U5LTImNDYtOTdjODIyM2VmZjU2'
110.      );
111.      $url = "https://onesignal.com/api/v1/notifications
/".$row->ONESIGNAL_ID."?app_id=34e90451-81b0-4aee-b841-
95dc798ae620";
112.
113.      $options = array (
114.        CURLOPT_URL => $url,
115.        CURLOPT_HTTPHEADER => $httpHeader,
116.        CURLOPT_RETURNTRANSFER => TRUE,
```

```
117.          CURLOPT_CUSTOMREQUEST => "DELETE",
118.          CURLOPT_SSL_VERIFYPEER => FALSE
119.      );
120.      curl_setopt_array($ch, $options);
121.      $response = curl_exec($ch);
122.      curl_close($ch);
123.      // end cancel notification onesignal //
124.
125.      // create notif onesignal //
126.      $headings = array(
127.          "en" => $nama_obat //Title: Nama obat
128.      );
129.
130.      $content = array(
131.          "en" => $message //Pesanan utama
132.      );
133.
134.      $fields = array(
135.          'app_id' => "34e90451-81b0-4aee-b841-
95dc798ae620",
136.          'url' => ".$base_url('konfirmasi/obat/'.$pres_
id.'').",
137.          'filters' => array(array("field" => "tag", "ke
y" => "pasien_id", "relation" => "=", "value" => "$pasien_id"
)),
138.          'headings' => $headings,
139.          'contents' => $content,
140.          'send_after' => $waktu_notifikasi_onesignal
141.      );
142.
143.      $fields = json_encode($fields);
144.
145.      $ch = curl_init();
146.      curl_setopt($ch, CURLOPT_URL, "https://onesignal.c
om/api/v1/notifications");
147.      curl_setopt($ch, CURLOPT_HTTPHEADER, array('Conten
t-Type: application/json; charset=utf-8',
148.          'Authorization: Basic ZDIyNzI2YTEtMGYzM00NzU5
LTlmNDYtOTdjODIyM2VmZjU2'));
149.      curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
150.      curl_setopt($ch, CURLOPT_HEADER, FALSE);
151.      curl_setopt($ch, CURLOPT_POST, TRUE);
152.      curl_setopt($ch, CURLOPT_POSTFIELDS, $fields);
```

```

153.         curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
154.
155.         $response = curl_exec($ch);
156.         curl_close($ch);
157.         //echo $response;
158.
159.         $json_response = json_decode($response, true);
160.         $onesignal_id = $json_response['id'];
161.         // end create notif onesignal //
162.
163.         //utk database
164.         $waktu_notifikasi = $waktu->format('Y-m-
d H:i:s');
165.         $data = array(
166.             'WAKTU_NOTIFIKASI' => $waktu_notifikasi,
167.             'ONESIGNAL_ID' => $onesignal_id
168.         );
169.     }
170.     else
171.     {
172.         //utk database
173.         $waktu_notifikasi = $waktu->format('Y-m-
d H:i:s');
174.         $data = array(
175.             'WAKTU_NOTIFIKASI' => $waktu_notifikasi
176.         );
177.     }
178.
179.         $this->M_preskripsi->updateJadwal($data, $row-
>NOTIF_ID);
180.
181.         $waktu->modify($range);
182.     }
183. }
184. echo '<script language="javascript">';
185. echo 'alert("Jadwal minum obat berhasil diubah");';
186. echo 'window.location.href = "' . site_url('preskripsi/d
etil/'.$pres_id.'')) . '"';
187. echo '</script>';
188. }

```

Kode Sumber 4. 28. Fungsi mengubah jadwal minum pada controller preskripsi

```

1. public function updateJadwal($data, $notif_id)
2. {
3.     $this->db->where('NOTIF_ID', $notif_id);
4.     $this->db->update('notifikasi', $data);
5.     return ($this->db->affected_rows() > 0);
6. }
```

Kode Sumber 4. 29. Fungsi model mengubah jadwal minum

Untuk pengubahan konfirmasi waktu minum, *Controller* yang digunakan adalah *controller* preskripsi dengan pemanggilan fungsi `ubahJadwal()` sebagai *view* dan fungsi `updateJadwal()` untuk mengatur proses pengambilan data melalui *model*.

```

1. public function ubahKonfirmasi($pres_id, $lap_id)
2. {
3.     $this->load->model('M_preskripsi');
4.     $this->load->model('M_laporan');
5.     $data['dataObat'] = $this->M_preskripsi-
>getDataObat($pres_id);
6.     $data['dataJadwal'] = $this->M_laporan-
>getDataLaporan($lap_id);
7.
8.     if($this->session->userdata('akses') == 'Pasien')
9.     {
10.        $this->load->view('header/header_pasien');
11.        $this->load-
>view('preskripsi/preskripsi_ubah_konfirmasi_pasien', $data);
12.    }
13.    else
14.    {
15.        $this->load->view('header/header_dokter');
16.        $this->load-
>view('preskripsi/preskripsi_ubah_konfirmasi_dokter', $data);
17.    }
18.    $this->load->view('footer/footer');
19. }
20.
21. public function updateKonfirmasi($pres_id, $lap_id)
22. {
23.     $tanggal = $this->input->post('tanggal');
24.     $waktu = $this->input->post('waktu');
```

```

25.
26.     $tanggal_notifikasi = DateTime::createFromFormat('d-m-
   Y', $tanggal);
27.     $tanggal_notifikasi = $tanggal_notifikasi->format('Y-m-
   d');
28.
29.     $waktu_notifikasi = $tanggal_notifikasi.' '.$waktu.':00';
30.
31.     $data = array(
32.         'WAKTU_MINUM' => $waktu_notifikasi
33.     );
34.
35.     $this->load->model('M_laporan');
36.
37.     if($this->M_laporan->updateKonfirmasi($data, $lap_id))
38.     {
39.         echo '<script language="javascript">';
40.         echo 'alert("Konfirmasi minum obat berhasil diubah");';
41.         echo 'window.location.href = "' . site_url('presripsi/de
   til/'.$pres_id.')' . "';";
42.         echo '</script>';
43.     }
44.     else
45.     {
46.         echo '<script language="javascript">';
47.         echo 'alert("Gagal mengubah konfirmasi. Konfirmasi yang d
   imasukkan sama dengan konfirmasi sebelumnya.");';
48.         echo 'window.history.back();';
49.         echo '</script>';
50.     }
51. }
```

Kode Sumber 4. 30. Fungsi mengubah konfirmasi waktu minum pada controller presripsi

```

1. public function updateKonfirmasi($data, $lap_id)
2. {
3.     $this->db->where('LAP_ID', $lap_id);
4.     $this->db->update('laporan', $data);
5.     return ($this->db->affected_rows() > 0);
6. }
```

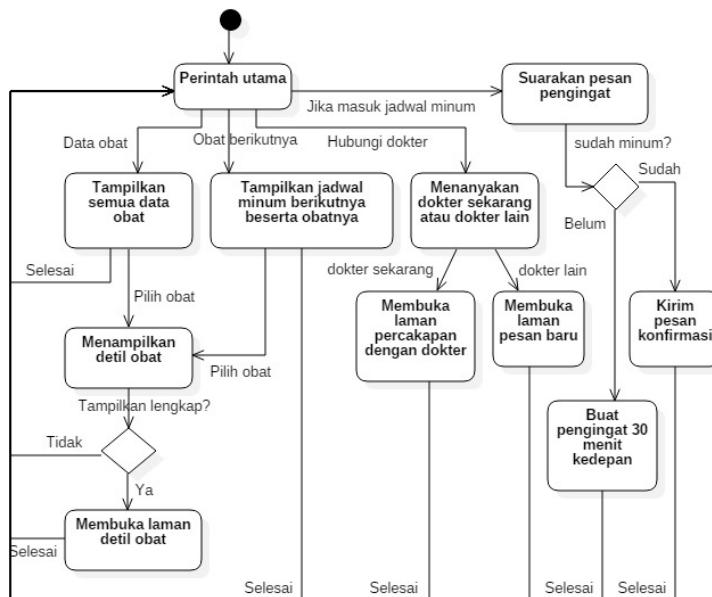
Kode Sumber 4. 31. Fungsi model mengubah konfirmasi waktu minum

4.6 Implementasi Sistem IVR

Sistem IVR yang dibangun tidak menggunakan teknologi telepon, namun menggunakan suara, teks (tertulis), dan tombol aksi. Tombol aksi mengganti tombol nada tekan telefon dan sebagai alternatif untuk menjalankan perintah. Isian teks sebagai alternatif lain ketika pengguna tidak memiliki mikrofon sebagai perekam suara.

Sistem IVR memiliki perintah yang dapat dijalankan oleh pengguna. Perintah didefinisikan saat implementasi perangkat lunak. Ketika pengguna menggunakan perintah yang tidak sesuai maka sistem akan menjawab dengan perintah anda tidak sesuai.

Sistem IVR menggunakan *third party* Artyom.js yang telah dijelaskan pada subbab 4.3.2. Menggunakan perintah dan *prompt*, perintah-perintah dapat didefinisikan dengan mudah. Berikut adalah struktur perintah sistem IVR yang diimplementasikan.



Gambar 4. 10. Skema perintah IVR dan balasan sistem

Pada awal pengguna menggunakan sistem IVR, terdapat perintah utama/awalan yaitu, “data obat”, “obat berikutnya”, dan “hubungi dokter”. Setelah perintah dipanggil, sistem akan membalas dengan aksi yang ditampilkan pada kotak (Gambar 4.10). Ketika perintah selesai dijalankan dan tidak memiliki perintah berikutnya, maka sistem IVR akan kembali menangkap perintah utama.

Kasus khusus pada saat jadwal minum telah tiba. Sistem IVR akan langsung memberikan suara pesan pengingat dengan pertanyaan kepada pengguna. Jika pengguna telah minum obat, sistem akan mengirimkan pesan konfirmasi, sebaliknya sistem akan membuat pengingat yang akan mengingatkan pasien 30 menit kedepan. Setelah salah satu perintah dijalankan, maka sistem IVR kembali mendengarkan pertntah utama.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Pada bab kelima ini akan dijelaskan mengenai skenario dan pengujian perangkat lunak yang telah dibangun. Pengujian menjelaskan tentang data yang digunakan, hasil yang didapatkan dari penggunaan perangkat lunak dan uji coba yang dilakukan pada perangkat lunak yang telah dikerjakan untuk menguji apa-kah kebutuhan fungsional perangkat lunak telah diimplementasi-sikan dengan benar dan berjalan sebagaimana mestinya.

5.1 Lingkungan Pengujian

Lingkungan pengujian menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

Tabel 5. 2. Spesifikasi lingkungan uji coba

Perangkat	Spesifikasi
Perangkat 1	Perangkat : Laptop Toshiba Satellite L-50 Prosesor : Intel Core i5-4200U CPU @ 1.60 GHz Memori : 4.00 GB Mikrofon : Conexant SmartAudio HD OS : Microsoft Windows 10 64-bit Browser : Google Chrome 63
Perangkat 2	Perangkat : Mobile OPPO R7 Lite (R7kf) Prosesor : Qualcomm® Snapdragon™ 625 Octa-core (ARM 64-bit) CPU @ 2.0 GHz Memori : 2.00 GB Mikrofon : Built-in microphone OS : Android Lollipop 5.1 Browser : Google Chrome 63

Pengujian dilakukan pada perangkat *desktop* dan perangkat bergerak dengan *browser* yang sama. Hal demikian dilakukan

untuk menguji fungsionalitas perintah suara. Perangkat bergerak untuk menguji berbagai fungsionalitas *website* ketika pengguna menggunakan perangkat bergerak.

5.2 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan untuk menguji semua kebutuhan fungsional dan sistem IVR. Pengujian menggunakan metode *blackbox*, yakni menguji dengan input serta *output* yang dihasilkan.

5.3 Skenario dan Hasil Uji Coba

Subbab ini akan dijelaskan skenario uji coba yang dilakukan dan hasil yang didapatkan.

5.3.1 Pengujian Fungsionalitas Mengelola Perpesanan

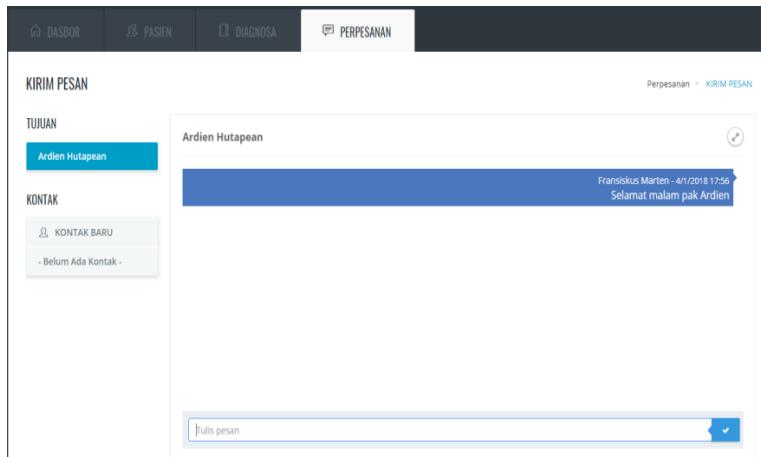
Kebutuhan fungsionalitas mengelola perpesanan dapat diakses oleh pengguna pasien dan dokter. Pengujian fungsionalitas dilakukan pada setiap pengguna dalam pengiriman dan penerimaan pesan. Tidak hanya pengiriman dan penerimaan. Kebutuhan juga mengakomodasi fitur pemilihan kontak tujuan.

Tabel 5. 2. Pengujian fungsionalitas mengelola perpesanan

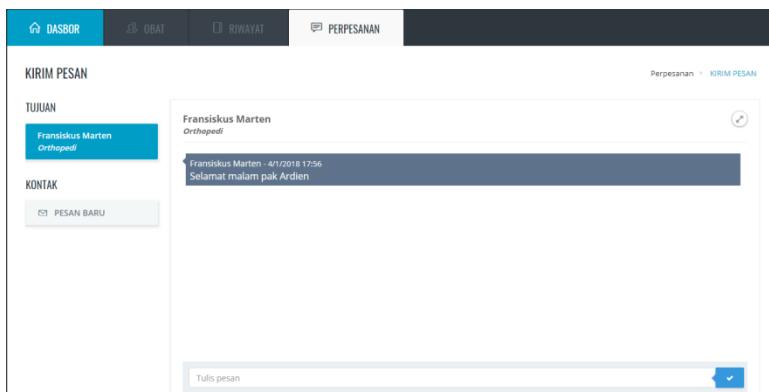
Kode	P-001
Referensi kasus penggunaan	UC-001
Nama	Mengelola Perpesanan
Tujuan	Menguji fungsi kontak tujuan dan pengiriman serta penerimaan pesan.
<i>Skenario 1 - Memilih kontak pasien</i>	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman perpesanan
Data Uji	Kontak pasien yang pernah didiagnosa oleh Dokter

Langkah pengujian	1. Pengguna memilih kontak pasien. 2. Sistem menampilkan hanya kontak pasien yang pernah didiagnosa oleh pengguna. 3. Pengguna memilih salah satu kontak.
Hasil yang diharapkan	Pengguna memilih kontak pasien yang pernah didiagnosa dokter. Kemudian sistem menampilkan halaman percakapan.
Hasil yang didapat	Sistem menampilkan halaman percakapan dengan pasien setelah tujuan dipilih.
Hasil pengujian	Berhasil
<i>Skenario 2 - Memilih kontak dokter</i>	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada halaman perpesanan
Data Uji	Kontak Dokter yang pernah mendiagnosa Pasien
Langkah pengujian	1. Pengguna memilih kontak dokter. 2. Sistem menampilkan hanya kontak dokter yang pernah mendiagnosa pengguna. 3. Pengguna memilih salah satu kontak.
Hasil yang diharapkan	Pengguna memilih kontak dokter yang pernah mendiagnosa pengguna dan sistem menampilkan laman percakapan.
Hasil yang didapat	Sistem menampilkan halaman percakapan dengan dokter setelah tujuan dipilih.
Hasil pengujian	Berhasil
<i>Skenario 3 - Mengirim pesan (pada skenario 3, pengirim adalah dokter)</i>	
Pengguna	Pasien/Dokter
Kondisi Awal	Pengguna berada pada halaman percakapan
Data Uji	Masukan pesan teks
Langkah pengujian	1. Pengguna menulis pesan teks pada <i>textbox</i> . 2. Pengguna memilih tombol kirim pada <i>textbox</i> .
Hasil yang diharapkan	- Pesan kosong tidak akan terkirim saat tombol kirim dipilih. - Pesan pada <i>textbox</i> ditampilkan dan dikirim setelah tombol kirim dipilih.
Hasil yang didapat	- Pesan kosong tidak terkirim. - Pesan yang dimasukkan tampil menandakan pesan telah terkirim.

Hasil pengujian	Berhasil
<i>Skenario 4 – Menerima pesan (pada skenario 4, penerima adalah pasien)</i>	
Pengguna	Pasien/Dokter
Kondisi Awal	Pengguna berada pada halaman percakapan
Data Uji	Pesan teks
Langkah pengujian	1. Pengguna menerima pesan dari dokter pada laman percakapan.
Hasil yang diharapkan	Pengguna menerima dan membaca pesan dokter
Hasil yang didapat	Pengguna menerima dan membaca pesan dokter
Hasil pengujian	Berhasil



Gambar 5. 1. Pengujian pengiriman pesan



Gambar 5. 2. Pengujian penerimaan pesan

5.3.2 Pengujian Fungsionalitas Membuat Diagnosa Pasien

Pengujian kasus kebutuhan fungsionalitas membuat diagnosa pasien melibatkan data pasien yang didiagnosa. Pengisian formulir data diagnosa menjadi skenario utama, sedangkan skenario alternatif akan dijalankan apabila pasien merupakan pasien baru yang belum terdaftar pada sistem.

Tabel 5. 3. Pengujian fungsionalitas mengelola perpesanan

Kode	P-002
Referensi kasus penggunaan	UC-002
Nama	Membuat Diagnosa Pasien
Tujuan	Menguji pengisian data diagnosa pasien
Skenario 1 – Mengisi formulir data diagnosa	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman diagnosa baru
Data Uji	Masukan data diagnosa
Langkah pengujian	<ol style="list-style-type: none"> Pengguna memilih pasien. Sistem menampilkan formulir isian diagnosa. Pengguna mengisi dan menyimpan data diagnosa.

Hasil yang diharapkan	Data diagnosa berhasil disimpan oleh sistem.
Hasil yang didapat	Data diagnosa disimpan oleh sistem.
Hasil pengujian	Berhasil
<i>Skenario 2 – Membuat data pasien baru</i>	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman diagnosa baru
Data Uji	Masukan data pasien
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu pasien baru. 2. Sistem menampilkan formulir isian pasien baru. 3. Pengguna mengisi dan menyimpan data pasien baru.
Hasil yang diharapkan	Data pasien baru disimpan oleh sistem dan isian data diagnosa ditampilkan.
Hasil yang didapat	Data pasien baru disimpan oleh sistem dan isian data diagnosa ditampilkan.
Hasil pengujian	Berhasil

Gambar 5. 3. Pengujian pembuatan data pasien baru

The screenshot shows a software interface with a dark header bar containing four tabs: 'DASBOR' (Dashboard), 'PASIEN' (Patient), 'DIAGNOSA' (Diagnosis), and 'PERESKRIPSIAN' (Prescription). The 'DIAGNOSA' tab is active.

PILIH PASIEN

Klik tombol pilih untuk memilih atau mengganti pasien.

Nama Pasien	Ardien Hutaapean	<input checked="" type="checkbox"/> Pilih
Data Pasien		
Nama Pasien:	Ardien Hutaapean	
Umur Pasien:	34	
Alamat:	Jl. Candi Mendut no. 45	
Keterangan:	Ada riwayat ginjal	

Pasien baru?

DETIL DIAGNOSA

Komplikasi *	<input type="text"/>
Keluhan *	<input type="text"/>
Tindakan *	<input type="text"/>
Saran	<input type="text"/>

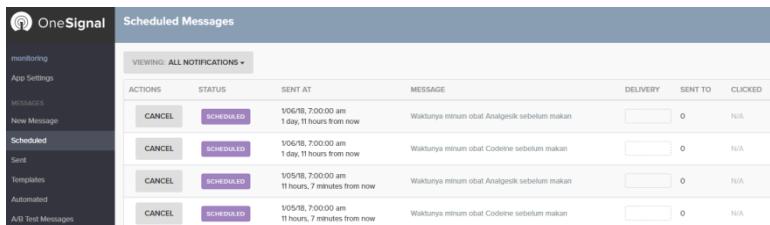
Gambar 5. 4. Pengujian pengisian data diagnosa

5.3.3 Pengujian Fungsionalitas Membuat Preskripsi Medikasi Pasien

Pengujian kebutuhan fungsionalitas membuat preskripsi medikasi pasien dilakukan setelah data diagnosa pasien disimpan. Pengujian difokuskan pada masukan data obat dan pembuatan jadwal minum sesuai dengan penggunaan obat yang telah dimasukkan.

Tabel 5. 4. Pengujian fungsionalitas membuat preskripsi medikasi pasien

Kode	P-003
Referensi kasus penggunaan	UC-003
Nama	Membuat Preskripsi Medikasi Pasien
Tujuan	Menguji pengisian data obat dan jadwal minum
Skenario 1 – Membuat preskripsi baru	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman diagnosa pasien
Data Uji	Masukan data obat, jadwal minum obat
Langkah pengujian	<ol style="list-style-type: none"> Pengguna memilih menu preskripsi baru. Sistem menampilkan formulir isian preskripsi. Pengguna mengisi dan menyimpan data obat.
Hasil yang diharapkan	<ul style="list-style-type: none"> Data obat berhasil disimpan oleh sistem. Jadwal minum berhasil disimpan di <i>database</i>. Jadwal minum untuk notifikasi berhasil dikirim ke <i>server OneSignal</i>.
Hasil yang didapat	<ul style="list-style-type: none"> Data obat disimpan oleh sistem. Jadwal minum disimpan di dalam <i>database</i>. Jadwal minum untuk notifikasi dikirim ke <i>server OneSignal</i>.
Hasil pengujian	Berhasil



Gambar 5. 5. Jadwal minum obat yang tersimpan pada server OneSignal

PRESKRIPSI	
Nama obat	Codeine
Dosis	30mg
Jumlah	3 tablet
Jenis obat	Tablet
Penggunaan	(1x1) 1 kali sehari, sebelum makan - Tidak ada keterangan obat -
Jangka terapi	3 hari
✓ UBAH DATA OBAT ✗ HAPUS DATA OBAT	

Nama obat	Analgesik
Dosis	30mg
Jumlah	3 Tablet
Jenis obat	Tablet
Penggunaan	(1x1) 1 kali sehari, sebelum makan - Tidak ada keterangan obat -
Jangka terapi	3 hari
✓ UBAH DATA OBAT ✗ HAPUS DATA OBAT	

Gambar 5. 6. Hasil pengujian membuat preskripsi medikasi pasien

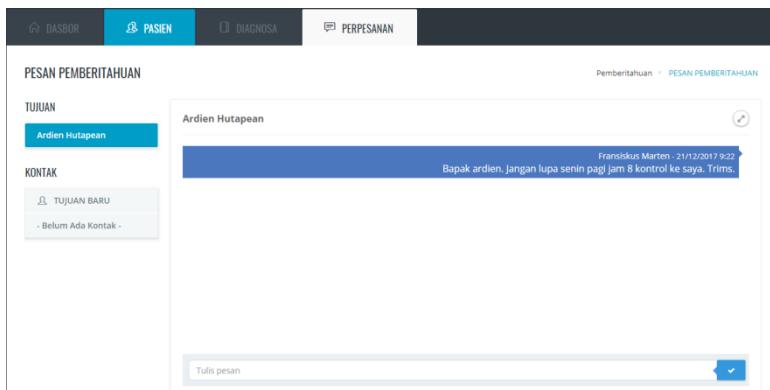
5.3.4 Pengujian Fungsionalitas Mengirim Pesan Pemberitahuan

Pesan pemberitahuan dikirim oleh dokter kepada pasien dan tidak dapat dibalas oleh pasien. Pasien yang dituju adalah pasien yang dipilih dan pasien yang pernah didiagnosa saja. Pengujian ini melibatkan pengguna dokter sebagai pengirim pesan dan pasien sebagai penerima pesan. Penerimaan pesan akan dibahas pada subbab pengujian berikutnya.

Tabel 5. 5. Pengujian fungsionalitas mengirim pesan pemberitahuan

Kode	P-004
Referensi kasus	UC-004

penggunaan	
Nama	Mengirim Pesan Pemberitahuan
Tujuan	Menguji pengiriman pesan pemberitahuan
Skenario 1 – Mengirim pesan pemberitahuan	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman pemberitahuan
Data Uji	Masukan pesan pemberitahuan
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih kontak pasien yang dituju. 2. Sistem menampilkan percakapan pemberitahuan. 3. Pengguna menulis pesan pada <i>textbox</i> kemudian memilih tombol kirim.
Hasil yang diharapkan	<ul style="list-style-type: none"> - Pesan kosong tidak terkirim saat tombol kirim dipilih. - Pesan pada <i>textbox</i> ditampilkan dan dikirim setelah tombol kirim dipilih.
Hasil yang didapat	<ul style="list-style-type: none"> - Pesan kosong tidak terkirim. - Pesan yang dimasukkan tampil menandakan pesan telah terkirim.
Hasil pengujian	Berhasil



Gambar 5. 7. Hasil pengujian mengirim pesan pemberitahuan

5.3.5 Pengujian Fungsionalitas Melihat Pesan Pemberitahuan

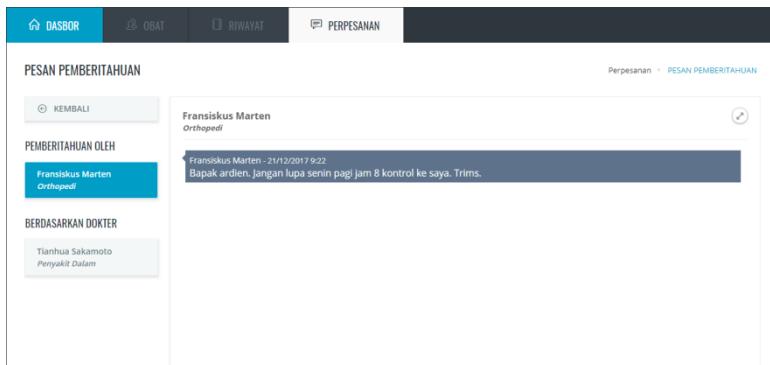
Pesan pemberitahuan yang dikirim oleh pengguna dokter kepada pasien dapat dilihat dan dibaca namun tidak dapat dibalas oleh pasien.

Tabel 5. 6. Pengujian fungsionalitas melihat pesan pemberitahuan

Kode	P-005
Referensi kasus penggunaan	UC-005
Nama	Melihat Pesan Pemberitahuan
Tujuan	Menguji penerimaan pesan pemberitahuan
<i>Skenario 1 – Melihat pesan pemberitahuan</i>	
Pengguna	Pasienn
Kondisi Awal	Pengguna berada pada halaman pemberitahuan
Data Uji	Masukan pesan pemberitahuan
Langkah pengujian	1. Pengguna memilih pesan pemberitahuan baru. 2. Sistem menampilkan pesan pemberitahuan.
Hasil yang diharapkan	Pesan pemberitahuan ditampilkan dalam bentuk percakapan pasien dengan dokter.
Hasil yang didapat	Pesan pemberitahuan ditampilkan dalam bentuk percakapan pasien dengan dokter.
Hasil pengujian	Berhasil



Gambar 5. 8. Menu utama pesan pemberitahuan



Gambar 5. 9. Isi pesan pemberitahuan dari dokter yang ditampilkan dalam bentuk percakapan

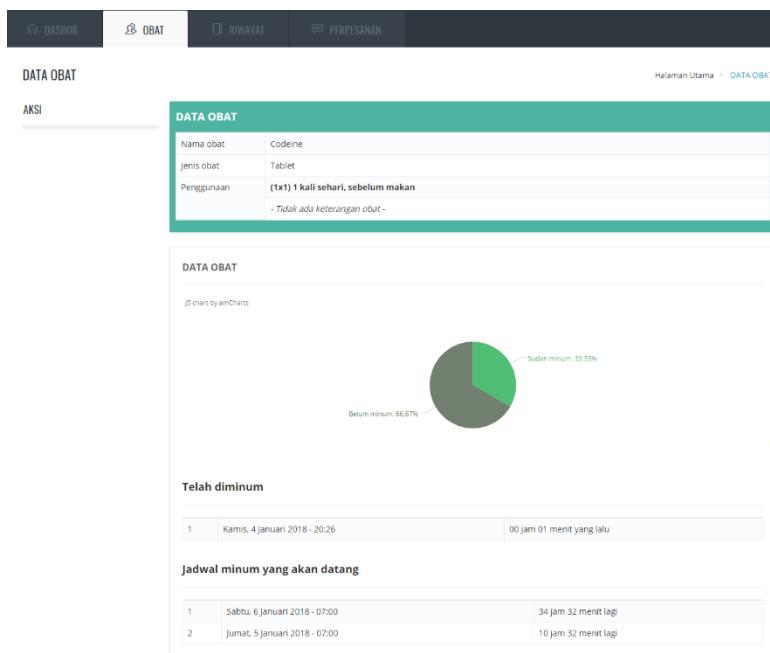
5.3.6 Pengujian Fungsionalitas Melihat Perkembangan Pasien

Jadwal obat yang telah dibuat, nantinya akan dikirim oleh sistem kepada pasien sebagai pesan pengingat. Setelah pasien mendapatkan pesan pengingat, pasien mengkonfirmasi minum obat. Data minum obat akan diuji pada subbab berikutnya. Pada subbab ini, sistem diharapkan dapat menampilkan data minum obat pasien beserta dengan grafik dan jadwal minum.

Tabel 5. 7. Pengujian fungsionalitas melihat perkembangan pasien

Kode	P-006
Referensi kasus penggunaan	UC-006
Nama	Melihat Perkembangan Pasien
Tujuan	Menguji data perkembangan pasien
<i>Skenario 1 – Melihat perkembangan pasien</i>	
Pengguna	Dokter
Kondisi Awal	Pengguna berada pada halaman detil diagnosa

Data Uji	Data minum obat pasien
Langkah pengujian	<ol style="list-style-type: none"> Pengguna memilih menu lihat data obat. Sistem menampilkan laman data obat beserta data minum obat pasien.
Hasil yang diharapkan	Data obat, data minum obat, dan grafik dapat ditampilkan dengan benar dan sesuai.
Hasil yang didapat	Data obat, data minum obat, dan grafik ditampilkan dengan benar dan sesuai.
Hasil pengujian	Berhasil



Gambar 5. 10. Hasil pengujian melihat perkembangan pasien

5.3.7 Pengujian Fungsionalitas Mengirim Pesan Konfirmasi

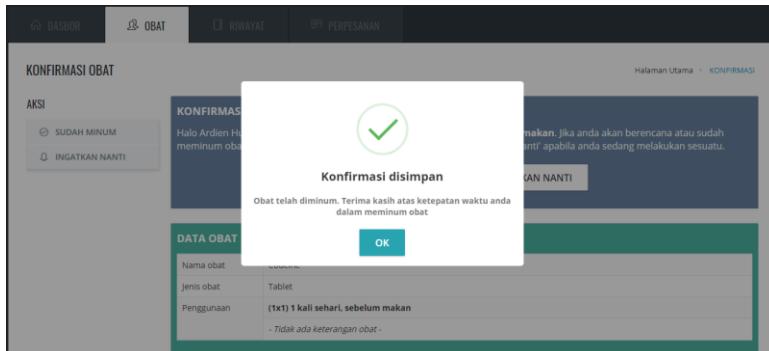
Saat jadwal minum obat telah tiba, sistem mengirimkan notifikasi kepada pasien berisi pesan pengingat minum obat.

Pasien kemudian mengirim pesan konfirmasi ketika notifikasi dibuka. Pasien dapat mengkonfirmasi minum obat sekarang atau memilih untuk diingatkan kembali dengan rentang waktu 30 menit kedepan. Data konfirmasi digunakan untuk melihat perkembangan pasien pada subbab sebelumnya.

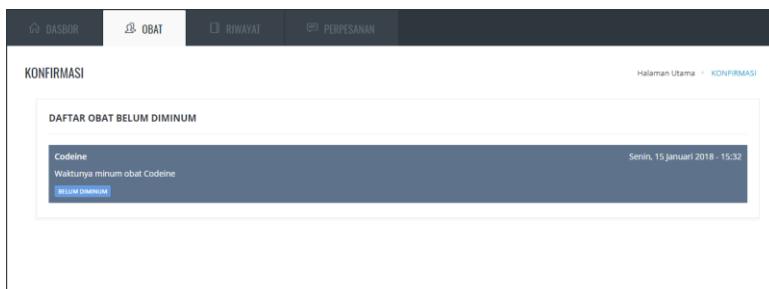
Tabel 5. 8. Pengujian fungsionalitas mengirim pesan konfirmasi

Kode	P-007
Referensi kasus penggunaan	UC-007
Nama	Mengirim Pesan Konfirmasi
Tujuan	Menguji data konfirmasi minum obat
<i>Skenario 1 – Konfirmasi minum obat</i>	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada konfirmasi minum obat
Data Uji	Konfirmasi minum obat pasien
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu sudah minum. 2. Sistem menyimpan data konfirmasi.
Hasil yang diharapkan	Data konfirmasi berhasil disimpan.
Hasil yang didapat	Data konfirmasi disimpan.
Hasil pengujian	Berhasil
<i>Skenario 2 – Memilih menu ingatkan nanti</i>	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada konfirmasi minum obat
Data Uji	Data jadwal minum obat pasien
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu ingatkan nanti. 2. Sistem membuat jadwal minum obat 30 menit kedepan.
Hasil yang diharapkan	Data jadwal minum obat berhasil dibuat dan sistem menampilkan data obat yang belum diminum pada halaman konfirmasi obat
Hasil yang didapat	Data jadwal minum obat berhasil dibuat dan sistem menampilkan data obat yang belum diminum pada

	halaman konfirmasi obat
Hasil pengujian	Berhasil



Gambar 5. 11. Hasil pengujian mengirim pesan konfirmasi



Gambar 5. 12. Hasil pengujian memilih menu ingatkan nanti

5.3.8 Pengujian Fungsionalitas Mengubah Waktu Minum

Pengujian fungsionalitas melibatkan dua skenario, pengubahan jadwal minum obat dan pengubahan waktu konfirmasi minum obat. Pengubahan jadwal minum obat mengubah data jadwal pada *database* dan layanan notifikasi OneSignal yang dapat dilakukan oleh pasien dan dokter (hanya kasus khusus jika pasien meminta perubahan jadwal minum saat jadwal tatap muka). Sedangkan pengubahan waktu konfirmasi minum obat

hanya dapat dilakukan oleh pasien untuk mengubah waktu konfirmasi minum obat. Waktu baru disimpan pada *database*.

Tabel 5. 9. Pengujian fungsionalitas mengubah waktu minum

Kode	P-008
Referensi kasus penggunaan	UC-008
Nama	Mengubah Waktu Minum
Tujuan	Menguji data konfirmasi minum obat
Skenario 1 – Mengubah Jadwal Minum Obat	
Pengguna	Pasien, Dokter
Kondisi Awal	Pengguna berada pada halaman data obat
Data Uji	Jadwal minum obat yang dipilih
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu ubah jadwal. 2. Sistem menampilkan halaman ubah jadwal minum obat. 3. Pengguna mengubah jadwal minum obat dan menyimpan jadwal. 4. Sistem menyimpan jadwal minum obat yang baru.
Hasil yang diharapkan	<ul style="list-style-type: none"> - Jadwal minum obat baru tersimpan di dalam <i>database</i>. - Jadwal minum obat baru tersimpan pada OneSignal.
Hasil yang didapat	<ul style="list-style-type: none"> - Jadwal minum obat baru tersimpan di dalam <i>database</i>. - Jadwal minum obat baru tersimpan pada OneSignal.
Hasil pengujian	Berhasil
Skenario 2 – Mengubah Waktu Konfirmasi Minum Obat	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada halaman data obat
Data Uji	Waktu konfirmasi minum obat yang dipilih
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu ubah pada konfirmasi minum obat yang dipilih. 2. Sistem menampilkan halaman ubah waktu konfirmasi minum obat.

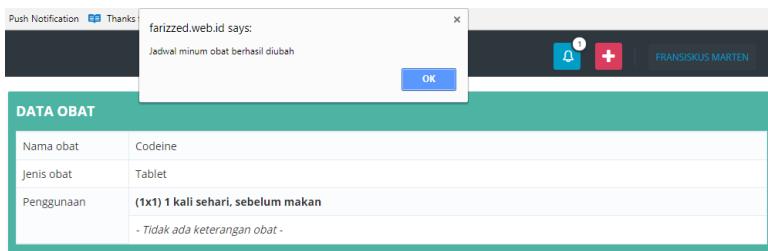
	3. Pengguna mengubah waktu konfirmasi minum obat dan menyimpan waktu. 4. Sistem menyimpan waktu konfirmasi minum obat yang baru
Hasil yang diharapkan	Waktu konfirmasi minum obat baru tersimpan di dalam <i>database</i>
Hasil yang didapat	Waktu konfirmasi minum obat baru tersimpan di dalam <i>database</i>
Hasil pengujian	Berhasil

Jadwal minum yang akan datang

UBAH

1	Sabtu, 20 Januari 2018 - 09:00	23 jam 07 menit lagi
2	Minggu, 21 Januari 2018 - 09:00	47 jam 07 menit lagi
3	Senin, 22 Januari 2018 - 09:00	71 jam 07 menit lagi

Gambar 5. 13. Kondisi awal pengujian mengubah jadwal minum obat satu kali sehari



The screenshot shows a form titled 'UBAH JADWAL MINUM'. It has two input fields: 'Tanggal mulai' with the value '20-01-2018' and 'Waktu mulai' with the value '21:00'. At the bottom right is a blue button labeled 'SIMPAN'.

Gambar 5. 14. Pengujian menyimpan jadwal minum obat satu kali sehari

Jadwal minum yang akan datang

UBAH

1	Sabtu, 20 Januari 2018 - 21:00	35 jam 04 menit lagi
2	Minggu, 21 Januari 2018 - 21:00	59 jam 04 menit lagi
3	Senin, 22 Januari 2018 - 21:00	83 jam 04 menit lagi

Gambar 5. 15. Hasil akhir pengujian mengubah jadwal minum obat satu kali sehari

Waktunya minum obat Amoxicillin sesudah makan		1/21/18, 10:00:00 pm 2 days, 12 hours from now	<input type="text"/>	0	N/A	
Waktunya minum obat Codeme sebelum makan		1/21/18, 9:00:00 pm 2 days, 11 hours from now	<input type="text"/>	0	N/A	
Waktunya minum obat Allopurinol sesudah makan		1/21/18, 8:00:00 pm 2 days, 10 hours from now	<input type="text"/>	0	N/A	

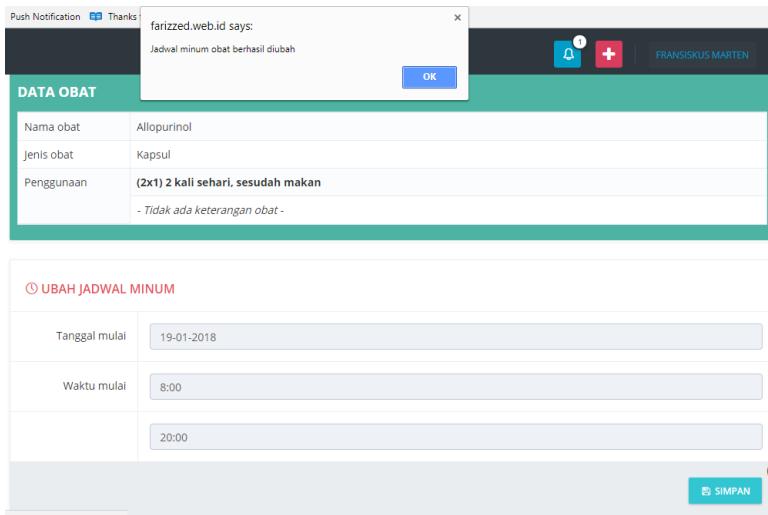
Gambar 5. 16. Hasil akhir pengujian mengubah jadwal minum obat satu kali sehari pada OneSignal

Jadwal minum yang akan datang

UBAH

1	Jumat, 19 Januari 2018 - 19:00	09 jam 16 menit lagi
2	Sabtu, 20 Januari 2018 - 07:00	21 jam 16 menit lagi
3	Sabtu, 20 Januari 2018 - 19:00	33 jam 16 menit lagi
4	Minggu, 21 Januari 2018 - 07:00	45 jam 16 menit lagi
5	Minggu, 21 Januari 2018 - 19:00	57 jam 16 menit lagi
6	Senin, 22 Januari 2018 - 07:00	69 jam 16 menit lagi
7	Senin, 22 Januari 2018 - 19:00	81 jam 16 menit lagi
8	Selasa, 23 Januari 2018 - 07:00	93 jam 16 menit lagi
9	Selasa, 23 Januari 2018 - 19:00	105 jam 16 menit lagi
10	Rabu, 24 Januari 2018 - 07:00	117 jam 16 menit lagi

Gambar 5. 17. Kondisi awal pengujian mengubah jadwal minum obat dua kali sehari



Gambar 5. 18. Pengujian menyimpan jadwal minum obat dua kali sehari

Jadwal minum yang akan datang		Ubah
1	Jumat, 19 Januari 2018 - 20:00	10 jam 14 menit lagi
2	Sabtu, 20 Januari 2018 - 08:00	22 jam 14 menit lagi
3	Sabtu, 20 Januari 2018 - 20:00	34 jam 14 menit lagi
4	Minggu, 21 Januari 2018 - 08:00	46 jam 14 menit lagi
5	Minggu, 21 Januari 2018 - 20:00	58 jam 14 menit lagi
6	Senin, 22 Januari 2018 - 08:00	70 jam 14 menit lagi
7	Senin, 22 Januari 2018 - 20:00	82 jam 14 menit lagi
8	Selasa, 23 Januari 2018 - 08:00	94 jam 14 menit lagi
9	Selasa, 23 Januari 2018 - 20:00	106 jam 14 menit lagi
10	Rabu, 24 Januari 2018 - 08:00	118 jam 14 menit lagi

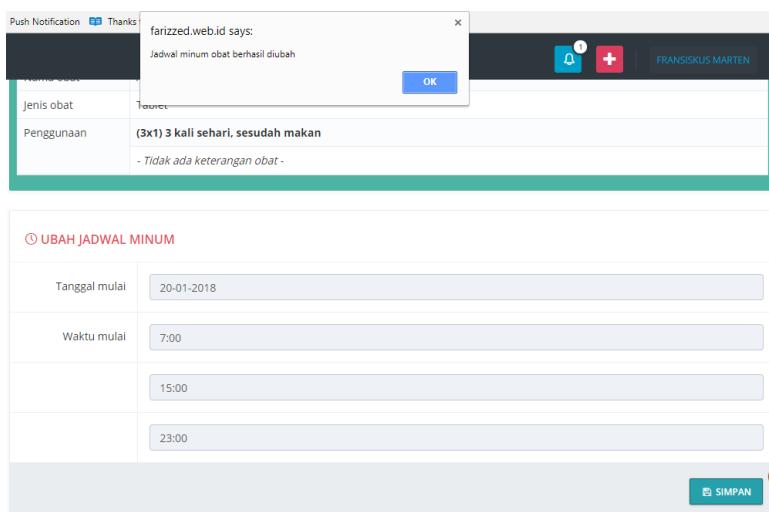
Gambar 5. 19. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari

MESSAGE	STATUS	SENT AT	DELIVERY	SENT TO	CLICKED	ACTIONS
Waktunya minum obat Allopurinol sesudah makan	SCHEDULED	1/24/18, 8:00:00 am 4 days, 22 hours from now		0	N/A	<button>CANCEL</button>
Waktunya minum obat Allopurinol sesudah makan	CANCELED	1/24/18, 7:00:00 am 4 days, 21 hours from now		0	N/A	<button>DELETE</button>
Waktunya minum obat Allopurinol sesudah makan	SCHEDULED	1/23/18, 8:00:00 pm 4 days, 10 hours from now		0	N/A	<button>CANCEL</button>
Waktunya minum obat Allopurinol sesudah makan	CANCELED	1/23/18, 7:00:00 pm 4 days, 9 hours from now		0	N/A	<button>DELETE</button>
Waktunya minum obat Allopurinol sesudah makan	SCHEDULED	1/23/18, 8:00:00 am 3 days, 22 hours from now		0	N/A	<button>CANCEL</button>
Waktunya minum obat Allopurinol sesudah makan	CANCELED	1/23/18, 7:00:00 am 3 days, 21 hours from now		0	N/A	<button>DELETE</button>
Waktunya minum obat Amoxicillin sesudah makan	SCHEDULED	1/22/18, 10:00:00 pm 3 days, 12 hours from now		0	N/A	<button>CANCEL</button>
Waktunya minum obat Amoxicillin sesudah makan	SCHEDULED	1/22/18, 8:00:00 pm 3 days, 10 hours from now		0	N/A	<button>CANCEL</button>
Waktunya minum obat Amoxicillin sesudah makan	CANCELED	1/22/18, 7:00:00 pm 3 days, 9 hours from now		0	N/A	<button>DELETE</button>

Gambar 5. 20. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari pada OneSignal

Jadwal minum yang akan datang			<button>UBAH</button>
1	Sabtu, 20 Januari 2018 - 06:00	19 jam 57 menit lagi	
2	Sabtu, 20 Januari 2018 - 14:00	27 jam 57 menit lagi	
3	Sabtu, 20 Januari 2018 - 22:00	35 jam 57 menit lagi	
4	Minggu, 21 Januari 2018 - 06:00	43 jam 57 menit lagi	
5	Minggu, 21 Januari 2018 - 14:00	51 jam 57 menit lagi	
6	Minggu, 21 Januari 2018 - 22:00	59 jam 57 menit lagi	
7	Senin, 22 Januari 2018 - 06:00	67 jam 57 menit lagi	
8	Senin, 22 Januari 2018 - 14:00	75 jam 57 menit lagi	
9	Senin, 22 Januari 2018 - 22:00	83 jam 57 menit lagi	
10	Selasa, 23 Januari 2018 - 06:00	91 jam 57 menit lagi	
11	Selasa, 23 Januari 2018 - 14:00	99 jam 57 menit lagi	
12	Selasa, 23 Januari 2018 - 22:00	107 jam 57 menit lagi	
13	Rabu, 24 Januari 2018 - 06:00	115 jam 57 menit lagi	
14	Rabu, 24 Januari 2018 - 14:00	123 jam 57 menit lagi	

Gambar 5. 21. Kondisi awal pengujian mengubah jadwal minum obat tiga kali sehari



Gambar 5. 22. Pengujian menyimpan jadwal minum obat tiga kali sehari

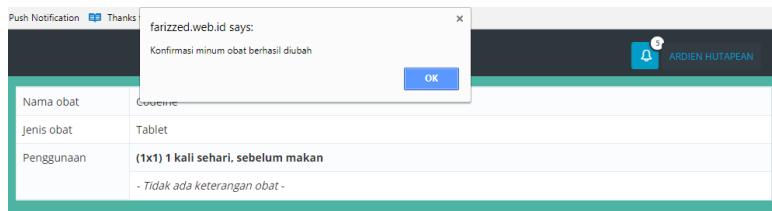
Jadwal minum yang akan datang		
		UBAH
1	Sabtu, 20 Januari 2018 - 07:00	20 jam 55 menit lagi
2	Sabtu, 20 Januari 2018 - 15:00	28 jam 55 menit lagi
3	Sabtu, 20 Januari 2018 - 23:00	36 jam 55 menit lagi
4	Minggu, 21 Januari 2018 - 07:00	44 jam 55 menit lagi
5	Minggu, 21 Januari 2018 - 15:00	52 jam 55 menit lagi
6	Minggu, 21 Januari 2018 - 23:00	60 jam 55 menit lagi
7	Senin, 22 Januari 2018 - 07:00	68 jam 55 menit lagi
8	Senin, 22 Januari 2018 - 15:00	76 jam 55 menit lagi
9	Senin, 22 Januari 2018 - 23:00	84 jam 55 menit lagi
10	Selasa, 23 Januari 2018 - 07:00	92 jam 55 menit lagi
11	Selasa, 23 Januari 2018 - 15:00	100 jam 55 menit lagi
12	Selasa, 23 Januari 2018 - 23:00	108 jam 55 menit lagi
13	Rabu, 24 Januari 2018 - 07:00	116 jam 55 menit lagi
14	Rabu, 24 Januari 2018 - 15:00	124 jam 55 menit lagi

Gambar 5. 23. Hasil akhir pengujian mengubah jadwal minum obat tiga kali sehari

MESSAGE	STATUS	SENT AT	DELIVERY	SENT TO	CLICKED	ACTIONS
Waktunya minum obat Amoxicillin sesudah makan	SCHEDULED	1/24/18, 11:00:00 pm 5 days, 12 hours from now		0	N/A	CANCEL
Waktunya minum obat Amoxicillin sesudah makan	SCHEDULED	1/24/18, 3:00:00 pm 5 days, 4 hours from now		0	N/A	CANCEL
Waktunya minum obat Allopurinol sesudah makan	SCHEDULED	1/24/18, 8:00:00 am 4 days, 21 hours from now		0	N/A	CANCEL
Waktunya minum obat Amoxicillin sesudah makan	SCHEDULED	1/24/18, 7:00:00 am 4 days, 20 hours from now		0	N/A	CANCEL

Gambar 5. 24. Hasil akhir pengujian mengubah jadwal minum obat dua kali sehari pada OneSignal

Hasil pengujian diatas untuk menguji bahwa jadwal baru tersimpan pada *database*. Setelah tersimpan pada *database*, kode unik notifikasi OneSignal untuk pengingat yang lama digunakan untuk membatalkan pengiriman pesan dan membuat pesan pengiriman baru yang sesuai dengan jadwal yang baru disimpan.



The screenshot shows a mobile application interface with a title '① UBAH KONFIRMASI MINUM OBAT'. It contains two input fields:

- Tanggal: 15-01-2018
- Waktu: 3:04

At the bottom right is a blue button labeled 'SIMPAN'.

Gambar 5. 25. Hasil pengujian mengubah waktu konfirmasi minum obat

5.3.9 Pengujian Sistem IVR

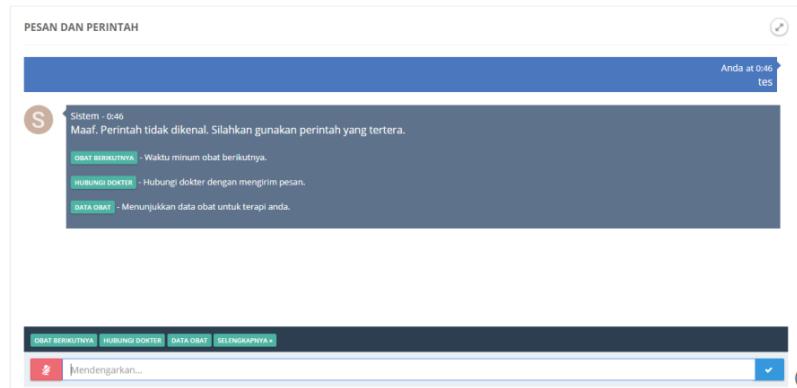
Pengujian sistem IVR meliputi pengujian input perintah yang ditangkap oleh sistem. Ada tiga cara untuk memasukkan perintah, yaitu menggunakan perintah suara, memilih perintah

dari *command box*, dan menulis perintah melalui *textbox*. Sistem selalu memberikan perintah yang dapat digunakan oleh pengguna pada *command box* dan menampilkan detil perintah pada balasan.

Tabel 5. 10. Pengujian sistem IVR

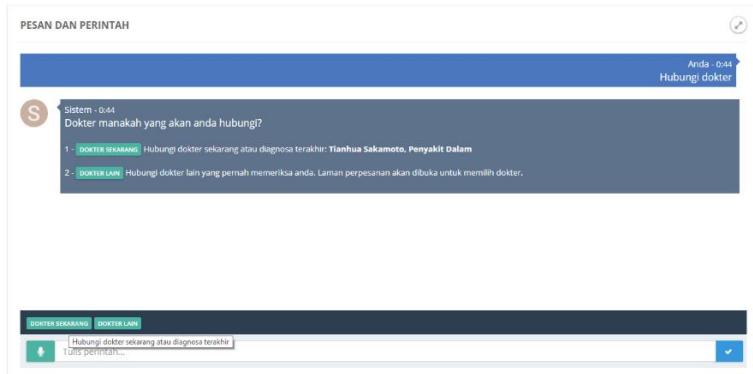
Kode	P-IVR
Nama	Pengujian Sistem IVR
Tujuan	Menguji <i>input</i> suara atau perintah sistem IVR dan menguji <i>output</i> yang dihasilkan
Skenario 1 – Pengguna menggunakan perintah suara	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada halaman utama sistem
Data Uji	Perintah suara pengguna (<i>voice recognition</i>) dan respon sistem (<i>speech synthesis</i>)
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih tombol mikrofon. 2. Sistem menyiapkan penangkapan suara. 3. Pengguna memberikan perintah suara. 4. Sistem menangkap suara, memproses balasan dan merespon balasan perintah yang sesuai.
Hasil yang diharapkan	<ul style="list-style-type: none"> - Sistem menyiapkan penangkapan suara dan menangkap perintah suara (<i>voice recognition</i>). - Sistem memberikan respon suara terhadap perintah suara (<i>speech synthesis</i>).
Hasil yang didapat	<ul style="list-style-type: none"> - Sistem menyediakan mode penangkapan suara dan memproses perintah suara. - Sistem memberikan respon suara.
Hasil pengujian	Berhasil
Skenario 2 – Memilih perintah melalui command box	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada halaman utama sistem
Data Uji	Data perintah
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih menu perintah dari <i>command box</i>. 2. Sistem menangkap perintah.
Hasil yang diharapkan	Data perintah ditangkap dan dibalas oleh sistem.
Hasil yang	Data perintah ditangkap dan dibalas oleh sistem.

didapat	
Hasil pengujian	Berhasil
Skenario 3 – Menulis perintah melalui textbox	
Pengguna	Pasien
Kondisi Awal	Pengguna berada pada halaman utama sistem
Data Uji	Data perintah
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna menulis perintah pada <i>textbox</i> dan memilih tombol kirim. 2. Sistem menangkap perintah.
Hasil yang diharapkan	Data perintah ditangkap dan dibalas oleh sistem.
Hasil yang didapat	Data perintah ditangkap dan dibalas oleh sistem.
Hasil pengujian	Berhasil



Gambar 5. 26. Pengujian sistem IVR skenario pertama, menggunakan perintah suara

Pengujian perintah suara meliputi perintah yang tersedia dan tidak tersedia. Ketika pengguna menyuarakan perintah yang tidak tersedia, maka sistem menampilkan perintah yang dapat digunakan saat ini. Perintah yang tersedia setelah disuarakan oleh pengguna dan ditangkap oleh sistem akan langsung diproses sesuai dengan konteks perintah yang diminta oleh pengguna.



Gambar 5. 27. Pengujian sistem IVR skenario kedua, memilih perintah dari *command box*



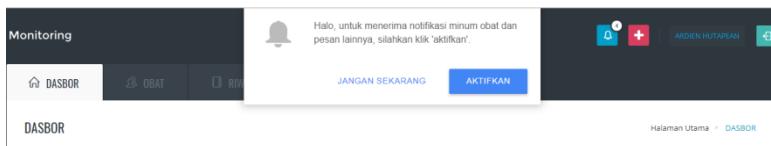
Gambar 5. 28. Pengujian sistem IVR skenario ketiga, menulis perintah pada *textbox*

5.3.10 Pengujian OneSignal Web Push Notification

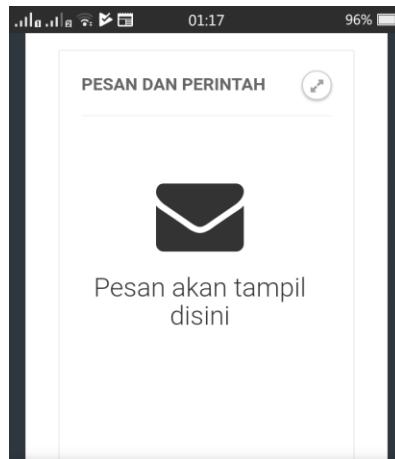
Pengujian OneSignal *push notification* melibatkan pendaftaran pengguna sebagai subscriber untuk menerima notifikasi dan pengiriman notifikasi kepada pengguna terdaftar.

Tabel 5. 11. Pengujian OneSignal web push notification

Kode	P-OneSignal
Nama	Pengujian OneSignal Web Push Notification
Tujuan	Menguji pengiriman notifikasi melalui OneSignal API
Skenario 1 – Pengguna belum terdaftar sebagai subscriber	
Pengguna	Pasien
Kondisi Awal	Pengguna belum masuk dalam daftar <i>subscriber</i>
Data Uji	Data <i>subscription</i> dan <i>user tag</i> .
Langkah pengujian	<ol style="list-style-type: none"> 1. Sistem menampilkan jendela <i>subscription</i>. 2. Pengguna memilih tombol aktifkan. 3. Sistem mengirim pasien_id sebagai <i>user tag</i> dan menambahkan perangkat pada <i>subscriber</i>.
Hasil yang diharapkan	OneSignal menyimpan <i>user tag</i> dan pengguna kedalam daftar <i>user</i> .
Hasil yang didapat	<i>User tag</i> berhasil disimpan dan pengguna telah terdaftar pada daftar <i>user</i> .
Hasil pengujian	Berhasil
Skenario 2 – Menerima notifikasi	
Pengguna	Pasien
Kondisi Awal	Pengguna telah terdaftar sebagai <i>subscriber</i> dan pada waktunya jadwal minum obat.
Data Uji	Pesan notifikasi
Langkah pengujian	<ol style="list-style-type: none"> 1. Pengguna berada pada waktu jadwal minum obat. 2. Sistem menampilkan notifikasi jadwal minum obat. 3. Pengguna memilih pesan notifikasi. 4. Sistem menampilkan laman konfirmasi minum obat.
Hasil yang diharapkan	<ul style="list-style-type: none"> - Sistem dapat menampilkan pesan notifikasi. - Sistem dapat navigasikan pengguna ke laman konfirmasi minum obat.
Hasil yang didapat	<ul style="list-style-type: none"> - Sistem menampilkan pesan notifikasi. - Sistem navigasikan pengguna ke laman konfirmasi minum obat.
Hasil pengujian	Berhasil



Gambar 5. 29. Pengujian skenario pertama, jendela *subscription* pada desktop



Halo, untuk menerima notifikasi minum obat dan pesan lainnya, silahkan klik 'aktifkan'.

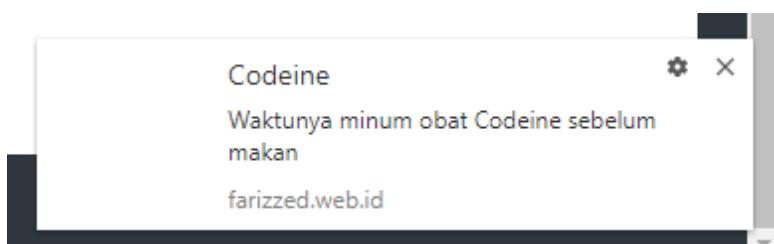
AKTIFKAN

JANGAN SEKARANG

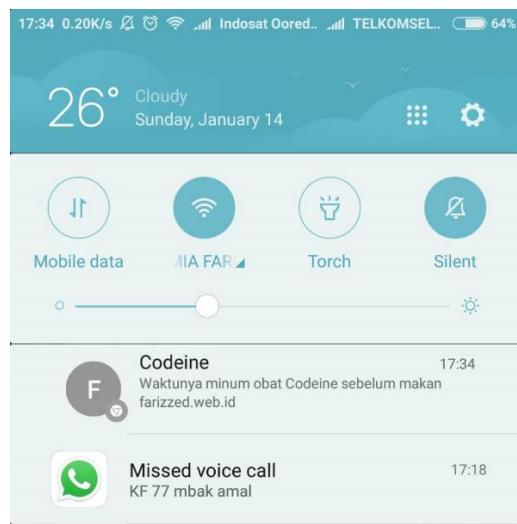
Gambar 5. 30. Pengujian skenario pertama, jendela *subscription* pada perangkat bergerak

Actions	Subscribed	Device	Platform	Player ID	Tags
OPTIONS	Yes	Win32 (63)	Web Push (Chrome)	1e850d8e-8b4e-40cd-93f1-ff89cf1659a0	{pasien_id: 13}
OPTIONS	Yes	Linux aarch64 (53)	Web Push (Chrome)	7a1ee92e-7737-4d85-b761-4e6beb546d30	{pasien_id: 1}

Gambar 5. 31. Hasil pengujian skenario pertama, kedua perangkat berhasil terdaftar pada OneSignal



Gambar 5. 32. Penerimaan notifikasi pada desktop



Gambar 5. 33. Penerimaan notifikasi pada perangkat bergerak

5.4 Evaluasi Pengujian

Berdasarkan hasil dari uji coba yang telah dilakukan pada kebutuhan fungsional dan fitur pada aplikasi, maka dapat dilakukan evaluasi sebagai berikut.

5.4.1 Evaluasi Pengujian Kebutuhan Fungsional

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.12. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik dan bekerja sesuai dengan yang diharapkan.

Tabel 5. 12. Rangkuman hasil pengujian kebutuhan fungsional

No.	Kode	Nama Kasus Penggunaan	Skenario	Hasil
1.	P-001	Mengelola Perpesanan	Skenario 1	Berhasil
			Skenario 2	
			Skenario 3	
			Skenario 4	
2.	P-002	Membuat Diagnosa Pasien	Skenario 1	Berhasil
			Skenario 2	
3.	P-003	Membuat Preskripsi Medikasi Pasien	Skenario 1	Berhasil
4.	P-004	Mengirim Pesan Pemberitahuan	Skenario 1	Berhasil
5.	P-005	Melihat Pesan Pemberitahuan	Skenario 1	Berhasil
6.	P-006	Melihat Perkembangan Pasien	Skenario 1	Berhasil
7.	P-007	Mengirim Pesan Konfirmasi	Skenario 1	Berhasil
			Skenario 2	
8.	P-008	Mengubah Waktu Minum	Skenario 1	Berhasil
			Skenario 2	

5.4.2 Evaluasi Pengujian Fitur

Rangkuman mengenai hasil pengujian fitur dapat dilihat pada Tabel 5.13. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik dan bekerja sesuai dengan yang diharapkan.

Tabel 5. 13. Rangkuman hasil pengujian fitur

No.	Kode	Nama Pengujian	Skenario	Hasil
1.	P-IVR	Pengujian Sistem IVR	Skenario 1	Berhasil
			Skenario 2	
			Skenario 3	
2.	P-OneSignal	Pengujian OneSignal Web Push Notification	Skenario 1	Berhasil
			Skenario 2	

Catatan: pada sistem IVR untuk perangkat bergerak (mobile). Balasan suara sistem (speech synthesis) mengikuti bahasa yang diatur dalam perangkat. Untuk menampilkan suara sistem dalam Bahasa Indonesia, bahasa perangkat harus diatur dalam Bahasa Indonesia pada pengaturan perangkat.

5.4.3 Evaluasi Kelebihan dan Kekurangan Sistem

Perangkat lunak yang dibangun menggunakan teknologi *web* memiliki beberapa kelebihan dan kekurangan. Beberapa perangkat lunak yang sudah ada tersedia dalam bentuk aplikasi *mobile* atau perangkat bergerak. Sebagai perbandingan, penulis memilih aplikasi Android bernama Medisafe Pill Reminder yang memiliki kegunaan yang sama dan nilai ulasan yang sangat baik. Adapun perbandingan kelebihan dan kekurangan dari sistem dan aplikasi *mobile* Medisafe Pill Reminder dapat dijabarkan sebagai berikut.

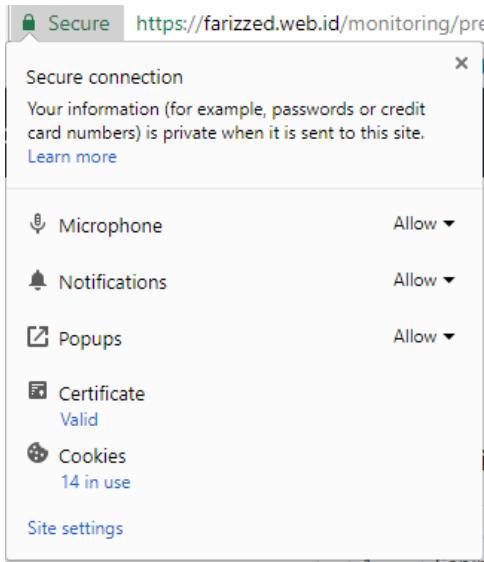
Tabel 5. 14. Perbandingan sistem dengan aplikasi Android

Sistem (<i>web</i>)	Medisafe Pill Reminder (Android)
<ul style="list-style-type: none"> - <i>Multiplatform</i>, dapat diakses oleh perangkat apa saja yang memiliki <i>browser</i> - Sistem dapat berjalan tanpa adanya konfigurasi dari pasien (<i>near zero configuration</i>) - Dapat diakses oleh perangkat lain yang terhubung oleh internet - Membutuhkan koneksi internet - Keterbatasan Web Notification API dalam pengiriman notifikasi pada versi <i>web</i> membutuhkan layanan pihak ketiga sebagai pengirim notifikasi agar dapat menerima notifikasi saat browser ditutup atau tabulasi halaman ditutup 	<ul style="list-style-type: none"> - Hanya dapat diakses pada perangkat dengan aplikasi terpasang dan memiliki sistem operasi Android versi tertentu - Pengguna diharuskan melakukan konfigurasi awal untuk mengatur penjadwalan minum obat - Hanya dapat diakses oleh perangkat dengan aplikasi terpasang - Tidak selalu membutuhkan koneksi internet - Aplikasi dapat langsung mengirimkan pesan notifikasi walau perangkat dalam keadaan <i>standby</i> atau aplikasi tidak dibuka

Selama sistem dibangun dan diuji, keterbatasan sistem dapat dijabarkan karena *platform web* yang memiliki fitur-fitur yang terbatas serta pengembangannya membutuhkan teknologi lain untuk bisa bekerja dengan baik. Keterbatasan-keterbatasan pada sistem ini dapat dijabarkan sebagai berikut:

1. Penggunaan teknologi wicara dan penangkapan suara menggunakan Web Speech API membutuhkan izin khusus. Seperti dicantumkan pada spesifikasi W3C [11], untuk dapat menggunakan API ini dibutuhkan sebuah situs *web* yang memiliki sertifikat SSL (*Secure Socket Layer*), biasa dikenal dengan ‘*https://*’ pada alamat situs yang menunjukkan bahwa situs

memiliki sertifikat keamanan yang dapat dipercaya untuk menggunakan mikrofon.



Gambar 5. 34. Koneksi aman dibutuhkan untuk mengakses mikrofon

2. Web Speech API baru mendukung beberapa *browser* terkini, seperti Google Chrome, Mozilla Firefox, dan Microsoft Edge [12].
3. Pengiriman pesan pengingat berupa notifikasi menggunakan Web Notification API harus menggunakan *service worker* pada situs aman menggunakan SSL (*Secure Socket Layer*) [16]. Agar notifikasi dapat diterima saat pengguna menutup situs *web* atau mematikan *browser*. Untuk *service worker*, penulis menggunakan OneSignal.
4. Mengikuti aturan layanan API OneSignal, pengguna harus mendaftar (*subscribe*) untuk menerima notifikasi. Pengguna cukup memilih menu ‘aktifkan’ tanpa mengisi formulir dan tanpa mengirimkan data pribadi kepada OneSignal.

BAB VI

KESIMPULAN DAN SARAN

Bab VI membahas kesimpulan yang diambil selama pengerjaan tugas akhir, saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang, dan kegunaan lanjutan dari hasil yang didapat dari tugas akhir ini.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Sistem pengingat dapat membantu pasien dalam meningkatkan tingkat kepatuhan minum obat yang diberikan oleh dokter. Dengan catatan, pasien lansia didampingi oleh seorang pendamping. Tata cara untuk mengkonfirmasi minum obat masih bisa dibilang cukup rumit karena membutuhkan perangkat *desktop* atau perangkat bergerak serta langkah-langkah yang masih sulit dimengerti oleh lansia.
2. Sistem dapat menjembatani komunikasi antara pasien dengan dokter diluar proses tatap muka dengan mengirim perpesan.
3. Web Speech API merupakan API yang baik untuk mengimplementasikan perintah suara dan penangkapan suara untuk aplikasi *website*. Hanya saja memiliki keterbatasan pada *browser* yang mendukung API ini.
4. OneSignal merupakan layanan pengiriman notifikasi melalui *web push*. Layanan ini dapat menerima perintah pengiriman notifikasi secara fleksibel dan mengirimkan notifikasi kepada target pengguna yang sesuai dengan cepat.

6.2 Saran

Saran yang diberikan untuk pengembangan perangkat lunak ini adalah:

1. Pengembangan penangkapan perintah suara dan teknologi wicara menggunakan Google Voice API yang mendukung berbagai *browser*. Kekurangan dari Google Voice API adalah API ini berbayar.
2. OneSignal sebagai layanan pengirim notifikasi dapat mengirimkan notifikasi menuju aplikasi perangkat bergerak seperti Android.
3. Untuk mempermudah proses konfirmasi minum obat oleh para lansia, aplikasi dapat dikembangkan lebih jauh menggunakan *SMS Gateway* dan *VoIP* [17]. Kedua teknologi tersebut dapat mengakomodasi berbagai macam perangkat telekomunikasi, termasuk handphone.
4. Perpesanan menggunakan *thread* agar perpesanan dapat dilakukan secara *real-time*.

DAFTAR PUSTAKA

- [1] Ana Yuda, Arie Sulistyarini, Catur D. Setiawan, Gesnita Nugraheni, Gusti N. V. Ahmad, Yunita Nita, "Profil Praktek Pengelolaan Obat pada Lansia di Surabaya," *Jurnal Farmasi Komunitas*, vol. 1, no. 1, pp. 23-28, 2014.
- [2] Menteri Kesehatan Republik Indonesia, *Kebijakan Obat Nasional. Keputusan Menteri Kesehatan Republik Indonesia nomor 189/MENKES/SK/III/2006*, Menteri Kesehatan Republik Indonesia, 2006.
- [3] Tim Penyusun IONI, 2014, "Buku Informatorium Obat Nasional Indonesia (IONI)," Badan Pengawas Obat dan Makanan, 2014. [Online]. Available: <http://pionas.pom.go.id/ioni>. [Diakses 12 Desember 2017].
- [4] M. Rifqi Rokhman, "Singkatan Latin dalam Resep (untuk Apoteker)," 9 Maret 2014. [Online]. Available: <http://m-rifqi-rokhman.staff.ugm.ac.id/2014/03/09/singkatan-latin-dalam-resep-untuk-apoteker/>. [Diakses 12 Desember 2017].
- [5] Yayasan Spiritia, "Info Dasar HIV," [Online]. Available: <http://spiritia.or.id/cst/bacacst.php?artno=1038>. [Diakses 12 Desember 2017].
- [6] Lika Aprilia Samiadi, "Demeclocycline obat apa?," HelloSEHAT, 14 Desember 2016. [Online]. Available: <https://hellosehat.com/obat/demeclocycline/>. [Diakses 12 Desember 2017].
- [7] Lika Aprilia Samiadi, "Efavirenz Obat Apa?," HelloSehat, 14 Desember 2016. [Online]. Available: <https://hellosehat.com/obat/efavirenz/>. [Diakses 12 Desember 2017].

- [8] marketing.co.id, "Teknologi dalam Call Center," 9 Juni 2010. [Online]. Available: <https://marketing.co.id/teknologi-dalam-call-center/>. [Diakses 13 Desember 2017].
- [9] Rian Feldinanto, "Cara Kerja Teknologi IVR Pada Phone Banking," 3 Desember 2011. [Online]. Available: <https://rianfeld.wordpress.com/2011/12/03/cara-kerja-teknologi-ivr-pada-phone-banking/>. [Diakses 13 Desember 2017].
- [10] Eric Bidelman, "Web apps that talk - Introduction to the Speech Synthesis API," Google, Januari 2014. [Online]. Available: <https://developers.google.com/web/updates/2014/01/Web-apps-that-talk-Introduction-to-the-Speech-Synthesis-api>. [Diakses 13 Desember 2017].
- [11] Hans Wennborg, Glen Shires, "Web Speech API Specification," W3C, 6 Juni 2014. [Online]. Available: <https://w3c.github.io/speech-api/webspeechapi.html>. [Diakses 13 Desember 2017].
- [12] Mozilla Developer Network, "Web Speech API," Mozilla, 16 Maret 2017. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. [Diakses 13 Desember 2017].
- [13] OneSignal, "OneSignal Documentation," OneSignal, [Online]. Available: <https://documentation.onesignal.com/docs>. [Diakses 13 Desember 2017].
- [14] Ahmed Fakhr Eldin, "Push Notification PHP via Onesignal," 28 Mei 2017. [Online]. Available: http://webeasystep.com/blog/view_article/Push_Notification_PHP_via_Onesignal. [Diakses 13 Desember 2017].
- [15] Carlos Delgado, "Artyom.js HTML5 Voice Control," 2016. [Online]. Available:

- <https://sdkcarlos.github.io/sites/artyom.html>. [Diakses 13 Desember 2017].
- [16] Jake Archibald, "Service workers explained," World Wide Web Consortium, 3 November 2017. [Online]. Available: <https://github.com/w3c/ServiceWorker/blob/master/explain.md>. [Diakses 13 Januari 2018].
- [17] Suthat Ronglong, Chaiyawut Sookplang, Chonlameth Arpnikanondt, Vajirasak Vanijja, "Design of a Medication Reminder and Feedback System for Thai Elders," *2012 International Conference on Computer & Information Science (ICCIS)*, pp. 43-49, 2012.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Fariz Aulia Pradipta, lahir di Malang pada tanggal 11 November 1994. Penulis menempuh pendidikan sekolah dasar (SD) di MIN Malang 1. Melanjutkan pendidikan formal sekolah menengah pertama (SMP) di SMPN 8 Malang. Menuntaskan pendidikan sekolah menengah atas (SMA) di SMAN 1 Malang. Selanjutnya penulis melanjutkan pendidikan jenjang sarjana (S1) di Jurusan Teknik Informatika, Fakultas Teknologi Informasi dan Komputer (FTIK), Institut Teknologi Sepuluh Nopember Surabaya.

Sebagai mahasiswa, penulis berperan aktif dalam beberapa organisasi kampus seperti staf wahana event ITS EXPO 2013, staf IECC partner BSO IECC BEM ITS 2012/2013, wakil koordinator NST Schematics 2013, staf media dan informasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) periode 2013-2014, staf ahli media dan informasi HMTC periode 2014-2015. Penulis dalam menyelesaikan pendidikan S1 mengambil bidang minat Manajemen Informasi (MI) dan memiliki ketertarikan di bidang pengembangan web. Penulis dapat dihubungi melalui surel farizauliapradipta@gmail.com