

**PENENTUAN KEDISIPLINAN MINUM OBAT DENGAN METODE
FUZZY C-MEANS CLUSTERING UNTUK PASIEN PENDERITA KUSTA**

TUGAS AKHIR



NIKEN SULISTYORINI

NIM : 311710016

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2022**

LEMBAR PENGESAHAN

TUGAS AKHIR

**PENENTUAN KEDISIPLINAN MINUM OBAT DENGAN METODE
FUZZY C-MEANS CLUSTERING UNTUK PASIEN PENDERITA KUSTA**

Oleh:

NIKEN SULISTYORINI

NIM. 311710016

Dari:

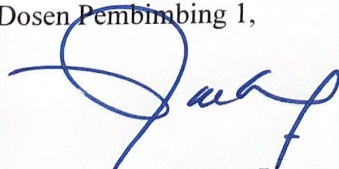
PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS dan TEKNOLOGI

UNIVERSITAS MA CHUNG

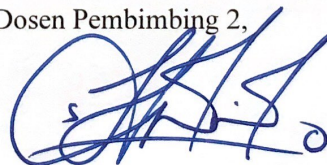
Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan
dan berhak mendapatkan gelar Sarjana Komputer

Dosen Pembimbing 1,



Paulus Lucky Tirma Irawan, S.Kom., MT.
NIP. 20100005

Dosen Pembimbing 2,



Mochamad Subianto, S.Kom., M.Cs.
NIP. 20100002

Dekan Fakultas Sains dan Teknologi,



Dr. Kestrelia Rega Prilianti, M.Si.
NIP. 20120035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi Tugas Akhir ini dengan judul “Penentuan Kedisiplinan Minum Obat Dengan Metode Fuzzy C-Means Clustering Untuk Pasien Penderita Kusta” adalah benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 28 Oktober 2022



Niken Sulistyorini

NIM. 311710016

PENENTUAN KEDISIPLINAN MINUM OBAT DENGAN METODE FUZZY C-MEANS CLUSTERING UNTUK PASIEN PENDERITA KUSTA

Niken Sulistyorini, Paulus Lucky Tirma Irawan, Mochamad Subianto

Universitas Ma Chung

ABSTRAK

Penyakit kusta merupakan salah satu penyakit menular yang disebabkan oleh infeksi *Mycobacterium Leprae*. Saat ini cara yang masih digunakan untuk menangani pasien penderita kusta adalah dengan pemberian obat secara teratur atau yang biasa disebut dengan pengobatan *Multi Drug Therapy* (MTD). Pengobatan MTD mengharuskan setiap pasien kusta dapat meminum obat secara teratur dengan pengawasan dokter maupun PMO. Terdapat pengaruh mengenai kedisiplinan minum obat dengan tingkat kesembuhan atau kesehatan pasien penderita kusta.

Pada penelitian ini PMO akan dibantu dengan kemudahan dalam memonitoring pasien dan pengelompokan pasien. Pasien akan dikelompokkan ke dalam kelompok disiplin dan tidak disiplin menggunakan metode *Fuzzy C-Means Clustering*. Data yang digunakan untuk klasifikasi adalah data konfirmasi minum obat dari setiap pasien yang telah dijadwalkan, dengan variabel tepat waktu, tidak tepat waktu, dan tidak respon. Selain itu PMO juga dapat memonitoring setiap pasien

Uji coba aplikasi monitoring menunjukkan metode *blackbox testing* menunjukkan bahwa seluruh pengujian berhasil sesuai dengan skenario yang telah dibuat. Lalu hasil yang diperoleh dari klasifikasi pasien adalah akurasi sebesar 75%, di mana terdapat 1 data pasien yang tidak berhasil di klasifikasi dengan benar sedangkan 3 data pasien lainnya dapat diklasifikasikan dengan benar.

Kata kunci : Kusta, Fuzzy C-Means Clustering, Monitoring, Kedisiplinan, Minum Obat

DETERMINATION OF DRUGS DRINKING DISCIPLINE USING THE FUZZY C-MEANS CLUSTERING METHOD FOR PATIENTS WITH LEPROSY

Niken Sulistyorini, Paulus Lucky Tirma Irawan, Mochamad Subianto
Universitas Ma Chung

ABSTRACT

Leprosy is an infectious disease caused by infection with *Mycobacterium Leprae*. Currently, the method that is still used to treat patients with leprosy is by administering drugs on a regular basis or commonly referred to as Multi Drug Therapy (MTD) treatment. MTD treatment requires that every leprosy patient can take medication regularly under the supervision of a doctor or PMO. There is an influence on the discipline of taking medicine with the rate of healing or the health of leprosy patients.

In this study, PMO will be assisted with ease in monitoring patients and grouping patients. Patients will be grouped into disciplined and undisciplined groups using the Fuzzy C-Means Clustering method. The data used for classification is data on confirmation of taking medication from each patient who has been scheduled, with the variables being on time, not on time, and not responding. In addition, PMO can also monitor every patient

The monitoring application trial shows the black box testing method showing that all tests were successful in accordance with the scenario that had been created. Then the results obtained from the patient classification is an accuracy of 75%, where there is 1 patient data that has not been classified correctly while the other 3 patient data can be classified correctly.

Keywords: Leprosy, Fuzzy C-Means Clustering, Monitoring, Discipline, Taking Medicine

KATA PENGANTAR

Segala puja dan puji syukur atas kehadiran Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat diselesaikan. Laporan ini berisi hasil dari pembuatan sebuah website yang mana memiliki tujuan agar dapat digunakan untuk memantau pasien kusta dalam meminum obat dan juga dapat digunakan untuk mengklasifikasikan setiap pasien kedalam kelompok disiplin dan tidak dalam meminum obat. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada pihak yang telah membantu dalam proses pembuatan Tugas Akhir, diantaranya:

1. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT. selaku dosen pembimbing,
2. Bapak Mochamad Subianto, S.Kom., M.Cs. selaku dosen pembimbing,
3. Ibu Dr.Kestrlia Rega Prilianti, M.Si selaku Dekan dari Fakultas Sains dan Teknologi Universitas Ma Chung,
4. Bapak Hendry Setiawan, ST, M.Kom, selaku Kepala Program Studi Teknik Informatika,
5. Kedua orang tua, yang telah memberikan dukungan dan semangat selama proses pengerjaan Tugas Akhir,
6. Teman-teman yang telah membantu dan mendukung selama proses pengerjaan Tugas Akhir.

Laporan ini disusun berdasarkan hasil Tugas Akhir yang berjudul “Penentuan Kedisiplinan Minum Obat Dengan Metode Fuzzy C-Means Clustering Untuk Pasien Penderita Kusta” sebagai salah satu prasyarat kelulusan.

Malang, 26 Oktober 2022

Niken Sulistyorini

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	5
1.3 Rumusan Masalah	5
1.4 Batasan Masalah.....	5
1.5 Tujuan Penelitian.....	6
1.6 Manfaat Penelitian.....	6
1.7 Luaran Tugas Akhir.....	7
1.8 Sistematika Penulisan.....	7
BAB II DASAR TEORI DAN TINJAUAN PUSTAKA.....	9
2.1 Kusta.....	9
2.2 HTML (Hypertext Markup Language).....	11
2.3 Laravel.....	15
2.4 MySQL.....	18
2.5 Plugin OneSignal.....	21
2.6 API (Application Programming Interface).....	25
2.7 Fuzzy Logic	27
2.8 <i>Clustering</i>	28
2.9 Fuzzy C-Means (FCM)	29
2.10 Blackbox Testing	31
2.11 Penelitian Terdahulu	32
BAB III METODE PENELITIAN.....	37
3.1 Alur Penelitian.....	37
3.2 Analisa Kebutuhan	37
3.3 Perancangan Sistem.....	38
3.4 Pengujian Aplikasi	54

BAB IV HASIL DAN PEMBAHASAN	57
4.1 Lingkungan Pengujian.....	57
4.2 Skenario Pengujian.....	57
4.3 Skenario dan Hasil Uji Coba	57
4.4 Evaluasi Pengujian	87
4.5 Diskusi Hasil	90
BAB V KESIMPULAN DAN SARAN.....	92
5.1 Kesimpulan.....	92
5.2 Saran	92
DAFTAR PUSTAKA	93
LAMPIRAN A	94
LAMPIRAN B	95
LAMPIRAN C	96
LAMPIRAN D	97

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi Penyakit Kusta.....	9
Gambar 2. 2 Memulai membuat aplikasi dengan <i>OneSignal</i>	21
Gambar 2. 3 Halaman pengisian nama aplikasi dan pemilihan <i>platform</i>	22
Gambar 2. 4 Halaman Web Configuration	22
Gambar 2. 5 Halaman Site Setup	23
Gambar 2. 6 Halaman Permission Prompt Setup.....	24
Gambar 2. 7 Halaman Add Code to Site.....	25
Gambar 2. 8 Ilustrasi cara kerja API.....	26
Gambar 3. 1 Alur Penelitian.....	37
Gambar 3. 2 Usecase Aplikasi Notifikasi Minum Obat Penderita Kusta	38
Gambar 3. 3 Activity Diagram Aplikasi	39
Gambar 3. 4 Sequence Diagram	40
Gambar 3. 5 Class Diagram	41
Gambar 3. 6 Flowchart Metode FCM.....	48
Gambar 3. 7 Mockup Halaman Login.....	50
Gambar 3. 8 Mockup Dashboard PMO.....	51
Gambar 3. 9 Mockup Melihat Data	51
Gambar 3. 10 Mockup Menu Mengolah Data Kedisiplinan	52
Gambar 3. 11 Mockup Dashboard Pasien.....	53
Gambar 3. 12 Mockup Menu Jadwal Minum Obat.....	53
Gambar 3. 13 Mockup Notifikasi	54
Gambar 4. 1 Pengujian T001 Skenario 1	59
Gambar 4. 2 Pengujian T001 Skenario 2	60
Gambar 4. 3 Pengujian T001 Skenario 3	61
Gambar 4. 4 Pengujian T002 Skenario 1	63
Gambar 4. 5 Pengujian T002 Skenario 2	64
Gambar 4. 6 Pengujian T002 Skenario 3	65
Gambar 4. 7 Pengujian T002 Skenario 4	66
Gambar 4. 8 Pengujian T003 Skenario 1	68
Gambar 4. 9 Pengujian T003 Skenario 2	69

Gambar 4. 10 Pengujian T003 Skenario 3	70
Gambar 4. 11 Pengujian T003 Skenario 4	71
Gambar 4. 12 Pengujian T004 Skenario 1	73
Gambar 4. 13 Pengujian T004 Skenario 2	74
Gambar 4. 14 Pengujian T004 Skenario 3	75
Gambar 4. 15 Pengujian T004 Skenario 4	76
Gambar 4. 16 Pengujian T005 Skenario 1	78
Gambar 4. 17 Pengujian T005 Skenario 2	79
Gambar 4. 18 Pengujian T005 Skenario 3	80
Gambar 4. 19 Pengujian T005 Skenario 4	81
Gambar 4. 20 Pengujian T006 Skenario 1	83
Gambar 4. 21 Pengujian T007 Skenario 1	84
Gambar 4. 22 Pengujian T008 Skenario 1	85
Gambar 4. 23 Pengujian T009 Skenario 1	87
Gambar 4. 24 Plot Cluster Data Latih.....	90

DAFTAR TABEL

Tabel 2. 1 Contoh Tag HTML	12
Tabel 2. 2 Jenis Atribut Khusus	13
Tabel 2. 3 Query DDL	20
Tabel 2. 4 Query DML.....	20
Tabel 2. 5 Query DCL.....	21
Tabel 2. 6 Detail field pada form site setup	23
Tabel 3. 1 Endpoint user	42
Tabel 3. 2 Endpoin pasien.....	43
Tabel 3. 3 Endpoint pmo	43
Tabel 3. 4 Endpoint obat	44
Tabel 3. 5 Endpoint jadwal	45
Tabel 3. 6 Endpoint konfirmasi.....	46
Tabel 3. 7 Data Latih.....	49
Tabel 3. 8 Kasus yang diujikan	55
Tabel 4. 1 Lingkungan Pengujian	57
Tabel 4. 2 Pengujian Fitur Login	58
Tabel 4. 3 Pengujian Fitur Mengolah Data PMO	62
Tabel 4. 4 Pengujian Fitur Mengolah Data Pasien.....	67
Tabel 4. 5 Pengujian Fitur Mengolah Data Obat	72
Tabel 4. 6 Pengujian Fitur Mengolah Data Jadwal	77
Tabel 4. 7 Pengujian Mengolah Data Kedisiplinan	82
Tabel 4. 8 Pengujian Menampilkan Data Jadwal.....	83
Tabel 4. 9 Pengujian Menampilkan Notifikasi	85
Tabel 4. 10 Pengujian Konfirmasi Notifikasi	86
Tabel 4. 11 Rangkuman Pengujian Fungsionalitas Fitur	87
Tabel 4. 12 Kelebihan dan Kekurangan Aplikasi	88
Tabel 4. 13 Hasil Uji FCM.....	89

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan salah satu negara tropis, penyakit yang banyak menyerang negara tropis sangatlah beragam. Salah satu yang ada di Indonesia adalah penyakit kusta. Penyakit kusta ini merupakan penyakit yang banyak diderita oleh masyarakat Indonesia. Menurut Direktur Pencegahan dan Pengendalian Penyakit Menular Langsung (P2PML) Kementerian Kesehatan (Kemenkes) Republik Indonesia yaitu dr. Wiendra Wawarontu, bahwa penyakit kusta masih menyebar di sembilan provinsi yang ada di Indonesia. Sembilan provinsi tersebut yakni Provinsi Sulawesi Tenggara, Sulawesi Barat, Sulawesi Selatan, Sulawesi Utara, Gorontalo, Maluku, Maluku Utara, Papua, dan Papua Barat, bahkan saat ini pun di Jawa Timur juga sudah mulai menyebar penyakit kusta tersebut. Kusta adalah salah satu penyakit menular yang disebabkan oleh infeksi *Mycobacterium leprae*. Bakteri tersebut menyerang kulit, syaraf tepi, dan jaringan tubuh lainnya kecuali susunan syaraf pusat. Gejala awal penyakit ini adalah adanya bercak putih atau kemerahan pada kulit yang mati rasa. Kusta dibagi menjadi dua, yakni kusta pausibasilar (PB) atau kusta tipe kering dan kusta multibasilar (MB) atau kusta tipe basah.

Saat ini yang menjadi perhatian dalam penanganan pasien kusta adalah tingkat keteraturan konsumsi obat oleh para pasien. Data keteraturan minum obat tersebut sangat berguna bagi Pengawas Minum Obat sebagai masukan terhadap penanganan pasien tersebut, sehingga akan lebih meminimalisir kesalahan penanganan terhadap setiap pasien. Namun, Pengawas Minum Obat mengalami kesulitan dalam mendapatkan data keteraturan tersebut karena minimnya pencatatan data keteraturan dalam minum obat setiap pasien, selain itu juga minimnya waktu dalam mengelompokkan pasien berdasarkan kedisiplinan minum obat melalui data tersebut membuat PMO menjadi membutuhkan waktu yang lebih lama untuk mengurus data kedisiplinan pasien. Kesulitan tersebut berakibat pada beberapa faktor yang dikhawatirkan yaitu seperti obat yang diberikan apakah sudah sesuai dengan kondisi pasien, atau bisa saja obat yang diberikan sudah sesuai hanya

saja pasien tidak teratur dalam mengonsumsi obat tersebut, dan juga penanganan yang diberikan selanjutnya tidak sesuai karena adanya faktor-faktor sebelumnya. Jadi, data keteraturan dan kedisiplinan tersebut sebenarnya benar-benar dibutuhkan dalam hal ini.

Kedisiplinan dalam meminum obat merupakan salah satu faktor penting untuk meningkatkan kesembuhan setiap pasien. Terdapat jurnal terdahulu yang membahas tentang kedisiplinan pasien dalam meminum obat. Seperti jurnal yang berjudul “Determinan Keberhasilan Pengobatan Multi Drug Therapy Pada Penderita Kusta Tipe Multibaciler” tahun 2018 oleh Muh. Zul Azhri Rustam. Pada penelitian tersebut dijelaskan bahwa terdapat pengaruh kedisiplinan setiap pasien yang meminum obat dengan adanya pengawasan dari pihak petugas kesehatan. Penelitian yang hampir sama dengan sebelumnya mengenai kepatuhan minum obat adalah penelitian yang berjudul “Hubungan Kepatuhan Minum Obat Antihipertensi Terhadap Tercapainya Target Terapi Pasien Hipertensi di Puskesmas Wirobrajan Yogyakarta”. Penelitian tersebut menjelaskan bahwa pasien yang memiliki tingkat kepatuhan yang tinggi merupakan pasien dengan penyakit penyerta DM, dengan adanya penyakit penyerta dan beberapa pengaruh lain membuat pasien termotivasi untuk patuh dalam meminum obat. Sebenarnya kepatuhan minum obat tidak hanya untuk para penderita kusta saja namun semua pasien yang diawasi oleh PMO juga harus patuh dalam meminum obat. Karena dengan patuh minum obat PMO dapat menganalisis segala sesuatu yang terjadi pada pasien tersebut dan penanganan yang diberikan akan lebih spesifik dan tepat.

Pada era modernisasi saat ini sangatlah mudah mengelompokkan suatu data menggunakan metode kecerdasan buatan. Terdapat banyak jenis kecerdasan buatan yang digunakan untuk mengelompokkan berbagai data sesuai dengan kebutuhan. Salah satu teknik yang digunakan untuk mengklasterkan data adalah Fuzzy C-Means Clustering. Fuzzy C-Means(FCM) adalah suatu teknik pengclusteran data yang mana keberadaan tiap-tiap titik data dalam suatu *cluster* (Kusumadewi & Purnomo, 2010). Jadi teknik ini merupakan teknik untuk meng-clusterkan data sesuai dengan kelompok data yang memiliki tingkat kesamaan yang tinggi. Proses peng-clusteran dengan teknik ini diawali dengan menentukan pusat cluster terlebih dahulu, kemudian dilanjutkan dengan menghitung derajat keanggotaan tiap

datanya. Nilai derajat keanggotaan akan digunakan untuk menentukan data tersebut masuk ke dalam *cluster* yang mana .

Terdapat penelitian yang menggunakan metode FCM untuk menentukan suatu kedisiplinan. Penelitian tersebut berjudul “Perbandingan Algoritma K-Means Dengan Algoritma Fuzzy C-Means Untuk Clustering Tingkat Kedisiplinan Kinerja Karyawan” pada tahun 2018 oleh Nova Agustina dan Prihandoko. Kemudian terdapat penelitian serupa yaitu berjudul “Perbandingan Metode K-Means dan Fuzzy C-Means Untuk Pengelompokan Pegawai Berdasarkan Nilai Kinerja dan Tingkat Kedisiplinan Pegawai” pada tahun 2018 oleh Wikarno dkk. Kedua penelitian tersebut berhasil menggunakan kedua metode tersebut, dan dijelaskan dalam kedua jurnal tersebut bahwa metode yang lebih baik adalah metode K-Means. Penelitian selanjutnya yang hampir sama yaitu berjudul “Klasterisasi kinerja karyawan menggunakan algoritma fuzzy c-means”. Penelitian yang dilakukan oleh Martin dan Yessica Nataliani bertujuan untuk mendapatkan tiga kelompok karyawan. Telah dibuktikan pada penelitian ini bahwa FCM dapat digunakan untuk mendapatkan kelompok karyawan berdasarkan kinerjanya. Adanya beberapa penelitian tersebut menjadi acuan untuk peneliti menggunakan metode fuzzy logic sebagai metode pengelompokan kedisiplinan para pasien penderita kusta dalam meminimum obat. Pengelompokan tersebut nantinya dapat digunakan oleh PMO dalam mengamati setiap efektivitas obat yang telah diberikan kepada pasien.

Kemajuan teknologi di bidang medis sudah terbilang sangat maju, dengan keterbatasan yang dialami dapat diselesaikan dengan adanya teknologi terkini. Terdapat beberapa jurnal yang memanfaatkan teknologi masa kini untuk dunia permedisan yang beberapa penelitian tersebut yaitu “Rancang Bangun Interactive Voice Response (IVR) Sebagai Peningkat Medikasi Pasien Lansia Berbasis Web” pada tahun 2018 oleh Fariz Aulia Pradipta. Selanjutnya yaitu berjudul “Rancang Bangun Smart Medicine Box Sebagai Peningkat Jadwal Minum Obat Berbasis Internet Of Things” oleh Alisya dkk. Kemudian terdapat sebuah jurnal yang berjudul “Desain Prototipe Sistem Monitoring Minum Obat Bagi ODHA” jurnal ini terbit pada tahun 2021 oleh Syawaludin, dkk. Jurnal selanjutnya adalah berjudul “Aplikasi Monitoring Untuk Pasien TBC Dewasa Berbasis Android” oleh

Muhammad Idris Musa. Kemudian jurnal yang memiliki tema yang sama yaitu “Rancang Bangun Aplikasi Lansia Care Untuk Monitoring Lansia Yang Tinggal Sendiri Berbasis Android” oleh Annisa Arum Mumtazah. Semua penelitian tersebut memiliki tema yang sama yaitu untuk mengawasi setiap pasien yang sedang dirawat. Dari semua penelitian yang dibahas, dapat disimpulkan bahwa dengan adanya teknologi sangat membantu untuk mengawasi atau mengawasi secara *real time* pasien yang sedang ditangani dengan adanya aplikasi yang sudah di buat.

Terdapat satu penelitian yang berjudul “Medicine Consumption Reminder and Monitoring Application for Patients with Leprosy Disease” disusun oleh Irawan, et al., 2022. Penelitian tersebut berhasil membuat aplikasi berbasis mobile yang memfasilitasi penderita kusta untuk mendapatkan notifikasi jadwal minum obat yang telah diberikan oleh PMO. Namun, pada penelitian tersebut aplikasi *mobile android* yang dibuat PMO masih kesulitan dalam mengawasi pasien. Karena pada penelitian tersebut aplikasi yang di buat berfokus pada android mobile sehingga mengalami keterbatasan dalam akses berbagai device. Selain itu pada penelitian ini aplikasi yang dibuat masih belum bisa mengklasterisasi karakteristik setiap pasien terhadap kedisiplinan minum obat.

Oleh sebab itu, dengan kemajuan teknologi maupun medis yang saat ini sudah berkembang dengan pesat, penulis ingin memenuhi kebutuhan tenaga medis untuk membantu dalam mempermudah merawat pasien penderita kusta. Khususnya dalam hal pencatatan atau pendataan keteraturan pasien dalam meminum obat yang sudah diberikan. Penulis berencana untuk membuat suatu website yang akan memudahkan tenaga medis maupun pasien penderita kusta dalam mengingatkan jadwal meminum obat yang mana obat tersebut harus di minum secara teratur. Kemudian website tersebut akan mencatat seluruh data keteraturan pasien dalam meminum obat, sehingga tenaga medis atau PMO dapat mengamati setiap aktivitas pasien dalam mengkonsumsi obat-obatan yang sudah diresepkan dan di jadwalkan oleh tenaga medis yang merawatnya. Website tersebut akan dibangun dengan menggunakan One Signal Push Notification dan untuk DBMS yang digunakan adalah MySQL. Aplikasi web ini nantinya akan mengirimkan notifikasi kepada pasien penderita kusta pada jam berapa dan obat apa saja yang perlu diminum saat itu. Kemudian website akan otomatis mencatat konfirmasi yang dilakukan oleh

pasien apakah pada jam tersebut sudah mengkonsumsi obat atau belum. Hasil pencatatan konfirmasi minum obat akan dikelompokkan sesuai tingkat kedisiplinannya menggunakan metode fuzzy logic yaitu Fuzzy C-Means. Teknik FCM ini digunakan karena banyak penelitian mengenai penentuan kluster kedisiplinan yang menggunakan teknik ini. Data kedisiplinan setiap pasien ini nantinya akan menjadi data untuk pemantauan ke efektifitas dari obat yang sudah diberikan oleh tenaga medis yang merawat pasien tersebut. Adanya website ini diharapkan dapat membantu dalam perawatan para pasien agar penyakit kusta ini dapat di tanggulangi dan ditangani dengan tepat sehingga penularannya akan berkurang.

1.2 Identifikasi Masalah

Identifikasi masalah yang akan didapat dari penelitian ini yaitu kesulitan tenaga medis dalam memantau setiap pasien untuk dapat selalu minum obat dengan teratur dan terjadwalkan. Selanjutnya tenaga medis juga mengalami kesulitan dalam pencatatan kedisiplinan setiap pasien dalam minum obat, karena nantinya data tersebut akan digunakan sebagai acuan tenaga medis untuk mengetahui keefektifitasan setiap obat yang diberikan kepada pasien dan meminimalisir salah tindakan akibat beberapa faktor yang menjadi perhatian dalam hal perawatan pasien.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, rumusan masalah yang digunakan pada penelitian ini adalah bagaimana merancang sebuah website yang dapat digunakan untuk mempermudah tenaga medis dalam mengingatkan pasien penderita kusta untuk minum obat tepat pada waktunya. Kemudian website ini nantinya juga dirancang untuk dapat mencatat data keteraturan setiap pasien dalam mengkonsumsi obat sesuai dengan jam dan obat yang telah diresepkan oleh tenaga medis yang menangani pasien tersebut.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Pengerjaan *web* dilakukan dengan menggunakan *framework* Laravel.
2. Menggunakan *plugin* OneSignal sebagai pengingat atau *notification* secara *real time*.
3. *Database Management System* (DBMS) yang digunakan adalah MySQL.
4. Aplikasi ini dirancang spesifik untuk penanganan pasien penderita kusta.
5. Variabel yang digunakan untuk peng-*cluster*-an adalah tepat waktu, tidak tepat waktu, dan tidak respon.

1.5 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah meng-*cluster*-kan pasien kedalam *cluster* disiplin dan tidak disiplin. Jika tenaga medis dapat melihat pasien mana yang disiplin dan tidak maka tenaga medis dapat memantau perkembangan setiap pasien penderita kusta dengan mengingatkan pasien penderita kusta untuk tetap selalu meminum obat secara teratur agar kusta yang diderita tidak semakin memburuk. Hal yang paling penting adalah dapat melakukan pencatatan tingkat keteraturan pasien dalam meminum obat yang telah diresepkan dan dijadwalkan, sehingga pasien akan mendapatkan perawatan yang tepat dari data tersebut. Jika pasien penderita kusta mendapatkan pengawasan yang baik, maka tingkat kesembuhan akan semakin membaik, dan penyebaran kusta tidak akan semakin bertambah parah.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Mempermudah tenaga medis dalam memantau dan mengingatkan pasien untuk selalu minum obat tepat pada waktunya.
2. Membantu dalam hal pencatatan data keteraturan konsumsi obat pasien penderita kusta, guna untuk pemantauan ke efektifitasan obat yang diberikan.
3. Membantu meningkatkan kedisiplinan minum obat untuk kesembuhan pasien penderita kusta.

1.7 Luaran Tugas Akhir

Luaran dari penelitian ini adalah sebuah website untuk membantu tenaga medis dalam penanganan pasien penderita kusta, website ini khususnya untuk pengingat minum obat bagi pasien penderita kusta dan untuk pencatatan data keteraturan pasien dalam meminum obat yang sudah diresepkan. Nantinya penelitian ini akan dilanjutkan dengan publikasi ilmiah ke dalam jurnal.

1.8 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut :

Bab I: Pendahuluan

Pada bab 1 ini penulis akan menjelaskan tentang latar belakang dari penelitian ini dibuat , rumusan masalah , batasan masalah , tujuan penelitian hingga manfaat dari penelitian yang akan dilaksanakan.

Bab II : Tinjauan Pustaka

Pada bab 2 akan dijelaskan teori-teori pendukung dari penelitian ini. Seperti penjelasan dasar mengenai penyakit kusta, framework Laravel, *Database Management System* yaitu MySQL, metode kecerdasan buatan Fuzzy Logic, kemudian *plugin* OneSignal yang digunakan untuk membuat notifikasi pengingat minum obat secara *real time*.

Bab III: Analisis dan Perancangan Sistem

Pada bab ini akan dijelaskan mengenai analisis perancangan awal sistem yang akan dibuat pada penelitian ini. Seperti perancangan usecase, desain tabel, API yang akan dibuat, user interface sisi admin, dan juga rancangan kecerdasan buatan fuzzy logic.

Bab IV: Hasil dan Pembahasan

Pada bab IV berisi mengenai pengujian yang telah di lakukan pada aplikasi yang sudah dibuat. Pengujian tersebut dilakukan agar dapat melihat apakah seluruh fitur yang dibuat sudah sesuai dengan apa yang telah dirancang dan diharapkan.

Bab V: Kesimpulan dan Saran

Bab terakhir ini menjelaskan tentang kesimpulan dari hasil yang telah diperoleh dari penelitian ini. Mulai dari pengujian, kebaharuan fitur yang merujuk dari jurnal sebelumnya, dan juga kesulitan yang didapat saat pengerjaan penelitian. Kemudian juga mencantumkan saran-saran yang mendukung untuk perbaikan penelitian agar menjadi lebih baik.

BAB II

DASAR TEORI DAN TINJAUAN PUSTAKA

2.1 Kusta

Kusta atau bisa juga disebut dengan lepra merupakan penyakit yang disebabkan oleh bakteri *Mycobacterium leprae*. Penyakit ini dikategorikan sebagai salah satu penyakit menular yang sampai saat ini masih ada di Indonesia. Kusta dapat menular melalui percikan dari saluran pernapasan seperti ludah atau dahak yang keluar saat batuk maupun bersin. Seseorang yang terus menerus terkena percikan tersebut akan beresiko tertular penyakit ini. Selain itu ada beberapa faktor yang dapat meningkatkan resiko tertular penyakit kusta yaitu bersentuhan dengan hewan pembawa penyakit kusta, menetap maupun berkunjung ke kawasan endemik kusta, dan memiliki gangguan sistem kekebalan tubuh. Namun, menurut Direktur Pencegahan dan Pengendalian Penyakit Menular Langsung Kemenkes, dr. Wiendra Waworuntu, M.Kes, bahwa penyakit kusta meskipun tergolong penyakit menular tetapi kusta bukanlah penyakit yang mematikan. Penderita kusta bisa dinyatakan sembuh jika menjalani pengobatan yang tepat.



Gambar 2. 1 Ilustrasi Penyakit Kusta

Penyakit kusta memiliki berbagai gejala yang awalnya tidak tampak dengan jelas, hingga pada beberapa kasus yang ada gejala kusta baru terlihat setelah 20-30 tahun. Berikut ini merupakan beberapa gejala kusta yang dirasakan oleh penderitanya:

1. Mati rasa di kulit, termasuk kehilangan kemampuan merasakan suhu, sentuhan, tekanan, atau rasa sakit

2. Muncul lesi pucat, berwarna lebih terang, dan menebal di kulit
3. Kulit tidak berkeringat (*anhidrosis*)
4. Muncul luka tapi tidak terasa sakit
5. Pembesaran saraf yang biasanya terjadi di siku dan lutut
6. Otot melemah, terutama otot kaki dan tangan
7. Kehilangan alis dan bulu mata
8. Mata menjadi kering dan jarang mengedip
9. Mimisan, hidung tersumbat, atau kehilangan tulang hidung

Berdasarkan dari tingkat keparahan dari gejala yang dialami, kusta dikelompokkan menjadi 6 jenis, yaitu:

1. *Intermediate leprosy*, ditandai dengan beberapa lesi datar berwarna pucat atau lebih cerah dari warna kulit sekitarnya yang kadang sembuh dengan sendirinya
2. *Tuberculoid leprosy*, ditandai dengan beberapa lesi datar yang kadang berukuran besar, mati rasa, dan disertai dengan pembesaran saraf
3. *Borderline tuberculoid leprosy*, ditandai dengan munculnya lesi yang berukuran lebih kecil dan lebih banyak dari *tuberculoid leprosy*
4. *Mid-borderline leprosy*, ditandai dengan banyak lesi kemerahan, yang tersebar secara acak dan asimetris, mati rasa, serta pembengkakan kelenjar getah bening setempat
5. *Borderline lepromatous leprosy*, ditandai dengan lesi yang berjumlah banyak bisa berbentuk datar, benjolan, nodul, dan terkadang mati rasa
6. *Lepromatous leprosy*, ditandai dengan lesi yang tersebar dengan simetris, umumnya lesi yang timbul mengandung banyak bakteri, dan disertai dengan rambut rontok, gangguan saraf, serta kelemahan anggota gerak

Penderita kusta yang mengalami gejala-gejala tersebut disarankan untuk segera memeriksakan diri ke dokter agar mendapatkan penanganan yang tepat. Semakin cepat kusta diketahui dan ditangani dengan tepat, maka semakin tinggi peluang untuk dapat sembuh. Saat ini untuk pengobatan kusta di Indonesia dilakukan dengan metode MTD (*Multi Drug Therapy*). MTD ini merupakan metode pemberian obat kepada pasien penderita kusta, di mana pasien tersebut harus meminum obat sesuai dengan arahan dari dokter yang menanganinya.

2.2 HTML (Hypertext Markup Language)

HTML atau *Hypertext Markup Language* adalah bahasa *markup* yang digunakan untuk membuat struktur halaman *website*. HTML terdiri dari kombinasi teks dan symbol yang disimpan dalam sebuah *file*. Pembuatan *website* dengan HTML harus mengikuti standar atau format khusus yang telah tertuang dalam standar kode internasional atau ASCII (*American Standard Code for Information Interchange*). HTML memudahkan untuk pembuatan *website* dengan menyusun heading, paragraph, gambar, link, dan komponen lainnya. Agar dapat diakses oleh umum, pengguna harus membuka lewat aplikasi *browser*, seperti Internet Explorer, Chrome, atau Mozilla Firefox.

Pada tahun 1991, tim Berners-Lee menciptakan HTML sebagai solusi untuk memudahkan para ilmuwan mengakses dokumen satu sama lain. Setelah satu tahun, HTML mulai dapat digunakan untuk umum. Sejarah ini tentunya tak lepas dari peran *World Wide Web* (WWW) dan HTML saling bersinergi agar informasi yang ingin disebarakan bisa diakses oleh banyak orang. Selama ini HTML berkembang sangat pesat mulai tahun 1991 dengan HTML1 hingga saat ini sudah menjadi HTML5 dengan perubahan yang sangat baik di HTML5 pengguna sudah dapat menggabungkan HTML dengan CSS, dan JavaScript. HTML5 menjadi versi terupdate dengan sistem paling canggih, bahkan saat ini sudah 87,4% *website* yang menggunakan HTML5.

HTML memiliki beberapa komponen, di mana komponen tersebut terdiri dari *tag*, elemen, dan atribut. Berikut ini adalah penjelasan singkat mengenai ketiga komponen tersebut.

1. *Tag*

Tag merupakan tanda awalan perintah dari HTML yang nantinya akan dibaca oleh *web browser*. *Tag* dibuat dengan menggunakan kurung siku “<....>” , di dalam *tag* nantinya akan berisi nama tag tersebut. Setiap tag memiliki fungsi dan perintah yang berbeda-beda, seperti membuat judul, paragraph, *heading*, cetak tebal, cetak miring, dan masih banyak lagi. *Tag* ditulis secara berpasangan, jadi *tag* terdiri dari *tag* pembuka dan *tag* penutup. Perbedaan dari *tag* pembuka dan penutup adalah pada *tag* penutup ditambahkan garis miring (/) didepan nama *tag*. Contoh tag HTML dapat dilihat pada tabel 2.1 berikut ini.

Tabel 2. 1 Contoh Tag HTML

Tag	Fungsi Tag
<code><html></code>	Untuk memulai membuat halaman HTML yang mencakup semua konten dan elemen.
<code><head></code>	Untuk membuat halaman HTML yang mencakup tampilan deskripsi di hasil pencarian Google, <i>style</i> konten (CSS), dan lainnya.
<code><title></code>	Untuk membuat judul <i>website</i> .
<code><body></code>	Untuk membuat bagian isi <i>website</i> .
<code><h1></code> sampai <code><h6></code>	Untuk membuat bagian heading pada artikel.
<code><p></code>	Untuk membuat paragraf.
<code><!--></code>	Untuk membuat komentar.

2. Elemen

Elemen merupakan komponen HTML yang berisi seluruh kode dari *tag* pembuka dan penutup. Elemen terdiri dari teks simbol yang berupa tag pembuka, isi atau konten, dan *tag* penutup. Berikut adalah contoh dari elemen dapat dilihat pada kode 2.1 dibawah ini.

```
<title>Judul</title>
```

Kode 2. 1 Contoh Kode Elemen

```
<html>
  <head>
    <title>Judul</title>
  </head>
  <body>
    <p>Tag Paragraf</p>
  <body>
</html>
```

Kode 2. 2 Contoh Kode Nested Elements

Kode 2.2 merupakan contoh kode *nested elements*. Nested Elements merupakan elemen yang berisi elemen lainnya, jadi di dalam satu elemen bisa berisi banyak *tag*.

3. Atribut

Atribut merupakan suatu informasi atau perintah tambahan yang berada pada sebuah elemen. Atribut memiliki fungsi sebagai penjelas perintah *tag* pada elemen. Berikut ini merupakan contoh dari atribut, dapat dilihat pada kode 2.3 dibawah ini

```

```

Kode 2. 3 Contoh Atribut

Setiap *tag* dapat memiliki lebih dari satu atribut, seperti pada gambar di atas *tag* `` memiliki atribut khusus yaitu (*scr*) dan (*alt*) yang berarti browser diperintahkan untuk menampilkan gambar.jpg dengan *alt text* “ikan kakap merah”. Namun, tidak semua atribut dapat digunakan dalam sebuah *tag*. Berikut ini jenis atribut khusus beberapa *tag*.

Tabel 2. 2 Jenis Atribut Khusus

Atribut	Dapat digunakan pada <i>Tag</i>
src	<code></code> , <code><embed></code> , <code><audio></code> , <code><iframe></code>
href	<code><a></code> , <code><link></code>
action	<code><form></code>
autoplay	<code><audio></code> , <code><video></code>

Selain atribut khusus tersebut, ada beberapa atribut yang disebut sebagai *atribut event*. Atribut ini nantinya akan diimplementasikan dengan pemrograman *javascript*. Atribut tersebut seperti (*onload*) saat *loading* selesai, (*onoffline*) saat tiba-tiba *offline*, (*onresize*) ukuran jendela.

2.1 PHP (*PHP Hypertext Preprocessor*)

PHP merupakan salah satu bahasa pemrograman yang sangat dibutuhkan dalam pembuatan *website*, karena PHP dapat membuat suatu *website* menjadi

dinamis. PHP adalah bahasa pemrograman *server side scripting* yang bersifat *open source*, bahasa pemrograman ini diciptakan oleh Rasmus Lerdorf pada tahun 1995. Karena PHP merupakan *scripting language*, maka PHP akan menjalankan instruksi pemrograman saat proses *runtime*. Hasil dari intruksi akan berbeda tergantung dari data yang diproses. *Script* dari PHP nantinya akan diproses di *server* dikarenakan PHP merupakan bahasa pemrograman *server-side*. Jenis *server* yang biasa digunakan dalam pemrosesan *script* PHP antara lain *Apache*, *Nginx*, dan *LiteSpeed*.

Hampir seluruh *website* yang ada di dunia menggunakan bahasa pemrograman PHP, bahkan *platform* besar seperti *facebook*-pun menggunakan PHP. Ada beberapa alasan mengapa PHP sangat populer dan banyak digunakan untuk pembuatan *website*. Berikut ini adalah beberapa alasannya.

1. PHP bahasa pemrograman yang cenderung mudah untuk dipahami
2. Materi belajar yang melimpah
3. PHP bersifat open source
4. Kecepatan tinggi
5. Banyak pilihan database
6. Kompatibilitas yang baik dengan HTML
7. Fleksibilitas tinggi
8. Multi-platform
9. Selalu diperbarui
10. Mendukung layanan cloud

Setiap bahasa pemrograman pasti memiliki aturan *coding* atau penulisan sintaks. PHP memiliki sintaks dasar yaitu `<?php` dan ditutup dengan `?>`. Contoh penulisan dapat dilihat pada kode 2.4 dibawah ini.

```
<?php
    echo    "Bahasa    Pemrograman
    PHP";
```

Kode 2. 4 Contoh Penulisan Sintaks PHP

Penjelasan dari sintaks di atas adalah `<?php` merupakan kode pembuka yang wajib ada pada saat membuat program PHP, dan `?>` merupakan kode penutupnya. *Echo* adalah perintah untuk menampilkan teks, dan teks yang akan ditampilkan

nantinya yaitu “Bahasa Pemrograman PHP”. Sintaks PHP memiliki sifat *case sensitive*. Jadi, pengguna harus memperhatikan besar kecilnya huruf yang digunakan karena akan mempengaruhi *output* yang diberikan. Semua kode *PHP* harus disimpan dalam *file* ekstensi *.php*. Penulisan kode PHP dibagi menjadi 2 yaitu *PHP Native* dan *PHP Framework*.

1. *PHP Native*, merupakan penulisan kode PHP yang dimulai dari 0 ketika melakukan perancangan sebuah *website*. Penulisan ini banyak digunakan oleh *developer* yang memiliki keahlian *coding* cukup baik atau memang *developer* menginginkan membuat alur kerangka PHP yang unik dengan fungsionalitas tinggi.
2. *PHP Framework*, dengan menggunakan *framework* akan memudahkan *developer* dalam penulisan kode PHP dengan memanfaatkan kerangka pengolahan *website* yang sudah jadi. *Framework* membantu *developer* agar pekerjaan yang dilakukan lebih efisien dan menyelesaikan pengembangan *website* dengan cepat. Beberapa *framework* yang biasa digunakan antara lain adalah CodeIgniter, Laravel, Yii, Symfony dan Zend *Framework*. Jika *developer* sudah mahir dalam penulisan *PHP Native*, sangat disarankan untuk mencoba beralih ke *PHP Framework*. Karena kode pada *framework* sudah dioptimasi sesuai standar, mulai dari segi kecepatan hingga keamanan.

PHP sering digunakan bersamaan dengan MySQL untuk membangun sebuah *website* dinamis. Pertama-tama PHP harus sudah terhubung dengan MySQL, ketika ada permintaan dari *browser* ke *web server* PHP akan menghubungi MySQL untuk mencari data yang dibutuhkan di *database*. Setelah data berhasil didapatkan, MySQL *server* akan meneruskan informasi tersebut ke *server* untuk dilanjutkan ke *browser*. Dengan adanya PHP, komunikasi dengan database sangat dimudahkan.

2.3 Laravel

Laravel merupakan *framework* yang dapat membantu untuk memaksimalkan penggunaan PHP dalam proses pengembangan suatu *website*. Adanya Laravel membuat PHP menjadi lebih *powerful*, cepat, aman, dan *simple*. Laravel dirilis sejak tahun 2011 dan mengalami pertumbuhan yang sangat baik.

Pada tahun 2015 Laravel menjadi *framework* yang banyak mendapatkan bintang di Github dan hingga saat ini Laravel telah menjadi *framework* yang sangat populer di dunia. Laravel memiliki fokus pada *end-user*, yang berarti fokus pada kejelasan dan kesederhanaan, baik berupa penulisan maupun tampilan serta menghasilkan fungsionalitas aplikasi *web* yang berjalan sebagaimana mestinya. Tidak hanya itu saja namun, Laravel juga mempermudah proses pengembangan *website* dengan bantuan fitur unggulannya yaitu, *template engine*, *routing*, dan *modularity*.

Terdapat beberapa keuntungan yang bisa didapatkan jika mengembangkan suatu *website* menggunakan *framework* laravel. Di antaranya yaitu *website* menjadi lebih mudah untuk dikembangkan. Kemudian terdapat *namespace* dan tampilan yang memudahkan untuk mengorganisir dan mengatur sumber daya *website*. Selanjutnya yang terakhir adalah proses pengembangan *website* menjadi lebih cepat dikarenakan Laravel dapat dikombinasikan dengan beberapa komponen dari *framework* lainnya. Laravel memiliki 2 *tools* yang menjadi andalannya yaitu.

1. Composer

Composer merupakan sebuah *tool* yang di dalamnya terdapat *dependencies* dan kumpulan *library*. Seluruh *dependencies* yang ada disimpan dalam format *file* *composer.json*, dengan format ini maka *dependencies* dapat ditempatkan di *folder* utama *website*. Composer sangatlah membantu dalam hal meng-*install* *library*, *library* yang di inginkan dapat kita *install* menggunakan composer. Tidak hanya meng-*install* saja namun, *composer* juga memudahkan untuk meng-*update* *library* yang telah di *install* sebelumnya.

2. Artisan

Artisan merupakan *command line interface* yang terdapat pada Laravel. Artisan dapat mencakup perintah-perintah untuk mempermudah dalam membangun suatu *website*. Pada Laravel versi 5.1 hingga saat ini, sudah terdapat fitur *add-on* pada artisan yang dapat digunakan untuk menambahkan berbagai macam fitur baru ke Laravel.

Selain memiliki kedua *tools* unggulan tersebut, Laravel juga memiliki fitur yang tidak semua *framework* menyediakannya. Karena Laravel adalah *framework* yang *modern* maka sangat memungkinkan untuk melakukan berbagai hal di dalam

membangun *website* menggunakan Laravel ini. Berikut ini adalah beberapa fitur dari Laravel.

- a) Blade *Template Engine*, blade merupakan sebuah *template engine* yang digunakan untuk mendesain *layout*. Blade memiliki kelebihan tidak membatasi *developer* dalam menggunakan kode PHP biasa didalam tampilan, kemudian desain tampilan blade akan tetap berada di-*cache* hingga adanya perubahan yang disimpan.
- b) Routing, semua request yang ada pada Laravel dipetakan dengan menggunakan route. Routing dasarnya adalah dengan merutekan request ke controller yang terkait. Terdapat 3 kategori routing dalam Laravel yaitu basic routing, route parameter, dan named routes.
- c) Modularity, fitur modularity sangat membantu dalam menyempurnakan dan meningkatkan fungsionalitas dari website yang telah dibangun, tidak hanya itu saja fungsi modularity dapat mempermudah proses update.
- d) Testability, framework Laravel memiliki fitur proses pengecekan yang lengkap. Proses pengecekan dari framework Laravel dapat dilakukan dengan PHPUnit atau phpunit.xml yang dapat disesuaikan dengan website yang sedang dibangun. Fitur yang ada memungkinkan untuk dapat mengecek website yang telah dibangun secara ekspresif.
- e) Query Builder and ORM, fitur ini digunakan untuk menjalankan berbagai operasi database yang ada di dalam website dan mendukung berbagai sistem database.
- f) Authentication, fitur ini membuat implementasi otentikasi menjadi lebih mudah. File konfigurasi Laravel terdapat pada 'config/auth.php', di dalam file ini terdapat beberapa pilihan otentikasi yang sudah terdokumentasi dan dapat disesuaikan dengan kebutuhan.
- g) Schema Builder, fitur ini menyediakan database agnostic untuk memanipulasi tabel. Schema dari Laravel berjalan dengan baik diberbagai tipe database yang didukung oleh Laravel dan memiliki API yang sama diseluruh sistem.
- h) Configuration Management Features, sebagai developer tidak perlu khawatir mengenai setiap konfigurasi yang dibuat. Karena Laravel menyediakan fitur di mana seluruh file konfigurasi disimpan di dalam direktori 'config'.

- i) E-mail Class, dengan fitur ini dapat memungkinkan untuk mengirim email dengan cepat melalui proses local maupun cloud. Laravel sudah menyediakan API beberapa library SwiftMailer sebagai pendukung fitur ini.
- j) Redis, framework Laravel menggunakan fitur redis untuk menghubungkan tiap sesi yang ada dengan cache general-purpose. Redis biasa dikenal dengan server struktur data yang dapat menyimpan key dengan berbagai tipe.
- k) Event and Command Bus, fitur ini menyediakan metode pengumpulan tugas yang dibutuhkan aplikasi agar dapat berjalan secara sederhana dan perintah yang digunakan mudah dimengerti.

Banyaknya fitur yang ada pada *framework* ini membuat Laravel banyak dipilih oleh *developer* Indonesia bahkan dunia. Tidak hanya karena fitur saja, Laravel juga banyak dipilih karena *template layout* yang ringan, memiliki banyak *library object oriented*, mendukung *framework* MVC, tersedia *tool* artisan, pembagian modul secara *independent*, dan pengecekan menggunakan *unit testing*. *Framework* Laravel adalah pilihan yang tepat untuk membuat suatu *website* yang bersifat kompleks.

2.4 MySQL

MySQL adalah sebuah *database management system* (manajemen basis data) yang menggunakan perintah dasar SQL (*Structured Query Language*). Banyak yang sudah menggunakan *database management system* (DBMS) MySQL multi pengguna dan multi alur ini yaitu kurang lebih 6 juta pengguna di seluruh dunia. MySQL merupakan DBMS yang *open source* dan memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Jadi MySQL merupakan *database* server yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi.

MySQL merupakan proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia), kemudian dikembangkan menjadi MySQL hingga saat ini. Dulunya UNIREG belum terlalu kompatibel dengan database dinamis yang digunakan di *website*. Kemudian TcX mencari alternatif lain dan menemukan miniSQL atau mSQL. Namun, mSQL masih belum sesuai dengan kebutuhan TcX dikarenakan mSQL tidak mendukung

indexing. Akhirnya muncul Kerjasama antara UNIREG, TcX, dan mSQL. Kerjasama yang dibuat ini bertujuan untuk mengembangkan sistem database yang baru, yaitu MySQL yang dirilis pada tahun 1995. Saat ini pengembangan MySQL berada di bawah Oracle.

Meskipun MySQL menjadi database yang sangat populer, tetapi MySQL juga memiliki kekurangan dan kelebihan dibandingkan dengan database server lainnya.

1. Kelebihan MySQL

MySQL memiliki beberapa kelebihan yang bisa dimanfaatkan untuk mengembangkan suatu *software*, oleh sebab itu dengan kelebihan ini MySQL banyak digunakan di dunia. Beberapa kelebihan tersebut adalah sebagai berikut.

- Mendukung integrasi dengan bahasa pemrograman lain.
- Tidak membutuhkan RAM besar.
- Mendukung multi user.
- Bersifat open source.
- Struktur tabel yang fleksibel.
- Tipe data yang bervariasi.
- Keamanan yang terjamin.

2. Kekurangan MySQL

Di sisi lain, MySQL memiliki beberapa kekurangan yang dapat dipertimbangkan sebelum memilih MySQL sebagai *database* server. Berikut ini merupakan kekurangan dari MySQL.

- Kurang cocok untuk aplikasi game dan mobile.
- Sulit mengelola database yang besar.
- Technical support yang kurang bagus.

Sesudah memilih MySQL sebagai database server, tentunya sebagai pengguna harus bisa mengoperasikan MySQL dengan mempelajari bahasa pemrogramannya yaitu SQL. SQL ini merupakan bahasa pemrograman yang dapat merelasikan antara beberapa tabel dengan *database* maupun antar *database*. Ada 3 bentuk SQL yang perlu diketahui, yaitu *Data Definition Language* (DDL), *Data Manipulation Language* (DML), dan *Data Control Language* (DCL). Berikut ini adalah penjelasan mengenai DDL, DML, dan DCL.

1. *Data Definition Language (DDL)*

DDL berguna untuk mendefinisikan data dari dalam tabel. Ada beberapa *query* yang dikelompokkan dalam DDL. Argumen yang ada pada DDL merupakan dasar dari penggunaan SQL pada bagian awal pembuatan database. *Query* tersebut dapat dilihat pada tabel dibawah ini.

Tabel 2. 3 Query DDL

CREATE	Dipakai untuk membuat tabel dan database.
DROP	Dipakai untuk menghapus database dan tabel.
ALTER	Dipakai untuk mengubah struktur tabel yang sudah ada. Alter dapat mengganti <i>field</i> menggunakan perintah “Change”, menambahkan <i>field</i> menggunakan perintah “Add”, atau menghapus <i>field</i> menggunakan perintah “drop”, dan mengubah namanya menggunakan perintah “Rename”.

2. *Data Manipulation Language (DML)*

DML digunakan saat perintah DML sudah selesai dijalankan. DML memiliki fungsi untuk memanipulasi, mengubah, atau mengganti isi dari *database*(tabel) yang sudah ada. Perintah DML dapat dilihat pada tabel berikut ini.

Tabel 2. 4 Query DML

INSERT	Dipakai untuk memasukkan data ke dalam tabel pada <i>database</i> .
UPDATE	Dipakai untuk mengubah data yang ada di dalam tabel pada <i>database</i> .
DELETE	Dipakai untuk menghapus data di dalam tabel pada <i>database</i> .

3. *Data Control Language (DCL)*

DCL berguna untuk memberikan hak akses *database*, mendefinisikan *space*, mengalokasikan *space*, dan melakukan audit penggunaan *database*. Jadi,

dengan DML ini pengguna dapat mengatur hak akses masing-masing *user*. Berikut ini perintah dari DCL terdapat pada tabel.

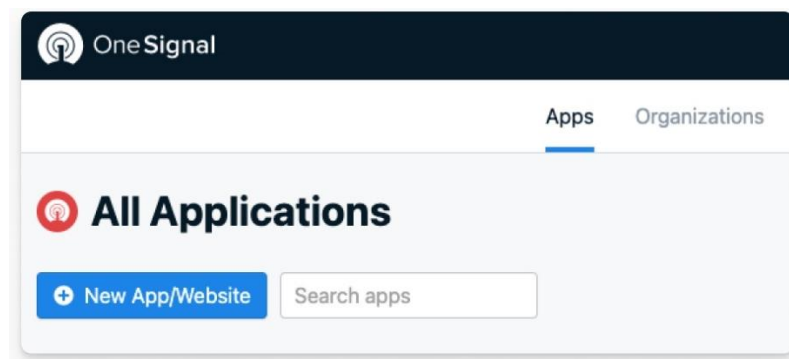
Tabel 2. 5 Query DCL

GRANT	Dipakai untuk memberikan izin kepada <i>user</i> untuk mengakses <i>database</i> .
REVOKE	Dipakai untuk membatalkan izin <i>user</i> untuk mengakses <i>database</i> .
COMMIT	Dipakai untuk menetapkan penyimpanan pada <i>database</i> .
ROOLBACK	Dipakai untuk membatalkan penyimpanan pada <i>database</i> .

2.5 Plugin OneSignal

OneSignal merupakan salah satu *plugin* yang digunakan untuk mengirim pemberitahuan notifikasi, *in-app messages*, *SMS*, dan *email*. Penggunaan plugin ini memudahkan *developer* dalam membuat suatu notifikasi pada aplikasi yang sedang dibangun. Untuk membuat suatu aplikasi dengan menggunakan plugin ini, tentunya plugin oneSignal menyediakan tutorial atau cara untuk memasang plugin ini dengan benar. Karena pada kasus penelitian ini adalah akan membangun suatu website, maka akan ada penjelasan mengenai cara untuk memasang atau menggunakan plugin ini pada sebuah website.

Sebelum melakukan langkah-langkah untuk mengkonfigurasi plugin dengan aplikasi yang akan dibuat, pengguna plugin harus mendaftarkan akun yang akan dibuat untuk mendapatkan akses plugin oneSignal. Kemudian jika aplikasi yang dibuat merupakan aplikasi pertama dengan oneSignal maka klik New App/Website seperti pada gambar 2.2 di bawah ini.



Gambar 2. 2 Memulai membuat aplikasi dengan *OneSignal*

Selanjutnya beri nama aplikasi sesuai dengan aplikasi yang sedang di buat, lalu pilih *Web* dari daftar *platform*. Klik "Next: Configure Your Platform" seperti pada gambar 2.3 di bawah ini.

Gambar 2. 3 Halaman pengisian nama aplikasi dan pemilihan *platform*

Setelah melalui tahap registrasi di atas, berikut ini merupakan langkah cepat penggunaan push notification yang disediakan oleh plugin oneSignal.

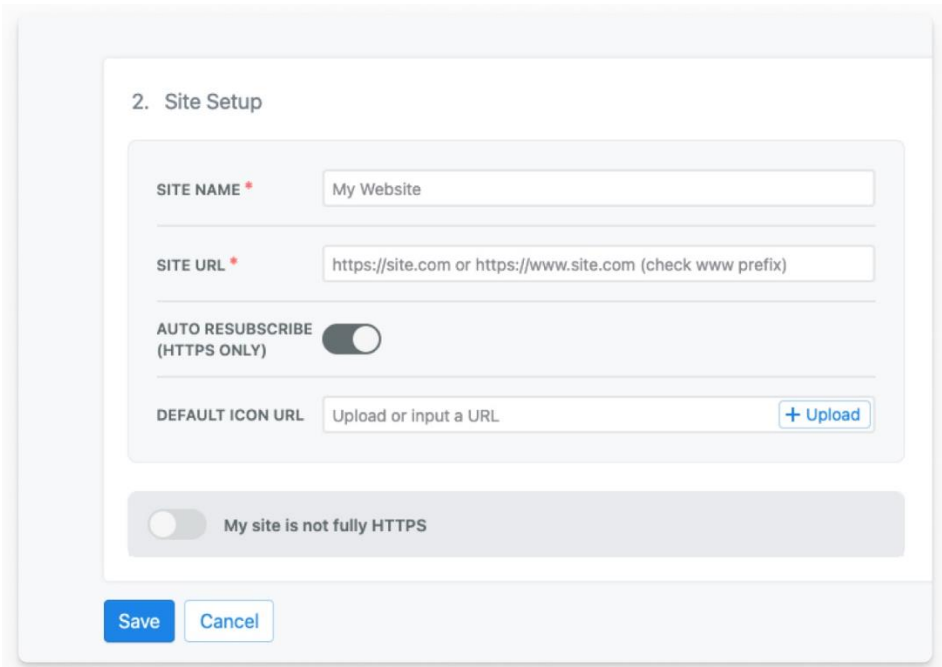
1. Choose Integration

Pada langkah ini, jika ingin menambahkan OneSignal pada web yang sudah di build sendiri maka pilih *Typical Site* atau pilih *Custom Code Setup*. Namun, jika menggunakan plugin *Wordpress*, *Shopify App*, atau *custom Website Builder*, maka pilih *Wordpress Plugin or Website Builder*. Lebih tepatnya dapat dilihat pada gambar 2.4 di bawah ini.

Gambar 2. 4 Halaman Web Configuration

2. Site Setup

Kemudian masukkan informasi dari situs atau website yang telah dibuat, seperti pada gambar 2.5 dibawah ini.



Gambar 2. 5 Halaman Site Setup

Berikut ini penjelasan lebih detail dari form yang harus diisi tersaji pada tabel 2.6 berikut ini.

Tabel 2. 6 Detail field pada form site setup

Nama Field	Deskripsi
<i>SITE NAME</i>	Nama yang digunakan untuk memanggil situs atau <i>website</i> dalam penggunaan <i>push notification</i>
<i>SITE URL</i>	URL untuk website yang telah dibuat. Apakah termasuk kedalam <code>http://</code> or <code>https://</code> , <code>www</code> atau non- <code>www</code> . Jika website masih dalam local, maka setup bisa dilanjutkan dengan cara yang disediakan oleh plugin OneSignal.
<i>AUTO RESUBSCRIBE (HTTPS ONLY)</i>	Direkomendasikan: Memungkinkan pengguna untuk berlangganan kembali secara otomatis

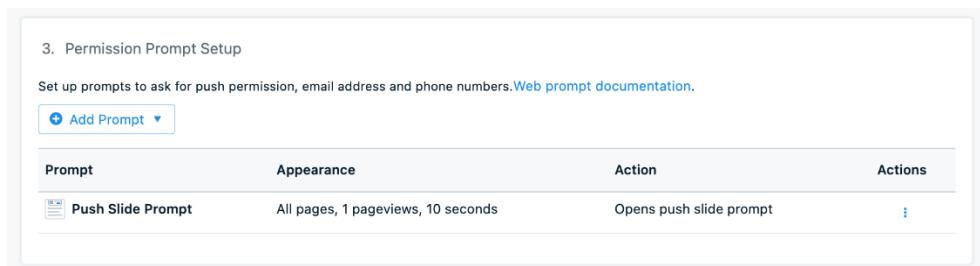
setelah kembali ke situs atau *website* tanpa diminta jika mereka menghapus data *browser* mereka atau jika *developer* mentransfer ke OneSignal dari penyedia layanan lainnya.

DEFAULT ICON URL

Menambahkan *icon* untuk notifikasi yang akan dibuat. Masukkan URL *icon* dengan url atau unggah file berukuran 256x256 piksel. File harus .png, .jpg, atau .gif.

3. Permission Prompt Setup

Secara default, *Push Slide Prompt* akan muncul pada halaman awal setelah 10 detik. Namun, langkah ini dapat dilewati saat ini dan nanti dapat kembali lagi untuk menambahkan *new prompt* atau memodifikasi *prompt* sebelumnya. Berikut ini gambar 2.6 adalah gambar halaman Permission Prompt Setup.



Gambar 2. 6 Halaman Permission Prompt Setup

4. Welcome Notification (Optional)

Push notification akan mengirimkan pesan pemberitahuan kepada pengguna saat ini setelah berlangganan situs atau website yang telah dibuat untuk pertama kalinya.

5. Advanced Push Settings (Optional)

Plugin OneSignal menyediakan beberapa opsi lanjutan yang dapat digunakan, contohnya adalah *Multiple Service Workers*, dan *Safari - Custom Certificate (Optional)*.

6. Upload Files

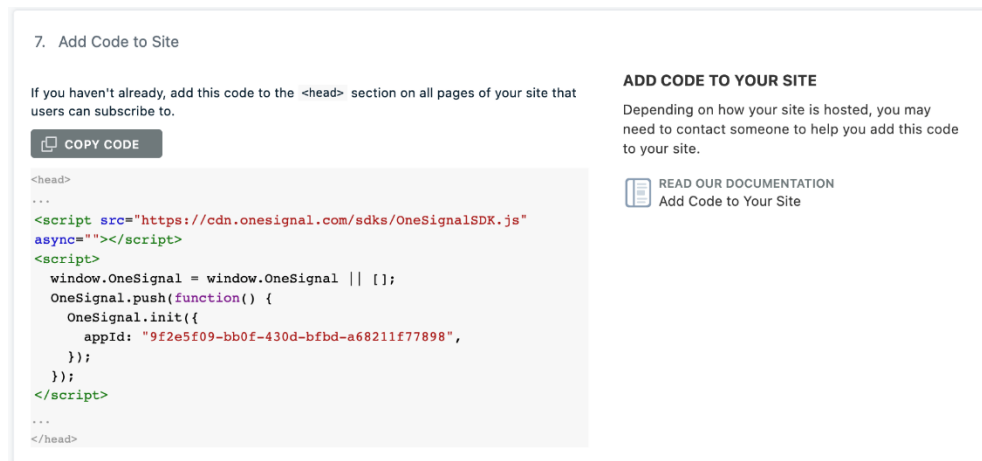
Langkah selanjutnya adalah upload OneSignal SDK Service Worker ke situs atau website yang telah dibuat. Terdapat beberapa opsi untuk melakukan

langkah ini, opsi yang pertama adalah buat file OneSignalSDKWorker.js pada public kemudian *copy* kode berikut ini pada *file* yang sudah dibuat sebelumnya `importScripts('https://cdn.onesignal.com/sdks/OneSignalSDKWorker.js');`.

Opsi yang kedua yaitu download OneSignal SDK, kemudian unzip file tersebut nantinya akan terdapat suatu file yang perlu untuk di-host pada url berikut: `https://yoursite.com/OneSignalSDKWorker.js`.

7. Add Code to Site

Untuk mengaktifkan OneSignal pada situs atau website yang telah dibuat, Tempelkan kode yang disediakan ke `<head>` pada website. Jika tidak memiliki akses untuk mengubah kode situs, hubungi pihak OneSignal untuk membantu menyelesaikan masalah tersebut. Berikut ini adalah gambar halaman untuk meng-copy kode yang dibutuhkan.



Gambar 2. 7 Halaman Add Code to Site

8. Subscribe and Send Yourself a Notification

Periksa *Audiens Dasbor OneSignal > All Users* untuk melihat *Device Record* dan menetapkan developer sebagai *User Test*. Kemudian buka *Messages > New Push* untuk Mengirim Pemberitahuan pertama.

2.6 API (Application Programming Interface)

API merupakan *interface* yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya. Bisa dikatakan bahwa API merupakan jembatan antar berbagai aplikasi berbeda, baik dalam satu *platform* yang sama maupun lintas *platform*. API

dapat digunakan untuk komunikasi dengan berbagai macam bahasa pemrograman yang berbeda. Hal ini sangat memudahkan para *developer*, karena tidak perlu menyediakan semua data yang dibutuhkan sendiri karena cukup dengan mengambil data yang dibutuhkan dari *platform* lain menggunakan API. Tidak berhenti disitu saja, namun API juga memungkinkan digunakan untuk mengembangkan suatu website dengan fitur yang lengkap. Bukan tanpa alasan bahwa API banyak digunakan oleh para *developer*, manfaat yang didapat antara lain adalah memudahkan dalam membangun aplikasi yang fungsional, pengembangan aplikasi menjadi lebih efisien, dan meringankan beban server. Cara kerja dari API sangatlah sederhana, yang pertama adalah aplikasi mengakses API terlebih dahulu, kemudian API melakukan *request* ke *server*, setelah itu *server* memberi respon ke API apakah data yang dicari sudah sesuai dengan permintaan, selanjutnya API menyampaikan respon ke aplikasi. Berikut ini adalah gambaran dari ilustrasi cara kerja dari API yang dapat dilihat pada gambar 2.8 di bawah ini.



Gambar 2. 8 Ilustrasi cara kerja API

API memiliki empat jenis sesuai dengan hak aksesnya, berikut ini merupakan jenis-jenis API sesuai hak aksesnya.

1. Public API, merupakan API yang dapat digunakan siapa saja dalam lintas platform.
2. Private API, biasanya digunakan untuk keperluan internal dalam pengembangan aplikasi tertentu.
3. Partner API, digunakan untuk kepentingan umum, namun hanya pihak yang memiliki izin saja yang dapat mengaksesnya.

4. Composite API, merupakan API yang menyimpan data dari berbagai server atau hosting dalam satu tempat.

Selain itu API memiliki tiga arsitektur yang sering digunakan oleh para developer. Arsitektur API ini berkaitan dengan bentuk data yang dikirim, di bawah ini merupakan penjelasan dari ketiga arsitektur tersebut.

1. RPC, merupakan arsitektur dengan konsep yang sederhana untuk membuat komunikasi antar *client side* dan *server side*. RPC memiliki dua jenis yaitu XML-RPC dan JSON-RPC.
2. SOAP (Simple Object Access Protocol), arsitektur ini menggunakan XML agar semua data memungkinkan untuk disimpan.
3. REST (Representational State Transfer), merupakan arsitektur yang banyak digunakan oleh developer karena tak perlu melakukan coding yang panjang untuk menggunakannya. REST menggunakan JSON sebagai bentuk datanya sehingga lebih ringan dan performa aplikasinya akan menjadi lebih baik.

2.7 Fuzzy Logic

Fuzzy Logic merupakan salah satu metode kecerdasan buatan yang digunakan untuk memecahkan suatu permasalahan dengan bantuan teknologi komputer. Logika ini pertama kali dikembangkan pada tahun 1965 oleh Prof. Lotfi A. Zadeh, beliau adalah seorang ilmuwan Amerika Serikat. Fuzzy Logic memiliki perbedaan dengan logika klasik atau Logika Boolean, saat logika klasik mengatakan bahwa segala hal hanya dapat diartikan dengan istilah biner seperti 0 atau 1, ya atau tidak. Namun, Fuzzy Logic menggantikan kebenaran boolean dengan tingkat kebenaran. Fuzzy Logic memungkinkan nilai keanggotaan antara 0 dan 1, atau konsep ketidakpastian seperti “buruk”, “baik”, dan “sangat baik”.

Terdapat beberapa alasan mengapa Fuzzy Logic ini banyak digunakan untuk memecahkan permasalahan (Cox, 1994), alasan tersebut antara lain:

1. Konsep logika fuzzy mudah dimengerti.
2. Logika fuzzy sangat fleksibel.
3. Logika fuzzy memiliki toleransi terhadap data yang tidak tepat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.

5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika fuzzy bekerjasama dengan teknik-teknik kendali secara konvensional
7. Logika fuzzy didasarkan pada bahasa alami .

Ada beberapa hal yang perlu diketahui dalam memahami fuzzy (Kusumadewi & Purnomo, 2010), yaitu:

1. Variabel Fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan, dan lain sebagainya.

2. Himpunan Fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy.

4. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy.

Fuzzy logic memiliki berbagai jenis teknik penyelesaian masalah yang dapat digunakan sesuai dengan kebutuhan. Seperti *Fuzzy Inference System*, *Fuzzy Clustering*, *Fuzzy Database*, *Fuzzy Quantification Theory* dan teknik-teknik lainnya. Setiap teknik tersebut memiliki cara penyelesaiannya masing-masing tergantung dari permasalahan yang akan diselesaikan.

2.8 Clustering

Clustering merupakan salah satu teknik dalam *machine learning* yang digunakan untuk pengelompokan data sehingga data yang ada pada *cluster* yang sama memiliki karakteristik yang mirip. Metode *clustering* ini termasuk dalam salah satu kelompok *unsupervised learning*, di mana data yang akan dikelompokkan tidak diberikan label sehingga proses training yang dilakukan sepenuhnya diserahkan kepada algoritma yang menangani proses tersebut. *Clustering* sudah banyak digunakan dalam berbagai aplikasi saat ini, sudah banyak

perusahaan yang menggunakan teknik *clustering* ini untuk mengelompokan data sesuai dengan apa yang dibutuhkan.

Hasil dari clustering yang baik akan menghasilkan tingkat kesamaan yang tinggi dalam satu kelompok, dan tingkat kesamaan yang rendah antar kelompok yang lain. Pada clustering ini, kesamaan yang dimaksud adalah berdasarkan pengukuran nilai numerik terhadap dua buah objek. Nilai kesamaan nantinya akan menjadi tinggi ketika dua objek yang dibandingkan memiliki kesamaan yang begitu tinggi, berlaku juga dengan keadaan sebaliknya. Kualitas dari hasil *clustering* bergantung pada teknik *clustering* yang digunakan, dalam pemilihan teknik harus diperhatikan data yang nantinya akan diproses.

2.9 Fuzzy C-Means (FCM)

Fuzzy C-Means atau FCM adalah suatu teknik pengclusteran data yang keberadaan tiap titik data dalam suatu *cluster* ditentukan oleh derajat keanggotaan (Kusumadewi & Purnomo, 2010). Teknik pengclusteran ini awalnya diperkenalkan oleh Jim Bezdek pada tahun 1981. Konsep dasar dari FCM pertama kali adalah menentukan pusat *cluster*, yang nantinya akan menandai lokasi rata-rata untuk tiap *cluster*. Pada kondisi awal, pusat cluster masih belum akurat. Dengan cara memperbaiki pusat cluster dan nilai keanggotaan tiap-tiap data secara berulang, maka dapat dilihat bahwa pusat cluster akan menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi objektif yang menggambarkan jarak dari titik data yang diberikan ke pusat cluster yang terbobot oleh derajat keanggotaan titik data tersebut. Berikut ini merupakan algoritma dari FCM adalah sebagai berikut.

1. Input data yang akan di cluster, data tersebut berupa matriks berukuran $n \times m$ dengan keterangan n adalah jumlah setiap sampel data dan m merupakan atribut setiap data. X_{ij} = data sampel ke- i ($i=1,2,3,\dots,n$), atribut ke- j ($j=1,2,3,\dots,n$).
2. Tentukan :
 - a) Jumlah *cluster* = c ;
 - b) Pangkat = w ;
 - c) Maksimum iterasi = MaxIter;
 - d) Error terkecil yang diharapkan = ξ

e) Fungsi objektif awal = $P_0 = 0$

f) Iterasi awal = $t = 1$

3. Bangkitkan bilangan *random* μ_{ik} , $i = 1, 2, \dots, n$; $k = 1, 2, \dots, c$; sebagai elemen-elemen matriks partisi awal U. Bilangan random yang dibangkitkan adalah bilangan antara 0 hingga 1. Bilangan random yang dibangkitkan ini nantinya akan menjadi derajat keanggotaan yang akan digunakan pada perhitungan selanjutnya.

Hitung jumlah setiap kolom:

$$Q_i = \sum_{k=1}^c \mu_{ik} \quad (2.1)$$

Dengan $j = 1, 2, \dots, n$.

Hitung:

$$\mu_{ik} = \frac{\mu_{ik}}{Q_i} \quad (2.2)$$

4. Hitung pusat cluster ke-k: V_{kj} , dengan $k = 1, 2, \dots, c$; dan $j = 1, 2, \dots, m$. Pusat cluster dihitung dengan cara mengalikan nilai data dengan nilai pangkat keanggotaan.

$$V_{kj} = \frac{\sum_{i=1}^n ((\mu_{ik})^w \times X_{ij})}{\sum_{i=1}^n (\mu_{ik})^w} \quad (2.3)$$

5. Fungsi objektif didapatkan dari perhitungan penjumlahan dari data dikurangi dengan pusat cluster kemudian dikali dengan nilai pangkat derajat keanggotaan. Rumus fungsi objektif pada iterasi ke- t , P_t :

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left(\left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] (\mu_{ik})^w \right) \quad (2.4)$$

6. Hitung perubahan matriks partisi, matriks partisi ini nantinya akan digunakan untuk derajat keanggotaan baru pada iterasi selanjutnya.

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \quad (2.5)$$

Dengan: $i = 1, 2, \dots, n$; dan $k = 1, 2, \dots, c$.

7. Cek kondisi berhenti

- a) Jika : $(|P_t - P_{t-1}| < \xi)$ atau $(t > MaxIter)$ maka berhenti;
- b) Jika tidak : $t = t + 1$, ulangi langkah 4.

2.10 Blackbox Testing

Blackbox testing merupakan salah satu metode pengujian untuk menguji *software*. Pengujian ini dilakukan untuk mengamati input dan output dari sebuah *software* tanpa mengetahui kode program. Blackbox testing ini biasanya dilakukan ketika pembuatan *software* sudah selesai untuk mengetahui apakah *software* yang dibuat sudah sesuai dengan apa yang diharapkan. Penguji yang melakukan pengujian tidak harus dapat memahami kode program dari *software* yang diuji. Metode pengujian ini memiliki tiga tipe, yaitu:

1. *Functional testing*, merupakan pengujian terhadap fungsi dari fitur yang ada pada *software*.
2. *Non-functional testing*, pengujian ini dilakukan pada aspek-aspek tambahan non-fungsional.
3. *Regression testing*, merupakan pengujian yang dilakukan untuk memastikan apakah terjadi regresi atau kemunduran ketika versi *software* di-*upgrade*

Selain itu *blackbox testing* memiliki beberapa kelebihan yaitu, tidak membutuhkan orang yang memiliki latar belakang teknis atau mengetahui kode program dari *software*. Kemudian *blackbox testing* ini juga dapat mengidentifikasi kekurangan dengan cepat pada awal pengujian. Namun, *blackbox testing* juga memiliki kekurangan yaitu, ketika penguji tidak mengetahui teknis program maka kemungkinan hal-hal dengan potensi kesalahan yang tidak diketahui atau terlewat begitu saja. Lalu jika pengujian dilakukan dalam waktu yang singkat maka terjadi

kemungkinan pengujian yang dilakukan kurang teliti akibat terlewatnya beberapa uji *input* dan *output*.

2.11 Penelitian Terdahulu

Terdapat beberapa penelitian yang menjadi acuan dari penelitian yang akan dibuat oleh penulis. Fokus tema jurnal yang pertama adalah mengenai kedisiplinan minum obat. Jurnal yang pertama berjudul “Determinan Keberhasilan Pengobatan Multi Drug Therapy Pada Penderita Kusta Tipe Multibaciler” oleh Muh. Zul Azhri Rustam. Menurut Muh. Zul Azhri, penelitian ini dilakukan untuk mengetahui faktor dominan yang mempengaruhi terhadap keberhasilan pengobatan MDT pada penderita kusta tipe MB di Provinsi Sulawesi Selatan. Hasil yang diperoleh dari penelitian ini adalah terdapat pengaruh antara kepatuhan minum obat terhadap pengobatan MDT (Rustam, 2018). Mayoritas responden yang patuh meminum obat adalah responden berjenis kelamin laki-laki yang bekerja, responden tersebut termotivasi untuk sembuh karena pekerjaannya demi menafkasi keluarga. Penelitian yang hampir sama dengan sebelumnya mengenai kepatuhan minum obat adalah penelitian yang berjudul “Hubungan Kepatuhan Minum Obat Antihipertensi Terhadap Tercapainya Target Terapi Pasien Hipertensi di Puskesmas Wirobrajan Yogyakarta”. Penelitian ini dilakukan untuk mengetahui hubungan kepatuhan minum obat antihipertensi terhadap tercapainya target terapi pasien hipertensi di Puskesmas Wirobrajan Yogyakarta. Hasil yang diperoleh dari penelitian yang telah dilakukan yakni adanya hubungan antara kepatuhan pasien dalam minum obat antihipertensi terhadap tercapainya target terapi pasien hipertensi (Cahyani, 2018). Pasien yang memiliki tingkat kepatuhan yang tinggi merupakan pasien dengan penyakit penyerta DM, dengan adanya penyakit penyerta dan beberapa pengaruh lain membuat pasien termotivasi untuk patuh dalam meminum obat.

Tema jurnal selanjutnya adalah mengenai pengelompokan suatu data menggunakan metode kecerdasan buatan *Fuzzy Logic*. Penelitian yang pertama berjudul “Perbandingan Algoritma K-Means Dengan Algoritma Fuzzy C-Means Untuk Clustering Tingkat Kedisiplinan Kinerja Karyawan”. Tujuan dibuatnya penelitian ini adalah untuk memudahkan penentuan tingkat kedisiplinan karyawan

dengan 3 kelompok yaitu tingkat kinerja tinggi, tingkat kinerja sedang dan tingkat kinerja rendah. Untuk mengelompokkannya peneliti menggunakan metode K-Means dengan Fuzzy C-Means. Hasil yang diperoleh dari penelitian ini adalah metode Fuzzy C-Means adalah metode yang lebih baik dibandingkan K-Means untuk melakukan pengklasteran data pada tingkat kinerja karyawan di STT Bandung karena nilai validasinya bernilai mendekati 1 (Agustina & Prihandoko, 2018). Penelitian serupa yaitu berjudul “Perbandingan Metode K-Means dan Fuzzy C-Means Untuk Pengelompokan Pegawai Berdasarkan Nilai Kinerja dan Tingkat Kedisiplinan Pegawai” pada tahun 2018 oleh Wikarno, et al. Penelitian ini memiliki tujuan untuk mengetahui klasifikasi setiap pegawai yang ada di Bank Indonesia dengan membandingkan performa metode K-Means dengan C-Means. Pengelompokan tersebut di bagi menjadi 5 kelompok pegawai yaitu kualifikasi terbaik, kualifikasi baik, kualifikasi cukup, kualifikasi kurang, dan kualifikasi buruk. Kesimpulan yang diperoleh pada penelitian tersebut adalah algoritma K-Means lebih baik daripada algoritma C-Means, karena metode K-Means unggul di beberapa pengujian yang dilakukan oleh peneliti. Penelitian selanjutnya yang hampir sama yaitu berjudul “Klasterisasi kinerja karyawan menggunakan algoritma fuzzy c-means”. Penelitian yang dilakukan oleh Martin dan Yessica Nataliani bertujuan untuk mendapatkan tiga kelompok karyawan dengan kriteria baik, sedang dan tidak baik. Pengelompokan tersebut berdasarkan kriteria penilaian seperti presensi, kedisiplinan, dan waktu penyelesaian pekerjaan. Telah dibuktikan pada penelitian ini bahwa FCM dapat digunakan untuk mendapatkan kelompok karyawan berdasarkan kinerjanya. Dengan menetapkan jumlah kelompok karyawan sebesar tiga kelompok, FCM bekerja dengan baik dalam melakukan klasterisasi atau pengelompokan kinerja karyawan (Martin & Nataliani, 2020).

Kemudian tema yang terakhir adalah mengenai pemanfaatan teknologi masa kini untuk bidang medis, khususnya membantu dalam pengawasan pasien. Jurnal pertama yaitu “Rancang Bangun Interactive Voice Response (IVR) Sebagai Pengingat Medikasi Pasien Lansia Berbasis Web” pada tahun 2018 oleh Fariz Aulia Pradipta. Pada penelitian tersebut, peneliti merancang sebuah website yang digunakan untuk mempermudah dokter dalam memantau setiap pasien yang sedang dirawatnya. Untuk sisi pasien, web ini mempermudah pasien untuk menghubungi

dokter tentang keadaan yang saat ini pasien rasakan dan juga pasien akan mendapatkan notifikasi pengingat minum obat yang sudah diresepkan oleh dokter yang menanganinya. Jurnal serupa selanjutnya yaitu berjudul “Rancang Bangun Smart Medicine Box Sebagai Pengingat Jadwal Minum Obat Berbasis Internet Of Things” oleh Alisya, et al. Seringnya pasien lansia lupa untuk meminum obat membuat penelitian ini dilakukan dengan memberikan solusi yang tepat untuk membuat sebuah Smart Medicine Box sebagai pengingat jadwal bagi para pasien. Alat ini terdiri dari dua kotak penyimpanan obat yang dapat berputar otomatis ketika pasien lansia/Caregiver mendekatkan tangannya ke touchless button, dan dilengkapi dengan aplikasi berbasis android yang digunakan untuk mengatur alarm dan mencatat data waktu ketika pasien lansia/Caregiver menekan touchless button untuk mengambil obat (Alisya, et al., n.d.). Selanjutnya terdapat sebuah jurnal yang berjudul “Desain Prototipe Sistem Monitoring Minum Obat Bagi ODHA” jurnal ini terbit pada tahun 2021 oleh Syawaludin, dkk. ODHA merupakan sebutan bagi orang yang terjangkit penyakit HIV/AIDS, untuk pengobatannya pemerintah menyarankan untuk melakukan terapi antiretroviral (ART). Terapi tersebut mengharuskan pasien untuk disiplin dalam mengkonsumsi obat yang telah diberikan untuk menghambat replikasi virus HIV, namun dalam kenyataannya banyak ODHA yang tidak mendapatkan hasil yang optimal dalam menjalankan terapi tersebut, karena kurangnya tingkat kepatuhan pasien dalam mengkonsumsi obat ARV dan juga tidak termonitoring dengan baik oleh para pendamping (Syawaludin, et al., 2021). Oleh karena itu peneliti membuat sebuah desain prototipe sistem monitoring minum obat bagi ODHA. Prototipe yang dibuat telah melewati uji tingkat kepuasan sistem oleh 31 responden, dan memperoleh presentase 74% yang menunjukkan hasil sangat baik. Sistem tersebut dapat membantu ODHA untuk patuh dalam meminum obat agar terapi yang dijalani dapat berhasil dengan baik. Jurnal selanjutnya adalah berjudul “Aplikasi Monitoring Untuk Pasien Tbc Dewasa Berbasis Android” oleh Muhammad Idris Musa. Penyakit TBC merupakan penyakit yang membahayakan jika pasiennya tidak meminum obat sesuai dengan arahan dokter. Oleh karena itu peneliti membuat sebuah aplikasi mobile yang dapat memudahkan tempat berobat atau petugas yang menangani pasien TBC dapat mengawasi pasien tersebut untuk dapat mengetahui

kepatuhan dalam meminum obat. Fitur yang ada pada aplikasi tersebut adalah mengelola data pasien, mengelola kunjungan, mengelola pesan, mengelola alarm presensi minum obat untuk admin (web), dan mengupdate presensi minum obat, melihat pesan, melihat biodata sendiri, melihat informasi tb untuk pasien (mobile). Adanya aplikasi ini diharapkan dapat membantu faskes dalam memantau dan membantu setiap pasien untuk patuh dalam meminum obat. Kemudian untuk jurnal selanjutnya yang memiliki tema yang sama yaitu “Rancang Bangun Aplikasi Lansia Care Untuk Monitoring Lansia Yang Tinggal Sendiri Berbasis Android” oleh Annisa Arum Mumtazah. Peneliti disini berfokus pada lansia yang tinggal sendiri, di mana rentannya lansia untuk terjatuh, ketakutan akan kematian, pengelihan yang berkurang, kepikunan, dan rasa terisolasi. Untuk mengatasi masalah tersebut peneliti membangun sebuah aplikasi android yang menjembatani hubungan antara pengasuh dengan lansia. Aplikasi Android ini dapat mendeteksi apakah lansia terjatuh dengan mengklasifikasi data *sensorviii accelerometer* dan *gyroscope* pada perangkat keras Android dan mengirimkan pesan darurat kepada pengasuh melalui SMS (Mumtazah, 2015). Aplikasi ini diharapkan dapat membantu para lansia dan juga pengasuh untuk dapat mengawasi para lansia yang tinggal sendirian.

Selanjutnya terdapat sebuah aplikasi yang menjadi referensi penulis untuk melakukan penelitian ini, aplikasi tersebut bernama Pillpack. Pillpack didirikan pada awal tahun 2013 dan memiliki markas di Cambridge. Aplikasi garapan amazon ini merupakan aplikasi apotek modern. Pillpack telah digunakan oleh pelanggan di 40 negara bagian di Amerika. Layanan yang disediakan Pillpack membantu pasien dalam hal meminum obat, Pillpack secara otomatis akan menyortir, mengatur, membagikan obat sesuai dosis, dan mengemasnya dengan memberikan label nama obat dan diurutkan sesuai tanggal dan waktu meminum agar pasien mengetahui jadwal meminum obat tersebut. Selain itu terdapat keunggulan dari aplikasi ini, yaitu menyediakan layanan isi ulang untuk pelanggan sehingga pasien tidak pernah kehabisan obat, interaksi dengan dokter dan asuransi, dan menyediakan apoteker yang siap sedia 24 jam. Pasien juga dimudahkan dalam mengontrol daftar obat secara online, melacak pengiriman, menghitung tagihan, dan mencatat jadwal penting pengiriman dan pengisian ulang. Pillpack tidak hanya melayani pemesanan obat sesuai resep dokter, namun Pillpack juga melayani

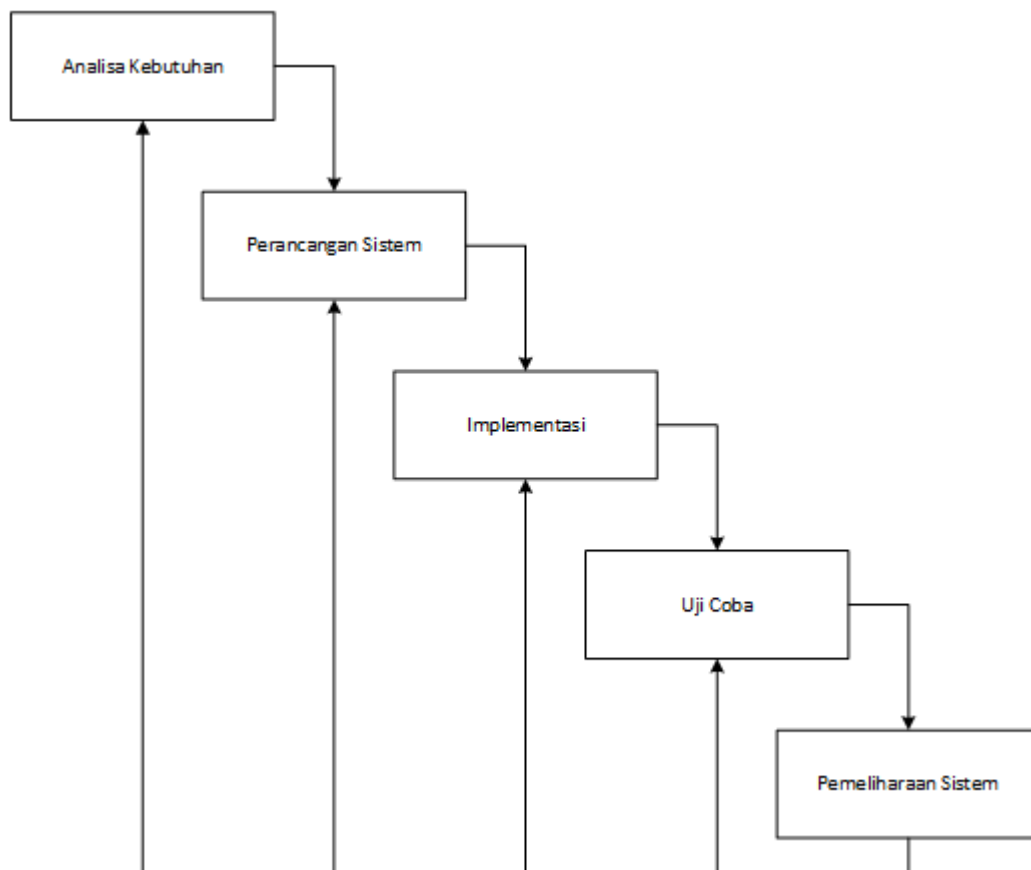
pemesanan obat tanpa resep seperti vitamin, obat gosok, pembalut, dan lain sebagainya.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Penelitian ini dilakukan untuk membuat suatu website yang dapat mengirimkan notifikasi pengingat minum obat kepada pasien serta mengelompokkan data pasien kedalam kelompok kedisiplinan dengan menggunakan algoritma Fuzzy. Oleh karena itu, penelitian ini dibagi menjadi beberapa proses tahapan seperti pada gambar 3.1 berikut ini.



Gambar 3. 1 Alur Penelitian

3.2 Analisa Kebutuhan

Sulitnya Pengawas Minum Obat atau PMO dalam mengawasi setiap pasien penderita kusta untuk meminum obat secara teratur membuat peneliti ingin membantu permasalahan tersebut. Solusi yang saat ini akan peneliti berikan adalah

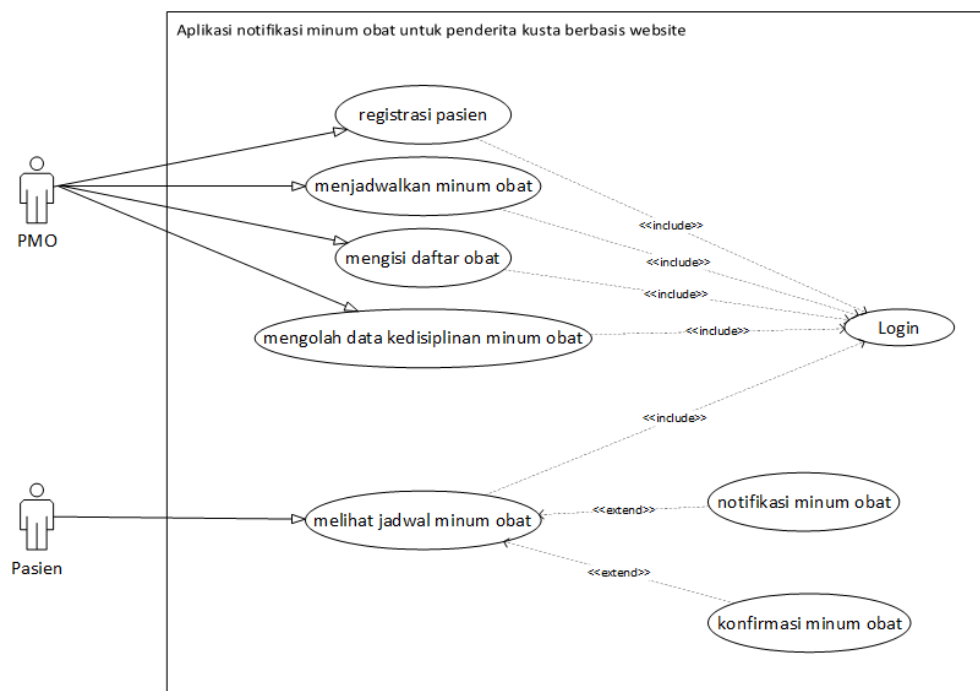
dengan pembuatan sebuah website. Website yang akan dibuat nantinya diharapkan dapat membantu PMO dalam mengawasi kepatuhan atau kedisiplinan pasien penderita kusta dalam meminum obat yang sudah diresepkan dan dijadwalkan. Mengingat PMO tidak dapat mengawasi secara langsung, maka website yang akan dibuat diharapkan dapat membantu mempermudah PMO dalam melakukan pengawasan dan juga pencatatan data setiap pasien. Data yang nantinya terkumpul akan dikelompokkan kedalam kelompok kedisiplinan supaya PMO dapat mengamati setiap pasien yang disiplin dan tidak agar dapat dilakukan penanganan tindak lanjut yang tepat.

3.3 Perancangan Sistem

Pada tahap perancangan sistem, akan dilakukan untuk merancang web yang akan dibuat nantinya. Berikut ini beberapa tahapan perancangan sistem yang dibutuhkan untuk membuat sebuah website notifikasi minum obat untuk penderita kusta.

1. Usecase diagram

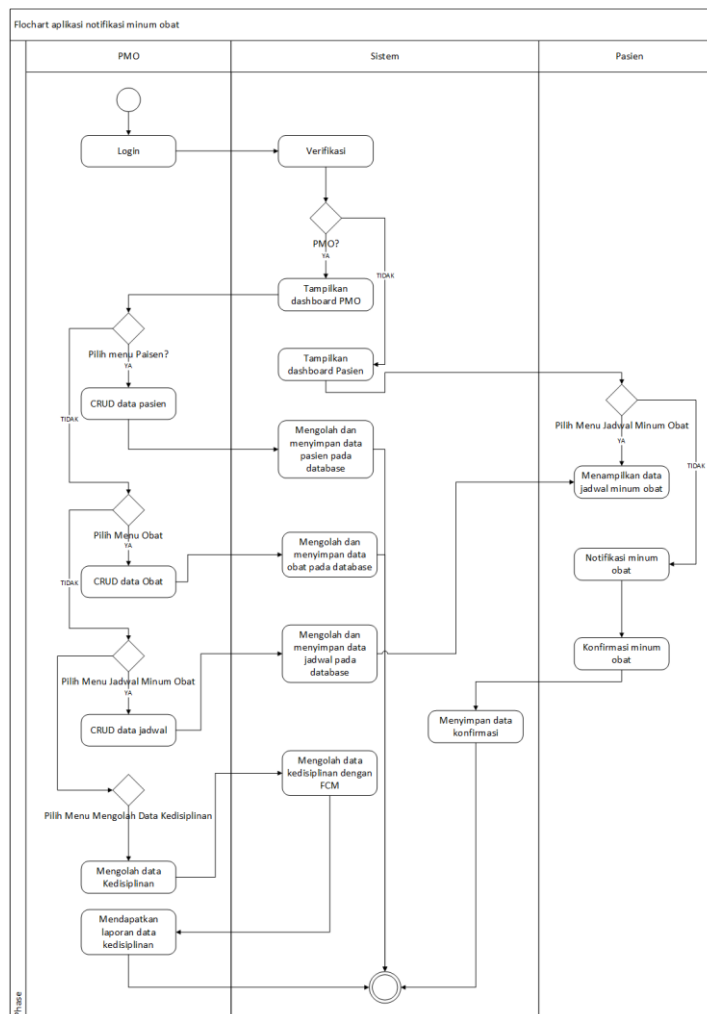
Tahap yang pertama dibuat adalah usecase diagram, berikut ini merupakan gambar usecase diagram yang telah dirancang.



Gambar 3. 2 Usecase Aplikasi Notifikasi Minum Obat Penderita Kusta

Gambar 3.2 merupakan usecase diagram dari aplikasi notifikasi minum obat penderita kusta berbasis web yang akan dibuat. Aktor yang terdapat pada usecase tersebut ada 2 yaitu PMO dan juga pasien. Pada usecase ini, PMO dapat melakukan proses registrasi pasien, menjadwalkan minum obat, mengisi daftar obat-obatan yang nantinya diberikan kepada pasien, kemudian mengolah data kedisiplinan setiap pasien yang ditangani. Aktor selanjutnya adalah pasien, di mana pasien hanya dapat melihat daftar jadwal minum obat, kemudian dari jadwal tersebut pasien akan mendapatkan notifikasi minum obat sesuai dengan jadwal dan pasien harus mengkonfirmasi setiap notifikasi yang didapat jika pasien sudah meminum obat yang telah dijadwalkan. Namun, untuk melakukan semua proses tersebut PMO dan pasien harus melakukan login terlebih dahulu.

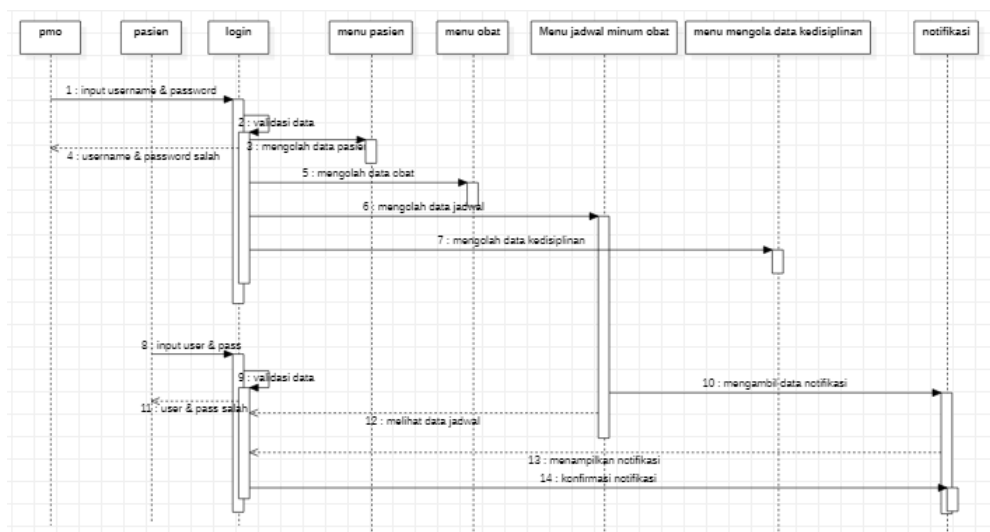
2. Activity diagram



Gambar 3. 3 Activity Diagram Aplikasi

Pada activity diagram ini menjelaskan bagaimana runtutan proses yang terjadi pada sistem aplikasi notifikasi minum obat ini. Pada gambar 3.3 tersebut merupakan model alur kerja sistem pada aplikasi yang akan dibuat, dalam proses tersebut terdapat hubungan aktivitas antara user dan sistem. Aktivitas yang dilakukan oleh user antara lain adalah login, kemudian mengolah data pasien, jadwal, dan mengolah kedisiplinan pasien. Tentunya setiap user memiliki level tersendiri yang dapat membedakan aktivitas mana saja yang dapat diakses oleh setiap level user tersebut. Gambar 3.3 untuk lebih jelasnya dapat dilihat pada LAMPIRAN A.

3. Sequence diagram

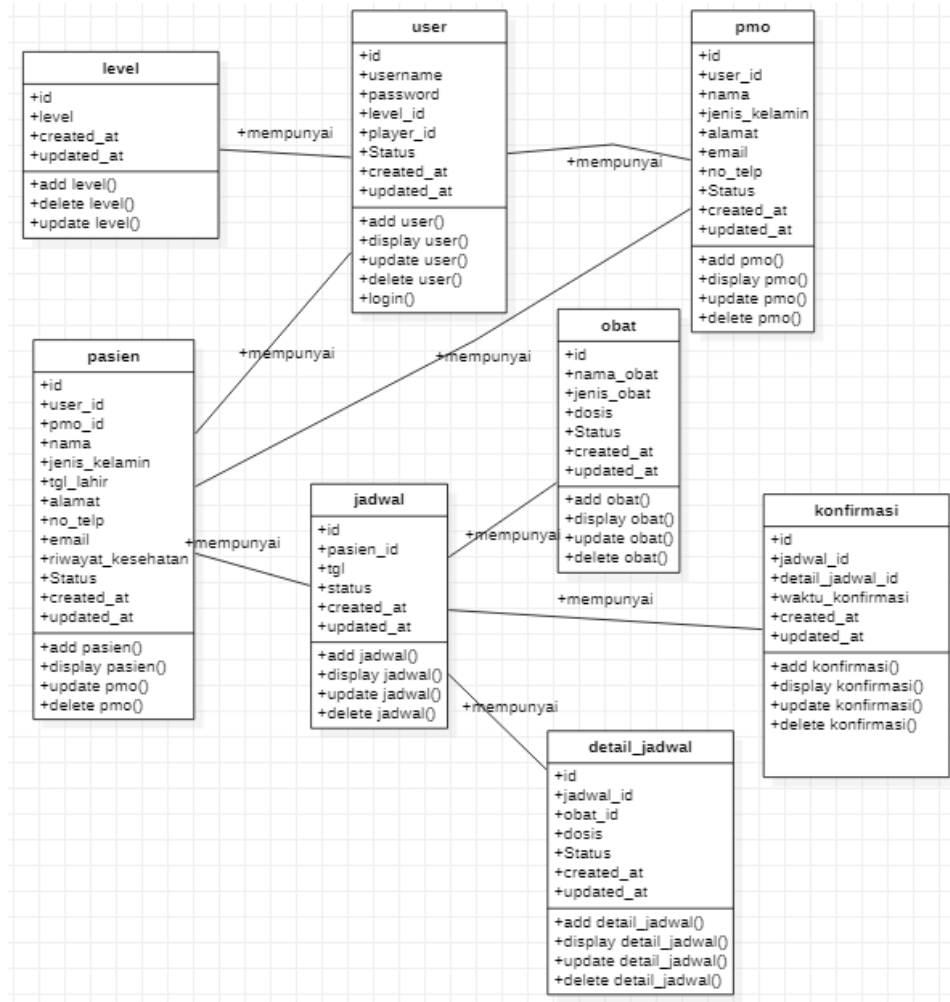


Gambar 3. 4 Sequence Diagram

Gambar 3.4 merupakan gambar sequence diagram yang telah dibuat, sequence diagram merupakan bagian dari UML oleh karena itu sequence dibuat untuk melengkapi UML yang ada. Pada sequence diagram ini terdapat aktor dan alur-alur yang menjelaskan lebih detail mengenai usecase diagram. Pada diagram ini menggambarkan alur apa saja yang dapat dilakukan oleh user. Alur yang digambarkan pada gambar 3.4 adalah alur mengenai hubungan antara user dan proses yang terjadi pada sistem. User dapat mengirimkan suatu inputan kemudian diproses pada sistem, selanjutnya sistem akan memberikan umpan balik atau respon dari setiap proses yang dilakukan oleh user tersebut. Namun, proses atau respon yang terjadi hanya dapat diakses oleh level tertentu saja

sesuai yang telah ditentukan. Gambar 3.4 untuk lebih jelasnya dapat dilihat di LAMPIRAN B.

4. Class diagram



Gambar 3. 5 Class Diagram

Guna class diagram ini adalah untuk membuat sketsa awal dari database yang akan digunakan. class diagram yang telah dibuat disajikan pada gambar 3.5. Pada rancangan class diagram yang sudah di buat terdapat 8 tabel, yaitu tabel level, user, pmo, pasien, obat, jadwal, detail_jadwal dan konfirmasi. Setiap tabel yang sudah dirancang memiliki relasi satu sama lain. Tabel yang paling penting pada sistem ini yaitu tabel pmo, pasien, obat, jadwal, detail_jadwal dan konfirmasi. Tabel-tabel tersebut saling memiliki relasi sehingga nantinya data yang masuk pada tabel tersebut akan saling terkait. Dengan adanya keterkaitan antar tabel tersebut akan memudahkan dalam pengolahan data nantinya.

5. Perancangan API

Pada pembuatan website ini nantinya akan dibuat juga API-nya menggunakan Restful API. Restful API digunakan karena tak perlu melakukan coding yang panjang untuk menggunakannya. REST menggunakan JSON sebagai bentuk datanya sehingga lebih ringan dan performa aplikasinya akan menjadi lebih baik. berikut ini merupakan perancangan API yang akan digunakan pada pembuatan website ini. Seluruh tabel yang sebelumnya dirancang akan masuk kedalam API tersebut. Sebelumnya akan dibuat perancangan *endpoint* yang diperlukan untuk website ini.

a) Pengolahan data user

Tabel 3. 1 Endpoint user

HTTP Method	URL	Input	Output	Keterangan
GET	/user	-	List seluruh object user	Untuk melihat seluruh data user
GET	/user{id}	-	Sebuah object dengan id tersebut	Untuk melihat data user dengan id tertentu
POST	/user	Body berisi user json object	Sebuah object dengan id	Untuk menambahkan data user baru
PUT	/user/{id}	Body berisi user json object	Sebuah object dengan id tersebut	Untuk mengubah data user yang ada
DELETE	/user/{id}	-	List object user	Untuk menghapus data user yang ada

b) Pengolahan data pasien

Tabel 3. 2 Endpoin pasien

HTTP Method	URL	Input	Output	Keterangan
GET	/pasien	-	List seluruh object pasien	Untuk melihat seluruh data pasien
GET	/pasien{id}	-	Sebuah json object pasien dengan id tersebut	Untuk melihat data pasien dengan id tertentu
POST	/pasien	Body berisi pasien json object	Sebuah json object pasien dengan id	Untuk menambahkan data pasien baru
PUT	/pasien/{id}	Body berisi pasien json object	Sebuah json object pasien dengan id tersebut	Untuk mengubah data pasien yang ada
DELETE	/pasien/{id}	-	List seluruh object pasien	Untuk menghapus data pasien yang ada

c) Pengolahan data PMO

Tabel 3. 3 Endpoint pmo

HTTP Method	URL	Input	Output	Keterangan
GET	/pmo	-	List seluruh object pmo	Untuk melihat seluruh data PMO

Tabel 3.3 Endpoint pmo(Lanjutan)

HTTP Method	URL	Input	Output		Keterangan
GET	/pmo{id}	-	Sebuah object dengan tersebut	json pmo id	Untuk melihat data PMO dengan id tertentu
POST	/pmo	Body berisi pmo json object	Sebuah object dengan id	json pmo	Untuk menambahkan data PMO baru
PUT	/pmo/{id}	Body berisi pmo json object	Sebuah object dengan tersebut	json pmo id	Untuk mengubah data PMO yang ada
DELETE	/pmo/{id}	-	List object pmo	seluruh pmo	Untuk menghapus data PMO yang ada

d) Pengolahan data obat

Tabel 3. 4 Endpoint obat

HTTP Method	URL	Input	Output		Keterangan
GET	/obat	-	List object obat	seluruh obat	Untuk melihat seluruh data obat
GET	/obat {id}	-	Sebuah object dengan tersebut	json obat id	Untuk melihat data obat dengan id tertentu

Tabel 3.4 Endpoint obat(Lanjutan)

HTTP Method	URL	Input	Output	Keterangan
POST	/obat	Body berisi obat json object	Sebuah object dengan id	Untuk menambahkan data obat baru
PUT	/obat/{id}	Body berisi obat json object	Sebuah object dengan id tersebut	Untuk mengubah data obat yang ada
DELETE	/obat/{id}	-	List object obat	Untuk menghapus data obat yang ada

e) Pengolahan data jadwal

Tabel 3. 5 Endpoint jadwal

HTTP Method	URL	Input	Output	Keterangan
GET	/jadwal	-	List object jadwal	Untuk melihat seluruh data jadwal
GET	/jadwal/{id}	-	Sebuah object dengan id tersebut	Untuk melihat data jadwal dengan id tertentu
POST	/jadwal	Body berisi jadwal json object	Sebuah object dengan id	Untuk menambahkan data jadwal baru

Tabel 3.5 Endpoint jadwal(Lanjutan)

HTTP Method	URL	Input	Output	Keterangan
PUT	/jadwal /{id}	Body berisi jadwal json object	Sebuah json object jadwal dengan id tersebut	Untuk mengubah data jadwal yang ada
DELETE	/jadwal /{id}	-	List seluruh object jadwal	Untuk menghapus data jadwal yang ada

f) Pengolahan data konfirmasi

Tabel 3. 6 Endpoint konfirmasi

HTTP Method	URL	Input	Output	Keterangan
GET	/konfirmasi	-	List seluruh object konfirmasi	Untuk melihat seluruh data konfirmasi
GET	/konfirmasi {id}	-	Sebuah json object konfirmasi dengan id tersebut	Untuk melihat data konfirmasi dengan id tertentu
POST	/konfirmasi	Body berisi konfirmasi json object	Sebuah json object konfirmasi dengan id	Untuk menambahkan data konfirmasi baru

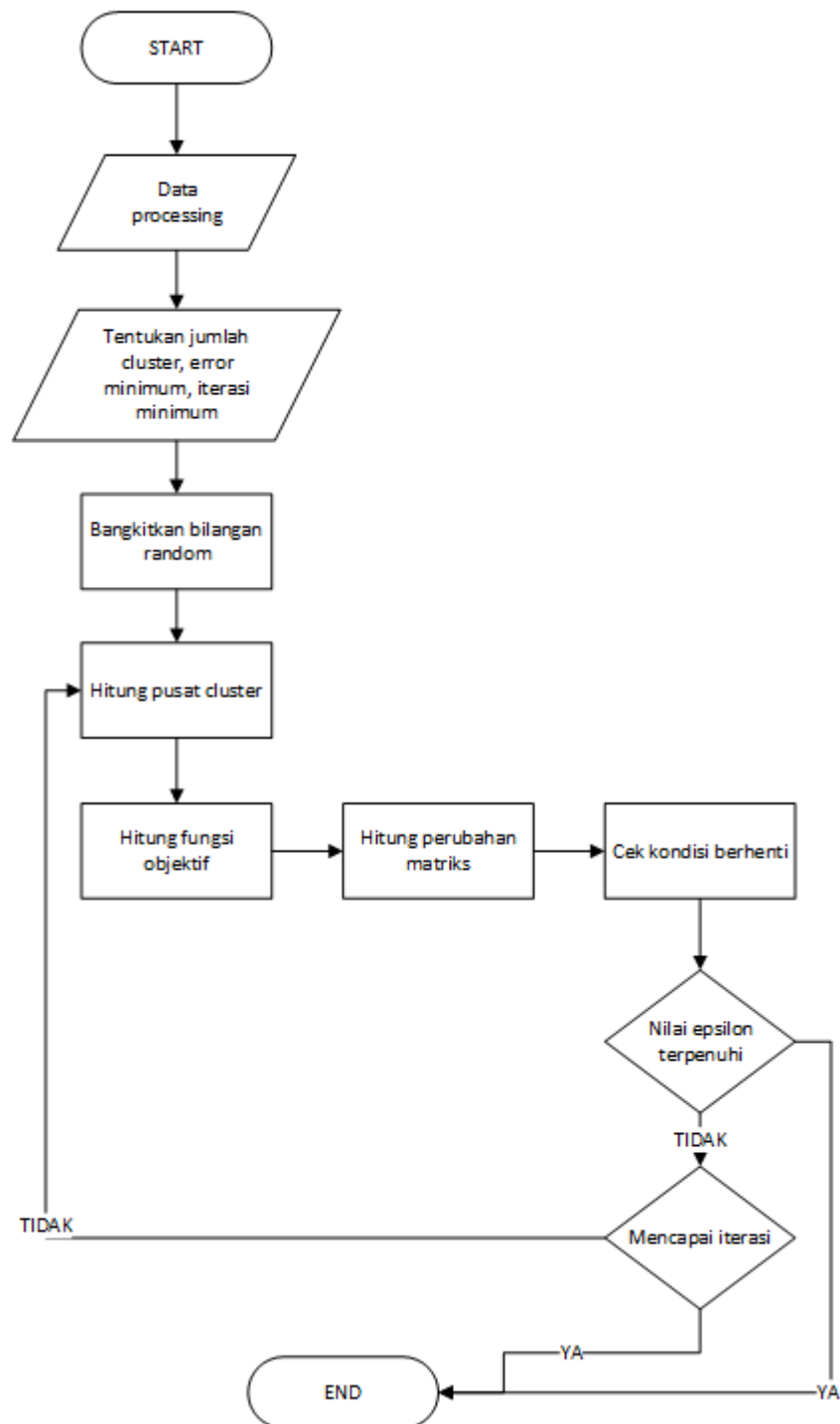
Tabel 3.6 Endpoint konfirmasi(Lanjutan)

HTTP Method	URL	Input	Output	Keterangan
PUT	/konfirmasi/{id}	Body berisi konfirmasi json object	Sebuah object konfirmasi dengan id tersebut	Untuk mengubah data konfirmasi yang ada
DELETE	/konfirmasi/{id}	-	List object konfirmasi	Untuk menghapus data konfirmasi yang ada

6. Perancangan model fuzzy logic

Selanjutnya adalah perancangan model fuzzy logic yang akan dibuat dengan menggunakan metode FCM (Fuzzy C-Means). Metode ini digunakan karena metode ini merupakan metode yang banyak digunakan di beberapa penelitian sebelumnya yang telah penulis temukan. Fuzzy C-Means merupakan suatu metode pengklasteran data yang ditentukan oleh derajat keanggotaan.

Gambar 3.9 merupakan flowchart metode FCM, awal dari proses pengklasteran adalah dengan menyiapkan data yang akan diklasterkan terlebih dahulu, data yang akan digunakan nantinya adalah data pencatatan tepat waktu, tidak tepat waktu, dan tidak respon dari setiap pasien yang telah terjadwalkan minum obat. Jika data sudah siap maka langkah selanjutnya adalah menentukan jumlah klaster, seperti pada kasus ini klaster akan dibagi menjadi 2 yaitu disiplin dan tidak disiplin. Lalu menentukan maksimum iterasi, pada penelitian ini menggunakan maksimum iterasi 100. Lalu menentukan error terkecil yang diharapkan, untuk saat ini error terkecil yang diharapkan yaitu 0.00001. setelah melakukan penentuan hal-hal tersebut maka kan dibangkitkan matriks random sebagai elemen-elemen matriks partisi awal U.



Gambar 3. 6 Flowchart Metode FCM

Kemudian menentukan titik pusat cluster, hitung fungsi objektif, hitung perubahan matriks partisi menggunakan rumus yang telah disediakan oleh metode FCM. Setelah proses penghitungan selesai maka di cek untuk nilai epsilon-nya apakah sudah memenuhi, jika belum akan di cek kembali untuk

maksimum iterasinya. Ketika maksimum iterasi masih belum tercapai maka akan kembali lagi ke proses awal yaitu penentuan pusat cluster. Semua alur tersebut di jalankan hingga mendapatkan nilai epsilon maupun maksimum iterasi yang sudah ditentukan.

Data training yang digunakan sebanyak 27 data, setiap satu data terdapat rincian nama, jumlah tepat waktu, jumlah tidak tepat waktu, dan jumlah tidak respon. Pasien dikatakan tidak tepat waktu apabila pasien meng-konfirmasi pada notifikasi dalam kurun waktu 0 sampai 30 menit dari awal notifikasi muncul. Lalu pasien dikatakan tidak tepat waktu apabila pasien meng-konfirmasi notifikasi dalam kurun waktu 31 menit sampai 1 jam dari awal notifikasi muncul. Kemudian untuk pasien yang tidak respon adalah ketika pasien sama sekali tidak menekan tombol konfirmasi pada notifikasi. Berikut merupakan bentuk data latih yang digunakan.

Tabel 3. 7 Data Latih

Nama	Tepat Waktu	Tidak Tepat Waktu	Tidak Respon
A	10	0	0
B	5	5	0
C	3	7	0
D	0	0	10
E	9	1	0
F	8	1	1
G	0	10	0
H	4	3	3
I	0	3	7
J	10	0	0
K	7	3	0
L	1	0	9
M	2	5	3
N	10	0	0
O	0	0	10
P	0	10	0
Q	3	7	0

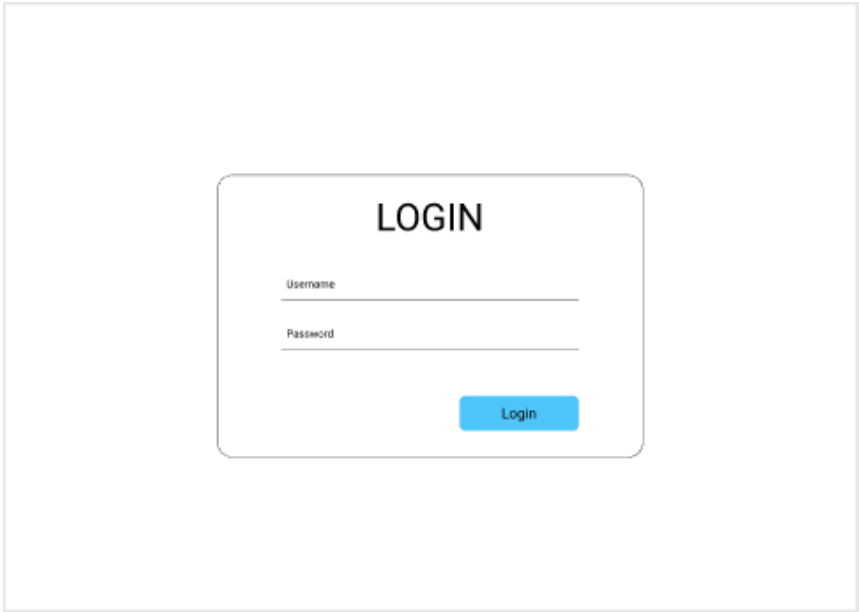
Tabel 3.7 Data Latih(Lanjutan)

Nama	Tepat Waktu	Tidak Tepat Waktu	Tidak Respon
R	10	0	0
S	10	0	0
T	5	5	0
U	4	2	4
V	2	8	0
W	9	1	0
X	8	1	1
Y	4	6	0
Z	10	0	0

7. Mockup tampilan website

Berikut ini merupakan mockup tampilan website, mockup yang ditampilkan adalah rancangan tampilan yang akan digunakan saat pembuatan website.

a) Halaman login

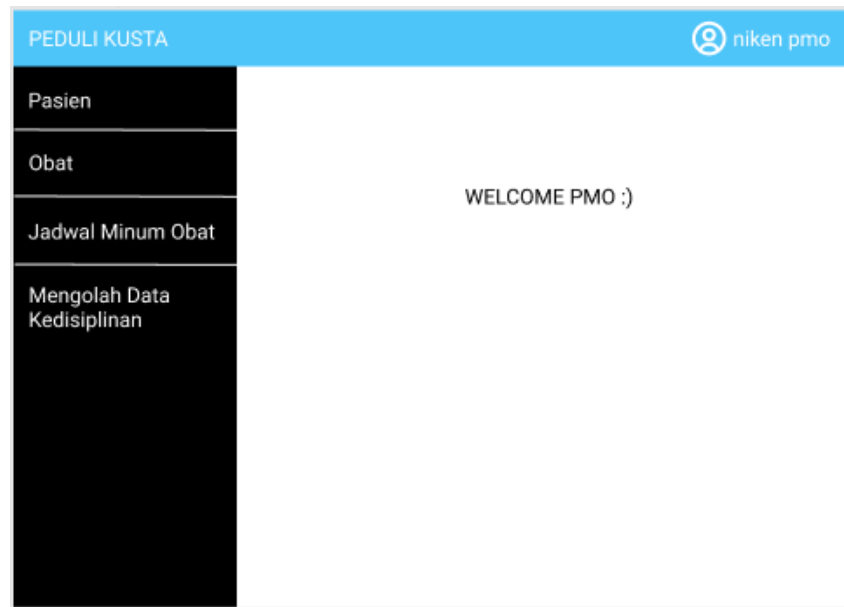


Gambar 3. 7 Mockup Halaman Login

Gambar 3.10 merupakan mockup halaman login, pada halaman ini dapat diakses oleh PMO maupun pasien. Sebelum login PMO dan pasien harus mengisi username dan password yang benar agar dapat mengakses halaman selanjutnya.

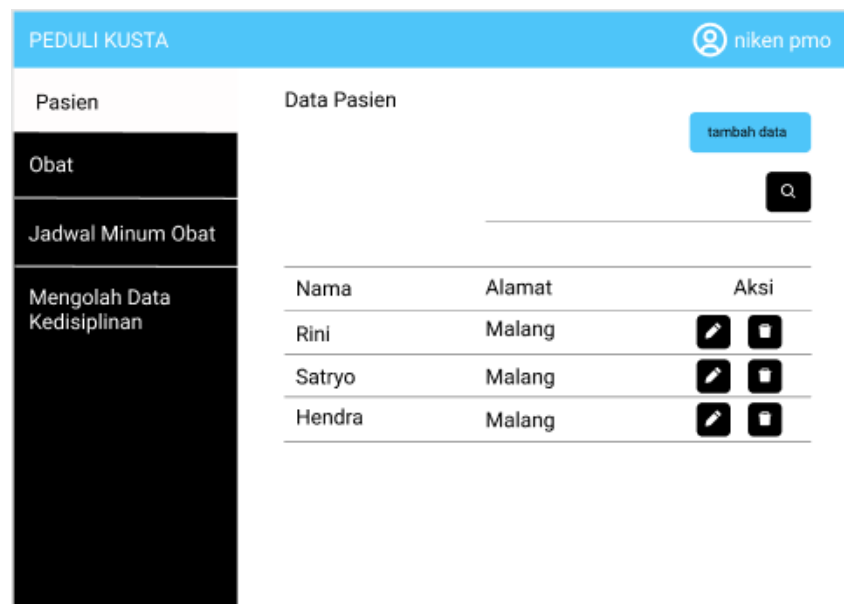
b) Dashboard PMO

Setelah melakukan proses login, PMO dapat mengakses halaman dashboard. Pada halaman ini PMO dapat mengakses menu yang disediakan yaitu menu pasien, obat, jadwal minum obat, dan mengolah data kedisiplinan pasien dalam meminum obat. Berikut ini merupakan gambar mockup dashboard PMO yang disediakan pada gambar 3.11.



Gambar 3. 8 Mockup Dashboard PMO

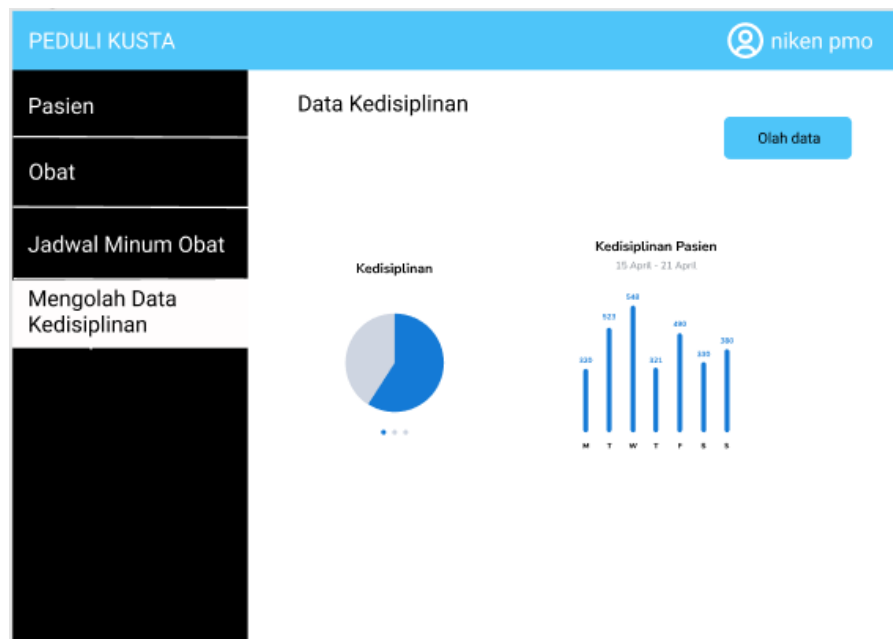
c) Tampilan garis besar melihat data



Gambar 3. 9 Mockup Melihat Data

Selanjutnya yaitu mockup dari isi setiap menu, gambar mockup yang disediakan pada gambar 3.12 merupakan rangkuman dari setiap menu yang ada karena tampilannya sama yang membedakan hanyalah isi datanya saja. Pada halaman ini PMO dapat melakukan proses *create*, *read*, *update*, dan *delete* data.

d) Tampilan menu mengolah data kedisiplinan

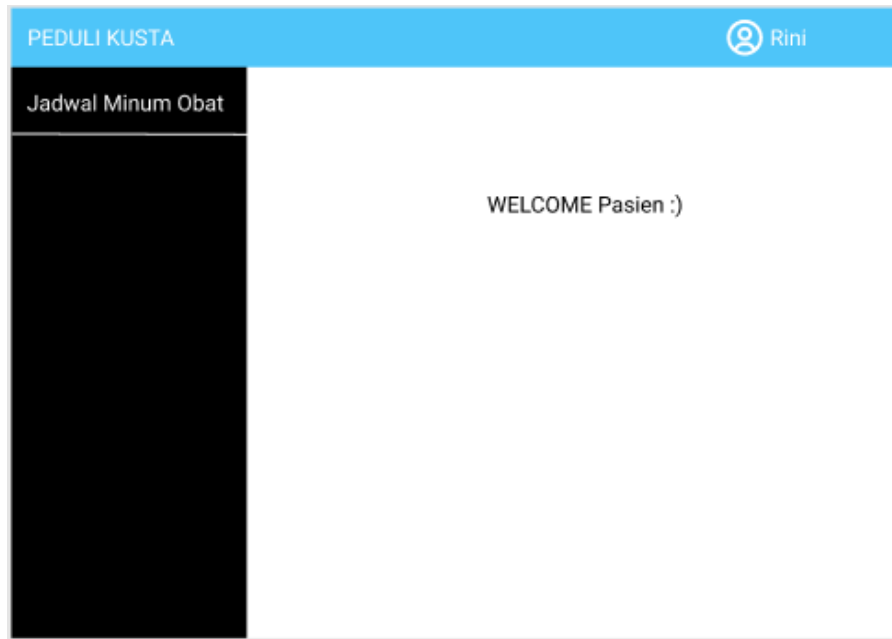


Gambar 3. 10 Mockup Menu Mengolah Data Kedisiplinan

Gambar 3.13 merupakan mockup dari menu mengolah data kedisiplinan. Pada halaman tersebut disajikan grafik dan juga *button* untuk mengolah data pasien yang sudah terekam dalam database. Grafik yang digunakan akan memudahkan pmo dalam melihat data-data yang dibutuhkan saja oleh pmo.

e) Dashboard pasien

Dashboard pasien merupakan tampilan awal ketika pasien telah berhasil login. Pada gambar 3.14 merupakan dashboard untuk menampilkan menu yang dapat dilihat oleh pasien yaitu menu jadwal minum obat. pada dashboard ini hanya menampilkan tulisan “welcome pasien” untuk menyapa pasien yang login pada web tersebut.



Gambar 3. 11 Mockup Dashboard Pasien

f) Tampilan menu jadwal minum obat



Gambar 3. 12 Mockup Menu Jadwal Minum Obat

Gambar 3.15 merupakan mockup menu jadwal minum obat. Ketika pasien melihat jadwal minum obat maka yang akan muncul adalah data jadwal yang berisi tanggal & jam, dan obat yang harus diminum.

g) Tampilan notifikasi

Gambar 3.16 adalah gambar mockup ketika notifikasi minum obat muncul. Notifikasi akan terus tampil hingga pasien mengkonfirmasi bahwa dirinya sudah meminum obat sesuai dengan apa yang diarahkan oleh PMO. Jika pasien dalam kurun waktu tertentu belum mengkonfirmasi maka sistem akan mengirimkan notifikasi hingga waktu timeout.



Gambar 3. 13 Mockup Notifikasi

3.4 Pengujian Aplikasi

Pengujian dilakukan untuk menunjukkan apakah aplikasi yang telah dibuat sesuai dengan apa yang telah dimaksudkan dan dirancang, selain itu pengujian aplikasi ini juga dilakukan untuk menemukan ketidaksempurnaan program sebelum digunakan. Proses pengujian aplikasi memiliki dua tujuan, yang pertama untuk menunjukkan bahwa aplikasi yang telah dibuat sudah memenuhi persyaratan. Tujuan yang kedua yaitu untuk menemukan ketidaksempurnaan aplikasi seperti crash, perhitungan yang salah, dan korupsi data. Terdapat banyak sekali metode pengujian aplikasi yang dapat digunakan. Namun, metode pengujian aplikasi yang akan digunakan pada penelitian ini adalah metode pengujian *blackbox*.

Pengujian Blackbox testing merupakan metode pengujian aplikasi yang dilakukan untuk mengamati hasil input dan output pada software tanpa mengetahui struktur kode dari software yang akan diuji. Untuk melakukan pengujian aplikasi perlu adanya kasus untuk dapat menguji hasil input dan output pada aplikasi tersebut. Kasus yang digunakan untuk menguji aplikasi ini megacu pada rancangan diagram *usecase* yang telah dibuat. Berikut ini merupakan pemaparan kasus yang akan diujikan disajikan pada tabel 3.8.

Tabel 3. 8 Kasus yang diujikan

No	Test ID	Deskripsi Pengujian	Keterangan	Hasil yang diharapkan
1	A001	Login	Memasukkan username & password untuk mengakses web	Berhasil login dengan username dan password yang sudah didaftarkan
2	A002	Mengolah data PMO	Melakukan olah data seperti <i>create, read, update, delete</i> data PMO	Berhasil melakukan <i>create, read, update, delete</i> data PMO
3	A003	Mengolah data pasien	Melakukan olah data seperti <i>create, read, update, delete</i> data pasien	Berhasil melakukan <i>create, read, update, delete</i> data pasien
4	A004	Mengolah data obat	Melakukan olah data seperti <i>create, read, update, delete</i> data obat	Berhasil melakukan <i>create, read, update, delete</i> data obat
5	A005	Mengolah data jadwal	Melakukan olah data seperti <i>create, read, update, delete</i> data jadwal	Berhasil melakukan <i>create, read, update, delete</i> data jadwal

Tabel 3.8 Kasus yang diujikan(Lanjutan)

No	Test ID	Deskripsi Pengujian	Keterangan	Hasil yang diharapkan
6	A006	Mengolah data kedisiplinan	Melakukan olah data kedisiplinan dari setiap pasien	Berhasil melakukan peng- <i>clusteran</i> data pasien yang meminum obat
7	A007	Menampilkan data jadwal	Pasien dapat melihat data jadwal yang sudah di berikan oleh PMO	Berhasil melihat data jadwal sesuai dengan data yang dijadwalkan PMO
8	A008	Menampilkan notifikasi	Notifikasi minum obat muncul sesuai dengan jadwal yang ditentukan	Notifikasi berhasil muncul sesuai dengan jadwal
9	A009	Konfirmasi notifikasi	Pasien dapat mengkonfirmasi bahwa sudah meminum obat melalui notifikasi	Berhasil melakukan konfirmasi melalui notifikasi

BAB IV

HASIL DAN PEMBAHASAN

4.1 Lingkungan Pengujian

Pengaturan lingkungan pengujian merupakan salah satu aspek dalam proses pengujian sebuah perangkat lunak. Oleh karena itu, pada proses pengujian ini ditentukanlah lingkungan pengujian yang akan dilakukan. Berikut ini merupakan lingkungan pengujian yang akan digunakan pada proses pengujian yang tersedia pada tabel .

Tabel 4. 1 Lingkungan Pengujian

Perangkat	Spesifikasi
Perangkat 1	Jenis Perangkat : Laptop Lenovo ideapad 110 RAM : 4 GB OS : Windows 10

Pengujian akan dilakukan pada perangkat yang sama dengan akses user sebagai super admin, PMO dan pasien. Proses ini dilakukan untuk menguji setiap fungsi yang ada pada *web* tersebut.

4.2 Skenario Pengujian

Proses pengujian yang dilakukan adalah untuk menguji fungsi dari setiap fitur yang disajikan pada *web* tersebut. Pengujian ini menggunakan metode *Blackbox testing* yang bertujuan untuk mengamati hasil *input* dan *output* pada *software* apakah sudah sesuai dengan apa yang telah diharapkan.

4.3 Skenario dan Hasil Uji Coba

Pada proses pengujian ini dilakukan beberapa skenario yang sudah dirancang sebelumnya untuk mengetahui fungsi setiap fitur yang diuji coba. Berikut ini adalah skenario dan hasil uji coba setiap fitur menggunakan metode uji coba *Blackbox testing*.

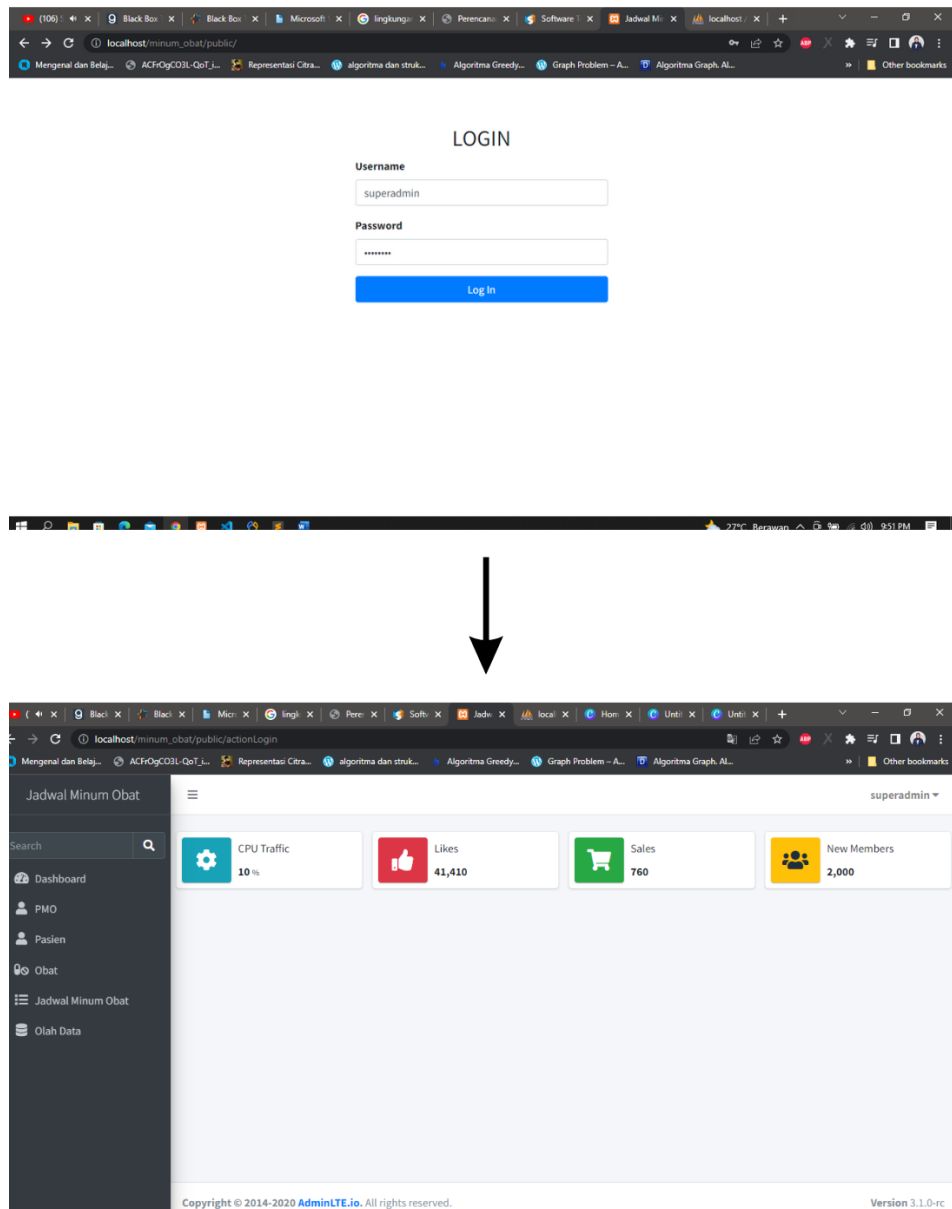
1. Pengujian Login

Fitur login dapat diakses oleh super admin, PMO, dan pasien. Namun, yang menjadi pembeda adalah level dari user tersebut untuk mengakses halaman

yang berbeda. Super admin memiliki kedudukan level tertinggi, di mana dapat mengolah data PMO, pasien, obat, jadwal, dan kedisiplinan. Sedangkan level PMO hanya dapat mengolah data pasien, obat, jadwal dan kedisiplinan. Level selanjutnya yaitu pasien yang hanya dapat mengakses jadwal minum obat saja.

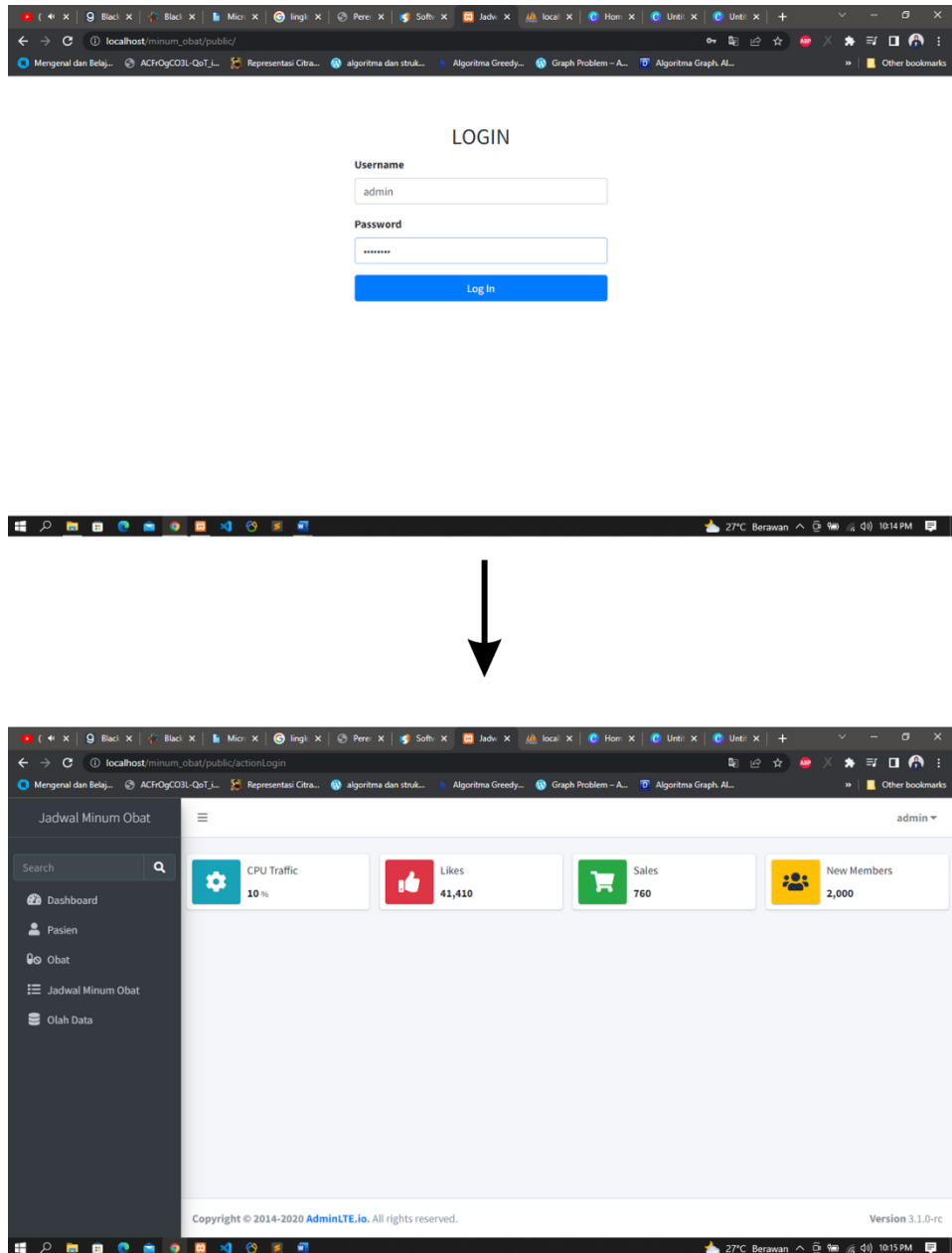
Tabel 4. 2 Pengujian Fitur Login

Kode	T001
Referensi kasus	A001
Deskripsi pengujian	Login
Tujuan	Menguji fungsi login pada setiap level <i>user</i>
Skenario 1 – Login Super Admin	
Pengguna	Super admin
Hasil yang diharapkan	<i>User</i> berhasil melakukan login dan <i>user</i> beralih ke <i>dashboard</i> super admin
Hasil yang didapat	Tampilan <i>web</i> beralih ke <i>dashboard</i> super admin
Hasil pengujian	Berhasil
Skenario 2 – Login PMO	
Pengguna	PMO
Hasil yang diharapkan	<i>User</i> berhasil melakukan login dan <i>user</i> beralih ke <i>dashboard</i> PMO
Hasil yang didapat	Tampilan <i>web</i> beralih ke <i>dashboard</i> PMO
Hasil pengujian	Berhasil
Skenario 3 – Login Pasien	
Pengguna	Pasien
Hasil yang diharapkan	<i>User</i> berhasil melakukan login dan <i>user</i> beralih ke <i>dashboard</i> Pasien
Hasil yang didapat	Tampilan <i>web</i> beralih ke <i>dashboard</i> Pasien
Hasil pengujian	Berhasil



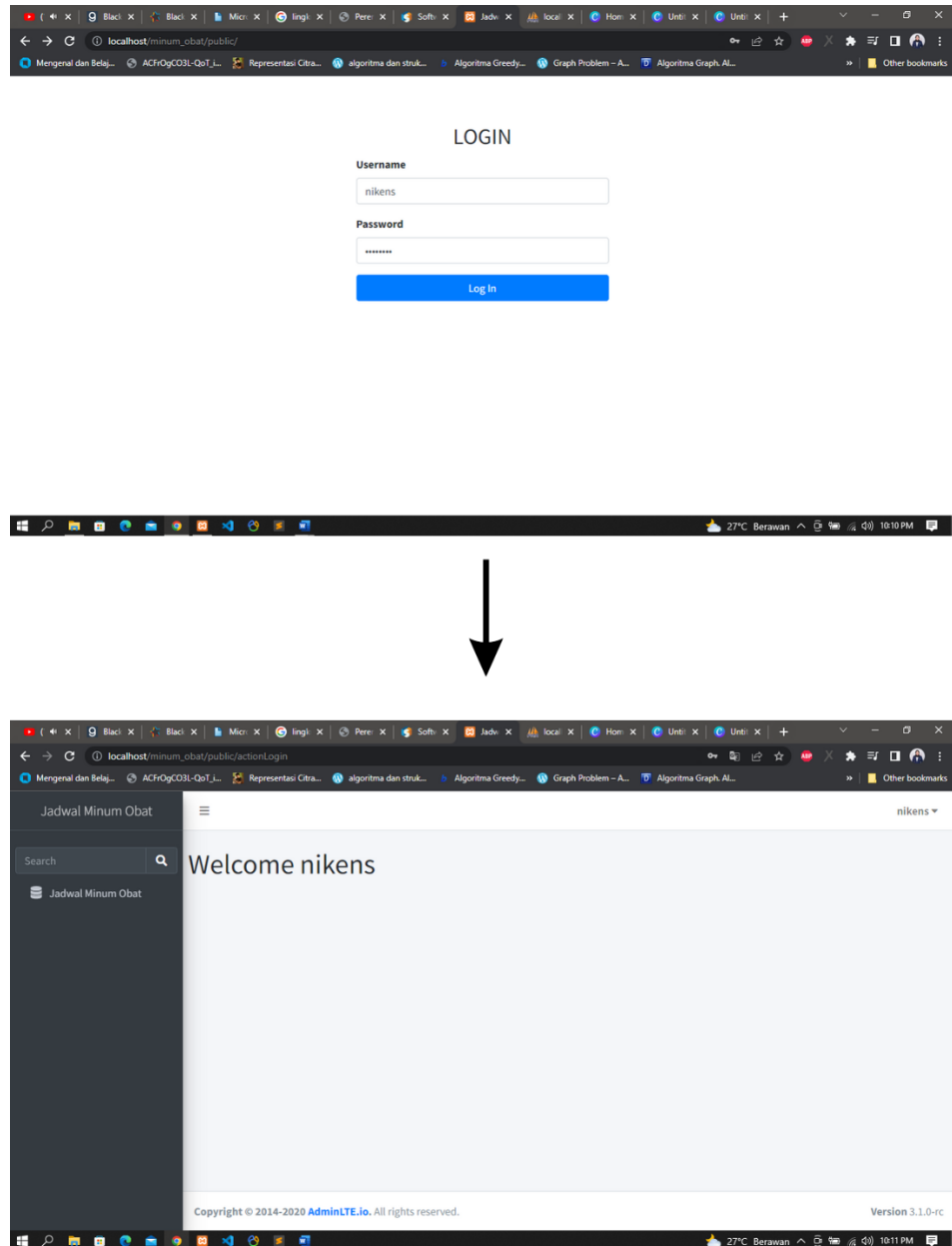
Gambar 4. 1 Pengujian T001 Skenario 1

Pada pengujian fitur login, yang perlu dilakukan adalah dengan memasukkan *username* dan *password* yang sudah terdaftar dalam *database* sebagai level super admin. Setelah memasukkan *username* dan *password*, lalu menekan *button* login, maka halaman akan dialihkan ke halaman *dashboard* untuk super admin. Mengacu pada gambar 4.1, pengujian fitur login dengan level *user* super admin telah berhasil dilakukan.



Gambar 4. 2 Pengujian T001 Skenario 2

Pada skenario 2 ini, pengujian dilakukan dengan memasukkan username dan *password* sesuai dengan *user* yang telah didaftarkan pada level PMO. Setelah memasukkan username dan password yang benar, maka halaman web akan dialihkan ke halaman dashboard untuk PMO. Pada gambar 4.2 pengujian telah dilakukan dan hasilnya adalah berhasil melakukan login dengan level *user* PMO.



Gambar 4. 3 Pengujian T001 Skenario 3

Saat skenario 3 dijalankan, yaitu dengan memasukkan *username* dan *password* yang telah didaftarkan sebagai level pasien, kemudian setelah memasukkan kedua inputan tersebut klik *button* login halaman login akan teralihkan ke halaman *dashboard* untuk level *user* pasien. Pada skenario pengujian yang telah di coba didapatkan hasil bahwa uji yang dilakukan berhasil untuk login dengan level *user* pasien.

2. Pengujian Mengolah Data PMO

Fitur mengolah data PMO memiliki beberapa bagian yaitu melihat data, menambah data, mengedit data dan menghapus data. Berikut ini adalah tabel pengujian fitur mengolah data yang disajikan pada tabel 4.3.

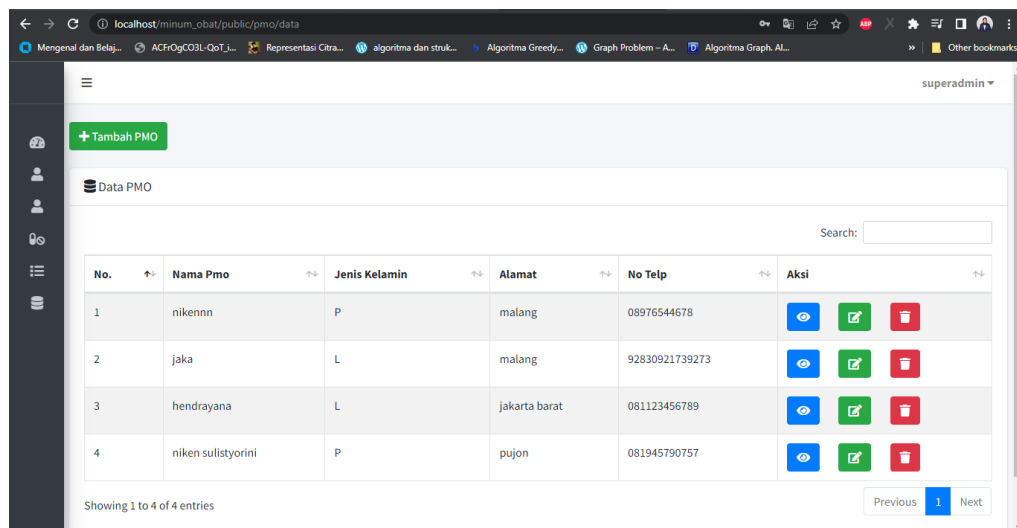
Tabel 4. 3 Pengujian Fitur Mengolah Data PMO

Kode	T002
Referensi kasus	A002
Deskripsi pengujian	Mengolah data PMO
Tujuan	Menguji fungsi lihat data, tambah data, <i>update</i> data dan hapus data
Skenario 1 – Melihat data PMO	
Pengguna	Super admin
Hasil yang diharapkan	Super admin dapat melihat data PMO yang sudah tersimpan pada <i>database</i>
Hasil yang didapat	Tampilan <i>list</i> data PMO yang disajikan pada sebuah tabel
Hasil pengujian	Berhasil
Skenario 2 – Menambah data PMO	
Pengguna	Super admin
Hasil yang diharapkan	Super admin dapat menambah data PMO kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data PMO tersimpan pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 3 – Mengupdate data PMO	
Pengguna	Super admin
Hasil yang diharapkan	Super admin dapat meng- <i>update</i> data PMO kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data PMO ter- <i>update</i> pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 4 – Menghapus data PMO	
Pengguna	Super admin
Hasil yang diharapkan	Super admin dapat menghapus data PMO

Tabel 4.3 Pengujian Fitur Mengolah Data PMO(Lanjutan)

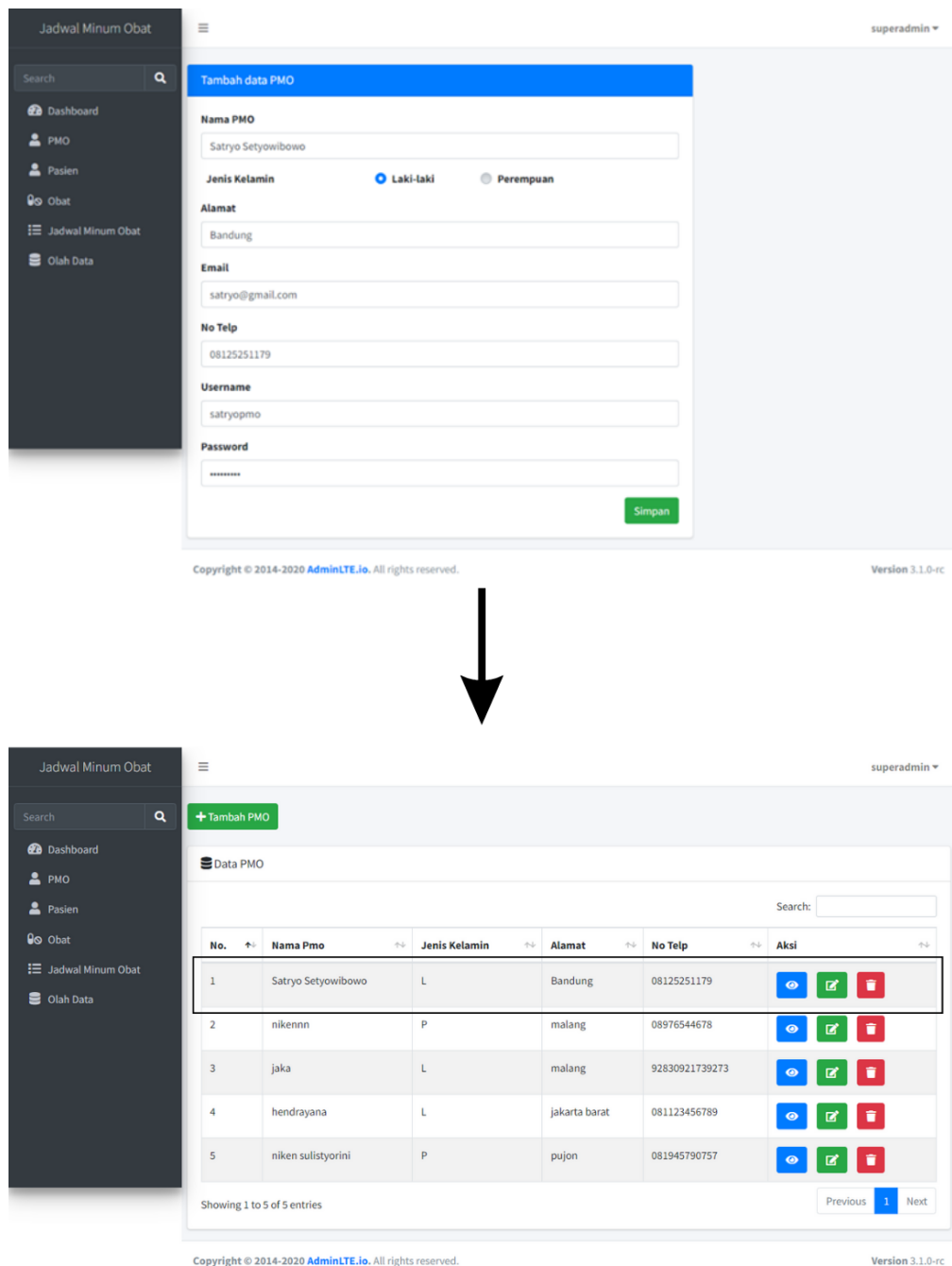
Hasil yang didapat	Data PMO terhapus dari list daftar PMO
Hasil pengujian	Berhasil

Gambar 4.4 menunjukkan pengujian T002 Skenario 1 yang mana pengujian tersebut sudah berhasil menampilkan daftar PMO yang ada pada *database*. Super admin dapat melihat daftar nama PMO kemudian terdapat button untuk menambahkan data PMO. Superadmin juga dapat melakukan aksi edit, lihat detail dan hapus yang sudah tersedia pada halaman tersebut.



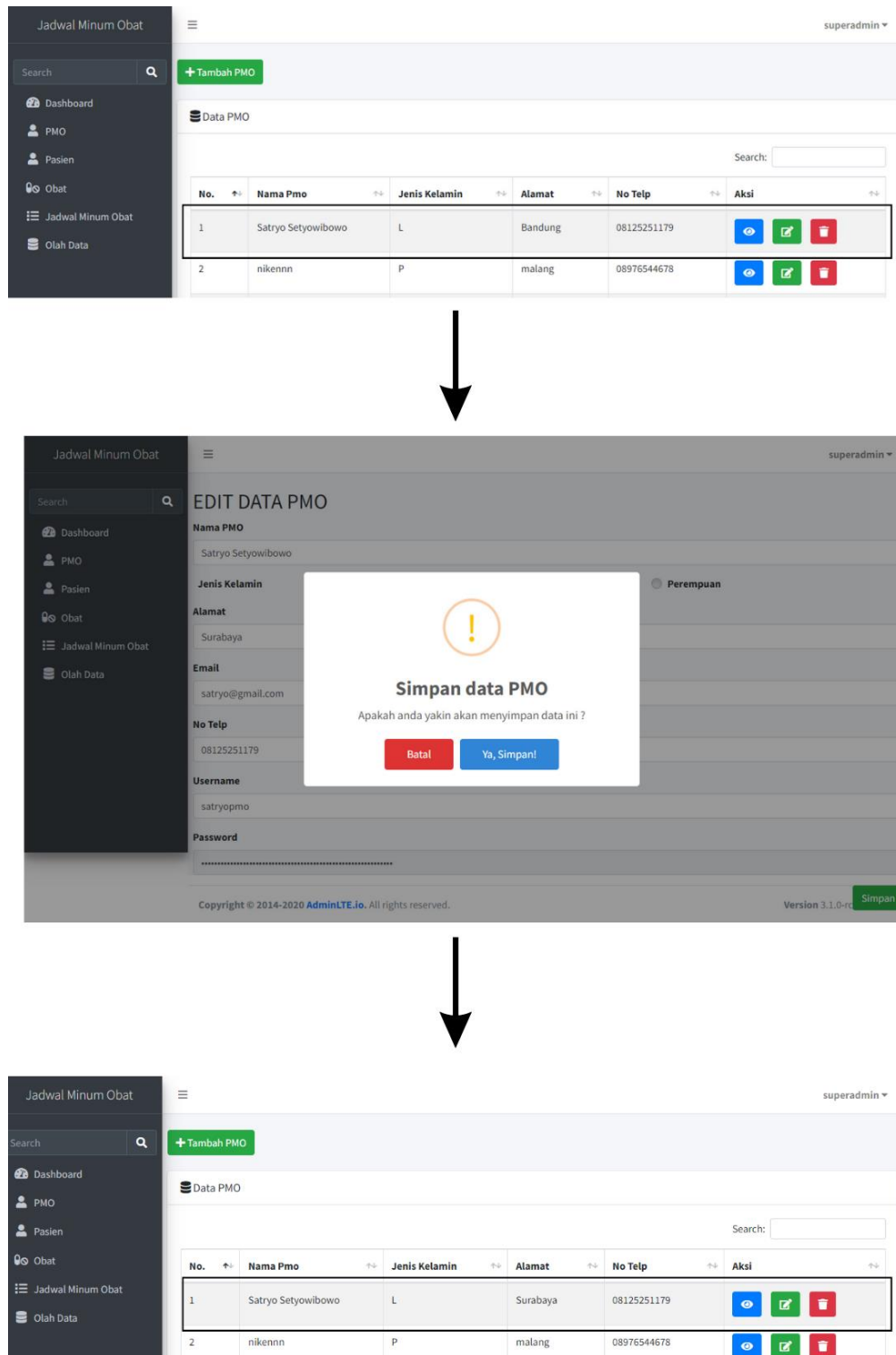
Gambar 4. 4 Pengujian T002 Skenario 1

Mengacu pada Pengujian T002 Skenario 2, bahwa pengujian tersebut dapat dikatakan berhasil. Bukti Pengujian T002 Skenario 2 berhasil dapat dilihat pada gambar 4.5. Pada gambar tersebut super admin dapat melakukan tambah PMO dengan mengisi data diri dari PMO yang terdiri dari nama, jenis kelamin, alamat, email, nomor telepon, *username*, dan juga *password*. Jika seluruh data diri dari PMO yang diisikan sudah lengkap, maka super admin dapat menekan simpan untuk dapat menyimpan data PMO ke dalam database. Seperti yang sudah di tampilkan pada gambar 4.5 bahwa penambahan data PMO yang berhasil masuk pada *database* akan tampil pada daftar PMO yang telah didaftarkan oleh super admin.



Gambar 4. 5 Pengujian T002 Skenario 2

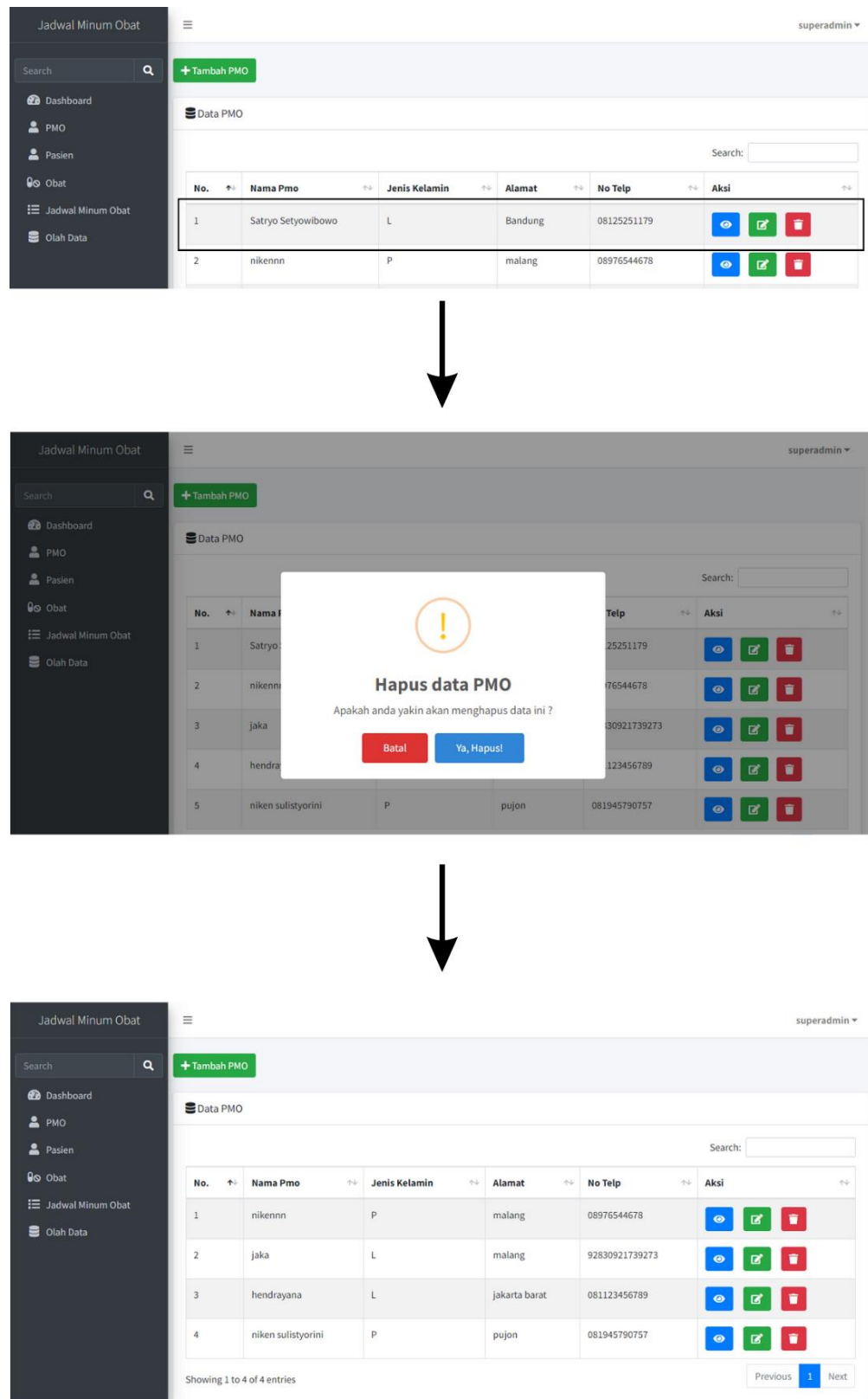
Pada pengujian selanjutnya yaitu Pengujian T002 Skenario 3, di mana pada skenario tersebut super admin dapat meng-edit atau meng-*update* data PMO jika ada yang salah atau mengalami perubahan data. Kesalahan peng-*input*-an dan *update* data diri yang menjadi faktor fitur edit data ini dibuat. Berikut ini merupakan gambar berhasilnya fitur edit data saat dilakukan percobaan untuk mengedit data, pengujian tersebut dapat dilihat pada gambar 4.6.



Gambar 4. 6 Pengujian T002 Skenario 3

Pengujian T002 Skenario 4 merupakan pengujian fitur hapus data PMO yang telah dibuat atau telah terdaftar pada database sebelumnya. Fitur ini dibuat

untuk menghapus data PMO yang mungkin sudah Pengujian fitur tersebut berhasil dan dapat dilihat pada gambar 4.7.



Gambar 4. 7 Pengujian T002 Skenario 4

3. Pengujian Mengolah Data Pasien

Pada fitur ini juga terdapat beberapa bagian yang perlu di uji, yaitu melihat daftar pasien, menambahkan data pasien, meng-update data pasien, dan menghapus data pasien. Berikut ini adalah skenario pengujian fitur mengolah data pasien pada tabel 4.4.

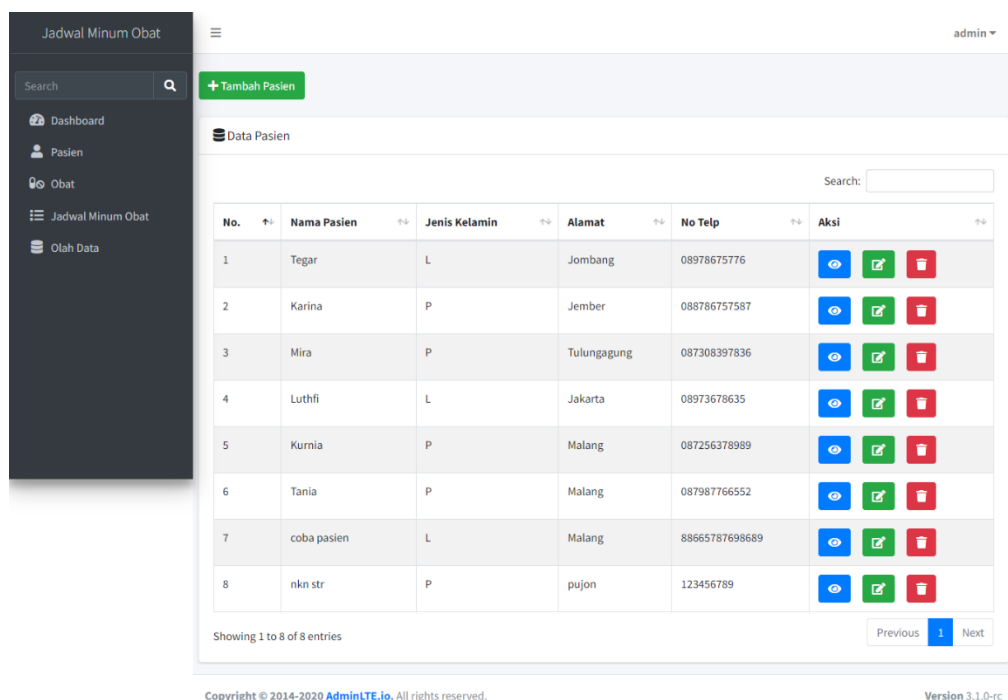
Tabel 4. 4 Pengujian Fitur Mengolah Data Pasien

Kode	T003
Referensi kasus	A003
Deskripsi pengujian	Mengolah data pasien
Tujuan	Menguji fungsi lihat data, tambah data, <i>update</i> data dan hapus data
Skenario 1 – Melihat data pasien	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat melihat data pasien yang sudah tersimpan pada <i>database</i>
Hasil yang didapat	Tampilan <i>list</i> data pasien yang disajikan pada sebuah tabel
Hasil pengujian	Berhasil
Skenario 2 – Menambah data pasien	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat menambah data pasien kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data pasien tersimpan pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 3 – Mengupdate data pasien	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat meng- <i>update</i> data pasien kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data pasien ter- <i>update</i> pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 4 – Menghapus data pasien	
Pengguna	Super admin dan PMO

Tabel 4.4 Pengujian Fitur Mengolah Data Pasien (Lanjutan)

Hasil yang diharapkan	User dapat menghapus data pasien
Hasil yang didapat	Data pasien terhapus dari list daftar pasien
Hasil pengujian	Berhasil

Pengujian yang dilakukan selanjutnya yaitu pengujian dengan kode T003. Pada pengujian ini dilakukan 4 skenario pengujian untuk menguji segala fitur yang ada pada menu pasien yang telah dibuat. Pengujian yang pertama yaitu Pengujian T001 Skenario 1. Hasil yang diharapkan pada skenario ini adalah super admin dan PMO dapat melihat data yang telah terdaftar pada database. Namun, kedua level user tersebut memiliki perbedaan dalam penyajian datanya. Level super admin dapat melihat seluruh data PMO yang telah terdaftar di database, sedangkan PMO hanya dapat melihat data pasien yang telah diinputkan oleh PMO itu sendiri. Gambar 4.8 di bawah ini merupakan bukti berhasilnya Pengujian T003 Skenario 1.



Gambar 4. 8 Pengujian T003 Skenario 1

Selanjutnya adalah Pengujian T003 Skenario 2, di mana super admin dan PMO dapat melakukan tambah data pasien. Super admin dan PMO dapat menambahkan data pasien dengan mengisi data diri dari pasien. Jika data diri

tersebut sudah benar maka dapat disimpan dengan menekan button simpan. Data pasien yang sudah diinputkan akan otomatis masuk kedalam database dan akan ditampilkan pada fitur lihat data pasien. Berhasilnya fitur tersebut dapat dilihat pada gambar 4.9.

The screenshot shows the 'Tambah data Pasien' form with the following data entered:

- Nama Pasien: Raya
- Jenis Kelamin: Perempuan (selected)
- Tanggal Lahir: 02/21/1979
- Alamat Pasien: Tulungagung
- Email Pasien: raya@gmail.com
- No Telp Pasien: 08976567897
- Riwayat Kesehatan Pasien: Demam, Pusing
- Username: rayapasien
- Password: (masked)

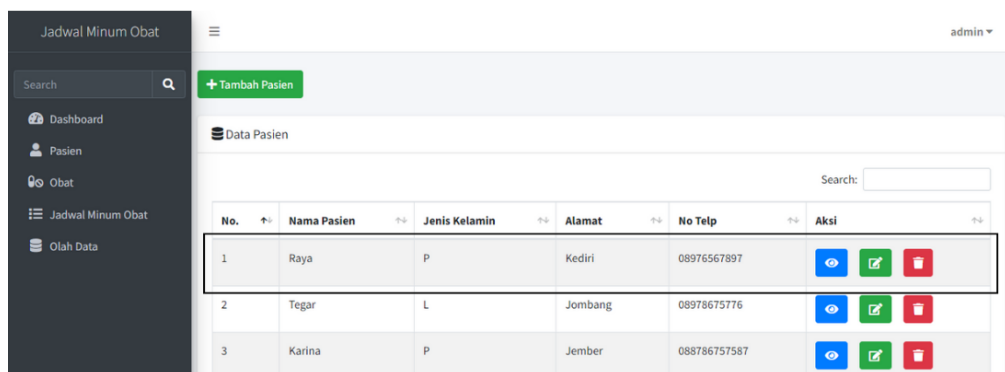
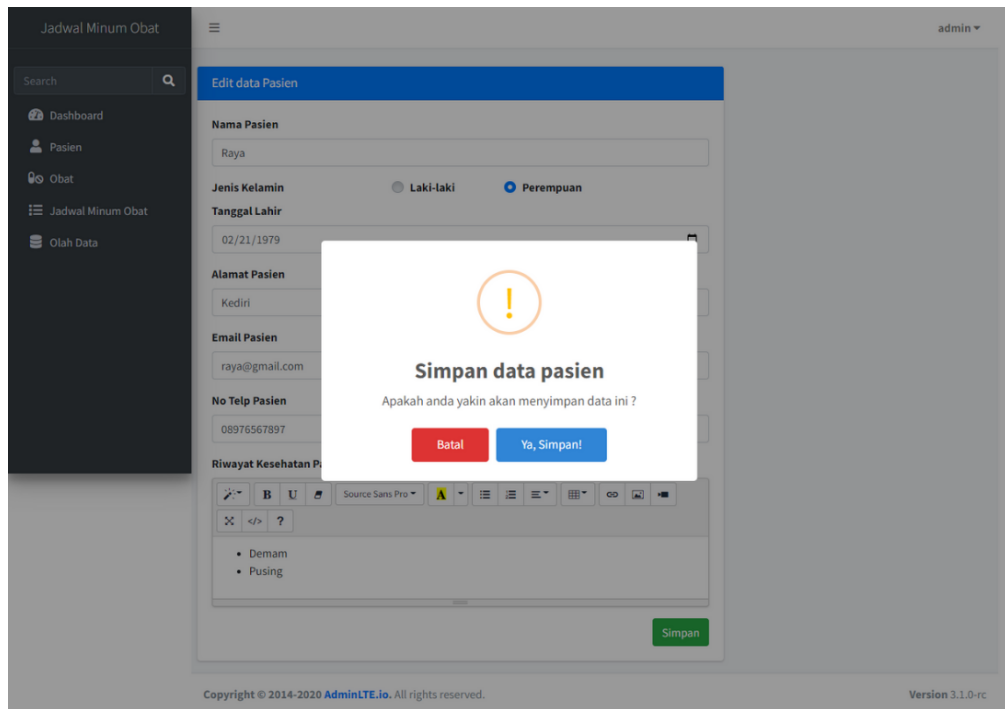
A green 'Simpan' button is located at the bottom right of the form.



The screenshot shows the 'Data Pasien' table with the following data:

No.	Nama Pasien	Jenis Kelamin	Alamat	No Telp	Aksi
1	Raya	P	Tulungagung	08976567897	View Edit Delete
2	Tegar	L	Jombang	08978675776	View Edit Delete
3	Karina	P	Jember	088786757587	View Edit Delete

Gambar 4. 9 Pengujian T003 Skenario 2

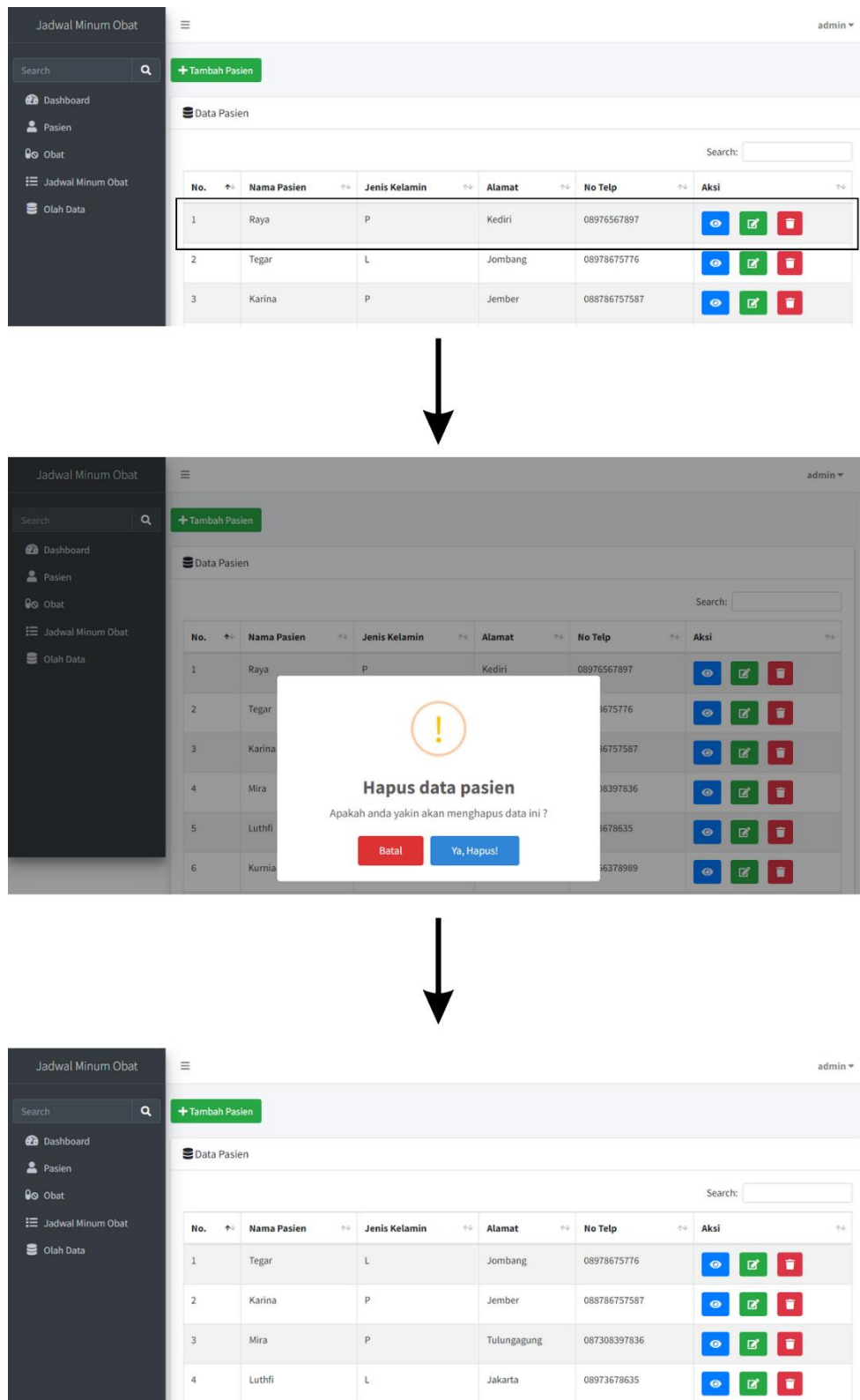


Gambar 4. 10 Pengujian T003 Skenario 3

Gambar 4.10 di atas merupakan gambar Pengujian T003 Skenario 3. Pada skenario ini PMO dapat melakukan edit atau meng-*update* data pasien. Super admin dan PMO dapat mengedit data pasien yang mengalami perubahan atau terjadi kesalahan saat penambahan data dilakukan. Pengujian ini telah dilakukan dan berhasil seperti apa yang telah diharapkan sebelumnya.

Pada gambar 4.11 merupakan hasil dari Pengujian T003 Skenario 4 yang telah berhasil dilakukan. Fitur yang diuji merupakan fitur hapus data, di mana fitur

tersebut dapat digunakan dengan baik, dan sesuai dengan apa yang diharapkan saat pengujian dilakukan.



Gambar 4. 11 Pengujian T003 Skenario 4

4. Pengujian Mengolah Data Obat

Pengujian fitur mengolah data obat ini memiliki bagian yang perlu diuji. Berikut ini adalah skenario pengujian yang akan di ujikan pada fitur ini yang tersaji pada tabel 4.5.

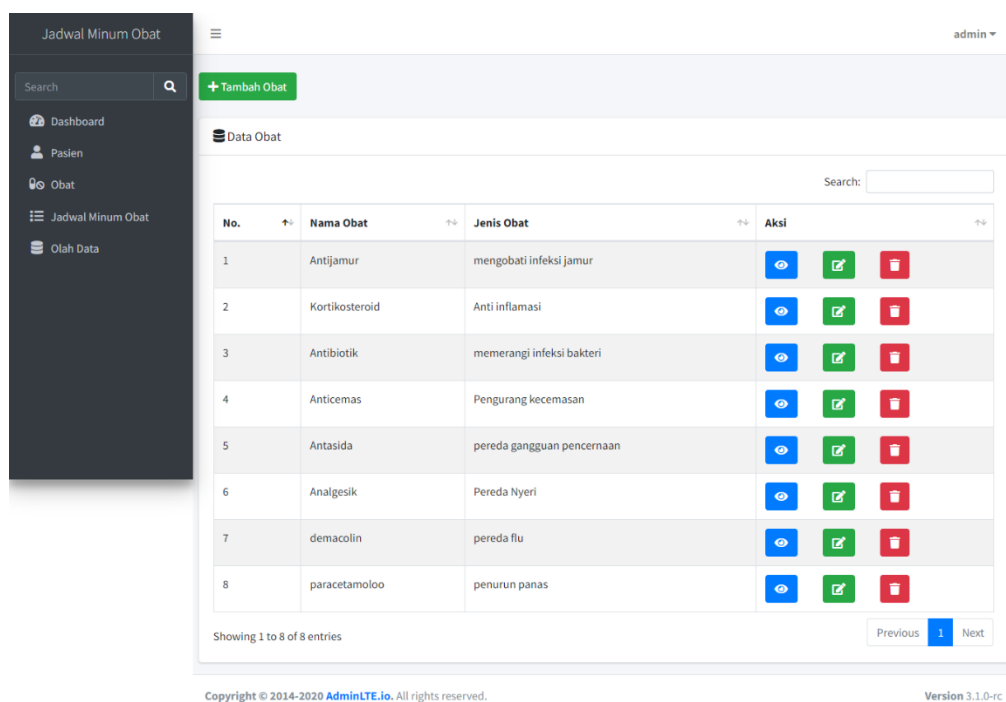
Tabel 4. 5 Pengujian Fitur Mengolah Data Obat

Kode	T004
Referensi kasus	A004
Deskripsi pengujian	Mengolah data obat
Tujuan	Menguji fungsi lihat data, tambah data, <i>update</i> data dan hapus data
Skenario 1 – Melihat data obat	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat melihat data pasien yang sudah tersimpan pada <i>database</i>
Hasil yang didapat	Tampilan <i>list</i> data obat yang disajikan pada sebuah tabel
Hasil pengujian	Berhasil
Skenario 2 – Menambah data obat	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat menambah data obat kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data pasien tersimpan pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 3 – Mengupdate data obat	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat meng- <i>update</i> data obat kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data obat ter- <i>update</i> pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 4 – Menghapus data obat	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat menghapus data obat

Tabel 4.5 Pengujian Fitur Mengolah Data Obat (Lanjutan)

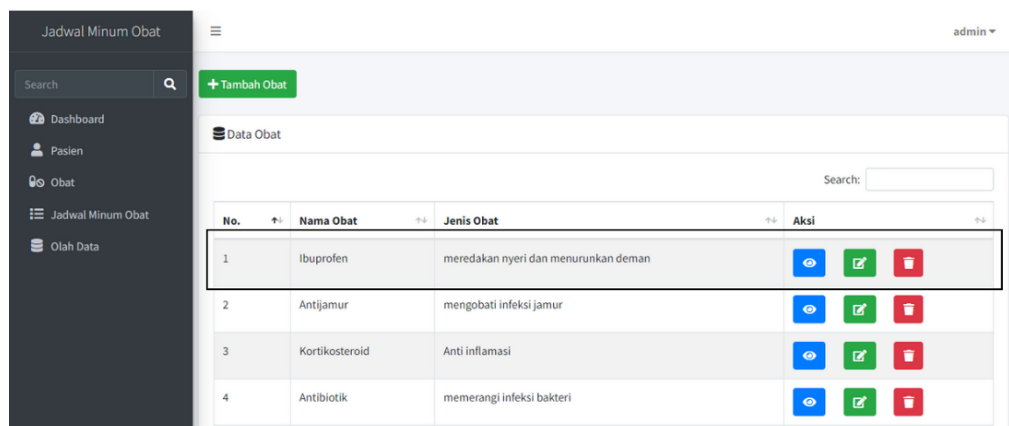
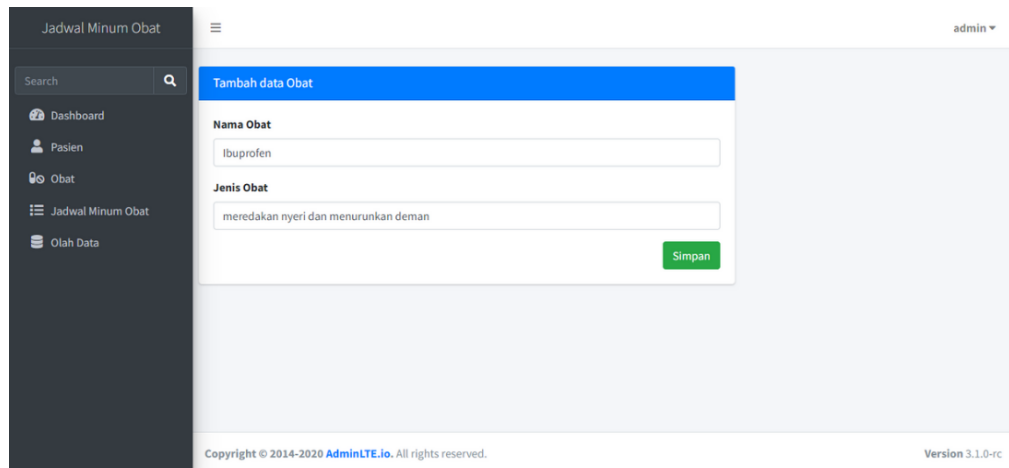
Hasil yang didapat	Data obat terhapus dari list daftar obat
Hasil pengujian	Berhasil

Hasil dari pengujian pada kode pengujian T004 Skenario 1 dapat dikatakan berhasil seperti gambar 4.12 yang telah disajikan di bawah ini. Pada pengujian tersebut level user yang dapat mengakses halaman *index* obat hanyalah super admin dan PMO saja. Super admin dan PMO dapat melihat data obat-obatan yang telah di-inputkan dan tersimpan dalam database master obat. halaman *index* obat ini juga menyediakan tombol aksi dari setiap obat yang ditampilkan.



Gambar 4. 12 Pengujian T004 Skenario 1

Pengujian selanjutnya yaitu pengujian T004 skenario 2, pada skenario ini super admin dan PMO dapat menambahkan data obat yang belum ada pada daftar obat. Setiap obat yang di-inputkan oleh super admin dan PMO akan tersimpan di database. Pengujian tersebut dapat dilihat pada gambar 4.13 dan pengujian ini dikatakan berhasil, karena data obat yang telah di inputkan dapat masuk kedalam database dan dapat ditampilkan pada index obat.

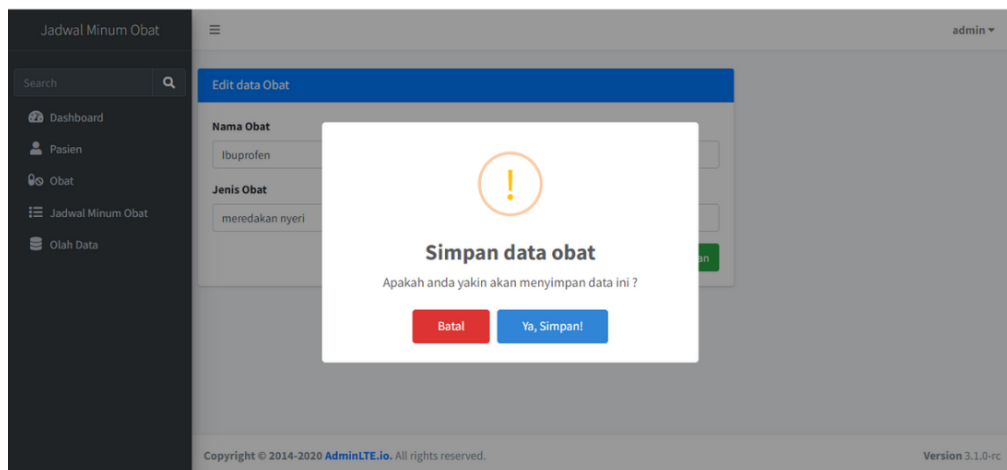
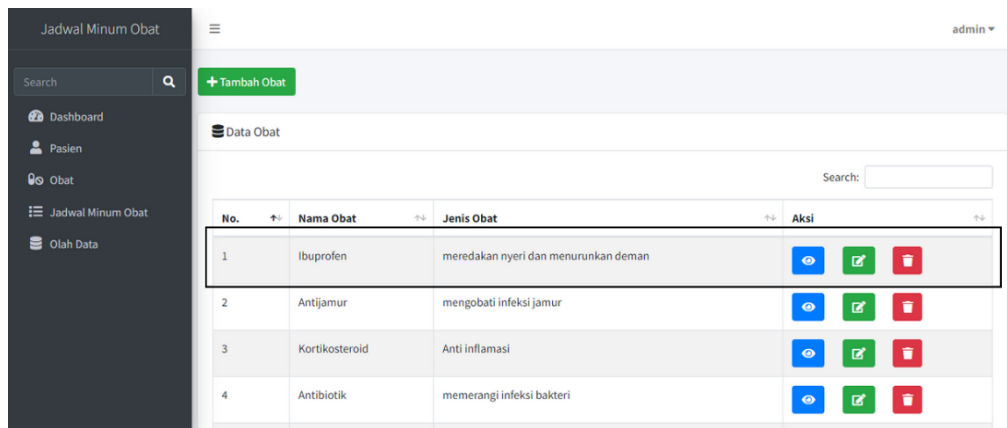


Gambar 4. 13 Pengujian T004 Skenario 2

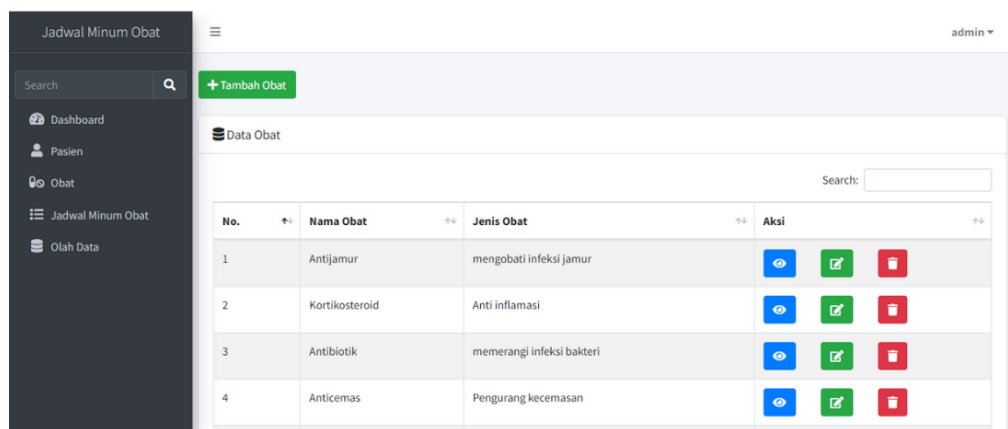
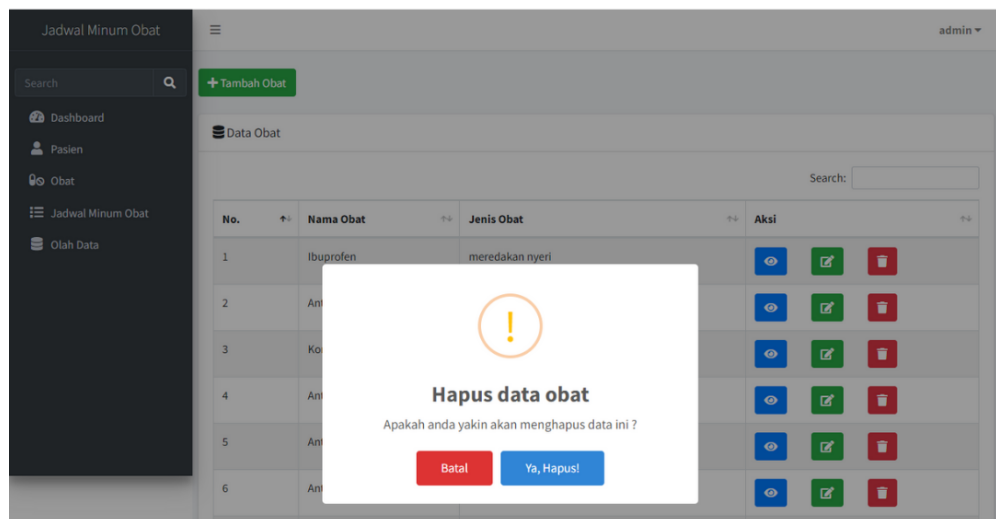
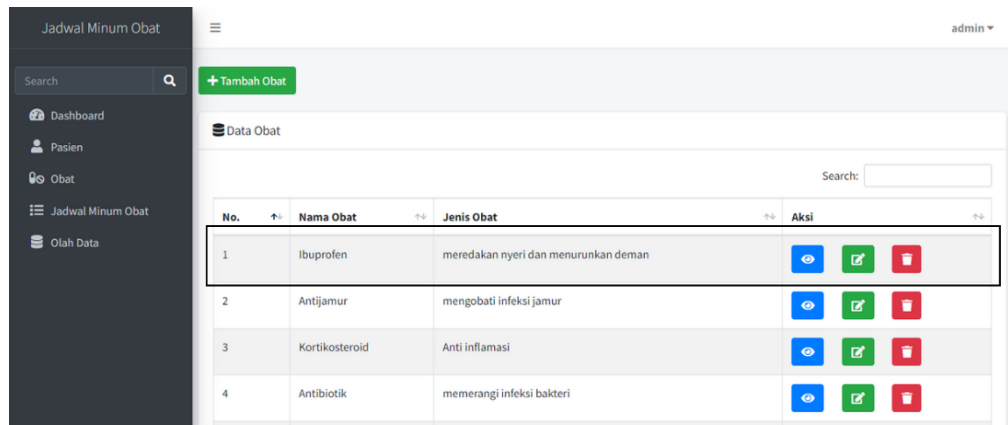
Selanjutnya dilakukan pengujian dengan skenario berbeda yaitu pengujian mengenai edit data obat. Pengujian tersebut diberikan kode Pengujian T004 Skenario 3 dan dapat dilihat pada gambar 4.14. Pada skenario ini pengguna yaitu super admin dan PMO dapat meng-*update* data obat yang telah tersimpan di *database*.

Pengujian T004 Skenario 5 merupakan pengujian fungsi tombol hapus data, di mana super admin dan PMO dapat menghapus data obat yang tidak digunakan lagi. Pada gambar 4.15 telah disajikan proses menghapus data obat

yang mana proses tersebut telah berhasil dijalankan, dan data yang dihapus juga akan menghilang dari daftar data obat.



Gambar 4. 14 Pengujian T004 Skenario 3



Gambar 4. 15 Pengujian T004 Skenario 4

5. Pengujian Mengolah Data Jadwal

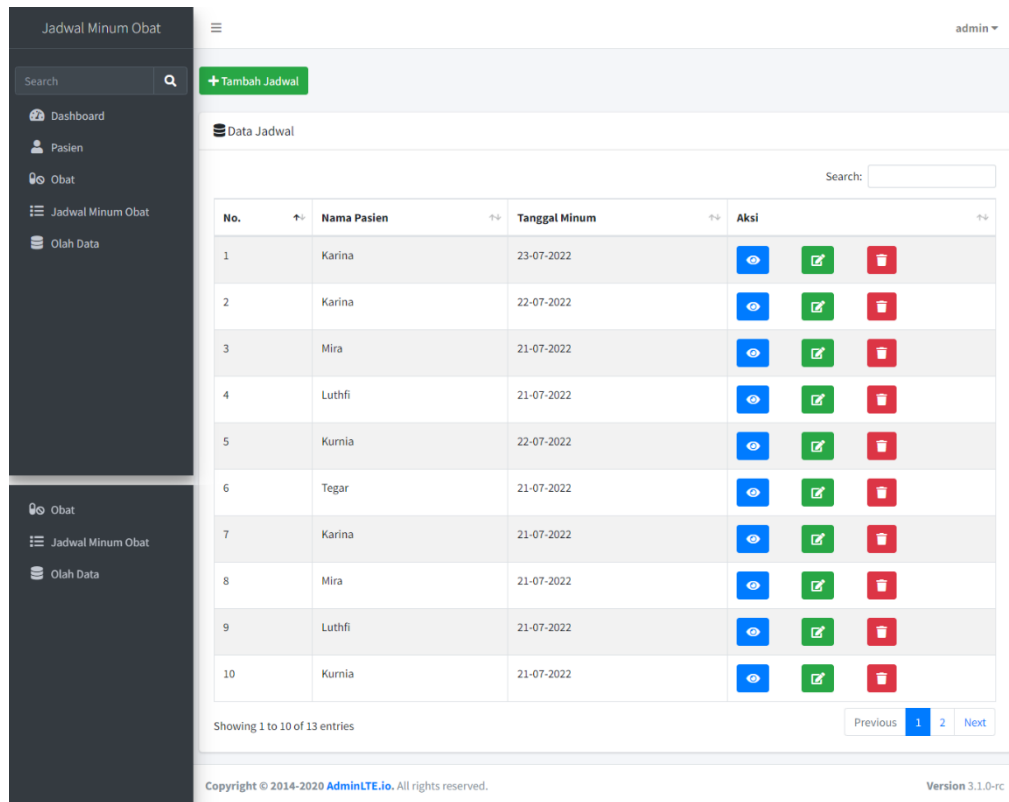
Pengujian selanjutnya yaitu menguji fitur mengolah data jadwal. Pada pengujian ini terdapat beberapa skenario yang dapat diakses oleh super admin dan PMO. Skenario yang digunakan tersaji pada tabel 4.6.

Tabel 4. 6 Pengujian Fitur Mengolah Data Jadwal

Kode	T005
Referensi kasus	A005
Deskripsi pengujian	Mengolah data jadwal
Tujuan	Menguji fungsi lihat data, tambah data, <i>update</i> data dan hapus data
Skenario 1 – Melihat data jadwal	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat melihat data jadwal yang sudah tersimpan pada <i>database</i>
Hasil yang didapat	Tampilan <i>list</i> data pasien yang disajikan pada sebuah tabel
Hasil pengujian	Berhasil
Skenario 2 – Menambah data jadwal	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat menambah data jadwal kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data pasien tersimpan pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 3 – Mengupdate data jadwal	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat meng- <i>update</i> data jadwal kemudian menyimpannya di <i>database</i>
Hasil yang didapat	Data jadwal ter- <i>update</i> pada <i>database</i>
Hasil pengujian	Berhasil
Skenario 4 – Menghapus data jadwal	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	<i>User</i> dapat menghapus data jadwal

Tabel 4.6 Pengujian Fitur Mengolah Data Jadwal (Lanjutan)

Hasil yang didapat	Data jadwal terhapus dari list daftar jadwal
Hasil pengujian	Berhasil



Gambar 4. 16 Pengujian T005 Skenario 1

Gambar 4.16 merupakan gambar pengujian halaman melihat data jadwal yang sudah dibuat. Pengujian ini berhasil dilakukan, user yang dapat mengakses halaman ini dapat melihat daftar jadwal yang telah dibuat sebelumnya. Halaman ini memuat seluruh data jadwal yang terekam pada *database*. Selain menampilkan seluruh daftar jadwal juga disediakan *button* untuk manajemen data jadwal.

Pengujian selanjutnya yaitu Pengujian T005 Skenario 2, pengujian ini dilakukan untuk memastikan bahwa jadwal yang telah diinputkan dapat terekam oleh *database*. Pada halaman tambah data jadwal *user* dapat memilih pasien dan tanggal untuk menentukan jadwal. Setelah itu *user* dapat memberikan obat apa saja yang harus di konsumsi untuk pasien tersebut, dan jam minum obat. Uji coba tersebut dapat dilihat pada gambar 4.17 di bawah ini.

Jadwal Minum Obat

Search

Dashboard
Pasien
Obat
Jadwal Minum Obat
Olah Data

Tambah data Jadwal

Pasien
Mira

Tanggal Minum
07/26/2022

Jadwal Obat

Obat
-- Pilih Obat --

Dosis

Waktu Minum
--:-- --

+ Tambah

Preview

No	Obat	Waktu	Dosis	Aksi
1	Analgesik	09:17:00	3x sehari	
2	Antasida	11:17:00	3x sehari	

Kembali

Simpan

Copyright © 2014-2020 AdminLTE.io. All rights reserved.

Version 3.1.0-rc

Jadwal Minum Obat

Search

Dashboard
Pasien
Obat
Jadwal Minum Obat
Olah Data

+ Tambah Jadwal

Data Jadwal

Search:

No.	Nama Pasien	Tanggal Minum	Aksi
1	Mira	26-07-2022	
2	Karina	23-07-2022	
3	Karina	22-07-2022	

Gambar 4. 17 Pengujian T005 Skenario 2

Selanjutnya adalah pengujian T005 Skenario 3, pengujian ini dilakukan untuk melihat keberhasilan peng-update-an data jadwal yang telah di lakukan. Jadwal dapat diedit mungkin karena kesalahan saat peng-input-an data sebelumnya. Data yang dapat diubah yaitu tanggal, obat yang telah di berikan

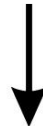
79

dan waktu minum obat. Pengujian T005 Skenario 3 dapat dilihat pada gambar 4.18.

Simpan data jadwal
Apakah anda yakin akan menyimpan data ini ?

Batal Ya, Simpan!

No	Obat	Waktu	Dosis	Aksi
1	Analgesik	09:17:00	3x sehari	
2	Antasida	11:17:00	3x sehari	
3	Anticemas	12:23:00	3x sehari	

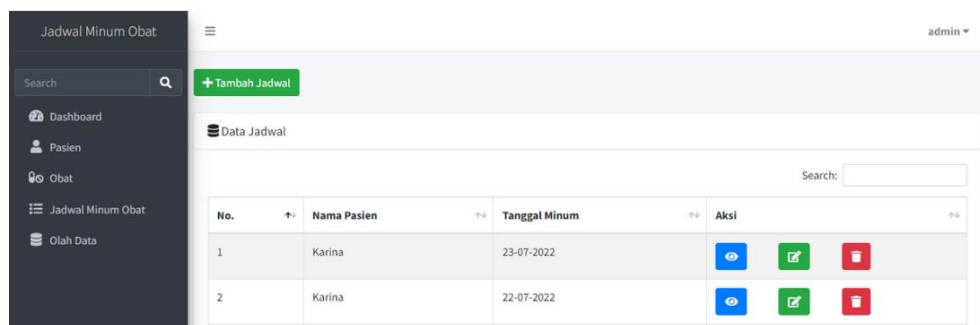
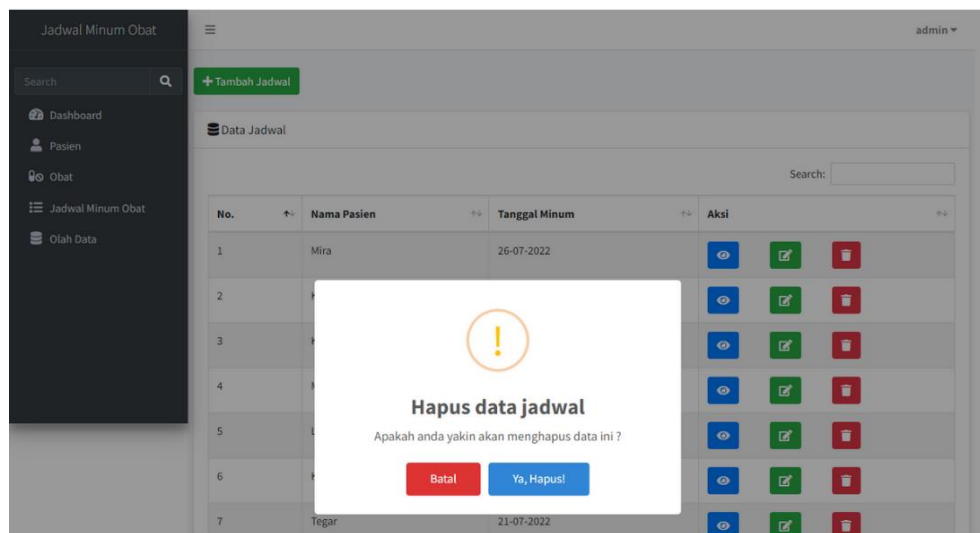
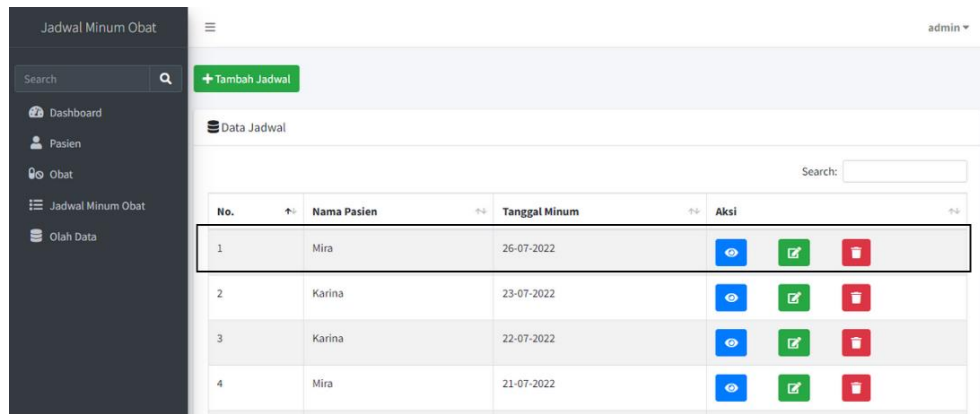


Detail Jadwal

Pasien: Mira
Tanggal Minum: 26-07-2022

No	Obat	Waktu	Dosis
1	Analgesik	09:17:00	3x sehari
2	Antasida	11:17:00	3x sehari
3	Anticemas	12:23:00	3x sehari

Gambar 4. 18 Pengujian T005 Skenario 3



Gambar 4. 19 Pengujian T005 Skenario 4

Gambar 4.19 merupakan gambar pengujian T005 Skenario 4, pada pengujian ini akan diuji fitur hapus data jadwal. Data jadwal yang sudah tidak diperlukan atau memang mengalami kesalahan yang tidak dapat di *update*, maka jadwal dapat dihapus dengan menekan *button* berwarna merah dengan *icon trash*.

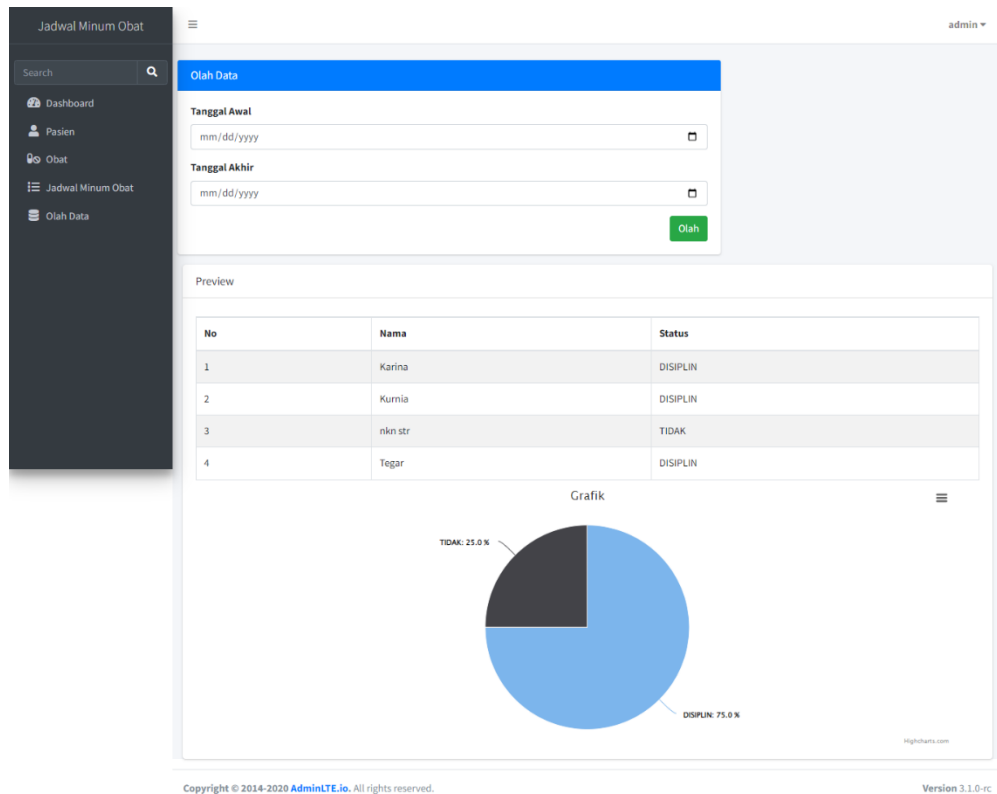
6. Pengujian Mengolah Data Kedisiplinan

Pengujian mengolah data jadwal dilakukan oleh super admin dan PMO. Pengolahan data kedisiplinan ini diambil dari data pasien kemudian di-clusterkan sesuai dengan cluster disiplin dan tidak disiplin. Berikut ini adalah skenario pengujian mengolah data kedisiplinan yang disajikan dalam tabel 4.7.

Tabel 4. 7 Pengujian Mengolah Data Kedisiplinan

Kode	T006
Referensi kasus	A006
Deskripsi pengujian	Mengolah data kedisiplinan
Tujuan	Menguji peng- <i>clusteran</i> data pasien kedalam cluster disiplin dan tidak disiplin
Skenario 1 – Menguji Pengclusteran	
Pengguna	Super admin dan PMO
Hasil yang diharapkan	User dapat melihat data pasien yang sudah di- <i>clusterkan</i> sesuai dengan <i>cluster</i> disiplin dan tidak
Hasil yang didapat	Tampilan <i>list</i> data pasien yang sudah ter- <i>cluster</i>
Hasil pengujian	Berhasil

Gambar 4.20 Merupakan gambar Pengujian T006 Skenario 1, pada pengujian ini pengguna super admin dan PMO dapat mengolah data kedisiplinan dari setiap pasien yang telah diberikan jadwal minum obat. Pengguna dapat memilih dari rentang tanggal berapa data yang akan diambil dan diolah untuk mengetahui setiap pasien disiplin atau tidak dalam meminum obat yang telah dijadwalkan.



Gambar 4. 20 Pengujian T006 Skenario 1

7. Pengujian Menampilkan Data Jadwal

Fitur menampilkan data jadwal ini dikhususkan untuk *user* level pasien. Pasien hanya dapat melihat jadwal yang sudah dibuatkan oleh PMO. Berikut ini merupakan skenario pengujian dari fitur menampilkan data jadwal yang dapat dilihat pada tabel 4.8.

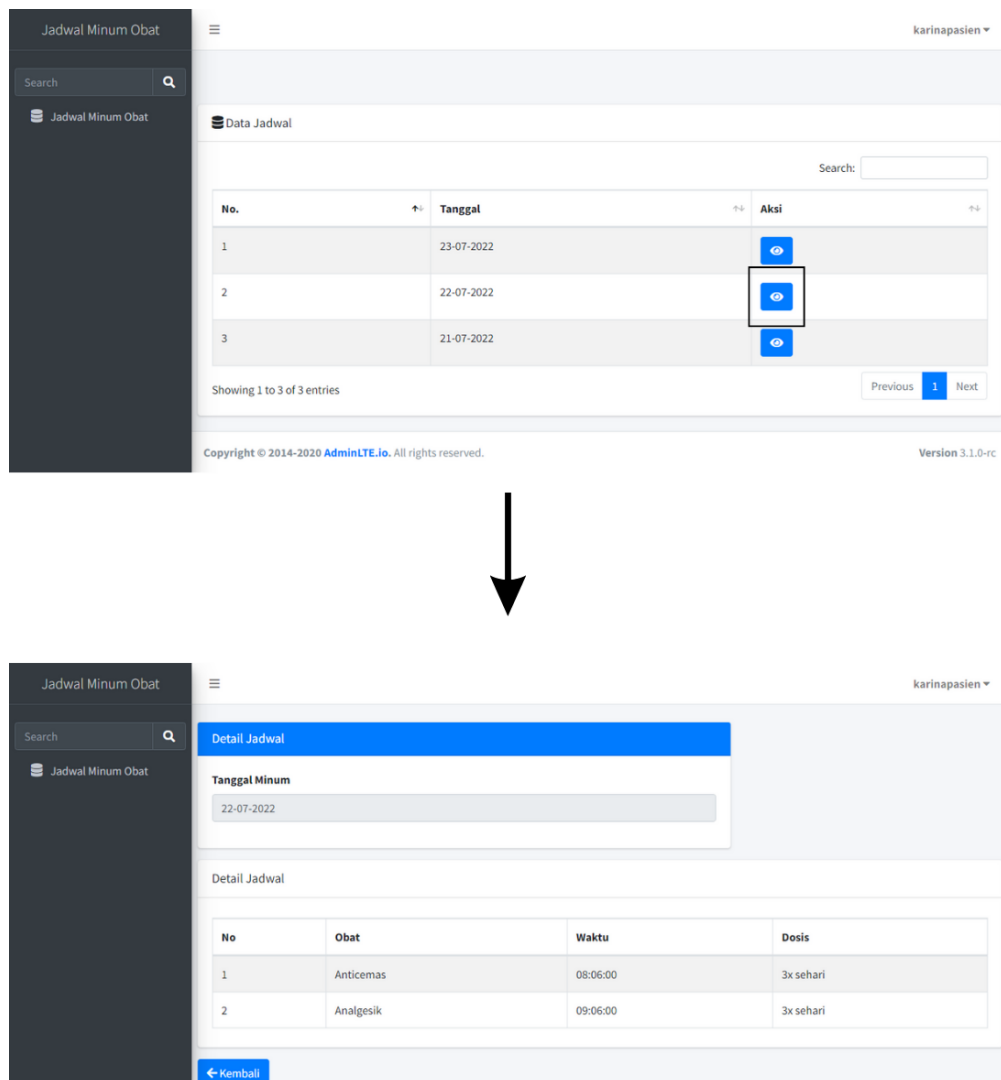
Tabel 4. 8 Pengujian Menampilkan Data Jadwal

Kode	T007
Referensi kasus	A007
Deskripsi pengujian	Menampilkan data jadwal
Tujuan	Menguji fungsi menampilkan data jadwal tiap pasien
Skenario 1 – Menguji Menampilkan data jadwal	
Pengguna	Pasien
Hasil yang diharapkan	User dapat melihat data jadwal yang sudah dijadwalkan oleh PMO

Tabel 4.8 Pengujian Menampilkan Data Jadwal(Lanjutan)

Hasil yang didapat	Tampilan <i>list</i> data jadwal dan detail jadwal
Hasil pengujian	Berhasil

Gambar 4.21 Merupakan gambar pengujian T007 Skenario 1, di mana user level pasien dapat melihat data jadwal minum obat yang telah dijadwalkan oleh masing-masing PMO dari setiap pasien. Pasien dapat melihat daftar jadwal dan melihat detail jadwal pada tanggal yang telah ditentukan. Pada detail jadwal terdapat obat apa saja yang telah diberikan oleh PMO dan waktu minum untuk setiap obat tersebut.



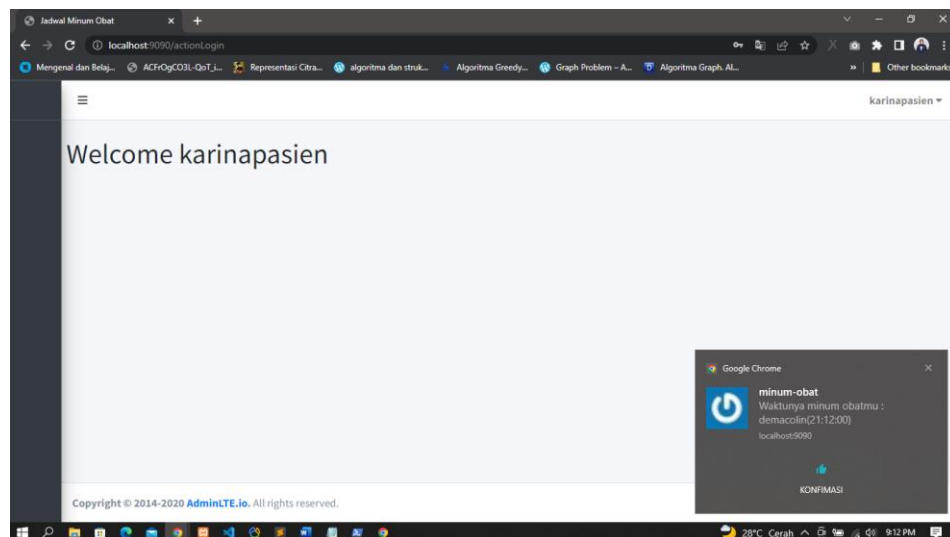
Gambar 4. 21 Pengujian T007 Skenario 1

8. Pengujian Menampilkan Notifikasi

Pengujian fitur tampil notifikasi digunakan untuk melihat apakah notifikasi yang muncul sudah sesuai dengan jadwal yang di inputkan oleh PMO kepada setiap pasien yang ditangani. Berikut ini skenario pengujian menampilkan notifikasi dapat dilihat pada tabel 4.9.

Tabel 4. 9 Pengujian Menampilkan Notifikasi

Kode	T008
Referensi kasus	A008
Deskripsi pengujian	Menampilkan notifikasi
Tujuan	Menguji notifikasi tampil sesuai jadwal
Skenario 1 – Menguji Notifikasi	
Pengguna	Pasien
Hasil yang diharapkan	yang User dapat menerima notifikasi sesuai dengan jadwal yang telah ditentukan PMO
Hasil yang didapat	Tampil notifikasi jadwal minum obat
Hasil pengujian	Berhasil



Gambar 4. 22 Pengujian T008 Skenario 1

Pengujian T008 Skenario 1 merupakan pengujian dalam menampilkan notifikasi minum obat untuk pasien. Nantinya pasien akan mendapatkan notifikasi sesuai dengan jadwal yang telah ditentukan oleh PMO. Notifikasi

tersebut akan mengingatkan pasien untuk meminum obat apa pada jam di mana notifikasi tersebut muncul. Gambar 4.22 merupakan gambar dimana notifikasi minum obat pada pasien akan muncul sesuai jadwal.

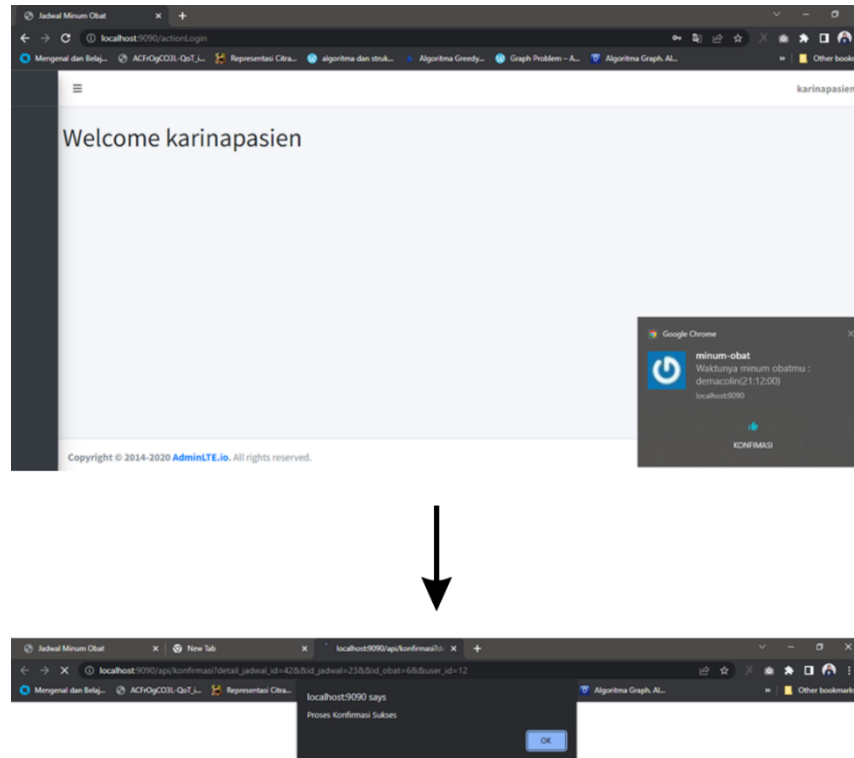
9. Pengujian Konfirmasi Notifikasi

Pengujian selanjutnya yaitu pengujian fungsi konfirmasi notifikasi minum obat. Konfirmasi pada notifikasi ini bertujuan untuk merekam data konfirmasi minum obat dari setiap pasien yang telah dijadwalkan setiap kali meminum obat oleh Pengawas Minum Obat. Notifikasi yang nantinya muncul akan dikonfirmasi oleh pasien, jika pasien sudah meminum obat yang telah dijadwalkan. Data konfirmasi yang didapatkan nantinya akan disimpan dalam database dan akan di-*clusterkan* apakah pasien disiplin atau tidak. Berikut adalah skenario dari pengujian tersebut yang tersaji pada tabel 4.10.

Tabel 4. 10 Pengujian Konfirmasi Notifikasi

Kode	T009
Referensi kasus	A009
Deskripsi pengujian	Konfirmasi notifikasi
Tujuan	Menguji fungsi konfirmasi notifikasi yang muncul
Skenario 1 – Menguji Konfirmasi Notifikasi	
Pengguna	Pasien
Hasil yang diharapkan	User dapat mengkonfirmasi notifikasi sesuai dengan jadwal yang telah ditentukan PMO
Hasil yang didapat	Data konfirmasi berhasil masuk kedalam <i>database</i>
Hasil pengujian	Berhasil

Pengujian T009 Skenario 1 merupakan pengujian mengenai fitur tombol konfirmasi pada setiap notifikasi yang akan muncul pada pasien. Tombol konfirmasi nantinya akan di klik oleh setiap pasien ketika pasien tersebut telah meminum obat sesuai dengan notifikasi jadwal minum obat yang akan muncul.



Gambar 4. 23 Pengujian T009 Skenario 1

4.4 Evaluasi Pengujian

1. Evaluasi Fungsionalitas Fitur

Hasil dari pengujian sebelumnya yang menggunakan metode *Blackbox testing* dirangkum pada tabel 4.11.

Tabel 4. 11 Rangkuman Pengujian Fungsionalitas Fitur

No	Kode	Deskripsi Pengujian	Skenario	Hasil
1	T001	Login	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
2	T002	Mengolah data PMO	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
			Skenario 4	Berhasil

Tabel 4.11 Rangkuman Pengujian Fungsionalitas Fitur(Lanjutan)

3	T003	Mengolah data pasien	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
			Skenario 4	Berhasil
4	T004	Mengolah data obat	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
			Skenario 4	Berhasil
5	T005	Mengolah data jadwal	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
			Skenario 4	Berhasil
6	T006	Mengolah data kedisiplinan	Skenario 1	Berhasil
7	T007	Menampilkan data jadwal	Skenario 1	Berhasil
8	T008	Menampilkan notifikasi	Skenario 1	Berhasil
9	T009	Konfirmasi notifikasi	Skenario 1	Berhasil

2. Evaluasi Kelebihan dan Kekurangan Aplikasi

Sebuah aplikasi yang dikembangkan tentunya memiliki kelebihan dan kekurangan. Oleh karena itu, aplikasi yang telah dibuat ini juga memiliki kelebihan dan kekurangan. Tabel 4.12 berikut ini merupakan kelebihan dan kekurangan dari aplikasi yang telah di buat.

Tabel 4. 12 Kelebihan dan Kekurangan Aplikasi

No	Kelebihan	Kekurangan
1	Aplikasi sudah dapat memfasilitasi PMO dalam mengawasi pasien penderita kusta	Notifikasi terkadang mengalami <i>delay</i> , sesuai dengan koneksi internet

Tabel 4.12 Kelebihan dan Kekurangan Aplikasi

2	Dapat menampilkan notifikasi sesuai dengan jadwal yang telah ditentukan	Sulit mendapatkan <i>player id</i> untuk dapat digunakan dalam pengiriman notifikasi
3	Dapat mengolah data pasien yang disiplin dan tidak bug berdasarkan data konfirmasi minum obat	Aplikasi masih memiliki beberapa bug
4	Aplikasi dapat dikatakan baik, karena setiap fitur berjalan dengan baik	Desain tampilan masih belum maksimal dalam penataan layout

3. Evaluasi hasil Fuzzy C-Means Clustering

Data konfirmasi yang telah terekam dalam database digunakan untuk pengujian kedisiplinan dengan teknik FCM. Hasil yang diperoleh dari peng-clusteran menggunakan data konfirmasi sesuai dengan apa yang telah diharapkan. Berpatokan dengan tiga jenis data konfirmasi yaitu tepat waktu, tidak tepat waktu, dan tidak respon, FCM dapat meng-*clusterkan* data konfirmasi setiap pasien dengan baik. Data yang diproses latih yang disimpan akan di latih terlebih dahulu sehingga menghasilkan dua cluster yang sudah ditentukan sejak awal. Kemudian data uji selanjutnya akan dilakukan proses *clustering* untuk mengetahui data yang diujikan masuk kedalam cluster mana. Hasil dari pengujian tersebut dapat dilihat pada tabel 4.13 berikut.

Tabel 4. 13 Hasil Uji FCM

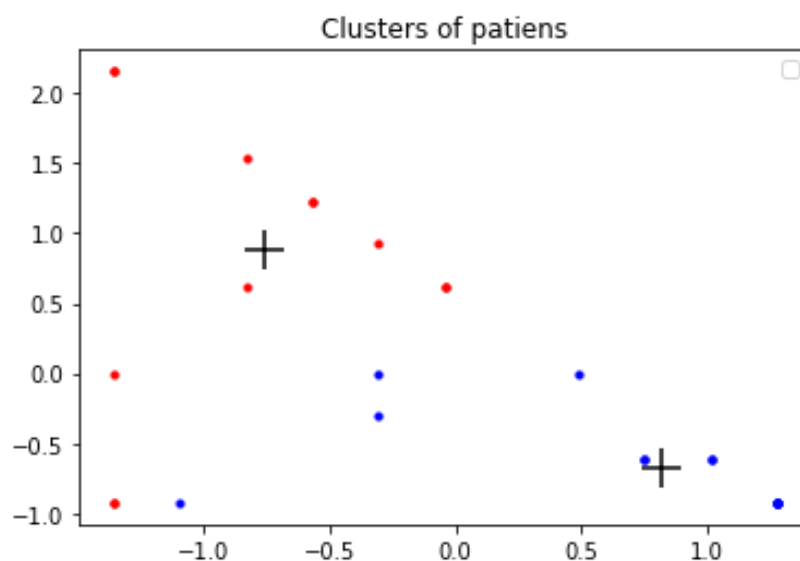
Nama	Tepat Waktu	Tidak Tepat Waktu	Tidak Respon	Prediksi	Sesuai dan Tidak Sesuai
Karina	3	0	1	Disiplin	Tidak Sesuai
Kurnia	7	0	0	Disiplin	Sesuai
Nkn str	1	0	0	Tidak Disiplin	Sesuai

Nama	Tepat Waktu	Tidak Tepat Waktu	Tidak Respon	Prediksi	Sesuai dan Tidak Sesuai
Tegar	7	2	1	Disiplin	Sesuai

Dapat dilihat pada tabel 4.13, bahwa hasil yang diperoleh dari pengujian yaitu berhasil meng-*cluster*-kan data yang diujikan. Hanya saja masih terdapat kekurangan, yaitu data latih yang digunakan masih minim, sehingga dalam proses peng-*clusteran* data uji digabung dengan data latih agar hasil yang muncul sesuai dengan apa yang diharapkan. Dari hasil tersebut dikatakan bahwa FCM berhasil melakukan clusterisasi sebesar 75% dari data yang diuji. Tabel 4.13 menyajikan bahwa terdapat satu data yang tidak sesuai dengan apa yang diharapkan, sisanya yaitu tiga data masuk *cluster* yang diharapkan.

4.5 Diskusi Hasil

Dari hasil yang proses peng-*clusteran* yang telah dilakukan mendapatkan hasil bahwa data yang diujikan telah berhasil di *cluster*kan dengan cukup baik. Namun, data uji yang masih sedikit mengakibatkan hasil pengujian kurang maksimal. Pada gambar 4.24 dapat dilihat bahwa persebaran *cluster*nya masih belum bisa terkumpul sesuai *cluster*.



Gambar 4. 24 Plot *Cluster* Data Latih

Jika saja data latih bisa lebih banyak, maka akan terlihat lebih jelas ketika hasil FCM di plotkan. Data yang digunakan sangat sulit untuk dicari karena belum mendapatkan data asli jumlah pasien kusta. Oleh karena itu, data latih yang digunakan merupakan data *dummy*. Mengapa ini menggunakan data *dummy*, karena dari informasi salah satu dokter yang telah diwawancarai oleh penulis menjelaskan bahwa tidak ada indikator atau variabel khusus penentu setiap pasien dikategorikan disiplin dan tidak disiplin. Selama ini dokter menganggap bahwa ketika pasiennya telah diresepkan dan diberi jadwal minum obat, satu kali saja pasien tersebut tidak minum sudah dikatakan tidak disiplin. Mengapa demikian mengingat bahwa penyakit yang memerlukan perawatan *Multi Drug Therapy* harus selalu tepat waktu dan rajin meminum obat sesuai anjuran dokter. Untuk informasi yang lebih jelas dapat dilihat pada LAMPIRAN C mengenai hasil wawancara dengan dokter tersebut.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil yang telah didapatkan selama proses perancangan, pembuatan dan pengujian, kesimpulan yang didapatkan adalah sebagai berikut:

1. Aplikasi yang sudah di buat dapat berjalan sesuai dengan apa yang telah dirancang sebelumnya. Aplikasi berhasil mengirimkan notifikasi minum obat kepada setiap pasien sesuai dengan jadwal yang telah ditentukan oleh PMO.
2. Seluruh pengujian berhasil dilakukan dengan skenario yang telah dibuat.
3. Aplikasi yang telah dibuat memiliki fitur kebaruan dari aplikasi pada penelitian sebelumnya yang berjudul “Medicine Consumption Reminder and Monitoring Application for Patients with Leprosy Disease”. Kebaruan tersebut yaitu, aplikasi dapat mengawasi data pasien secara luas dengan adanya penyajian data pasien dan jadwal. Selain itu pada aplikasi ini dapat mengklasterisasi data pasien kedalam kelompok disiplin dan tidak disiplin sesuai dengan keteraturan minum obat.
4. Sedikit sulit mengimplementasikan *plugin* OneSignal pada kasus aplikasi *web-base* ini, karena sulitnya mendapatkan *player id* untuk dapat mengirimkan notifikasi pada setiap *user* (pasien), dan terdapat *delay* pada saat notifikasi muncul apabila koneksi internet tidak stabil.

5.2 Saran

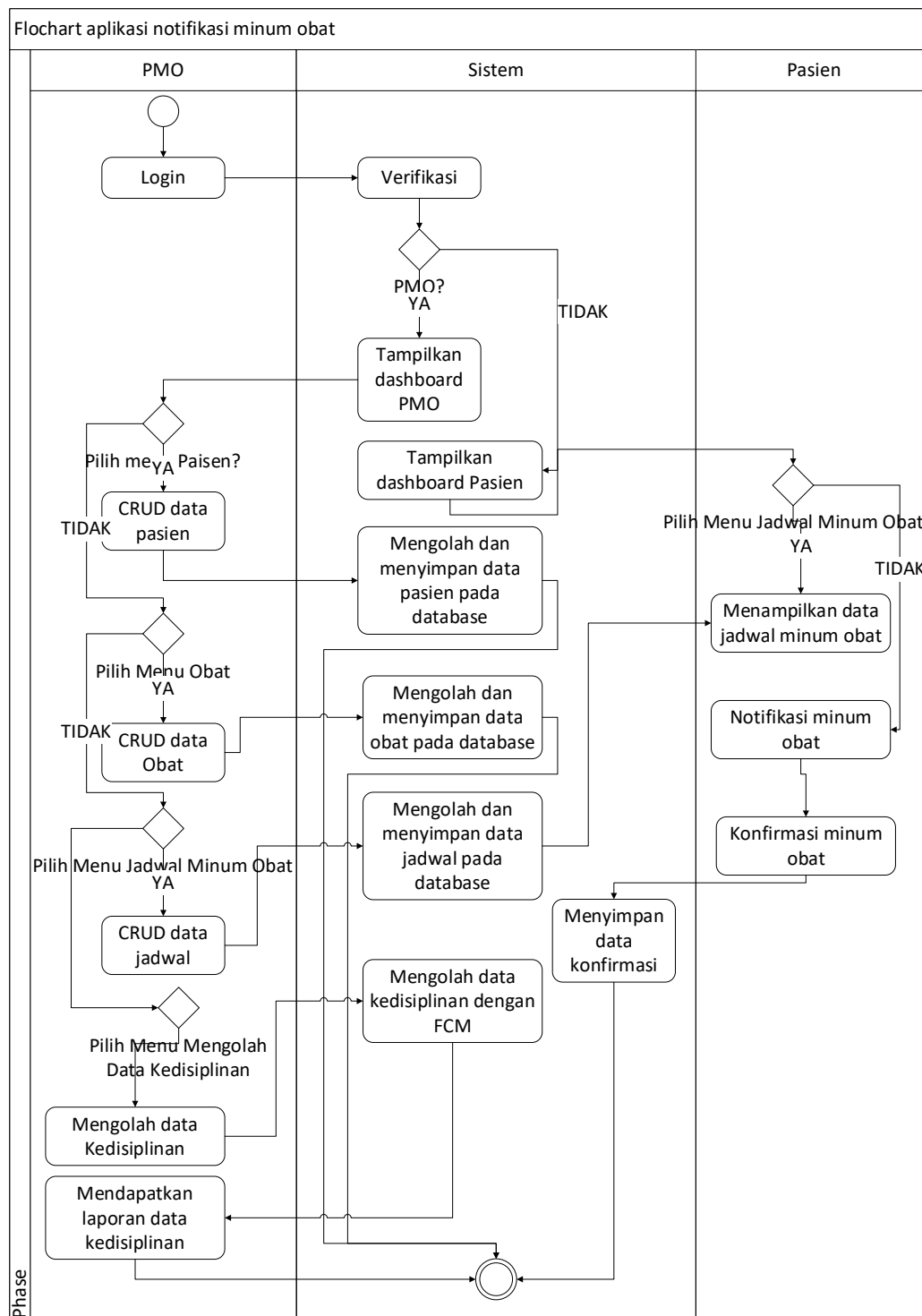
Saran untuk pengembangan aplikasi ini selanjutnya adalah sebagai berikut :

1. Menambahkan fitur lain sesuai dengan kebutuhan PMO seperti log kunjungan pasien.
2. Mencari solusi untuk memunculkan notifikasi menggunakan *plugin* Onesignal agar lebih stabil dengan berbagai kesulitan yang telah ditemukan
3. Menambah lebih banyak dataset untuk pengujian kedisiplinan pasien agar pengklasteran menjadi lebih baik.

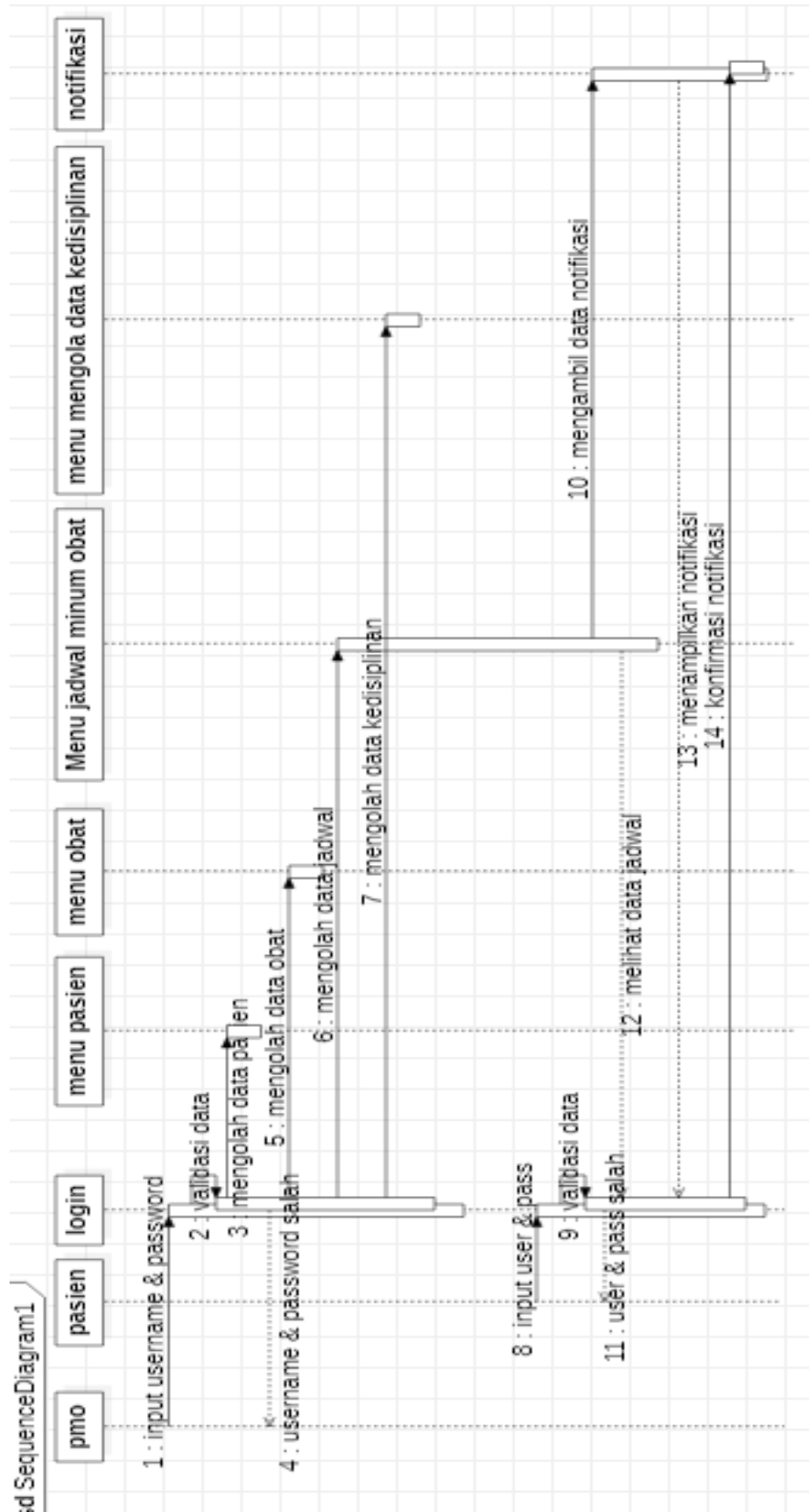
DAFTAR PUSTAKA

- Agustina, N. & Prihandoko, 2018. Perbandingan Algoritma K-Means Dengan Algoritma Fuzzy C-Means Untuk Clustering Tingkat Kedisiplinan Kinerja Karyawan. JURNAL RESTI, Volume II, pp. 621-626.
- Alisya, C. A., Nurdin, A. & Salamah, I., n.d. Rancang Bangun Smart Medicine Box Sebagai Pengingat Jadwal Minum Obat Berbasis Internet Of Things. Jurnal Teknologi Elekterika, Volume XVIII, pp. 1-9.
- Cahyani, F. M., 2018. Hubungan Kepatuhan Minum Obat Antihipertensi Terhadap Tercapainya Target Terapi Pasien Hipertensi di Puskesmas Wirobrajan Yogyakarta. Journal of Pharmaceutical Science and Medical Reseaarch (PHARMED), Volume I, pp. 10-16.
- Cox, E., 1994. The Fuzzy Systems Handbook A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems. United States of America: Academic Press, Inc.
- Irawan, P. L. T., Hartanto, C. B. S. & Kelana, O. H., 2022. Medicine Consumption Reminder and Monitoring Application for Patients with Leprosy Disease. Journal of Community Practice and Social Welfare, Volume II, pp. 37-49.
- Kusumadewi, S. & Purnomo, H., 2010. Aplikasi Logika Fuzzy untuk Pendukung Keputusan. 2nd ed. Yogyakarta: Graha Ilmu.
- Martin & Nataliani, Y., 2020. Klasterisasi kinerja karyawan menggunakan algoritma fuzzy c-means. AITI: Jurnal Teknologi Informasi, Volume XVII, pp. 118-129.
- Mumtazah, A. A., 2015. Rancang Bangun Aplikasi Lansia Care Untuk Monitoring Lansia Yang Tinggal Sendiri Berbasis Android, Surabaya: Institut Teknologi Sepuluh Nopember.
- Musa, M. I., 2019. Aplikasi Monitoring Untuk Pasien TBC Dewasa Berbasis Android, Yogyakarta: Universitas Islam Indonesia.
- Pradipta, F. A., 2018. Rancang Bangun Intervace Voice Response(IVR) Sebagai Pengingat Medikasi Pasien Lansia Berbasis Web, Surabaya: Institut Teknologi Sepuluh November.
- Rustam, M. Z. A., 2018. Determinan Keberhasilan Pengobatan Multi Drug Therapy Pada Penderita Kusta Tipe Multibaciler. Jurnal Manajemen Kesehatan Yayasan RS.Dr.Soetomo, Volume IV, pp. 61-70.
- Syawaludin, M., Muhimmah, I. & Kurniawan, R., 2021. Desain Prototipe Sistem Monitoring Minum Obat Bagi ODHA. JIKO (Jurnal Informatika dan Komputer), Volume IV, pp. 76-85.
- Wikarno, Malani, R. & Suprpty, B., 2018. Perbandingan Metode K-Means dan Fuzzy C-Means Untuk Pengelompokan Pegawai Berdasarkan Nilai Kinerja dan Tingkat Kedisiplinan Pegawai. Prosiding Seminar Ilmu Komputer dan Teknologi Informasi, Volume III, pp. 45-52.

LAMPIRAN A



LAMPIRAN B



LAMPIRAN C

Narasumber : dr. Rachmad Ari Irmawan, Sp. PD.

Jabatan :

1. Direktur Rumah Sakit dr. Etty Asharto Batu
2. Dokter Spesialis Penyakit Dalam Rumah Sakit dr. Etty Asharto Batu.

Wawancara dilakukan tatap muka pada tanggal 27 September 2022 di Rumah Sakit dr. Etty Asharto Batu. Berikut merupakan hasil wawancara dengan beliau.

1. Apakah terdapat indikator atau variabel khusus dalam penentuan setiap pasien disiplin dalam meminum obat?

Jawaban : tidak ada indicator atau variabel khusus dalam penentuan pasien disiplin dan tidaknya meminum obat. Sebenarnya disiplin atau tidaknya dalam meminum obat bersumber pada setiap pasien itu sendiri.

2. Bagaimana pasien dikatakan tepat waktu dalam meminum obat?

Jawaban : pasien dikatakan tepat waktu meminum obat, jika pasien tersebut meminum obat dalam waktu yang telah di jadwalkan oleh dokter.

3. Semisal dalam kurun waktu satu minggu pemberian obat, berapa kali jumlah minum obat secara tepat waktu yang hasilnya dapat dikatakan bahwa pasien tersebut disiplin?

Jawaban : kalau idealnya setau saya seluruh obat yang dijadwalkan harus diminum semuanya, satu kali saja tidak minum sudah dikatakan tidak disiplin. Karena untuk kasus-kasus penyakit yang membutuhkan Multi Drug Therapy ini memang harus selalu mengkonsumsi obat yang telah diresepkan dokter secara tepat waktu. Menurut saya tidak ada patokan disiplin dalam minum obat, namun menurut saya kalau disiplin itu ya semua obat yang diresepkan dan dijadwalkan harus diminum.

4. Apa saja faktor yang mempengaruhi pasien tidak disiplin dalam meminum obat yang sering terjadi?

Jawaban : banyak faktor yang mempengaruhi mengapa pasien tidak disiplin meminum obat, yang pertama yaitu dari edukasi minum obat pada pasien yang kurang, efek samping obat, motivasi yang kurang dari pasien itu sendiri.

LAMPIRAN D

Source code FCM di Laravel

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
import shutil
import os
warnings.filterwarnings('ignore')
from fcmeans import FCM
from sklearn.preprocessing import StandardScaler
import json
import csv
nKelas = 4
headers = 'Nama,tepat waktu,tidak tepat waktu,tidak respon,class'

def loadCsv(filename):
    data = pd.read_csv(filename)
    data_baru = data.drop(['Nama'],axis=1)
    data_baru = data_baru.drop(['class'],axis=1)

    kelas = pd.DataFrame(data,columns=['class'])
    data = pd.DataFrame(data,columns=headers.split(","))

    scalarModel = StandardScaler()
    data_new = scalarModel.fit_transform(data_baru) #normalisasi
    data supaya std deviasi sama, spy variabel tdk memberikan
    kontribusi berbeda terhadap model

    return data_new, kelas
def normClass(pred, kelas):
    pattern = np.array(kelas)[0][0]
    isDiff = False
    if ( pred[0] != pattern):
        patInde = (np.where(pred == pred[0]))
        patNeg = (np.where(pred != pred[0]))
        pred[patNeg] = pred[0]
        pred[patInde] = pattern
        isDiff = True
    return pred, isDiff

def normDiff(pred,isDiff):
    pattern = pred[0]
    if ( isDiff):
        patInde = (np.where(pred == pred[0]))
        patNeg = (np.where(pred != pred[0]))
        pred[patNeg] = pred[0]
        pred[patInde] = pattern
    return pred

global kelas , fcmModel
data_norm, kelas = loadCsv('dataset1.csv')

fcmModel = FCM(n_clusters =nKelas,error=0.00001,max_iter=100)
fcmModel.fit(data_norm)
center = fcmModel.centers
```

```

pred = fcmModel.predict(data_norm)
pred,isDiff = normClass(pred, kelas)

print('Predicted Value for fcmModel is : ' , pred)

from http.server import BaseHTTPRequestHandler, HTTPServer

import logging

class ServerFCM(BaseHTTPRequestHandler):

    def _set_response(self):
        self.send_response(200)
        self.send_header('Content-type', 'application/json')
        self.end_headers()

    def do_GET(self):
        logging.info("GET request,\nPath: %s\nHeaders:\n%s\n",
str(self.path), str(self.headers))
        self._set_response()
        self.wfile.write("GET request for
{}".format(self.path).encode('utf-8'))

    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        post_data = self.rfile.read(content_length)
        logging.info("POST request,\nPath:
%s\nHeaders:\n%s\n\nBody:\n%s\n",
            str(self.path), str(self.headers),
post_data.decode('utf-8'))

        arrayUser = post_data.decode('utf-8')
        data = json.loads(arrayUser)
        shutil.copyfile('dataset1.csv', "temp")

        file = open('temp', 'a')
        # file.write(headers)
        countData = 0
        for line in data['data']:
            file.write("\n{0},0".format(line))
            countData+=1
        file.close()

        global kelas , fcmModel

        data_user, c = loadCsv('temp')
        predUser = fcmModel.predict(data_user)
        predUser,isDiff = normClass(predUser, kelas)
        predUser = normDiff(predUser,isDiff)

        print('Predicted user data : ' , predUser)
        strResponse =
"{ "+"\"data\":[{}]"}.format(", ".join(str(predUser[i]) for i in
range(len(predUser)-countData,len(predUser)) )) +"}"
        print(strResponse)
        self._set_response()

```

```

        self.wfile.write( strResponse.encode('utf-8'))

def run(server_class=HTTPServer, handler_class=ServerFCM,
port=8080):
    logging.basicConfig(level=logging.INFO)
    server_address = ('', port)
    httpd = server_class(server_address, handler_class)
    logging.info(': Starting FCM httpd...\n')
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        pass
    httpd.server_close()
    logging.info(': Stopping FCM httpd...\n')

if __name__ == '__main__':
    run(port=9191)

```