

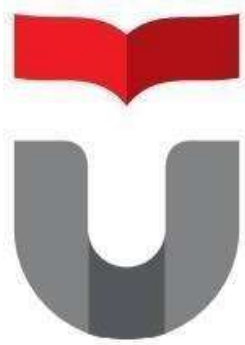
TUGAS AKHIR

**PENGEMBANGAN FRONT-END SISTEM MONITORING
PASIEN COVID-19 BERDASARKAN PROSES
PENANGANAN COVID-19 DI RUMAH SAKIT**

Oleh:

Alaric Rasendriya Aniko

1202194162



PROGRAM STUDI STRATA 1 SISTEM INFORMASI

FAKULTAS REKAYASA INDUSTRI

UNIVERSITAS TELKOM

2023

ABSTRAK

Seiring merebaknya Virus Corona (Covid-19) secara global, situasi kesehatan dunia terguncang oleh krisis besar yang disebabkan oleh penyebarannya yang cepat dan dampak serius terhadap kesehatan masyarakat. Dalam konteks ini, menjadi semakin penting untuk mengembangkan solusi teknologi yang dapat mendukung upaya penanganan Covid-19, terutama di lingkungan rumah sakit. Penelitian ini bertujuan untuk mengembangkan sebuah perancangan sistem pemantauan berupa *dashboard* pasien Covid-19 yang akan terintegrasi dengan sistem informasi manajemen rumah sakit dan *Internet of Things* (IoT) melalui *Application Programming Interface* (API). Pendekatan pengembangan yang digunakan adalah *Prototyping process* model. Sistem ini bertujuan untuk memberikan informasi secara *real-time* tentang kondisi pasien kepada para tenaga medis, sekaligus mengawasi tanda-tanda vital pasien secara berkelanjutan. Harapannya sistem ini dapat meringankan beban kerja rumah sakit dalam menangani pasien Covid-19. Evaluasi kegunaan sistem dilakukan melalui penggunaan *System Usability Scale* (SUS) guna mengukur sejauh mana sistem ini dapat dimanfaatkan. Hasil evaluasi dengan skor SUS sebesar 81.5 mengindikasikan bahwa *dashboard* memiliki tingkat penerimaan yang tinggi di antara pengguna. Pengguna secara umum merasa nyaman dan mampu menggunakan sistem dengan baik. Hal itu menunjukkan bahwa sistem memiliki potensi untuk diterima dengan baik oleh pengguna dan memenuhi kebutuhan pengguna.

Kata Kunci: Covid-19, *Website Development*, *Prototyping process* model, *System Usability Scale*.

ABSTRACT

With the global outbreak of the coronavirus (Covid-19), the world health situation has been shaken by a major crisis caused by its rapid spread and serious impact on public health. In this context, it is becoming increasingly important to develop technological solutions that can support efforts to deal with Covid-19, especially in the hospital environment. This study aims to design and develop a monitoring system in the form of a Covid-19 patient dashboard that will be integrated with hospital management information systems and the Internet of Things (IoT). The development approach used is the Prototyping process model. This system aims to provide real-time information about the patient's condition to medical personnel while continuously monitoring the patient's vital signs. The hope is that this system can ease the workload of hospitals in treating Covid-19 patients. Evaluation of system usability is carried out through the use of the System Usability Scale (SUS) to measure the extent to which this system can be utilized. The evaluation results with a SUS score of 81.5 indicate that the dashboard has a high level of acceptance among users. Users generally feel comfortable and able to use the system well. This shows that the system has the potential to be well-received by users and meet user needs.

Keywords: Covid-19, Website Development, Prototyping process model, System Usability Scale.

LEMBAR PENGESAHAN

Tugas Akhir dengan judul:

PENGEMBANGAN FRONT-END SISTEM MONITORING PASIEN COVID-19 BERDASARKAN PROSES PENANGANAN COVID-19 DI RUMAH SAKIT

Telah disetujui dan disahkan pada sidang Tugas Akhir

Program Studi Strata 1 Sistem Informasi

Fakultas Rekayasa Industri Universitas Telkom

Oleh:

Alaric Rasendriya Aniko

1202194162

Bandung, 24 Agustus 2023

Disetujui oleh,

Pembimbing 1,

Pembimbing 2,

Dr. Tien Fabrianti Kusumasari, S.T., M.T.
14790008

Dr. Sinung Suakanto, S.T., M.T.
NIP: 20820007

LEMBAR PERNYATAAN ORISINALITAS



Nama : Alaric Rasendriya Aniko
NIM : 1202194162
Alamat : Kp. Rawa RT 01/RW 08, Kec. Bojong
Gede, Kab. Bogor, Jawa Barat
No. Telepon : 081290482972

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri. Atas pernyataan ini, saya siap menanggung risiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 24 Agustus 2023



Alaric Rasendriya Aniko

KATA PENGANTAR

Alhamdulillah, puji serta syukur penulis panjatkan ke hadirat Allah Subhaanahu wa ta'aala. Berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir berjudul "PENGEMBANGAN FRONT-END SISTEM MONITORING PASIEN COVID-19 BERDASARKAN PROSES PENANGANAN COVID-19 DI RUMAH SAKIT". Tak lupa, Shalawat serta salam selalu terlimpahkan kepada Nabi Muhammad Shallallaahu 'alaihi wa sallam. Setelah memuji Allah ta'aala dan bershalawat kepada Nabi Shallallaahu 'alaihi wa sallam, penulis berterima kasih sebesar-besarnya kepada:

1. Bapak Eko Purnomo, Ibu Anik Andarwati, Ravi Ali Purnomo, Dean Ihsan Adhikari, dan Nisina Kayyisa yang telah memberikan dukungan kepada penulis. Baik itu dukungan yang terlihat dan tidak terlihat selama penulis hidup, khususnya dalam penyusunan tugas akhir ini.
2. Ibu Dr. Tien Fabrianti Kusumasari, S.T., M.T. dan Pak Dr. Sinung Suakanto, S.T., M.T. selaku pembimbing tugas akhir yang telah membimbing, mengarahkan, serta memotivasi penulis selama penyusunan tugas akhir ini.
3. Daffa Rifki Maula dan Barajati Syakurnia sebagai teman kamar yang selalu memotivasi dan menasihati penulis dalam perkara dunia dan akhirat.
4. Seluruh teman, rekan, dan pihak yang semoga Allah ta'aala rahmati.

Penulis berharap Tugas Akhir ini dapat bermanfaat bagi seluruh pihak. Bila masih didapati kekurangan dalam Tugas Akhir ini, maka penulis berharap kritik dan saran rekan-rekan agar penulis dapat memperbaiki kekurangan tersebut dan tidak terjadi kesalahan yang sama di masa mendatang.

DAFTAR ISI

ABSTRAK	i
<i>ABSTRACT</i>	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN ORISINALITAS	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang	1
I.2 Perumusan Masalah.....	2
I.3 Tujuan Penelitian.....	2
I.4 Batasan Penelitian	3
I.5 Manfaat Penelitian.....	3
I.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	5
II.1 Covid-19	5
II.2 Sistem <i>Monitoring</i>	10
II.3 <i>Software Engineering</i>	11
II.4 <i>Website Architecture</i>	16
II.5 <i>Website Development</i>	17
II.6 <i>Internet of Things (IoT)</i>	22
II.7 <i>Dashboard</i>	22
II.8 <i>Unified Modeling Language</i>	23

II.9	<i>System Usability Scale</i>	28
II.10	<i>State of The Art</i>	31
BAB III	METODOLOGI PENELITIAN	33
III.1	Model Konseptual	33
III.2	Sistematika Penyelesaian Masalah	34
BAB IV	ANALISIS DAN PERANCANGAN	36
IV.1	Fase <i>Communication</i>	36
IV.2	Fase <i>Quick Plan</i>	38
IV.2.1	Analisis Kebutuhan Sistem.....	38
IV.2.2	Proses Bisnis <i>Targeting</i>	40
IV.2.3	<i>Use Case Diagram</i>	41
IV.2.4	<i>Use Case Scenario</i>	41
IV.2.5	<i>Activity Diagram</i>	45
IV.2.6	<i>Class Diagram</i>	48
IV.2.7	<i>Sequence Diagram</i>	50
IV.2.8	<i>Deployment Diagram</i>	55
IV.3	Fase <i>Modelling Quick Design</i>	56
IV.3.1	Perancangan Desain Antarmuka <i>Dashboard</i>	56
BAB V	IMPLEMENTASI DAN PENGUJIAN	61
V.1	Implementasi	61
V.1.1	Iterasi 1	61
V.1.2	Iterasi 2	65
V.2	Pengujian	71
BAB VI	KESIMPULAN DAN SARAN	73
VI.1	Kesimpulan.....	73
VI.2	Saran	73

DAFTAR PUSTAKA	74
LAMPIRAN.....	79

DAFTAR GAMBAR

Gambar II.1 Proses pelayanan pasien Covid-19 di Rumah Sakit Indonesia (Direktorat Jenderal Pelayanan Kesehatan; Direktorat Pelayanan Kesehatan Rujukan, 2021).....	7
Gambar II.2 Proses pelayanan pasien Covid-19 di Rumah Sakit Malaysia (Kementrian Kesehatan Malaysia, 2020).....	8
Gambar II.3 Jenis-jenis Kasus Pasien Covid-19 (Kementrian Kesehatan Malaysia, 2020)	9
Gambar II.4 Kriteria Penerimaan Pasien <i>Confirmed</i> Covid (Kementrian Kesehatan Malaysia, 2020).....	9
Gambar II.5 Tabel <i>Early Warning Scoring</i> (Carter dkk., 2020).....	11
Gambar II.6 Layer <i>Software Engineering</i> (PRESSMAN & MAXIM, 2020).....	11
Gambar II.7 <i>Source code</i> HTML	17
Gambar II.8 <i>Source code</i> JavaScript.....	18
Gambar II.9 <i>Inline Css</i>	19
Gambar II.10 <i>Internal Css</i>	19
Gambar II.11 <i>Eksternal Css</i>	19
Gambar II.12 Baris 10 untuk memanggil <i>file css</i> eksternal	20
Gambar II.13 <i>Class Diagram</i> (Bhatt & Nandu, 2021)	24
Gambar II.14 Notasi <i>Multiplicity</i> (Hadiprakoso, 2021).....	28
Gambar II.15 Skala Likert (Kurniawan dkk., 2022)	29
Gambar II.16 <i>Acceptability, Grade Scale, dan Adjective Rating</i> (Welda dkk., 2020).	30
Gambar III.1 Model Konseptual	33
Gambar III.2 Sistematisa Penyelesaian Masalah.....	34
Gambar IV.1 Hasil <i>Card Sorting</i> dari Fitur yang dijukan	37
Gambar IV.2 Alur Perawatan Pasien Covid-19 Rawat Inap Eksisting.....	37
Gambar IV.3 Alur Perawatan Pasien Covid-19 Rawat Inap <i>Targeting</i>	40
Gambar IV.4 <i>Use Case Diagram</i>	41
Gambar IV.5 <i>Activity Diagram</i> Mengakses Data Pasien	45
Gambar IV.6 <i>Activity Diagram</i> Mengakses Data Ruangan.....	45
Gambar IV.7 <i>Activity Diagram</i> Mengakses <i>Early Warning System</i>	46

Gambar IV.8 <i>Activity</i> Diagram Menandai Penanganan Pasien Kondisi Darurat	46
Gambar IV.9 <i>Activity</i> Diagram Mengakses Master Data.....	47
Gambar IV.10 <i>Activity</i> Diagram Mengakses Bantuan	48
Gambar IV.11 <i>Class</i> Diagram.....	48
Gambar IV.12 <i>Sequence</i> Diagram Mengakses Data Pasien.....	50
Gambar IV.13 <i>Sequence</i> Diagram Mengakses Data <i>Ward</i>	51
Gambar IV.14 <i>Sequence</i> Diagram Mengakses <i>Early Warning System</i>	52
Gambar IV.15 <i>Sequence</i> Diagram Menandai Penanganan Pasien Kondisi Darurat	53
Gambar IV.16 <i>Sequence</i> Diagram Mengakses Master Data <i>Patient</i>	54
Gambar IV.17 <i>Sequence</i> Diagram Mengakses <i>Help</i>	55
Gambar IV.18 <i>Deployment</i> Diagram	55
Gambar IV.19 Perancangan Desain Halaman <i>Patient</i> Data.....	56
Gambar IV.20 Perancangan Desain Halaman Data Ruang.....	57
Gambar IV.21 Perancangan Desain Halaman <i>Early Warning System</i>	57
Gambar IV.22 Perancangan Desain Halaman Master Data	58
Gambar IV.23 Perancangan Desain Halaman Master Data <i>Patient</i>	58
Gambar IV.24 Perancangan Desain Halaman Master Data <i>Ward</i>	59
Gambar IV.25 Perancangan Desain Halaman Master Data <i>Doctor</i>	59
Gambar IV.26 Perancangan Desain Halaman <i>Help</i>	60
Gambar V.1 Implementasi Halaman <i>Patient</i> Data.....	61
Gambar V.2 Code Halaman <i>Patient</i> Data	62
Gambar V.3 Implementasi Halaman <i>Ward</i> Data	63
Gambar V.4 Code Halaman <i>Ward</i> Data.....	63
Gambar V.5 Implementasi Halaman <i>Early Warning System</i> dan Menandai Penanganan Pasien Kondisi Darurat	64
Gambar V.6 Code Halaman <i>Early Warning System</i> dan Menandai Penanganan Pasien Kondisi Darurat	65
Gambar V.7 Implementasi Halaman Master Data	65
Gambar V.8 Code Halaman Master Data	66
Gambar V.9 Implementasi Halaman Master Data Pasien.....	66
Gambar V.10 Halaman Master Data Pasien	67

Gambar V.11 Implementasi Halaman Master Data <i>Ward</i>	67
Gambar V.12 Code Halaman Master Data <i>Ward</i>	68
Gambar V.13 Halaman Master Data <i>Doctor</i>	69
Gambar V.14 Code Halaman Master Data <i>Doctor</i>	69
Gambar V.15 Halaman <i>Help</i>	70
Gambar V.16 Code Halaman <i>Help</i>	70

DAFTAR TABEL

Tabel II.1 Perbandingan metode pengembangan perangkat lunak (Narayan, STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES, 2021)	15
Tabel II.2 Notasi - Notasi UML (Hadiprakoso, 2021).....	25
Tabel II.3 Pertanyaan <i>System Usability Testing</i>	28
Tabel II.4 <i>State of The Art</i>	31
Tabel IV.1 Daftar Fitur <i>Card Sorting</i>	36
Tabel IV.2 Daftar Kelompok Fitur <i>Card Sorting</i>	36
Tabel IV.3 Kebutuhan Sistem <i>Dashboard</i>	39
Tabel IV.4 <i>Use case scenario</i> Mengakses Data Pasien	42
Tabel IV.5 <i>Use case scenarion</i> Mengakses Data Ruangan.....	42
Tabel IV.6 <i>Use case scenario</i> Mengakses <i>Early Warning System</i>	43
Tabel IV.7 <i>Use case scenario</i> Menandai Penanganan Pasien Kondisi Darurat..	43
Tabel IV.8 <i>Use case scenario</i> Mengakses Master Data.....	44
Tabel IV.9 <i>Use case scenario</i> Mengakses Bantuan	44
Tabel V.1 Daftar Pertanyaan <i>System Usability Scale</i>	71
Tabel V.2 Jawaban <i>System Usability Scale</i>	72

DAFTAR LAMPIRAN

Lampiran A. Formulir <i>System Usability Scale</i>	79
Lampiran B. Jawaban <i>System Usability Scale</i> Responden.....	79
Lampiran C. Pelaksanaan <i>System Usability Scale</i>	84
Lampiran D. Pelaksanaan <i>Card Sorting</i>	85

BAB I PENDAHULUAN

I.1 Latar Belakang

Pada awal tahun 2020 Virus Corona (Covid-19) pertama kali menyerang dunia. Virus ini dapat menyebar melalui udara. Akibatnya, banyak orang yang terjangkit Covid-19, serta menyebabkan banyak rumah sakit dan tenaga kesehatan yang kewalahan. Pada tahun 2021, Sekretaris Jenderal (Sekjen) Perhimpunan Rumah Sakit Indonesia (Persi), Lia G. Partakusuma mengaku, bahwa saat itu berbagai rumah sakit sudah kewalahan karena banyaknya pasien yang terus berdatangan, baik yang positif Covid-19 maupun bukan Covid-19. "Banyak hal-hal yang menyedihkan terjadi seputar pasien di rumah sakit. Akibatnya tenaga kesehatan pun juga sering kewalahan" (Situmorang, 2021). Lia juga mengatakan, "Pekerjaan RS cukup berat, dengan kondisi tenaga eksisting yang berkurang. Terus terang saja para tenaga kesehatan sudah kewalahan. Para tenaga kesehatan sudah menunggu tambahan tenaga tapi tidak semudah itu, karena tenaga kesehatan yang kerja di ruang isolasi pun harus ada tambahan kompetensi. Jadi tidak semua orang yang bekerja di kesehatan mampu bekerja di ruang isolasi atau ICU" (KH, 2021). Bahkan tidak hanya kewalahan, petugas rumah sakit juga menjadi korban dari Covid-19. Menurut laporan WHO dari April 2020, telah terjadi hampir 22.000 kasus Covid-19 di antara petugas kesehatan di 52 negara. Pada Juni 2020, sebuah laporan oleh Dewan Perawat Internasional menyatakan bahwa 7% dari semua kasus Covid-19 secara global adalah di antara petugas kesehatan. Hal ini menunjukkan bahwa perawat dan petugas kesehatan lainnya berada pada risiko yang besar dalam penularan Covid-19 (Ejin, Yew, Mazlina Mamat, Chekima, & Chung, 2022).

Saat ini, kasus Covid-19 sudah stabil. Berdasarkan laporan (raharjo, 2023), bahwa Direktur *Surveilans* dan Karantina Kesehatan Kementerian Kesehatan Achmad Farchanny mengatakan, "tren kasus Covid-19 secara nasional sejak 1 Juli 2023 terbilang landai dengan laju *positivity rate* sebesar 0,31 persen dan tren *hospitalisasi* serta kematian yang hampir menyentuh angka nol". Adapun berdasarkan laporan Organisasi Kesehatan Dunia (WHO), per 16 Agustus 2023, Covid-19 telah menginfeksi 769.806.130 orang yang mencakup kematian

sebanyak 6.955.497 di seluruh dunia. Di Indonesia sendiri berdasarkan data yang dikutip dari covid19.go.id, per 24 Agustus 2023 kasus positif Covid-19 terkonfirmasi sebanyak 6.811.945 kasus dan kasus kematian akibat Covid-19 sebanyak 161.870 kasus.

Meskipun kasus Covid-19 dapat dikatakan stabil, Covid-19 terus berkembang dan menghasilkan varian-varian baru seperti varian *Alpha*, *Delta*, dan *Omicron* (Setiawati, 2023). Oleh karena itu, dirancanglah sistem *monitoring* pasien rawat inap berupa *dashboard* berbasis *website* untuk menekan risiko terjadinya tenaga kesehatan yang menjadi korban Covid-19 di masa mendatang. Harapannya *dashboard* ini dapat mengurangi kontak tenaga medis khususnya dokter dengan pasien. Sehingga dapat menekan penyebaran virus Covid-19.

I.2 Perumusan Masalah

Berdasarkan masalah pada latar belakang, maka rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana proses bisnis sistem *monitoring* pasien Covid-19 rawat inap di rumah sakit?
2. Bagaimana analisis proses bisnis alur pelayanan pasien Covid-19 di Rumah Sakit dalam memenuhi kebutuhan sistem *monitoring* pasien Covid-19?
3. Bagaimana pengembangan *dashboard monitoring* pasien Covid-19 yang terintegrasi dengan sistem informasi manajemen rumah sakit?

I.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, penelitian ini bertujuan untuk

1. Merancang proses bisnis *targeting* dari perawatan pasien Covid-19 rawat inap di Rumah Sakit.
2. Menganalisis proses bisnis alur pelayanan pasien Covid-19 di Rumah Sakit dalam memenuhi kebutuhan sistem *monitoring* pasien Covid-19.
3. Mengembangkan *dashboard monitoring* pasien Covid-19 dengan pengimplementasian *Application Programming Interface* (API) sebagai integrasi *dashboard* dengan sistem informasi manajemen rumah sakit dan *Internet of Things* (IoT).

I.4 Batasan Penelitian

Berikut adalah batasan-batasan masalah pada penelitian ini.

1. Penelitian ini menggunakan Nextjs dan Laravel sebagai *framework* untuk pengembangan *website*.
2. Data-data pada *dashboard* diperoleh dari sistem yang sudah ada melalui API.
3. Penelitian ini tidak membahas perancangan *IoT* yang digunakan untuk memantau tanda-tanda vital pasien.
4. Pengujian sistem yang pada penelitian ini ada *System Usability Scale* (SUS).

I.5 Manfaat Penelitian

Manfaat penelitian ini:

1. Memudahkan proses *monitoring* pasien Covid-19 yang melakukan rawat inap di Rumah Sakit.
2. Bagi pasien Covid-19, penelitian ini akan membuat pasien merasa selalu terawasi kondisi kesehatannya di setiap saat.
3. Bagi peneliti lain, penelitian ini bermanfaat dalam menjelaskan pendekatan yang paling tepat dalam membangun upaya digitalisasi aktivitas pemantauan pasien di rumah sakit.

I.6 Sistematika Penulisan

Pada subbab ini, akan dijelaskan sistematika penulisan tugas akhir.

BAB I: PENDAHULUAN

Bab ini memaparkan informasi mengenai latar belakang permasalahan, perumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Bab ini memaparkan kajian literatur serta teori-teori yang digunakan dalam penelitian.

BAB III: METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah dari metode pengembangan sistem yang digunakan pada penelitian.

BAB IV: ANALISIS DAN PERANCANGAN

Bab ini menerangkan proses analisis dan perancangan yang sesuai dengan kebutuhan (*requirement*) yang ada.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Bab ini berisikan pengimplementasian dari rancangan sistem yang telah dirancang pada bab sebelumnya, serta berisikan hasil pengujian dari sistem.

BAB VI: KESIMPULAN DAN SARAN

Bab ini menguraikan simpulan berdasarkan penelitian yang dilakukan serta saran untuk penelitian selanjutnya.

BAB II TINJAUAN PUSTAKA

II.1 Covid-19

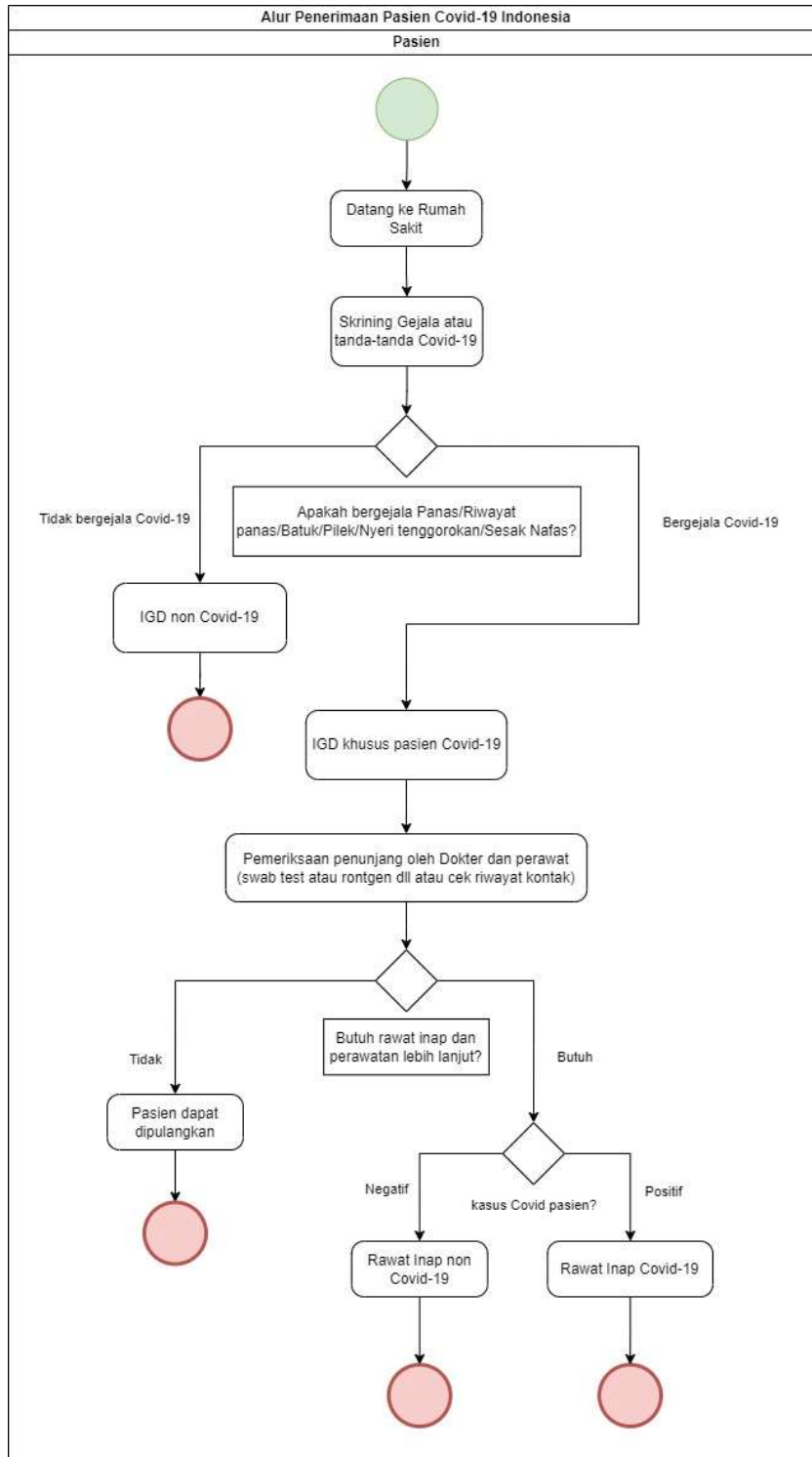
Virus Corona berasal dari Kota Wuhan, China. Pada awal tahun 2020 virus ini menyebar ke berbagai negara, termasuk Indonesia (Astari *et al.*, 2020). Virus Corona atau *severe acute respiratory syndrome coronavirus 2* (SARS-CoV-2) adalah virus yang menyerang sistem pernapasan. Penyakit akibat infeksi virus ini disebut Covid-19 (Pittara, 2022). Virus corona adalah virus yang ditularkan oleh hewan dan manusia sebagai inang. Transmisi pertama melalui agen zoonosis dari hewan ke manusia. Transmisi ke dua dari manusia ke manusia. Kemudian yang sedang diteliti adalah transmisi ke tiga dari manusia ke hewan, khususnya hewan peliharaan seperti kucing dan anjing di dalam rumah. Pusat penyebaran Covid-19 di berbagai negara dengan kasus tertinggi mempunyai iklim subtropis dan terjadi di musim dingin hingga musim semi. Covid-19 dapat menyebar melalui cipratan liur, fomite, dan aerosol dari penderita. Fomite adalah objek atau media yang dapat membawa virus, seperti pakaian, peralatan dan furnitur. Virus Covid-19 dapat menempel di berbagai media dan dapat bertahan hidup dalam beberapa jam hingga hari pada suhu 21-23°C dan 40-65% kelembaban (Firman Alamsyah, 2020).

Gejala yang ditimbulkan oleh virus corona mirip dengan flu, namun virus corona berkembang lebih cepat sehingga menyebabkan infeksi yang lebih parah dan berujung pada kegagalan organ. Tanda dan gejala yang signifikan adalah anoreksia, sesak nafas, nyeri faring, dan vertigo. Pada setiap negara dalam penentuan kriteria tanda dan gejala Covid-19 merujuk pada ketentuan WHO. Di Indonesia pedoman dalam penentuan gejala Covid-19 salah satunya merujuk pada pedoman diagnosis dan penatalaksanaan Pneumonia dan Covid-19 yaitu seseorang dikatakan Covid-19 jika mengalami tiga gejala utama demam, batuk kering (sebagian kecil berdahak) dan sulit bernafas atau sesak. Ada pada sekelompok orang tidak didapati gejala demam pada usia geriatri atau pada orang yang imunokompromais. Gejala Covid-19 juga ada beberapa tambahan gejala seperti nyeri kepala, nyeri otot, lemas, diare dan batuk darah. Selain itu juga adanya tanda dan gejala infeksi saluran nafas berat dengan ketentuan riwayat

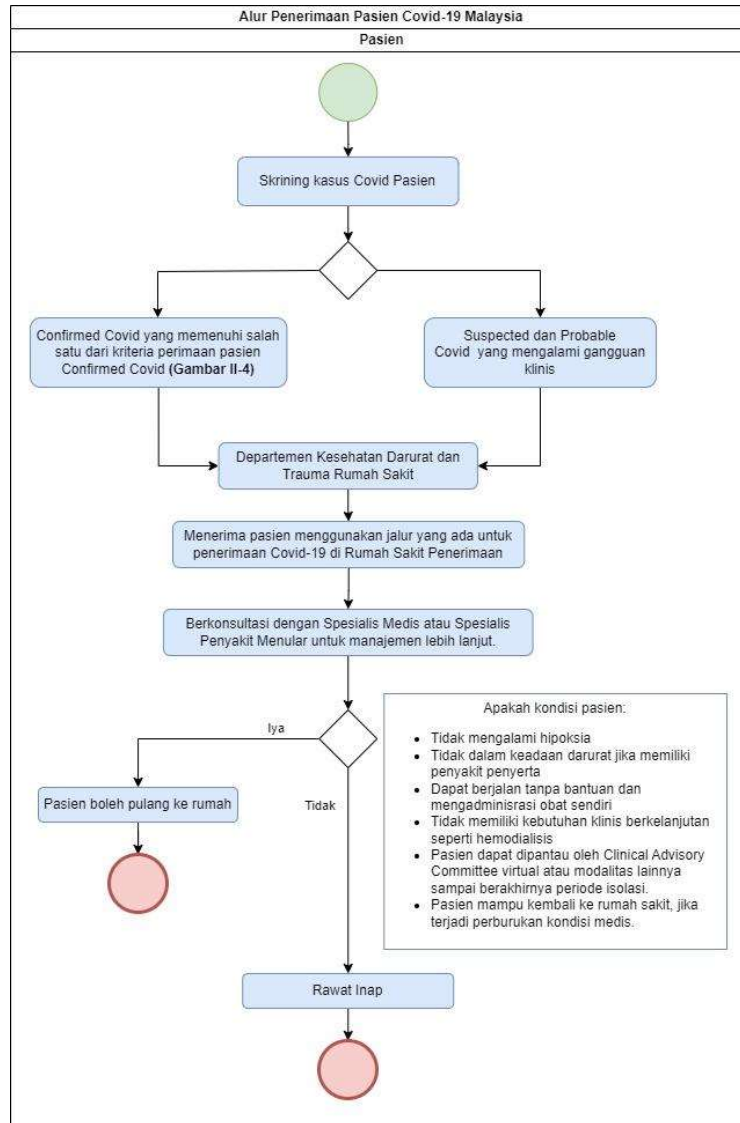
demam $\geq 38^{\circ}$ C dan batuk dengan serangan 10 hari terakhir serta perlunya rehabilitasi dirawat di Rumah Sakit. Meskipun tanda gejala Covid-19 sudah ditetapkan dengan minimal 3 gejala tetapi sebaiknya agar tetap waspada dengan senantiasa mematuhi protokol kesehatan dan menjaga Perilaku Hidup Bersih dan Sehat (PHBS) serta melakukan *screening* dengan PCR atau *rapid test*. Hal ini karena banyaknya orang tanpa gejala (OTG) yang dinyatakan positif setelah melakukan *test* (Hidayani, 2020).

Masyarakat dan individu memiliki peran penting dalam memutus mata rantai penularan Covid-19 agar tidak menimbulkan sumber penularan baru. Dengan penularan berdasarkan penularan cipratan liur dari satu orang ke orang lain, penularan dapat terjadi di rumah, saat bepergian, di tempat kerja, di tempat ibadah, di tempat wisata tempat wisata dan di tempat lain masyarakat berinteraksi sosial. Khususnya masyarakat yang berprofesi sebagai tenaga medis di rumah sakit. Berdasarkan dokumen Pedoman Pencegahan Dan Pengendalian *Coronavirus Disease (Covid-19)* Kementerian Kesehatan RI, pencegahan dan pengendalian Covid-19 yang dapat dilakukan oleh individu adalah melakukan tindakan, seperti membatasi diri terhadap interaksi / kontak dengan orang lain yang tidak diketahui status kesehatannya (Kementerian Kesehatan RI, 2020).

Agar mengurangi interaksi petugas kesehatan dengan pasien Covid-19, pemerintah juga merancang alur khusus untuk menangani pasien Covid-19 di rumah sakit sebagaimana Gambar II.1. Pasien yang datang ke rumah sakit, akan diperiksa kondisi tubuhnya. Bila menunjukkan gejala-gejala Covid-19, maka pasien akan diarahkan menuju IGD khusus pasien Covid-19. Lalu pasien akan melakukan tes Swab dan dilakukan pemeriksaan lebih lanjut. Bila pasien membutuhkan perawatan lebih lanjut dan positif Covid-19, maka pasien dirawat inap khusus pasien Covid-19.



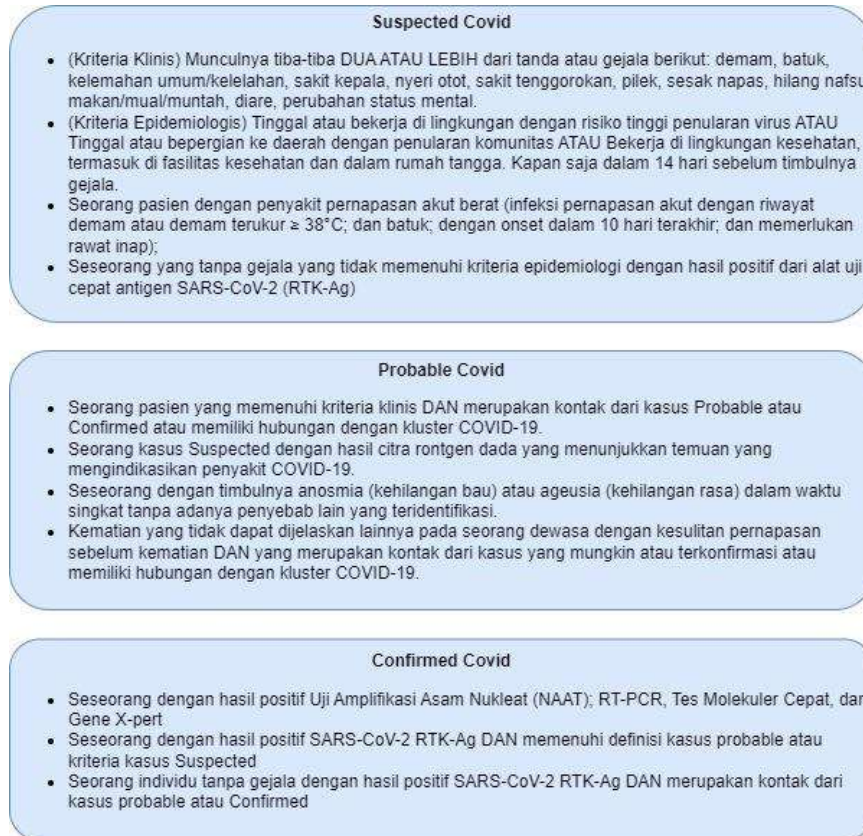
Gambar II.1 Proses pelayanan pasien Covid-19 di Rumah Sakit Indonesia (Direktorat Jenderal Pelayanan Kesehatan; Direktorat Pelayanan Kesehatan Rujukan, 2021)



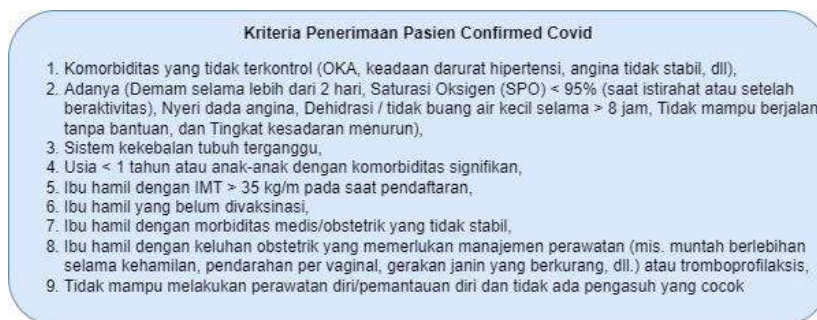
Gambar II.2 Proses pelayanan pasien Covid-19 di Rumah Sakit Malaysia
(Kementerian Kesehatan Malaysia, 2020)

Tidak hanya di Indonesia, Negeri Jiran (Malaysia) juga merancang alur khusus untuk menangani pasien Covid-19 di rumah sakit sebagaimana Gambar II.2. Kementerian kesehatan Malaysia membagi pasien menjadi berdasarkan kasus Covid-19 menjadi tiga sebagaimana yang diilustrasikan pada Gambar II.3. Pasien akan diterima oleh Departemen Kesehatan Darurat dan Trauma Rumah Sakit bila memenuhi kriteria pasien yang ditentukan. Bagi pasien *Confirmed* Covid-19, akan diterima bila memenuhi salah satu kriteria pada Gambar II.4. Adapun bagi pasien *Suspected* dan *Probable* Covid akan diterima bila mengalami gangguan klinis.

Dalam Departemen Kesehatan Darurat dan Trauma Rumah Sakit, pasien akan diterima melalui jalur khusus pasien Covid-19. Setelah itu, pasien akan berkonsultasi dengan dokter atau spesialis medis. Pasien akan dirawat inap bila kondisinya tidak memenuhi kondisi pasien yang diperbolehkan pulang.



Gambar II.3 Jenis-jenis Kasus Pasien Covid-19 (Kementerian Kesehatan Malaysia, 2020)



Gambar II.4 Kriteria Penerimaan Pasien *Confirmed Covid* (Kementerian Kesehatan Malaysia, 2020)

II.2 Sistem *Monitoring*

Sistem *Monitoring* adalah suatu sistem yang dirancang untuk melakukan proses pengumpulan data *real-time* dan melakukan analisis terhadap data-data tersebut dari berbagai sumber daya. Menurut (Fahyurisandi & Neforawati, 2020) proses sistem *monitoring* sendiri terbagi menjadi tiga proses utama, yaitu Proses di dalam pengumpulan data *monitoring*, Proses di dalam analisis data *monitoring*, dan Proses di dalam menampilkan data hasil *monitoring*.

Bagi pasien Covid-19, di antara tanda-tanda vital yang harus dipantau adalah SpO₂ (*Oxygen Saturation*), *heartrate*, *resporation rate*, tekanan darah, serta suhu badan. Menurut (Carter dkk., 2020), hal-hal yang penting untuk diperhatikan pada pasien adalah seberapa cepat pernapasannya, kadar oksigen dalam darah, denyut jantung, tekanan darah, suhu tubuh, tingkat kesadaran yang dinilai dengan cara apakah pasien responsif terhadap suara, rabaan, atau nyeri, serta skor rasa nyeri dan produksi urine. Covid-19 telah mengubah situasinya, dengan banyak pasien sekarang datang dengan masalah pernapasan yang parah yang dapat memburuk dengan cepat.

Proses dalam analisis data tanda vital pasien akan diolah dalam sistem web. Sistem akan memberikan hasil kondisi pasien berdasarkan parameter *Early Warning Scoring* (EWS). EWS merupakan alat yang membantu tim perawatan di ruang rawat untuk mengidentifikasi kondisi pasien dengan menggabungkan pengamatan teratur dan perhitungan skor fisiologis. Pada kasus Covid-19 yang parah, banyak pasien sekarang datang dengan kondisi yang buruk. Hal itu mengakibatkan EWS pasien-pasien ini akan mendapatkan skor tinggi. Akibatnya, perawat sekarang perlu secara teratur mencatat tanda-tanda vital, mengamati tanda dan gejala klinis tambahan, serta mengidentifikasi tren dari tanda-tanda vital pasien (Carter dkk., 2020).

Physiological parameter	Score						
	3	2	1	0	1	2	3
Respiration rate (per minute)	≤8		9–11	12–20		21–24	≥25
SpO ₂ Scale 1 (%)	≤91	92–93	94–95	≥96			
SpO ₂ Scale 2 (%)	≤83	84–85	86–87	88–92 ≥93 on air	93–94 on oxygen	95–96 on oxygen	≥97 on oxygen
Air or oxygen?		Oxygen		Air			
Systolic blood pressure (mmHg)	≤90	91–100	101–110	111–219			≥220
Pulse (per minute)	≤40		41–50	51–90	91–110	111–130	≥131
Consciousness				Alert			CVPU
Temperature (°C)	≤35.0		35.1–36.0	36.1–38.0	38.1–39.0	≥39.1	

Gambar II.5 Tabel *Early Warning Scoring* (Carter dkk., 2020).

II.3 *Software Engineering*

Software Engineering atau biasa disebut rekayasa perangkat lunak memiliki banyak pengertian. Menurut Stephen R. Schach rekayasa perangkat lunak adalah sebuah disiplin dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tepat waktu serta memuaskan keinginan pemakai. Menurut Fritz Bauer rekayasa perangkat lunak adalah penetapan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata. Menurut IEEE 610.12 rekayasa perangkat lunak adalah sebuah studi pendekatan dan aplikasi secara sistematis, disiplin pengembangan operasi dan pemeliharaan perangkat lunak yang semuanya itu merupakan aplikasi rekayasa yang berkaitan dengan perangkat lunak (HERI SANTOSO, 2019).



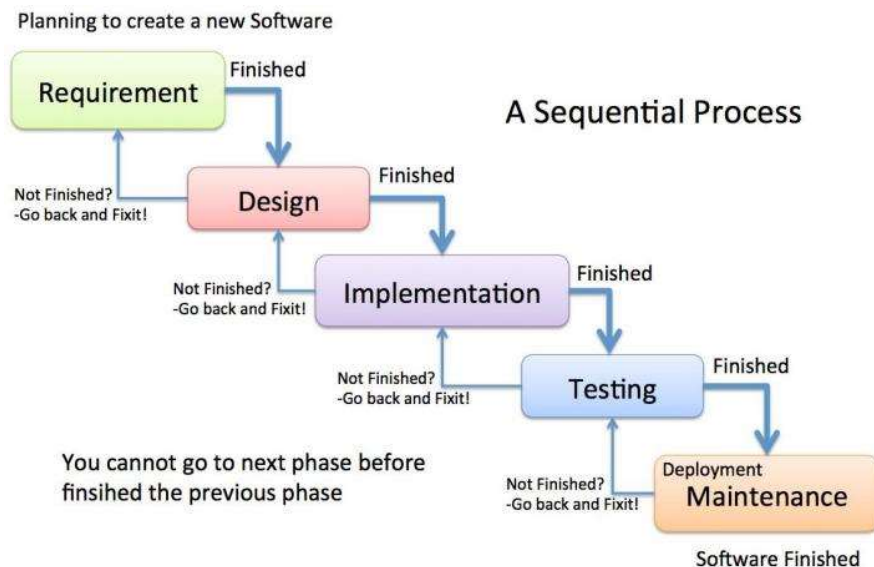
Gambar II.6 Layer *Software Engineering* (PRESSMAN & MAXIM, 2020)

Rekayasa perangkat lunak adalah teknologi berlapis, yaitu *Quality Focus*, *Process*, *Methods*, dan *Tools*. Ke-lima lapisan teknologi itu harus bertumpu pada

komitmen organisasi terhadap kualitas. Anda mungkin pernah mendengar tentang manajemen kualitas total (*Total Quality Management*) atau *Six Sigma*, dan filosofi serupa yang menumbuhkan budaya perbaikan proses yang berkelanjutan. Budaya inilah yang pada akhirnya mengarah pada pendekatan yang lebih efektif untuk rekayasa perangkat lunak. Landasan yang mendukung rekayasa perangkat lunak adalah fokus kualitas. Proses rekayasa perangkat lunak adalah perekat yang menyatukan lapisan teknologi dan memungkinkan pengembangan perangkat lunak komputer yang rasional dan tepat waktu. Metode rekayasa perangkat lunak menyediakan cara teknis untuk membangun perangkat lunak. Metode mencakup beragam tugas yang mencakup komunikasi, analisis persyaratan, pemodelan desain, konstruksi program, pengujian, dan dukungan. Alat rekayasa perangkat lunak memberikan dukungan otomatis atau semi otomatis untuk proses dan metode (PRESSMAN & MAXIM, 2020).

Dalam pengembangan dan perancangan *website* tentunya dibutuhkan metode atau proses model. Proses model ini memiliki kelebihan dan kekurangan masing-masing. Di antara proses model tersebut adalah

1. *Waterfall Process Model*

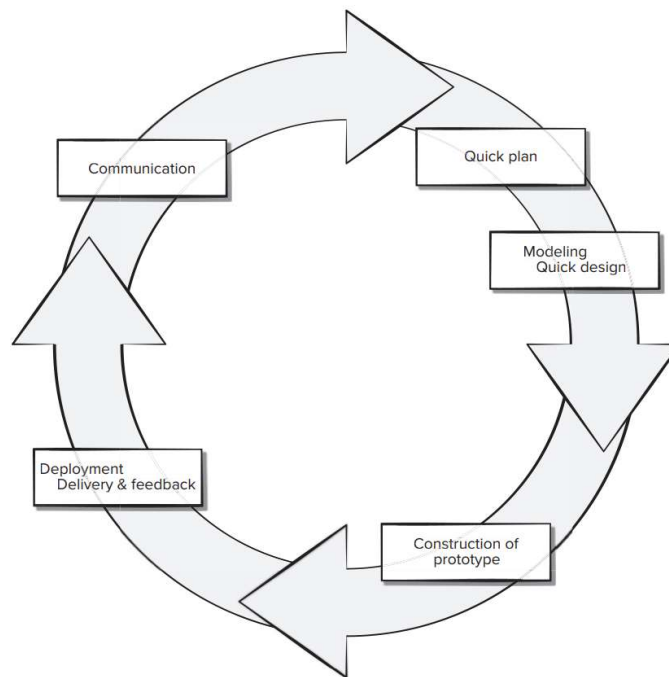


Gambar II.5 Waterfall Model (Halvorsen, 2020)

Model *Waterfall* (Gambar II.5) cukup populer di awal pengembangan perangkat lunak. Menurut (Halvorsen, 2020), model *Waterfall* terdiri dari fase-fase berikut,

Spesifikasi kebutuhan (Analisis kebutuhan), Desain perangkat lunak, Implementasi dan Integrasi, Pengujian (atau Validasi), Penerapan (atau Instalasi), dan Pemeliharaan. Model *Waterfall* menyatakan bahwa suatu proyek memiliki beberapa tahap persyaratan, analisis, desain, kode, pengujian, dan pemeliharaan dan implementasinya berurutan, artinya tahap berikutnya tidak dapat dimulai sebelum tahap sebelumnya ditutup, didokumentasikan, dan disetujui (Filipova & Vilão, 2018).

2. *Prototyping Process Model*

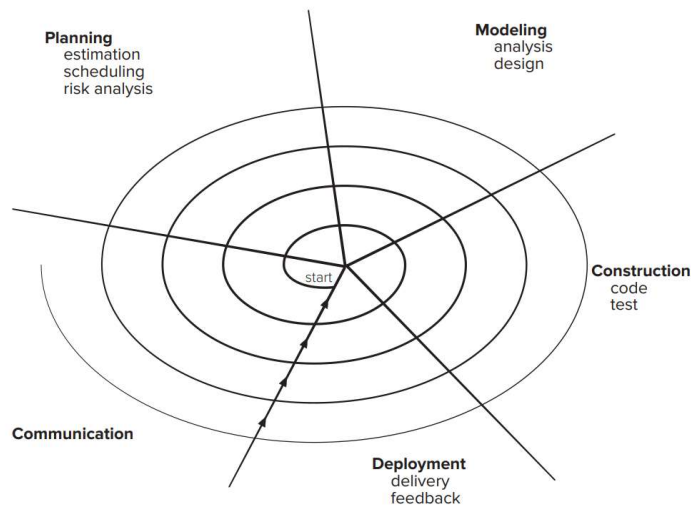


Gambar II.6 *Prototyping Process Model* (Pressman, 2020)

Model *prototyping* membantu pengguna dan pemangku kepentingan lainnya untuk lebih memahami apa yang akan dibangun saat persyaratan tidak jelas. Idealnya, prototipe berfungsi sebagai mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Proses model *prototyping* dimulai dengan komunikasi. Pengembang bertemu dengan *user* untuk menentukan tujuan keseluruhan perangkat lunak, mengidentifikasi persyaratan apa pun yang diketahui, dan garis besar area di mana definisi lebih lanjut wajib. Iterasi pembuatan prototipe direncanakan dengan cepat, dan pemodelan (dalam bentuk "desain cepat") terjadi. Desain cepat berfokus pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh pengguna akhir

(misalnya, tata letak antarmuka manusia atau format tampilan keluaran). Desain cepat mengarah pada pembangunan prototipe. Prototipe disebar dan dievaluasi melalui umpan balik yang pengembang berikan. Umpan balik tersebut akan digunakan untuk menyempurnakan proses-proses berikutnya. Iterasi terjadi saat prototipe diatur untuk memenuhi kebutuhan berbagai pemangku kepentingan yang belum terpenuhi, sekaligus memungkinkan pengembang memahami dengan lebih baik apa yang perlu dilakukan. (PRESSMAN & MAXIM, 2020).

3. Spiral Proses Model



Gambar II.7 *Prototyping Process Model* (Pressman, 2020)

Model Spiral merupakan salah satu bentuk penting dari Siklus Hidup Pengembangan Perangkat Lunak (*Software Development Life Cycle*) yang memiliki peran yang sangat signifikan. Model ini memiliki fokus yang kuat pada pengelolaan risiko dalam proses pengembangan. Setiap siklus spiral dalam model ini merujuk pada tahapan dalam proses pengembangan perangkat lunak. Jumlah langkah yang diperlukan dalam pengembangan produk bisa disesuaikan oleh pengelola proyek berdasarkan tingkat risiko yang diidentifikasi dalam proyek tersebut. Karena penentuan jumlah langkah ini merupakan hal yang krusial, peran manajer proyek sangat penting dalam penerapan model Spiral dalam pengembangan perangkat lunak (Doshi dkk., 2021).

Setelah memaparkan dan mendefinisikan tiga metode pengembangan perangkat lunak, akan dikomparasikan kelebihan dan kekurangan metode-metode tersebut.

Tabel II.1 Perbandingan metode pengembangan perangkat lunak (Narayan, STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES, 2021)

Metode pengembangan perangkat lunak	Kelebihan	Kekurangan
<i>Waterfall Process Model</i>	<ul style="list-style-type: none"> • Mudah dipahami dan direncanakan. • Cocok untuk proyek-proyek kecil yang sudah dipahami dengan baik. • Analisis dan pengujian dilakukan dengan jelas. • Modelnya kaku tapi mudah ditangani. • Berguna dan lugas • Membantu menghemat waktu. • Memfasilitasi pengujian dan analisis. 	<ul style="list-style-type: none"> • Tidak dapat menyesuaikan perubahan dengan baik. • Pengujian dilakukan terlambat dalam proses. • Persetujuan dari pelanggan dilakukan pada akhir proses. • Tergantung pada kebutuhan yang tepat. • Tidak membantu dalam mempertahankan proyek. • Tidak ada pilihan untuk memahami hasil proyek. • Ini tidak baik untuk proyek perangkat lunak yang diperpanjang dan berkelanjutan.
<i>Prototyping Process Model</i>	<ul style="list-style-type: none"> • Dampak perubahan kebutuhan berkurang. • Pelanggan sering terlibat. • Cocok untuk proyek-proyek kecil. • Kemungkinan penolakan produk berkurang. • Ide proses fungsional perangkat lunak harus jelas. • Metode ini membantu dalam mengumpulkan 	<ul style="list-style-type: none"> • Keterlibatan pelanggan dapat menyebabkan keterlambatan. • Ada godaan untuk "mengirimkan" prototipe. • Pekerjaan hilang dalam prototipe yang dibuang. • Sulit untuk merencanakan dan mengelola.

Metode pengembangan perangkat lunak	Kelebihan	Kekurangan
	kebutuhan informasi dan menganalisisnya <ul style="list-style-type: none"> • Risiko kegagalan fungsi perangkat lunak akan berkurang. 	<ul style="list-style-type: none"> • Inferensi pelanggan sangat memengaruhi pemrosesan. • Banyak perubahan yang sering mempengaruhi proses kerja perangkat lunak. • Biaya manajemen dapat meningkat.
Spiral Proses Model	<ul style="list-style-type: none"> • Terdapat keterlibatan pelanggan secara berkelanjutan. • Risiko pengembangan dikelola dengan baik. • Cocok untuk proyek-proyek besar dan kompleks. • Cocok untuk produk yang dapat diperluas. 	<ul style="list-style-type: none"> • Kegagalan analisis risiko dapat mengancam proyek. • Proyek dapat sulit untuk dikelola. • Membutuhkan tim pengembangan yang ahli.

II.4 Website Architecture

Arsitektur web adalah mekanisme atau struktur konseptual yang memungkinkan komponen aplikasi *client* dan server saling terhubung. Sistem web pada awalnya didukung oleh arsitektur klien-server dan tiga standar (URL, HTTP, dan HTML), dan memiliki evolusi yang cukup besar dalam dua dekade terakhir. Sistem web dapat dianggap sebagai semacam model arsitektur klien-server. Dalam skenario ini, browser Web mewakili klien yang menafsirkan kode HTML, CSS, dan JavaScript. Selain itu, ia berkomunikasi dengan server menggunakan URL dan protokol HTTP. Pada awalnya, setiap halaman Web dikirim ke browser sebagai dokumen statis dan tanggung jawab server hanya untuk menerima permintaan untuk menemukan dan mengirim *file*. Namun, server sekarang dapat menghasilkan halaman dinamis untuk setiap permintaan dengan menjalankan perangkat lunak, mengakses *database*, atau berintegrasi dengan sistem lain. Selain itu, halaman Web juga dapat mengeksekusi kode di sisi klien (Kulesza dkk., 2020).

Arsitektur sistem web pada awalnya, sistem dikembangkan menggunakan arsitektur CGI (*Common Gateway Interface*). Setelah itu, sejumlah arsitektur muncul, seperti arsitektur "Model 2" MVC, yang kemudian menjadi salah satu model sistem Web utama dan mendorong teknologi seperti *Struts*, *Tapestry*, dan *Java Server Pages* (JSF). Juga pada saat itu, kerangka kerja dikembangkan untuk memfasilitasi pemetaan antara model berorientasi objek dan model relasional, seperti *Hibernate*, yang berfungsi sebagai dasar untuk arsitektur 3-tier (presentasi, logika bisnis, dan data) (Kulesza dkk., 2020).

II.5 Website Development

Dalam mengembangkan situs web, diperlukan beberapa unsur-unsur untuk merancangannya seperti bahasa pemrograman, *software* pendukung, metode pengembangan *software*, dll. Terdapat tiga hal utama dalam mengembangkan *website*, yaitu HTML, CSS, dan JavaScript (Js). HTML (*Hypertext Markup Language*) merupakan gabungan dari dua istilah *hypertext* dan *markup language*. *Hypertext* yaitu dokumen berisi tautan yang memungkinkan pengguna terhubung ke halaman lain. *markup language* merupakan bahasa komputer yang terdiri dari sekumpulan kode untuk mengatur struktur dan menyajikan informasi. HTML merupakan *building block* web paling dasar (Sama & Hartanto, 2021). HTML disebut dengan tag. Tag digunakan untuk menentukan tampilan dari dokumen HTML. Tag HTML berfungsi untuk mendefinisikan bahwa isi dalam *file* tersebut adalah dokumen (Mariko, 2019). Gambar II.7 merupakan contoh struktur pemrograman HTML.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Gambar II.7 Sorce code HTML

Adapun JavaScript, adalah bahasa pemrograman web berbentuk kumpulan *script* yang berjalan pada suatu dokumen HTML JavaScript dapat menyempurnakan tampilan dan sistem pada halaman *web-based application* yang dikembangkan. Adapun karakteristik dari bahasa pemrograman *JavaScript* adalah: (1) Bahasa pemrograman berjenis *high-level programming*; (2) Bersifat *client-side*; (3) Berorientasi pada objek, dan (4) Bersifat *loosely typed* (Mariko, 2019). JavaScript bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*. Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti PHP, dimana untuk server *side* seluruh kode program dijalankan di sisi server (Setiawan, 2019). Berikut adalah Struktur dari pemrograman JavaScript.

```
9 <body>
10 <script>
11   console.log('Assalamualaikum')
12 </script>
13 </body>
```

Gambar II.8 Source code JavaScript

Pada JavaScript sendiri terdapat beberapa *framework* yang membantu developer untuk merancang atau mengembangkan *website*, di antaranya *NextJS*.

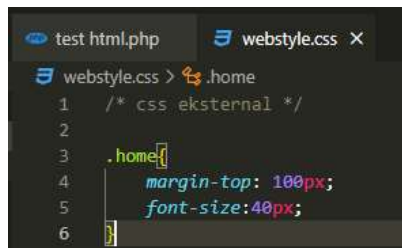
Sedangkan CSS merupakan kependekan dari *Cascading Style Sheet* merupakan suatu bahasa pemrograman untuk mengatur *style* pada bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen sistem sehingga tampilan dapat dibuat *mobile* ataupun web yang akan dapat membuat lebih rapi, terstruktur serta seragam (Reza & Putra, 2021). CSS berfungsi untuk mengatur tampilan dengan kemampuan jauh lebih baik dari tag maupun atribut standar HTML. CSS sebenarnya adalah suatu kumpulan atribut untuk fungsi format tampilan dan dapat digunakan untuk mengontrol tampilan banyak dokumen secara bersamaan. Keuntungan menggunakan CSS yaitu jika ingin mengubah format dokumen, maka tidak perlu mengedit satu persatu (Saed Novendri dkk., 2019). CSS dibagi menjadi 3 berdasarkan penempatannya. *Inline*, *Internal*, dan *Eksternal* CSS. Berikut *syntax* dari CSS


```
<div style="margin-top: 100px; font-size:40px;">
  <p>
    Ini adalah websiteku
  </p>
</div>
```

Gambar II.9 *Inline Css*

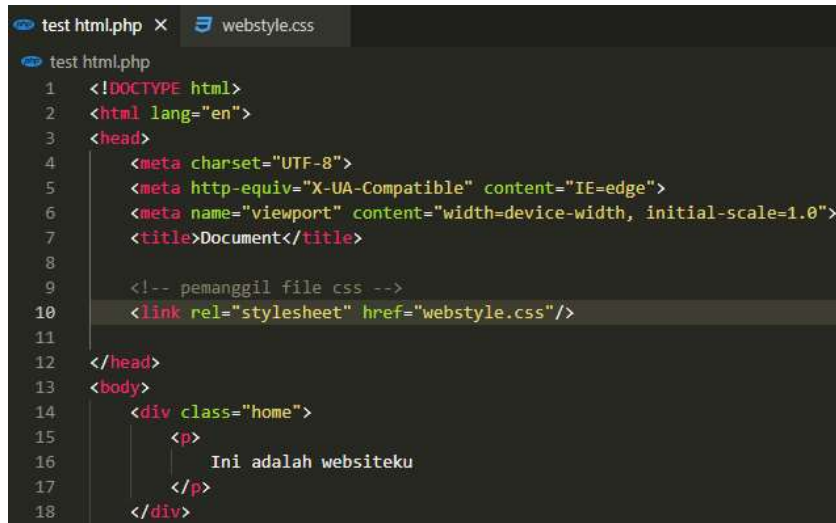
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <!-- Ini CSS Internal -->
10  <style |
11    .home{
12      margin-top: 100px;
13      font-size:40px;
14    }
15  </style>
16 </head>
17 <body>
18   <div class="home">
19     <p>
20       Ini adalah websiteku
21     </p>
22   </div>
```

Gambar II.10 *Internal Css*



```
test.html.php  webstyle.css x
webstyle.css > .home
1 /* css eksternal */
2
3 .home{
4   margin-top: 100px;
5   font-size:40px;
6 }
```

Gambar II.11 *Eksternal Css*



```
test html.php x webstyle.css
test html.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8
9   <!-- pemanggil file css -->
10  <link rel="stylesheet" href="webstyle.css"/>
11
12 </head>
13 <body>
14   <div class="home">
15     <p>
16       Ini adalah websiteku
17     </p>
18   </div>
```

Gambar II.12 Baris 10 untuk memanggil *file css* eksternal

Agar bahasa pemrograman tersebut dapat menampilkan suatu tampilan *website*, diperlukan *software* pendukung untuk mengelola bahasa pemrograman tersebut menjadi suatu tampilan *website* seperti Visual Studio Code, Web Browser, Xampp, MySQL, dan Git Hub.

Visual Studio Code merupakan sebuah aplikasi editor *code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS. Visual Code memudahkan dalam penulisan *code* yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. Visual Code memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. Visual Studio Code juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di Visual Studio Code (Ramdhan & Nufriana, 2019).

Web browser adalah sebuah perangkat lunak atau *software* yang berfungsi untuk menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web. Dengan web browser kita dapat memperoleh informasi yang disediakan oleh server web (Khorib, 2021). Dalam pengembangan aplikasi web, web browser sangat berguna untuk membantu developer dalam mengembangkan tampilan dari suatu *website*.

Istilah SQL dapat diartikan sebagai suatu bahasa yang digunakan untuk mengakses suatu data dalam *database* relasional dan terstruktur sedangkan MySQL adalah *software* atau *tools* untuk mengelola atau manajemen SQL seperti Oracle, MS SQL, PostgreSQL dan lainnya dengan menggunakan Query atau bahasa khusus. MySQL juga merupakan salah satu *software* yang bersifat *open source*. MySQL berkemampuan *multi-platform* dan berlisensi GPL, sehingga dapat digunakan oleh komputer hampir di semua OS. Beberapa fitur lain yang ada pada MySQL saat ini tersedianya tipe data yang sangat beragam seperti Float, Double, Char, Date dan lain – lain. MySQL juga mendukung penggunaan *field* sebagai *index* serta memiliki tingkat keamanan yang cukup bagus dengan adanya *Subnetmask*, nama Host serta sandi yang terenkripsi (Setiawan, 2019).

XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. XAMPP itu sendiri merupakan sebuah perangkat lunak yang fungsinya digunakan sebagai server dimana di dalamnya terdapat beberapa jenis perangkat lainnya. Dalam XAMPP v3.2.2 yang digunakan untuk windows terdapat beberapa perangkat lunak yang ada di dalamnya yaitu Apache, MySQL, FileZilla, Mercury, dan Tomcat (Setiawan, 2019).

Github merupakan salah satu lokasi populer dalam mempublikasikan *project open source*, karena sifatnya yang gratis dan memiliki komunitas yang besar. Oleh karena itu, banyak perangkat lunak developer baik dari yang sifatnya kecil sampai perusahaan besar, mempublikasikan *project open sourcenya* dalam Github (Herwanto, 2019). Github memfasilitasi para *programmer* untuk mengembangkan aplikasi-aplikasi dalam satu platform yang bisa digunakan untuk mempermudah kegiatan berkolaborasi *project*, di antaranya : (1) Menyediakan *free hosting* bagi proyek *open source*, (2) Bisa digunakan untuk bekerja secara tim dan tidak di satu tempat, (3) Mempunyai fitur seperti sosial media, bisa memfollow profil *programmer* yang lain di GitHub, (4) Mendukung semua bahasa pemrograman, (5) Memonitor repository tertentu. Jika ada perubahan pada *Repository* maka akan muncul notifikasi, (6) *History* untuk penanda siapa yang mengubah, menambahkan, menggabungkan *source code*, (7) Bisa melihat dan membagikan *project* yang telah kita buat ke pada *programmer* yang lain, (8) Bisa memperbaiki

source code secara bersama dengan tim dalam satu media GitHub tersebut,(9) dan lain sebagainya. Dengan begitu GitHub bisa digunakan sebagai situs *repository source code* yang dirancang untuk melayani pendistribusian *source code* hasil belajar siswa secara terbuka, dan memonitor *project* sejak awal sehingga dapat mempercepat waktu dalam hal manajemen *project* perangkat lunak (Sari & Ekohariadi, 2021).

II.6 Internet of Things (IoT)

Internet of things atau biasa disingkat IoT adalah suatu objek atau perangkat yang dapat melakukan pertukaran data dengan perangkat lain melalui jaringan tanpa ada campur tangan manusia. Istilah IoT (*Internet of Things*) diperkenalkan pertama kali oleh Kevin Ashton, *cofounder and executive director of the Auto-ID Center* di MIT pada tahun 1999. IoT sangat erat hubungannya dengan komunikasi *machine-to-machine* (M2M). Perangkat yang dapat melakukan komunikasi M2M akan menyelesaikan tugas-tugasnya tanpa ada campur tangan manusia. Dikarenakan kemampuannya itu, perangkat yang memiliki kemampuan komunikasi M2M lebih dikenal dengan istilah perangkat cerdas (*smart device*) (Fahyurisandi & Neforawati, 2020).

Internet of Things (IoT) di sektor kesehatan umumnya terdiri dari berbagai sensor yang terkoneksi dengan server, yang memungkinkan pemantauan langsung terhadap lingkungan atau individu (Vedaei dkk., 2020). Sensor berperan sebagai penghasil *input* dari lingkungan fisik, yang kemudian dipindahkan melalui jaringan (Kumar dkk., 2020). Di antara sensor-sensor tersebut adalah sensor detak jantung dan Spo2 (MAX30100), dan sensor tekanan darah (BP0001) (Bhardwaj dkk., 2022). Sensor DS18B20 digunakan sebagai sensor suhu tubuh (Islam dkk., 2020). Adapun sensor ADS1292R digunakan untuk mendeteksi laju pernapasan (Yakubu & Wereko, 2021).

II.7 Dashboard

Dashboard diartikan sebagai antarmuka visual yang menampilkan informasi krusial yang dibutuhkan untuk mencapai satu atau lebih tujuan; informasi ini disatukan dan disusun pada satu layar agar pengguna dapat dengan cepat memantaunya (Bach dkk., 2023). Tujuan dasbor adalah untuk mengonversi data

asli yang tersimpan dalam penyimpanan organisasi menjadi informasi yang dapat diakses (Storey & Treude, 2019). *Dashboard* memiliki banyak kegunaan seperti mendukung pengambilan keputusan di tingkat eksekutif (strategis), meringkas data tentang departemen (taktis), dan memberikan informasi untuk pekerja garis depan (operasional). *Dashboard* dapat memberikan konsistensi sehubungan dengan indikator kinerja utama dalam suatu organisasi, membantu memantau kinerja, memfasilitasi perencanaan, dan mendukung komunikasi (Bach dkk., 2023).

Dashboard kini menjadi metode yang banyak digemari untuk menghadirkan data penting dengan tampilan yang mudah dilihat, dengan tujuan menciptakan pemahaman visual yang lebih jelas saat berkomunikasi (Noonpakdee dkk., 2018). Pengembangan *dashboard* juga tengah berlangsung dengan cepat, dan digunakan secara luas di berbagai industri (Noonpakdee dkk., 2018). Dalam industri kesehatan, *dashboard* adalah aplikasi yang dapat menyimpan data kesehatan dari aplikasi lain serta memberikan tinjauan terhadap data kesehatan pengguna (Bol dkk., 2018). Sehingga pengguna mendapatkan gambaran lengkap tentang kesehatannya.

II.8 Unified Modeling Language

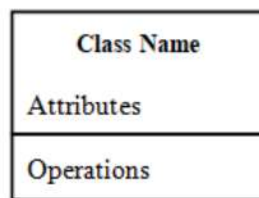
Unified Modeling Language (UML) merupakan salah satu bahasa standar yang secara luas digunakan dalam industri untuk merumuskan kebutuhan, melakukan analisis dan desain, serta mengilustrasikan struktur dalam pemrograman berbasis objek (Putra & Andriani, 2019). UML telah muncul dan memantapkan dirinya sebagai standar dalam pengembangan industri sistem perangkat lunak. Hal ini disebabkan oleh fleksibilitasnya, yang memungkinkan penggunaannya sebagai tujuan umum, dan juga kemampuannya untuk disesuaikan melalui mekanisme yang secara langsung mendukung konsep khusus untuk domain individu, organisasi, atau proyek (Ciccozzi dkk., 2019). UML didukung oleh banyak sekali alat pemodelan, misalnya, Enterprise Architect, Visual Paradigm, MagicDraw, Modelio, IBM Rational Rhapsody, dll (Ozkaya & Erata, 2020). Dengan menggunakan alat-alat ini, praktisi dapat memodelkan sistem perangkat lunak mereka dalam UML dan melakukan berbagai operasi seperti pengelolaan model,

analisis, simulasi, kolaborasi pengguna, pengelolaan proyek, transformasi model, dokumentasi, dll (Ozkaya & Erata, 2020). UML terdiri dari dua jenis, yaitu *Behavioral Modeling Diagrams* (*use case* diagram, *activity* diagram, dan *sequence* diagram) dan *Structural Modeling Diagrams* (*class* diagram dan *deployment* diagram) (Fu'adi dkk., 2022).

Use Case diagram merupakan gambaran tentang fungsionalitas yang diinginkan dari sebuah sistem, dan mewakili interaksi antara aktor dan sistem. Dalam *use case*, terdapat aktor yang menggambarkan entitas manusia atau sistem yang beroperasi dalam sistem (M Teguh Prihandoyo, 2018).

Activity diagram merupakan gambaran alur dari aktivitas-aktivitas yang terjadi dalam sistem (M Teguh Prihandoyo, 2018). Menurut Putra & Andrian (2019), *Activity* diagram adalah representasi visual yang menggambarkan alur kerja atau aktivitas dari suatu sistem yang ada dalam perangkat lunak.

Class diagram merupakan ilustrasi struktur dan karakteristik dari *class*, *package*, dan objek yang saling terhubung melalui relasi seperti pewarisan dan asosiasi (M Teguh Prihandoyo, 2018). Sebagaimana Gambar II.13, *class* diagram memiliki tiga elemen utama, yaitu nama kelas, atribut, dan operasi. Kelas-kelas didefinisikan menggunakan persegi panjang dengan nama mereka ditulis di bagian atas-tengah persegi panjang tersebut. Nama kelas ini harus bermakna. Persegi panjang ini dibagi menjadi dua bagian, di mana bagian atas berisi semua atribut kelas dan bagian bawah berisi semua operasi (Bhatt & Nandu, 2021).








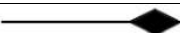
Gambar II.13 *Class* Diagram (Bhatt & Nandu, 2021)






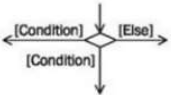

Sequence diagram menunjukkan interaksi antara objek dalam dan sekitar sistem melalui pesan yang ditampilkan pada sumbu waktu (M Teguh Prihandoyo, 2018). Semua interaksi pesan yang sudah dicakup pada *Sequence* diagram akan mencakup banyak *use case* yang telah didefinisikan, sehingga semakin banyak


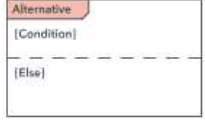
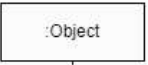



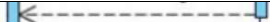

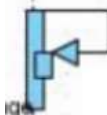

use case yang didefinisikan, maka semakin banyak pula *sequence* diagram yang harus dibuat (Putra & Andriani, 2019).

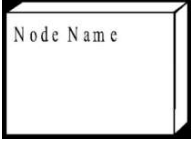
Deployment diagram adalah bentuk diagram struktur yang digunakan dalam merancang arsitektur pelaksanaan sistem dan alokasi komponen perangkat lunak pada elemen-elemen sistem (*node*) (Pérez-Castillo & Piattini, 2022). Diagram ini menghubungkan antara komponen logis dan/atau fisik dari sistem serta artefak teknologi informasi (Pérez-Castillo & Piattini, 2022). *Deployment* diagram dimanfaatkan untuk mengilustrasikan struktur fisik perangkat lunak dan perangkat keras pada perangkat atau sistem (Shi & Ge, 2020).

Tabel II.2 Notasi - Notasi UML (Hadiprakoso, 2021)

No.	Notasi	Nama Notasi	Penjelasan
1.		<i>Association</i>	Merupakan garis lurus yang memiliki panah terbuka yang bersifat opsional, artinya dapat ditambahkan atau tidak. Bentuk relasi antara dua kelas ini umumnya disertai dengan catatan dan notasi <i>multiplicity</i> (Gambar II.14).
2.		<i>Inheritance</i>	Menggambarkan hubungan antara <i>superclass</i> dengan <i>subclass</i> .
3.		<i>Implementation/Realization</i>	Menggambarkan hubungan antara kelas <i>interface</i> dengan kelas yang mengimplementasikan <i>interface</i> tersebut
4.		<i>Dependency</i>	Menggambarkan hubungan ketergantungan. Arah tanda panah berasal dari kelas yang memiliki ketergantungan.
5.		<i>Aggregation</i>	Menggambarkan hubungan “memiliki”
6.		<i>Composition</i>	Menggambarkan hubungan “komposisi” dengan

No.	Notasi	Nama Notasi	Penjelasan
			bentuk saling ketergantungan
7.		<i>Actor</i>	Aktor adalah sesuatu yang berinteraksi dengan sistem. Aktor yang mewakili manusia digambarkan menggunakan simbol sederhana orang, sedangkan aktor yang mewakili sistem komputer atau entitas selain manusia digambarkan menggunakan simbol kotak dengan label <<actor>>
8.		<i>Use Case</i>	Aktivitas yang dapat aktor jalankan dalam suatu sistem
9.	<<include >>	<i>Include</i>	Menggunakan kembali behaviour yang digunakan oleh use case lain
10.	<<exclude>>	<i>Exclude</i>	Hubungan antar use case yang bersifat pilihan atau hanya berlaku dalam situasi tertentu.
11.		<i>Initial Node</i>	Simpul atau titik awal dari suatu diagram
12.		<i>Activity Final Node</i>	Simpul akhir yang menunjukkan bahwa suatu urutan aktivitas telah selesai.
13.		<i>Activity</i>	Langkah individu dalam suatu activity
14.		<i>Decission Node</i>	Menggambarakan action dalam bentuk percabangan atau kondisi.
15.		<i>Swim Lane</i>	Merupakan garis lurus vertikal yang membagi sebuah diagram aktivitas ke dalam beberapa bagian.

No.	Notasi	Nama Notasi	Penjelasan
16.		<i>Frame</i>	Wadah yang digunakan untuk menggambarkan <i>sequence diagram</i>
17.		<i>Alternative</i>	Mengilustrasikan pilihan antara dua atau lebih rangkaian pesan.
18.		<i>Object</i>	Entitas yang terlibat dalam interaksi atau proses yang dijelaskan dalam skenario
19.		<i>Activation Bar</i>	Waktu di mana suatu objek sedang aktif dalam melakukan operasi atau interaksi dengan objek lain.
20.		<i>Lifeline</i>	"umur" objek selama interaksi di skenario yang diberikan
21.		<i>Synchronous Message</i>	Pesan yang dikirim dari satu entitas ke entitas lain, di mana entitas pengirim harus menunggu respons dari entitas penerima sebelum bisa beraksi lagi atau mengirim pesan berikutnya.
22.		<i>Return Message</i>	Respons balik dari objek yang telah dikirim pesan,
23.		<i>Self Message</i>	Objek yang mengirimkan pesan untuk dirinya sendiri
24.		<i>Recursive Message</i>	<i>Self message</i> atau <i>Synchronous message</i> yang bersifat rekursif/berulang.
25.		<i>Component</i>	Komponen mengatasi tampilan implementasi statis dari suatu sistem. Komponen digunakan untuk memodelkan fitur fisik dari suatu sistem. Aspek fisik meliputi elemen-elemen seperti

No.	Notasi	Nama Notasi	Penjelasan
			pustaka, berkas eksekusi, berkas, dokumen, dll. yang ditempatkan dalam sebuah <i>node</i> .
26.		<i>Node</i>	Sumber daya atau media di mana artefak dikembangkan untuk dieksekusi

Notasi	Keterangan
0	Tidak ada
0..1	Tidak ada atau satu
1	Tepat satu
0..*	Tidak ada atau beberapa
1..n	Satu sampai n
1..*	Satu atau beberapa

Gambar II.14 Notasi *Multiplicity* (Hadiprakoso, 2021)

II.9 System Usability Scale

System Usability Scale (SUS) didefinisikan sebagai sebuah metode pengujian kebermanfaatan atau kebergunaan (*usability*) yang memanfaatkan sepuluh pertanyaan baku sebagai instrumen evaluasi. Keistimewaan dari metode ini adalah kemampuannya untuk memberikan hasil yang relevan tanpa memerlukan jumlah sampel yang besar, yang pada akhirnya dapat mengurangi biaya yang diperlukan (Welda dkk., 2020). Terdapat 10 pertanyaan yang digunakan untuk mengimplementasikan metode SUS. Pertanyaan nomor genap memiliki nuansa negatif, sedangkan pertanyaan nomor ganjil memiliki nuansa positif.

Tabel II.3 Pertanyaan *System Usability Testing*

No.	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.
3	Saya merasa sistem ini mudah untuk digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.

No.	Pertanyaan
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Setiap pertanyaan akan dijawab dengan 5 poin skala Likert, yaitu dimulai dari "Sangat Tidak Setuju=1", "Tidak Setuju=2", "Netral=3", "Setuju=4", hingga "Sangat Setuju=5" (Kurniawan dkk., 2022). Responden diminta untuk menilai 10 pernyataan dalam instrumen SUS sesuai dengan perspektif subjektif mereka seperti yang terlihat dalam ilustrasi di bawah ini:

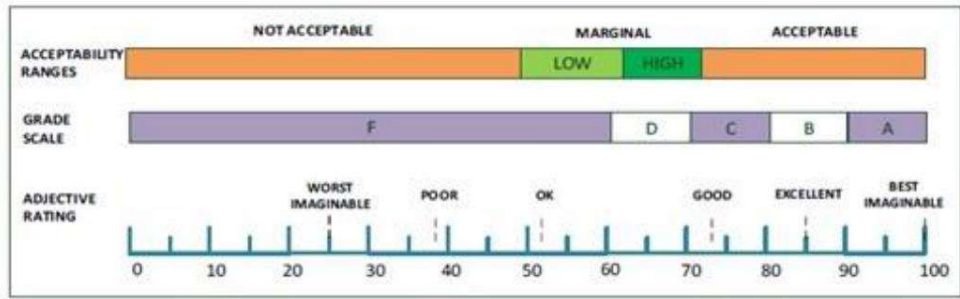


Gambar II.15 Skala Likert (Kurniawan dkk., 2022)

Masing-masing pernyataan memiliki nilai kontribusi yang akan diberikan skor, rentang skor kontribusi pernyataan adalah antara 0 hingga 4. Pada pernyataan 1, 3, 5, 7, dan 9, skor kontribusinya setara dengan skala dikurangi 1. Sementara itu, pada pernyataan 2, 4, 6, 8, dan 10, skor kontribusinya adalah 5 dikurangi posisi skala. Hasil skor kontribusi ini akan dikalikan dengan faktor 2.5 untuk menghitung nilai keseluruhan *usability* sistem (Kurniawan dkk., 2022). Nilai skor SUS akan berkisar antara 0 hingga 100. Berikut adalah rumus perhitungannya:

$$SUS\ Score = \{(Q1-1) + (5-Q2) + (Q3-1) + (5-Q4) + (Q5-1) + (5-Q6) + (Q7-1) + (5-Q8) + (Q9-1) + (5-Q10)\} * 2.5.$$

Setelah nilai *SUS score* didapat, *SUS score* akan dibandingkan dengan tabel penentuan *Acceptability*, *Grade Scale*, dan *Adjective Rating* agar diketahui *perspective* pengguna terhadap software yang dikembangkan (Welda dkk., 2020).



Gambar II.16 *Acceptability, Grade Scale, dan Adjective Rating* (Welda dkk., 2020).

II.10 State of The Art

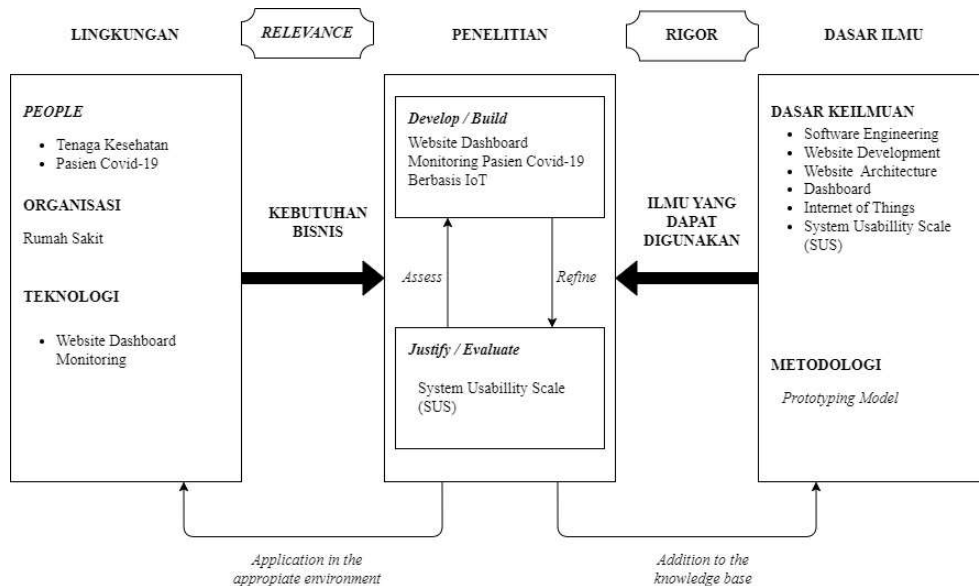
Tabel II.4 State of The Art

No	Judul Penelitian	Penulis	Rangkuman	Tahun
1	<i>Internet of things based real-time coronavirus 2019 disease patient health monitoring system</i>	Abraham Ninian Ejin, Hoe Tung Yew, Mazlina Mamat, Farrah Wong, Ali Chekima, Seng Kheau Chung	Sistem pemantauan kesehatan <i>real-time</i> berbasis <i>Internet of Things</i> (IoT) untuk pasien COVID-19 telah diusulkan. Ini memberikan pemantauan lima parameter fisiologis: elektrokardiogram (EKG), detak jantung (HR), laju pernapasan (RR), saturasi oksigen (SpO2) dan suhu tubuh. Tanda-tanda vital ini diproses oleh pengontrol utama dan dikirim ke <i>cloud</i> untuk disimpan. Dasbor berbasis web dengan layanan pemberitahuan memungkinkan profesional medis membaca data pasien vital secara <i>real time</i> . Sistem yang diusulkan dapat secara akurat mengirimkan dan menampilkan semua parameter secara <i>real time</i> tanpa kehilangan paket atau kesalahan transmisi. Akurasi pengukuran suhu, RR, SpO2, dan HR mencapai 99,7%, 100%, 97,97%, dan 98,34%. Dengan sistem yang diusulkan, petugas kesehatan dapat lebih mudah memantau pasien COVID-19 dari jarak jauh, mengurangi paparan patogen, dan meningkatkan pengawasan pasien.	2022
2	<i>COVID-19 disease: assessment of a critically ill patient</i>	Chris Carter, Helen Aedy, Joy Notter	Dalam respons terhadap peningkatan pasien COVID-19 yang parah, perawat perlu dengan cepat menilai pasien. Meskipun banyak pasien datang dengan tanda-tanda pneumonia virus dan risiko gagal pernapasan, komplikasi sistemik juga harus diidentifikasi. Karena jumlah pasien yang memerlukan perawatan kritis sangat besar, beberapa awalnya harus diurus di unit gawat darurat dan ruang rawat akut. Artikel ini memberikan panduan penilaian pasien COVID-19 yang parah, baik di ruang rawat maupun perawatan kritis, menggunakan pendekatan <i>Airway, Breathing, Circulation, Disability,</i>	2020

No	Judul Penelitian	Penulis	Rangkuman	Tahun
3	Sistem Monitoring Pasien Robot Covid dengan Parameter Suhu, Detak Jantung, Dan Saturasi Oksigen Berbasis Website.	Adi Viyo Nugroho	<p>dan <i>Exposure</i> (ABCDE). Artikel ini bertujuan membantu perawat dalam penilaian terstruktur dan prioritas perawatan.</p> <p>Untuk mengurangi intensitas kontak langsung antara tenaga medis dengan pasien positif Covid-19 banyak perguruan tinggi yang melakukan penelitian untuk menciptakan robot agar dapat dikontrol dari jarak jauh dan dapat berkomunikasi dengan pasien, termasuk Universitas Muhammadiyah Surakarta ikut melakukan penelitian robot tersebut. Robot covid ini dilengkapi dengan sensor – sensor yang berfungsi untuk menghimpun data kesehatan pasien. Data yang dihimpun meliputi ID pasien, suhu, detak jantung, dan saturasi oksigen. Sensor tersebut diprogram dengan mikrokontroler arduino dan esp82266 yang kemudian data yang terbaca oleh sensor akan dikirim secara <i>wireless</i> ke <i>website</i> dengan <i>realtime</i>. <i>Website</i> difungsikan untuk menampilkan data sensor yang dikirim secara <i>wireless</i> dari mikrokontroler robot tersebut yang kemudian akan disimpan pada <i>database webserver</i>.</p>	2020
4	<i>Digital Health Technologies Respond to the COVID-19 Pandemic In a Tertiary Hospital in China: Development and Usability Study</i>	Wanmin Lian, Li Wen, Qiru Zhou, Weijie Zhu, Wenzhou Duan, Xiongzhi Xiao, Florence Mhungu, Wenchen Huang, Chongchong Li, Weibin Cheng, Junzhang Tian	Platform layanan COVID-19 berbasis web digunakan dalam sistem perawatan kesehatan Rumah Sakit Umum Provinsi Kedua Guangdong dan Rumah Sakit Internet. Fungsi fokus dari sistem ini termasuk skrining COVID-19 otomatis, pemantauan gejala terkait, konsultasi berbasis web, dan dukungan psikologis; itu juga berfungsi sebagai pusat pengetahuan COVID-19.	2020
5	<i>Personalizing symptom monitoring and contact tracing efforts through a COVID-19 web-app</i>	Kevin Yi Lwern Yap, Qihuang Xie	Aplikasi web COVID-19 <i>Symptom Monitoring and Contact Tracing Record</i> (CoV-SCR) (http://bit.ly/covscrapp) memungkinkan individu untuk menyimpan catatan pribadi kontak dekat mereka dan memantau gejala mereka setiap hari, sehingga mereka dapat memberikan detail yang relevan dan akurat ketika mereka menemui dokter dan selama proses pelacakan kontak.	2020

BAB III METODOLOGI PENELITIAN

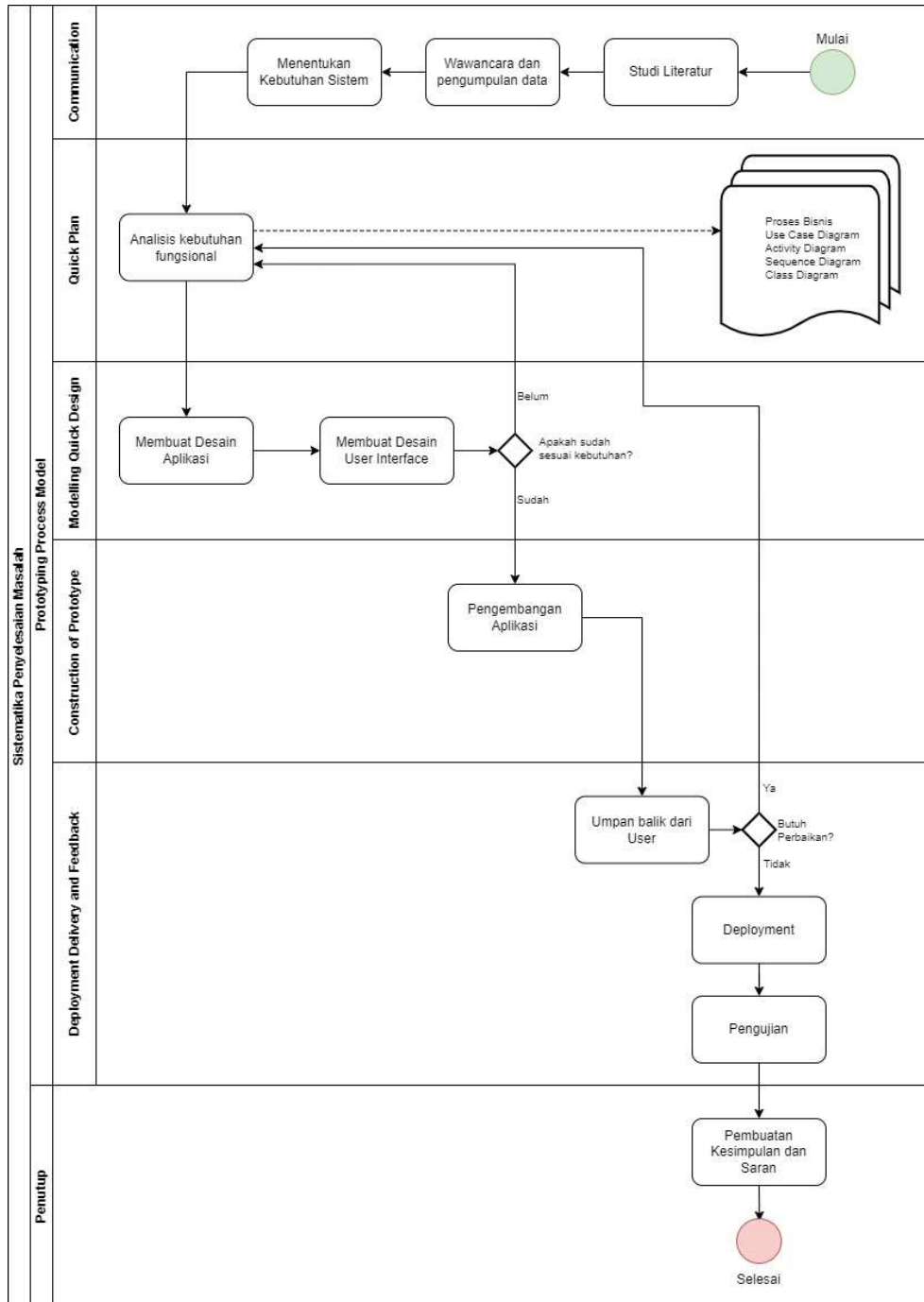
III.1 Model Konseptual



Gambar III.1 Model Konseptual

Berdasarkan ilustrasi dalam Gambar III.1, terlihat bahwa tantangan dalam lingkungan melibatkan tiga elemen utama, yakni *people*, organisasi, dan teknologi. Di dalam komponen *people*, terdapat tenaga medis dan pasien Covid-19, di mana permasalahan muncul terkait situasi di rumah sakit yang belum optimal dalam penanganan pasien Covid-19, terutama saat jumlah pasien positif meningkat. Oleh karena itu, diperlukan penerapan teknologi berupa sebuah *dashboard* berbasis *website* yang terintegrasi dengan *Internet of Things* (IoT) untuk memantau kondisi pasien di lingkungan rumah sakit. Lebih dari itu, penelitian ini mengandalkan landasan ilmiah dari disiplin seperti rekayasa perangkat lunak, pengembangan *website*, arsitektur *website*, *dashboard*, *Internet of Things*, dan *System Usability Scale* (SUS). Pendekatan metodologi yang diterapkan dalam perancangan sistem ini adalah model *prototyping*. Sementara untuk proses pengujian, *System Usability Scale* (SUS) digunakan untuk menilai aplikasi yang telah dikembangkan. Secara keseluruhan, penelitian ini diharapkan akan menghasilkan sistem *dashboard* pemantauan pasien Covid-19.

III.2 Sistematika Penyelesaian Masalah



Gambar III.2 Sistematika Penyelesaian Masalah

Dalam pengerjaan tugas akhir ini, dilakukan langkah sistematika masalah menggunakan *Prototyping Process Model*. Gambar III.2 adalah ilustrasi dari alur sistematika penyelesaian masalah pada penelitian ini. Alasan menggunakan

metode ini adalah karena *requirement* untuk fitur kurang spesifik dan dibutuhkan banyak interaksi dengan *user*. *Prototyping process* model dapat membantu untuk memahami apa yang akan dibangun ketika *requirement* kurang jelas. Tahap pengembangan aplikasi diawali dengan fase *communication* untuk menentukan kebutuhan sistem *monitoring* pasien Covid-19. Untuk menentukan kebutuhan sistem dilakukan wawancara kepada beberapa tenaga medis untuk mengetahui lebih lanjut mengenai alur penanganan dan perawatan pasien Covid-19. Selanjutnya adalah fase *quick plan* dimana pada fase ini dilakukan analisis untuk kebutuhan fungsional dari aplikasi yang akan dibuat dengan menggunakan artefak-artefak yang mendukung untuk melakukan pengembangan *dashboard* seperti proses bisnis rawat inap, *usecase* diagram, *activity* diagram, *class* diagram, *sequence* diagram, dan *deployment* diagram. Berikutnya adalah fase *modelling quick design* dimana pada fase ini dilakukan pembuatan desain *dashboard* yang terdiri dari perancangan desain *prototype*. Apabila desain *prototype* sudah sesuai dengan kebutuhan maka selanjutnya adalah fase *Construction of Prototype*. Pada fase *Construction of Prototype* dilakukan pengembangan *dashboard* sesuai dengan desain yang telah dirancang, setelah *dashboard* berhasil dikembangkan maka akan dilakukan fase *deployment*. Pada fase ini *dashboard* yang telah dikembangkan ada *user* beri *feedback*. Jika pengembangan telah sesuai, maka akan dilanjutkan dengan pengujian. Pengujian dilakukan menggunakan metode *System Usability Scale* untuk mengevaluasi pengalaman pengguna terhadap *dashoard* yang telah dibuat. Lalu pada tahap penutup, akan disusun kesimpulan dan saran dari penelitian yang dilakukan.

BAB IV ANALISIS DAN PERANCANGAN

IV.1 Fase *Communication*

Pada fase ini, dilakukan komunikasi dengan tiga tenaga kesehatan menggunakan *open card sorting* sebagai wadah pengumpulan data dan informasi yang pengguna berikan. Data dan informasi tersebut selanjutnya akan dianalisis untuk menentukan kebutuhan sistem *monitoring* Pasien Covid-19. Dengan demikian sistem yang dikembangkan dapat sesuai dengan kebutuhan pengguna. Pada *open card sorting* responden diarahkan untuk mengelompokkan kartu yang berisi fitur berdasarkan kelompok yang diajukan pada Tabel IV.2. Responden juga diperbolehkan untuk mengajukan fitur *dashboard* tambahan selain yang diajukan pada Tabel IV.1.

Tabel IV.1 Daftar Fitur *Card Sorting*

No	Daftar Fitur
1	Manajemen pasien
2	<i>Monitoring</i> tanda-tanda vital setiap Pasien
3	Hasil <i>Monitoring</i> seluruh pasien
4	Catatan Perkembangan pasien terintegrasi
5	Manajemen Bangsal
6	<i>Discharge Planning</i>
7	Order Lab
8	Order Radiologi
9	<i>Early Warning System</i>

Fitur-fitur pada Tabel IV.1 diajukan berdasarkan fitur-fitur yang terdapat pada sistem informasi rumah sakit setempat.

Tabel IV.2 Daftar Kelompok Fitur *Card Sorting*

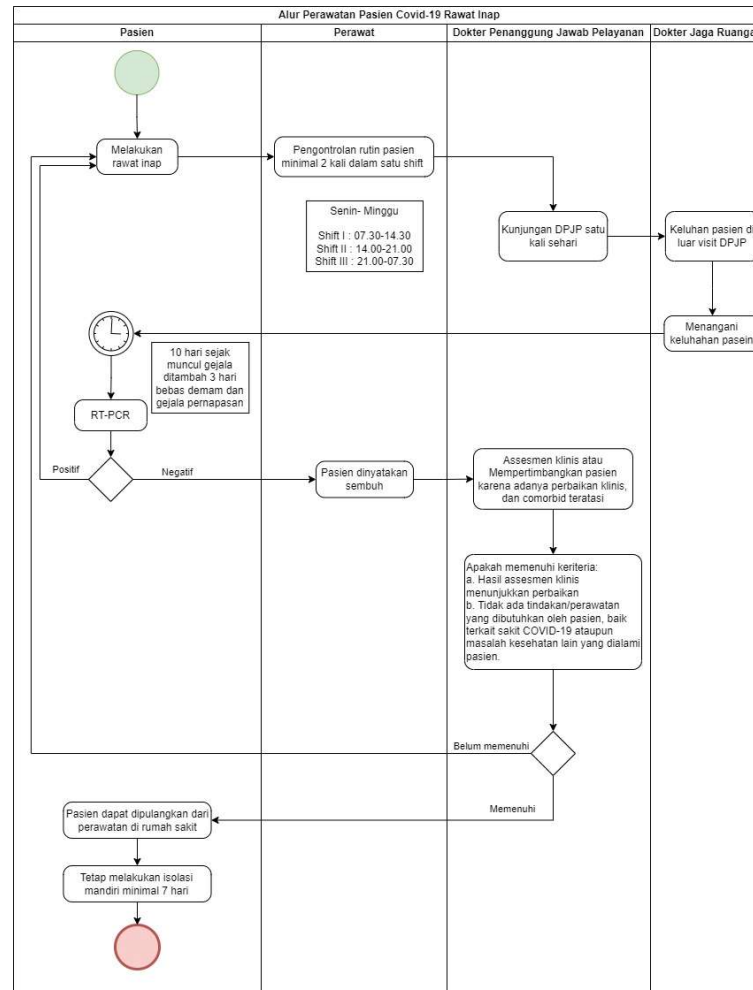
No	Daftar Kelompok Fitur
1	<i>Dashboard</i> utama
2	Rawat inap (<i>inpatient management</i>)
3	Pemeriksaan dokter
4	Pemeriksaan perawat

Setelah dilakukan *open card sorting* kepada tenaga kesehatan, hasil *card sorting* tersebut dikumpulkan dalam tabel yang diilustrasikan melalui Gambar IV.1.

No	Nama Fitur	Card Sort 1	Card Sort 2	Card Sort 3
1	Manajemen pasien	Rawat Inap	Rawat Inap	Rawat Inap
2	Monitoring TTV setiap Pasien	Pemeriksaan Dokter	Rawat Inap	Rawat Inap
3	Hasil Monitoring seluruh pasien	Dashboard Utama	Dashboard Utama	Dashboard utama
4	Catatan Perkembangan pasien terintegrasi	Pemeriksaan Dokter, Pemeriksaan Perawat	Pemeriksaan Dokter, Pemeriksaan Perawat	Pemeriksaan Dokter, Pemeriksaan Perawat
5	Manajemen Bangsal	Rawat Inap	Rawat Inap	Rawat Inap
6	Discharge Planning	Rawat Inap	Pemeriksaan Dokter	Pemeriksaan Dokter, Pemeriksaan Perawat
7	Order Lab	Pemeriksaan Penunjang	Rawat Inap	Pemeriksaan Dokter
8	Order Radiologi	Pemeriksaan Penunjang	Rawat Inap	Pemeriksaan Dokter
9	Early Warning System	Pemeriksaan Perawat	Rawat Inap	Dashboard utama
10	Grafik Jumlah Pasien	Dashboard Utama	Dashboard Utama	Dashboard Utama

Gambar IV.1 Hasil *Card Sorting* dari Fitur yang diujikan

Dari hasil *card sorting* di atas, dapat diketahui apa saja fitur-fitur yang dibutuhkan pengguna dalam sistem *monitoring* Pasien Covid-19.



Gambar IV.2 Alur Perawatan Pasien Covid-19 Rawat Inap Eksisting

Selanjutnya dilakukan analisis proses bisnis *eksisting* alur Perawatan pasien Covid-19 Rawat Inap di Rumah Sakit. Setiap saat pasien rawat inap akan dipantau oleh perawat dan dokter. Perawat akan memantau pasien dengan minimal 2 kali dalam 1 *shift*. Adapun dokter hanya akan mengunjungi pasien 1 kali sehari bila pasien tidak ada keluhan tambahan. Pasien akan dinyatakan sembuh setelah 13 hari melakukan rawat inap dan hasil tes RT-PCR negatif. Selanjutnya dokter akan melakukan assesmen klinis pasien. Pasien yang memenuhi kriteria akan diperbolehkan pulang. Namun pasien harus melakukan isolasi mandiri selama 7 hari setelah kepulangannya dari perawatannya di rumah sakit. Berdasarkan hasil *card sorting* dan proses bisnis *eksisting* alur Perawatan pasien Covid-19 Rawat Inap di Rumah Sakit, maka dapat disimpulkan bahwa peran pengguna yang akan berinteraksi dengan sistem adalah Perawat dan Dokter.

IV.2 Fase *Quick Plan*

Setelah melakukan fase *Communication* dengan pengguna, dilakukan fase *Quick Plan*. Pada fase ini akan dilakukan analisis kebutuhan sistem berdasarkan hasil *card sorting*, analisis proses bisnis *eksisting* alur perawatan pasien Covid-19 rawat inap di rumah sakit, serta merancang artefak-artefak yang mendukung untuk melakukan pengembangan *dashboard* berdasarkan analisis kebutuhan sistem itu sendiri.

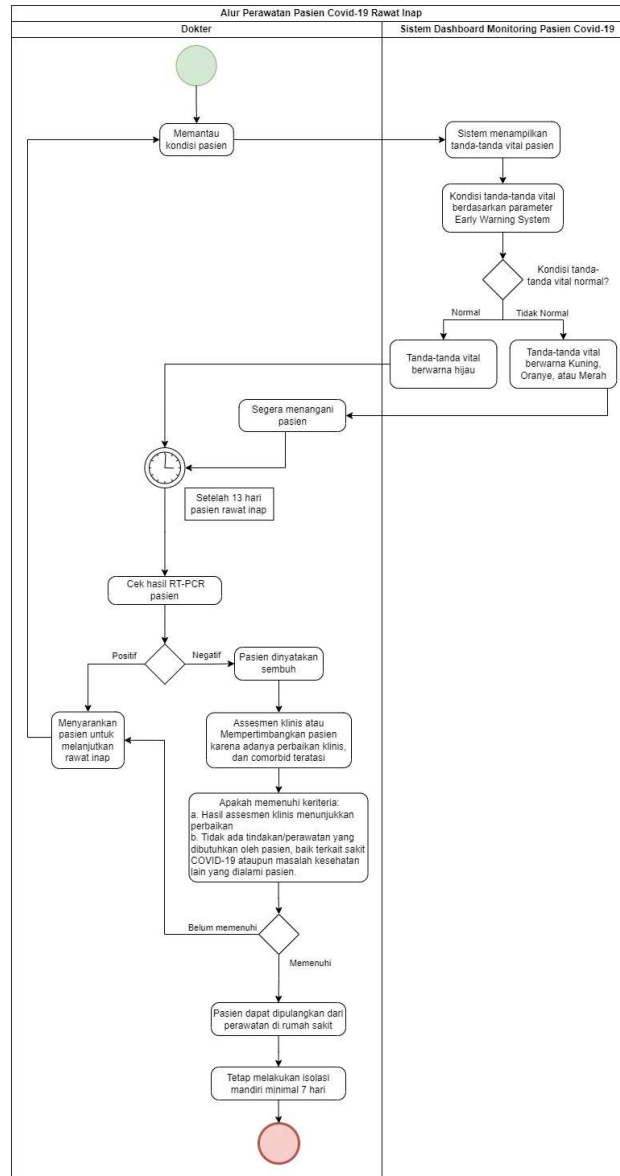
IV.2.1 Analisis Kebutuhan Sistem

Dalam bagian ini, dipaparkan hasil analisis kebutuhan sistem *monitoring* Pasien Covid-19, yang didasarkan pada pemahaman terkait hasil dari *card sorting* dan proses bisnis *eksisting* alur perawatan pasien Covid-19 rawat inap di rumah sakit. Analisis kebutuhan ini akan menjadi landasan penting dalam merancang solusi sistem yang sesuai dengan kebutuhan pengguna, sehingga dapat meningkatkan efisiensi dan efektivitas dalam penanganan pasien Covid-19 di lingkungan rumah sakit.

Tabel IV.3 Kebutuhan Sistem *Dashboard*

No	Kebutuhan Sistem	Deskripsi	Aktor
1.	Tampilan <i>Dashboard</i>	Melihat tampilan <i>dashboard</i> yang terdiri dari total pasien, grafik perbandingan kasus pasien, serta tabel EWS pasien.	Perawat dan Dokter
2.	Tampilan <i>Patient Data</i>	Melihat tampilan data pasien yang memuat informasi terkait kondisinya berdasarkan parameter EWS.	Perawat dan Dokter
3.	Tampilan <i>Detail Pasien</i>	Melihat tampilan grafik dari parameter kondisi pasien sesuai dengan data yang telah didapatkan dari API.	Perawat dan Dokter
4.	Tampilan <i>Ward Data</i>	Melihat tampilan data dari bangsal, ruangan, dan fasilitas dari ruangan yang didapat oleh pasien	Perawat dan Dokter
5.	Tampilan <i>Early Warning System (EWS)</i>	Melihat tampilan dari pasien yang memiliki kondisi yang tergolong parah karena memiliki parameter kondisi yang tidak normal, sehingga perlu mendapatkan perhatian khusus.	Perawat dan Dokter
6.	Tampilan master data	Melihat data master dari pasien, dokter, dan ruangan	Perawat dan Dokter

IV.2.2 Proses Bisnis *Targeting*

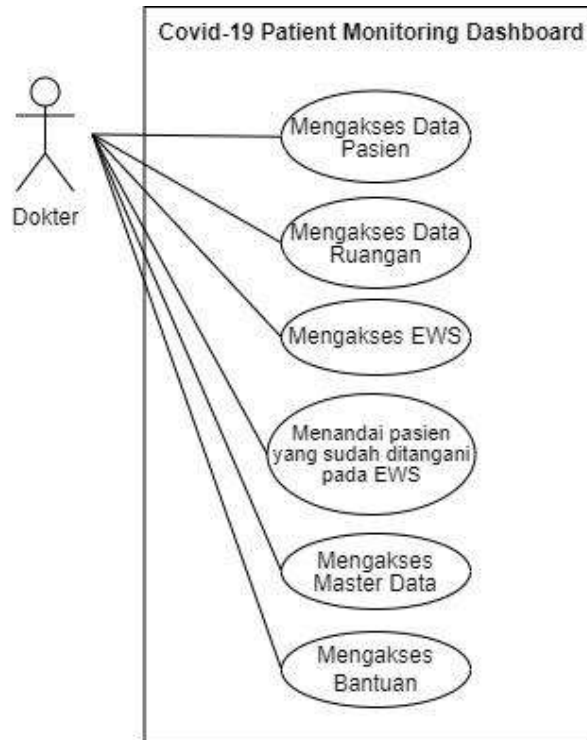


Gambar IV.3 Alur Perawatan Pasien Covid-19 Rawat Inap *Targeting*

Pada Gambar IV.3 diilustrasikan proses bisnis *targeting* alur perawatan pasien Covid-19 rawat inap. Perbedaan yang ada pada proses bisnis *targeting* dengan eksisting adalah diterapkannya Sistem *monitoring* Pasien Covid-19. Sistem akan menampilkan tanda-tanda vital pasien. Lalu tanda-tanda vital yang diterima sistem akan di analisis menggunakan parameter EWS sehingga dokter mengetahui kondisi pasien. Sistem akan menampilkan warna merah, oranye, atau kuning bila

tanda-tanda vital pasien tidak normal. Sehingga dokter dapat segera memberikan penanganan.

IV.2.3 Use Case Diagram



Gambar IV.4 Use Case Diagram

Pada *use case* diagram akan diilustrasikan interaksi antara aktor yang terlibat dan fungsi-fungsi yang dapat dilakukan oleh sistem. Aktor *dashboard monitoring* Pasien Covid-19 dapat mengakses data pasien, mengakses data ruangan, mengakses halaman EWS, menandai penanganan pasien darurat pada halaman EWS, mengakses master data, dan mengakses bantuan.

IV.2.4 Use Case Scenario

Setelah merancang *use case* diagram, akan dirancang *use case scenario* penggunaan sistem serta hubungan antara pengguna (aktor) dengan fungsionalitas yang ada.

Tabel IV.4 *Use case scenario* Mengakses Data Pasien

Nama Use Case	:	Mengakses Data Pasien
Aktor	:	Dokter
Pre-condition	:	Aktor berada di <i>dashbord</i> utama
Post condition	:	Sistem berhasil menampilkan data pasien
Skenario Awal		
Aksi Aktor		Reaksi Sistem
Aktor memilih menu <i>IPD Management</i>		Sistem menampilkan sub-menu dari <i>IPD Management</i>
Aktor memilih sub-menu <i>Patient Data</i>		Sistem menampilkan data pasien yang ada pada database
Skenario Alternatif		

Tabel IV.4 merupakan *Use case scenario* bagi perawat dan dokter untuk mengakses data pasien.

Tabel IV.5 *Use case scenarion* Mengakses Data Ruangan

Nama Use Case	:	Mengakses Data Ruangan (<i>Ward</i>)
Aktor	:	User
Pre-condition	:	Aktor berada di <i>dashbord</i> utama
Post condition	:	Sistem berhasil menampilkan data ruangan
Skenario Awal		
Aksi Aktor		Reaksi Sistem
Aktor memilih menu <i>IPD Management</i>		Sistem menampilkan sub-menu dari <i>IPD Management</i>
Aktor memilih sub-menu <i>Ward Data</i>		Sistem menampilkan data ruangan yang ada pada database
Skenario Alternatif		

Tabel IV.6 merupakan *Use case scenario* bagi perawat dan dokter untuk mengakses data-data ruangan pasien rawat inap.

Tabel IV.6 *Use case scenario* Mengakses *Early Warning System*

Nama Use Case	:	Mengakses <i>Early Warning System</i>
Aktor	:	Dokter
Pre-condition	:	Aktor berada di <i>dashbord</i> utama
Post condition	:	Sistem berhasil menampilkan data <i>Early Warning System</i>
Skenario Awal		
Aksi Aktor		Reaksi Sistem
Aktor memilih menu <i>Early Warning System</i>		Sistem menampilkan data <i>Early Warning System</i>
Skenario Alternatif		

Tabel IV.7 merupakan *Use case scenario* bagi perawat dan dokter untuk mengakses halaman *Early Warning System*

Tabel IV.7 *Use case scenario* Menandai Penanganan Pasien Kondisi Darurat

Nama Use Case	:	Menandai Penanganan Pasien Kondisi Darurat
Aktor	:	Dokter
Pre-condition	:	Aktor berada di <i>dashbord</i> utama
Post condition	:	Aktor berhasil memberikan tanda pada pasien kondisi darurat yang telah ditangani pada halaman <i>Early Warning System</i>
Skenario Awal		
Aksi Aktor		Reaksi Sistem
Aktor memilih menu <i>Early Warning System</i>		Sistem menampilkan data <i>Early Warning System</i>
Aktor menekan tombol <i>checkbox</i> pada pasien yang telah ditangani		Sistem menampilkan data <i>Early Warning System</i> dengan <i>checkbox</i> yang menyala pada pasien yang telah ditangani
Skenario Alternatif		

Tabel IV.8 merupakan *Use case scenario* bagi perawat dan dokter untuk menandai penanganan pasien kondisi darurat pada halaman *Early Warning System*.

Tabel IV.8 *Use case scenario* Mengakses Master Data

Nama Case	Use	:	Mengakses Master Data
Aktor		:	Dokter
Pre-condition		:	Aktor berada di <i>dashbord</i> utama
Post condition		:	Sistem berhasil menampilkan master data
Skenario Awal			
Aksi Aktor		Reaksi Sistem	
Aktor memilih menu Master Data pada sidebar		Sistem menampilkan master data yang tersedia	
Aktor memilih salah satu master data		Sistem menampilkan data dari master data yang dipilih oleh aktor	
Skenario Alternatif			

Tabel IV.9 merupakan *Use case scenario* bagi perawat dan dokter untuk mengakses master data.

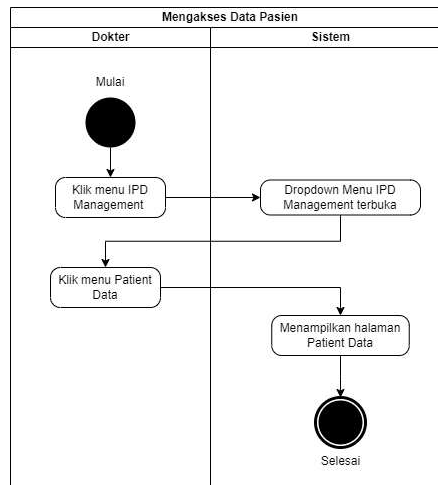
Tabel IV.9 *Use case scenario* Mengakses Bantuan

Nama Case	Use	:	Mengakses Bantuan
Aktor		:	Dokter
Pre-condition		:	Aktor berada di <i>dashbord</i> utama
Post condition		:	Sistem berhasil menampilkan halaman bantuan
Skenario Awal			
Aksi Aktor		Reaksi Sistem	
Aktor memilih menu <i>Help</i>		Sistem menampilkan halaman <i>help</i> yang berisi tentang instruksi penggunaan <i>dashboard monitoring</i>	
Skenario Alternatif			

Tabel IV.10 merupakan *Use case scenario* bagi perawat dan dokter untuk menggunakan fitur *help*.

IV.2.5 Activity Diagram

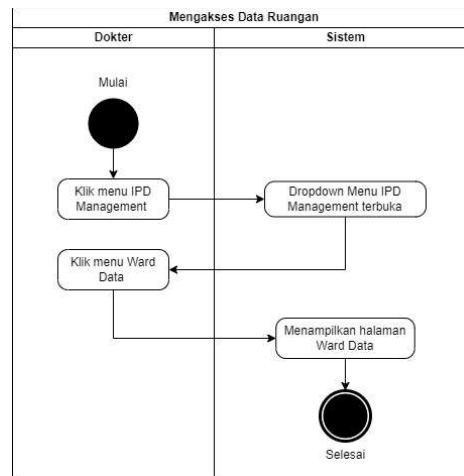
1. Activity Diagram Mengakses Data Pasien



Gambar IV.5 Activity Diagram Mengakses Data Pasien

Bila ingin mengakses data tanda-tanda pasien, dokter dapat klik menu IPD *management* pada *sidebar*. Lalu akan menu IPD *management*. Lalu sistem akan memunculkan submenu *Patient Data* dan *Ward Data*. Dokter memilih submenu *Patient Data* dan sistem akan memunculkan halaman *Patient Data*.

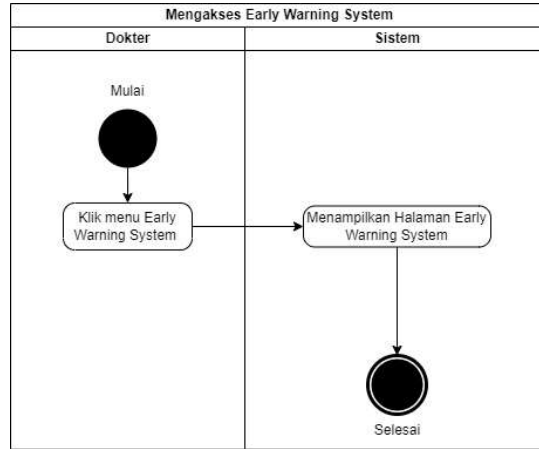
2. Activity Diagram Mengakses Data Ruangan



Gambar IV.6 Activity Diagram Mengakses Data Ruangan

Berdasarkan Gambar IV.6, dokter dapat memilih menu IPD *Management* bila ingin mengakses data ruangan. Sistem akan menampilkan submenu *Patient Data* dan *Ward Data*. Dokter memilih submenu *Ward Data* dan sistem akan menampilkan halaman *Ward Data*.

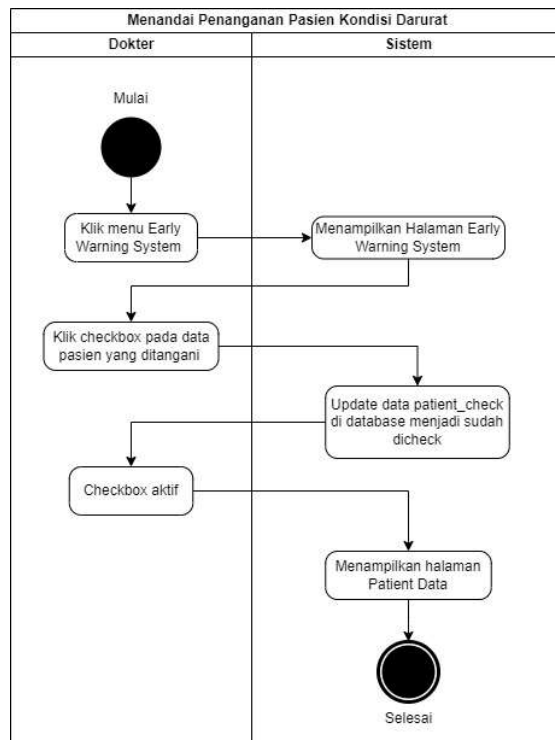
3. Activity Diagram Mengakses *Early Warning System*



Gambar IV.7 Activity Diagram Mengakses *Early Warning System*

Dokter yang ingin mengakses *Early Warning System*, dapat memilih menu *Early Warning System* yang tertera pada *sidebar*. Lalu sistem akan menampilkan halaman *Early Warning System*.

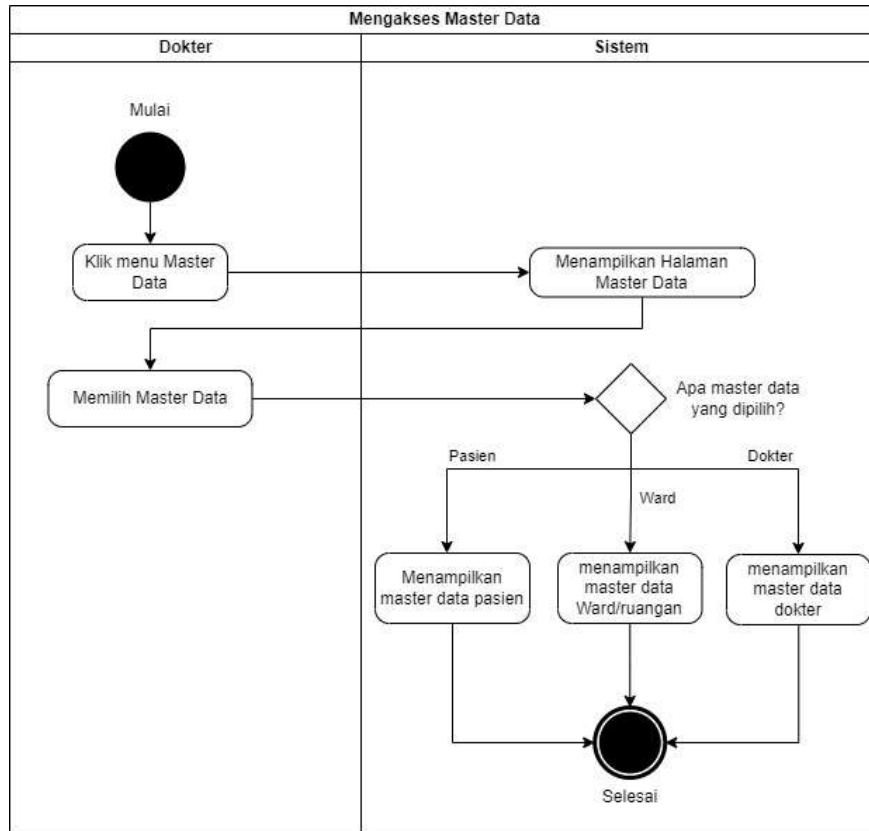
4. Activity Diagram Menandai Penanganan Pasien Kondisi Darurat



Gambar IV.8 Activity Diagram Menandai Penanganan Pasien Kondisi Darurat

Bila dokter ingin menandai pasien kondisi darurat yang sudah ditangani, dokter dapat memilih menu *Early Warning System* yang tertera pada *sidebar*. Lalu sistem akan menampilkan halaman *Early Warning System*. Pada halaman tersebut, dokter dapat menekan *checkbox* yang tersedia.

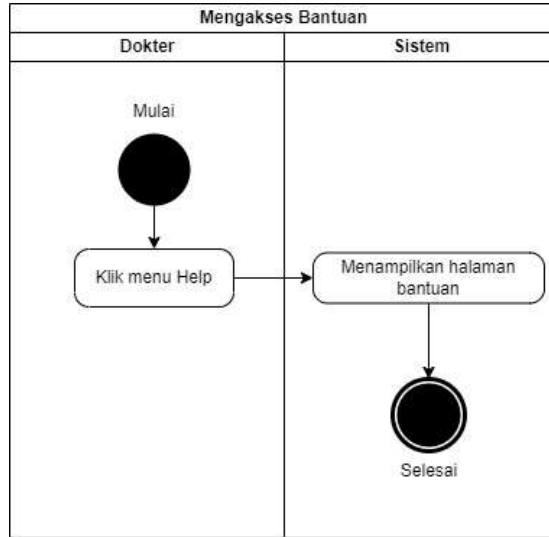
5. Activity Diagram Mengakses Master Data



Gambar IV.9 Activity Diagram Mengakses Master Data

Dokter dapat memilih menu Master Data pada *sidebar*. Lalu sistem akan menampilkan halaman Master Data. Bila Dokter ingin mengakses halaman master data pasien, maka Dokter dapat memilih menu *Patient* pada halaman Master Data. Lalu sistem akan menampilkan halaman Master Data *Patient*. Sedangkan bila Dokter ingin mengakses halaman master data ruangan, maka Dokter dapat memilih menu *Ward* pada halaman Master Data. Lalu sistem akan menampilkan halaman Master Data *Ward*. Adapun bila Dokter ingin mengakses halaman master data dokter, maka Dokter dapat memilih menu *Doctor* pada halaman Master Data. Lalu sistem akan menampilkan halaman Master Data *Doctor*.

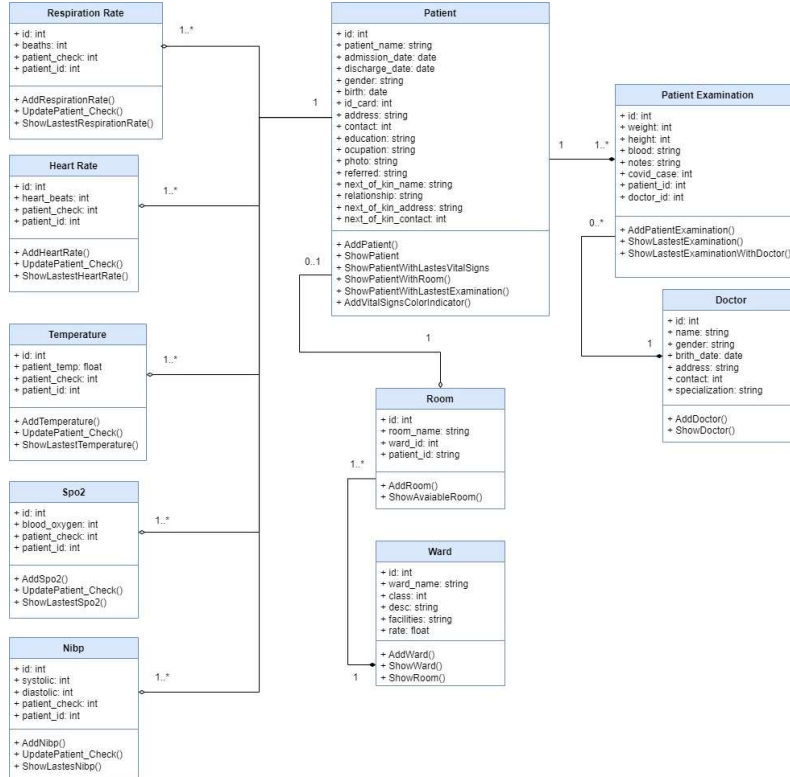
6. Activity Diagram Mengakses Bantuan



Gambar IV.10 Activity Diagram Mengakses Bantuan

Berdasarkan Gambar IV.10, dokter dapat mengakses bantuan dengan memilih menu *Help* pada *sidebar*. Lalu sistem akan menampilkan halaman *Help*.

IV.2.6 Class Diagram



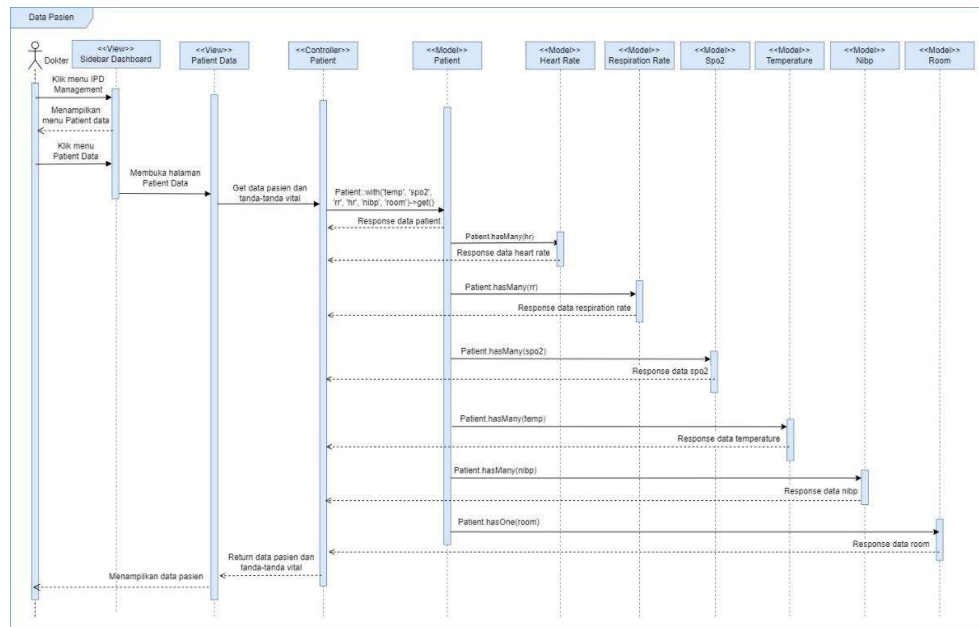
Gambar IV.11 Class Diagram

Class diagram dashboard monitoring pasien memiliki class patient, patient examination, doctor, room, ward, heart rate, respiration rate, temperature, spo2, dan class Nibp (Non Invasive Blood Pressure). Class patient memiliki relasi Agregation dengan class tanda-tanda vital (class heart rate, respiration rate, temperature, spo2, dan class Nibp), dan class Room. Class patient minimal memiliki 1 Class tanda-tanda vital dan class tanda-tanda vital hanya memiliki 1 class patient. Class patient juga minimal memiliki 1 class room dan class room maksimal memiliki 1 class patient. Adapun class room memiliki relasi Composition dengan class ward. Class room hanya memiliki 1 class ward dan class ward minimal memiliki 1 class room.

Class patient selain memiliki relasi Agregation, juga memiliki relasi Composition. Class patient berelasi Composition dengan class patient examination. Class patient minimal memiliki 1 class patient examination dan class patient examination hanya memiliki 1 class patient. Selain itu class patient examination juga memiliki relasi Composition dengan class doctor. Class patient examination hanya memiliki 1 class doctor. Sementara class doctor dapat memiliki banyak class patient examination.

IV.2.7 Sequence Diagram

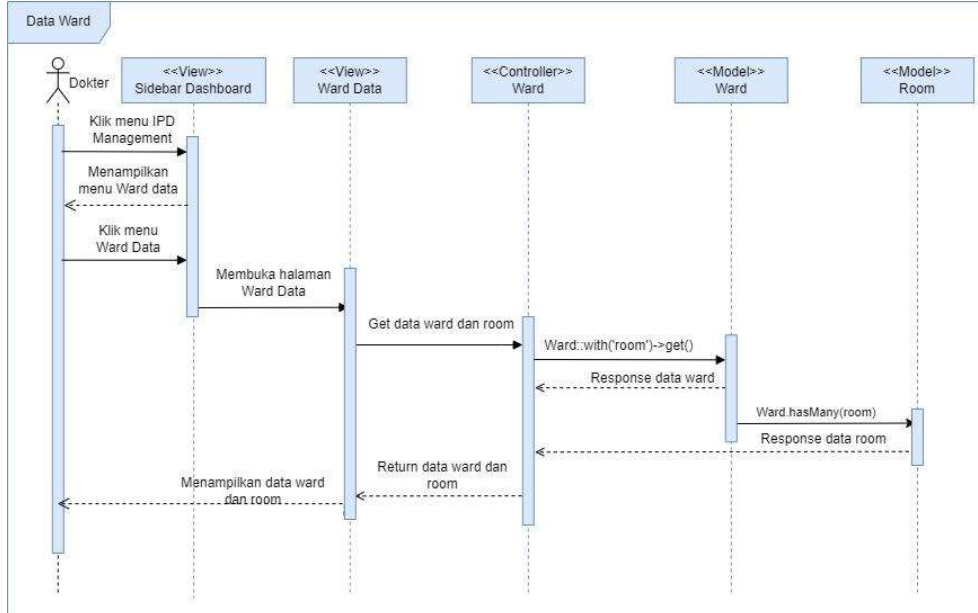
1. Sequence Diagram Mengakses Data Pasien



Gambar IV.12 Sequence Diagram Mengakses Data Pasien

Proses *Sequence Diagram* untuk mengakses data pasien dimulai dengan Dokter yang membuka halaman *Patient data*. *View Patient Data* akan melakukan *request* dengan metode *Get* kepada *Controller Patient* melalui API. *Controller* akan menjalankan fungsi untuk mengambil data dari *database* melalui model *Heart rate*, *Respiration Rate*, *Spo2*, *Temperature*, *Nibp*, dan model *Room*. Seluruh data yang telah terkumpul pada *Controller Patient* akan dikembalikan kepada *View Patient Data*. Sehingga Dokter dapat mengakses data pada halaman *Patient data*.

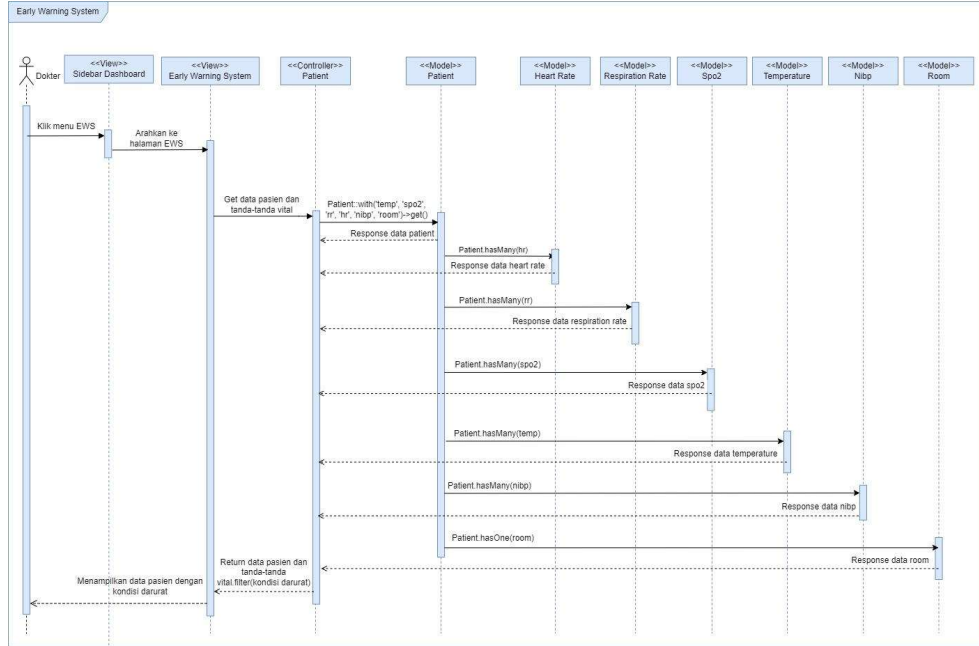
2. Sequence Diagram Mengakses Data Ward



Gambar IV.13 Sequence Diagram Mengakses Data Ward

Mula-mula dokter klik menu IPD *Management* pada *sidebar*. Dokter klik menu Ward data pada menu IPD *Management*, lalu sistem akan membuka halaman Ward Data. Melalui API, *View Ward Data* akan melakukan *request* dengan metode *Get* ke *Controller Ward*. *Controller* akan menjalankan fungsi untuk mengambil data dari *database* melalui model *Ward*, dan model *Room*. Seluruh data yang telah terkumpul pada *Controller Ward* akan dikembalikan kepada *View Ward Data*. Sehingga dokter dapat mengakses data pada halaman *Ward Data*.

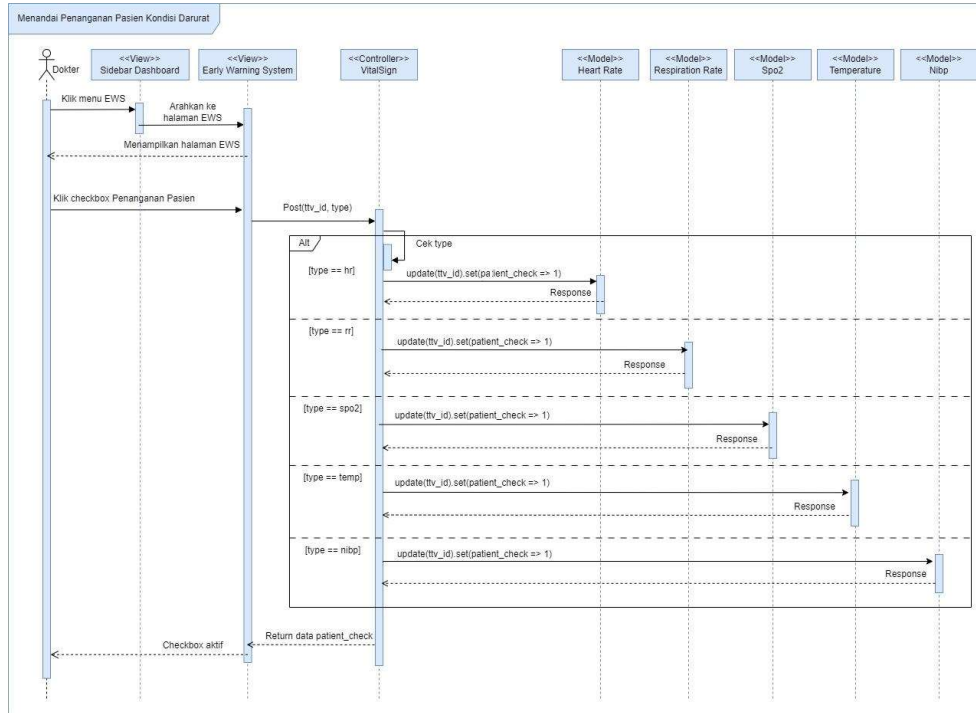
3. Sequence Diagram Mengakses *Early Warning System*



Gambar IV.14 Sequence Diagram Mengakses *Early Warning System*

Proses *Sequence Diagram* untuk Mengakses *Early Warning System* bermula saat dokter klik menu EWS pada *sidebar*. Lalu sistem akan membuka halaman *Early Warning System*. Melalui API, *View Early Warning System* akan melakukan *request* dengan metode *Get* kepada *Controller Patient*. *Controller* akan menjalankan fungsi untuk mengambil data dari *database* melalui model *Heart rate*, *Respiration Rate*, *Spo2*, *Temperature*, *Nibp*, dan model *Room*. Seluruh data yang telah terkumpul pada *Controller Patient* akan dikembalikan kepada *View Early Warning System*. Sehingga dokter dapat mengakses data pada halaman *Early Warning System*.

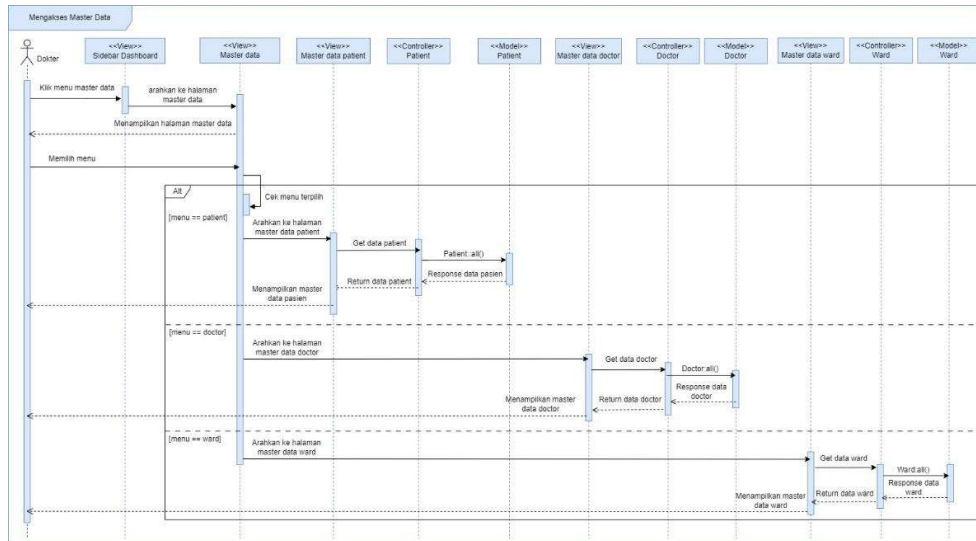
4. Sequence Diagram Menandai Penanganan Pasien Kondisi Darurat



Gambar IV.15 Sequence Diagram Menandai Penanganan Pasien Kondisi Darurat

Dokter melakukan klik menu *Early Warning System* pada sidebar. Lalu akan tampil halaman *Early Warning System* dan klik *checkbox* penanganan pasien. *View Early Warning System* akan melakukan *request* dengan metode *Post* kepada *Controller VitalSign* melalui API. *Controller* akan melakukan pengondisian berdasarkan *type* yang diterima. Berdasarkan pengondisian yang dilakukan, *controller* menjalankan fungsi *update* melalui model *Heart rate*, *Respiration Rate*, *Spo2*, *Temperature*, atau *Nibp*. Setelah proses *update* berhasil, model akan memberi *respon*. Lalu *Controller* akan mengembalikan data yang telah di-*update*. Sehingga *checkbox* penanganan pasien menjadi aktif.

5. Sequence Diagram Mengakses Master Data



Gambar IV.16 Sequence Diagram Mengakses Master Data *Patient*

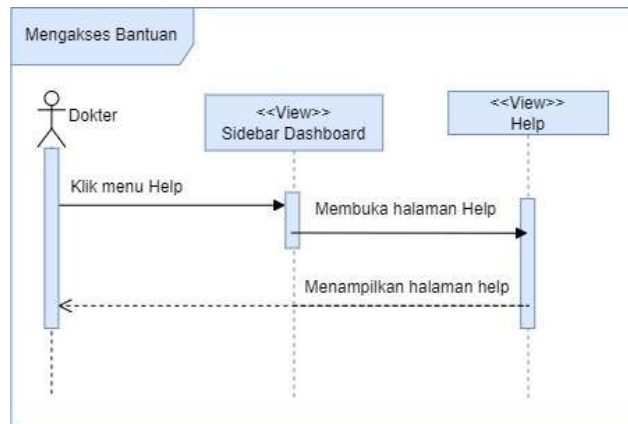
Mula-mula *user/dokter* memilih menu master data pada *sidebar*. Lalu sistem akan menampilkan halaman master data. Dokter dapat memilih 3 menu pada halaman master data, yaitu menu *patient*, *doctor*, dan *ward*. Bila dokter memilih menu *patient*, maka dokter akan diarahkan ke halaman Master Data *Patient*. *View Master Data Patient* akan melakukan *request* dengan metode *Get* ke *Controller Patient* melalui API. *Controller* akan menjalankan fungsi untuk mengambil data dari *database* melalui model *Patient*. Model akan memberikan *respon* kepada *Controller*. Lalu *Controller* akan mengembalikan data kepada *View Master Data Patient*. Lalu *user* dapat mengakses data pada halaman Master Data *Patient*.

Bila dokter memilih menu *doctor*, maka dokter akan diarahkan ke halaman Master Data *Doctor*. *View Master Data Doctor* akan melakukan *request* dengan metode *Get* kepada *Controller Doctor* melalui API. *Controller* akan menjalankan fungsi untuk mengambil data dari *database* melalui model *Doctor*. Model akan memberikan *respon* kepada *Controller*. Lalu *Controller* akan mengembalikan data kepada *View Master Data Doctor*. Lalu *user* dapat mengakses data pada halaman Master Data *Doctor*.

Bila dokter memilih menu *ward*, maka dokter akan diarahkan ke halaman Master Data *Ward*. *View Master Data Ward* akan melakukan *request* dengan metode *Get* kepada *Controller Ward* melalui API. *Controller* akan menjalankan fungsi untuk

mengambil data dari *database* melalui model *Ward*. Model akan memberikan *respon* kepada *Controller*. Lalu *Controller* akan mengembalikan data kepada *View* Master Data *Ward*. Lalu *user* dapat mengakses data pada halaman Master Data *Ward*.

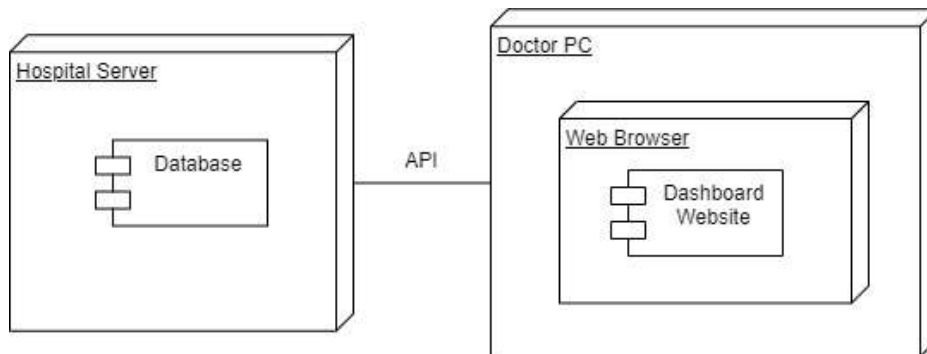
6. Sequence Diagram Mengakses Help



Gambar IV.17 Sequence Diagram Mengakses Help

Untuk mengakses halaman *help*, dokter dapat melakukan klik pada menu *help*. Sistem akan membuka halaman *help*. Lalu sistem akan memberi *respon* dengan menampilkan halaman *help*.

IV.2.8 Deployment Diagram



Gambar IV.18 Deployment Diagram

Dashboard monitoring pasien memiliki dua node yang dihubungkan dengan API. Node hospital server memiliki komponen *database*. *Database* ini akan menyimpan data-data pasien, tanda-tanda vital, dokter, dll. API akan mengirimkan data-data pada *database* ke *dashboard* sehingga dokter dapat

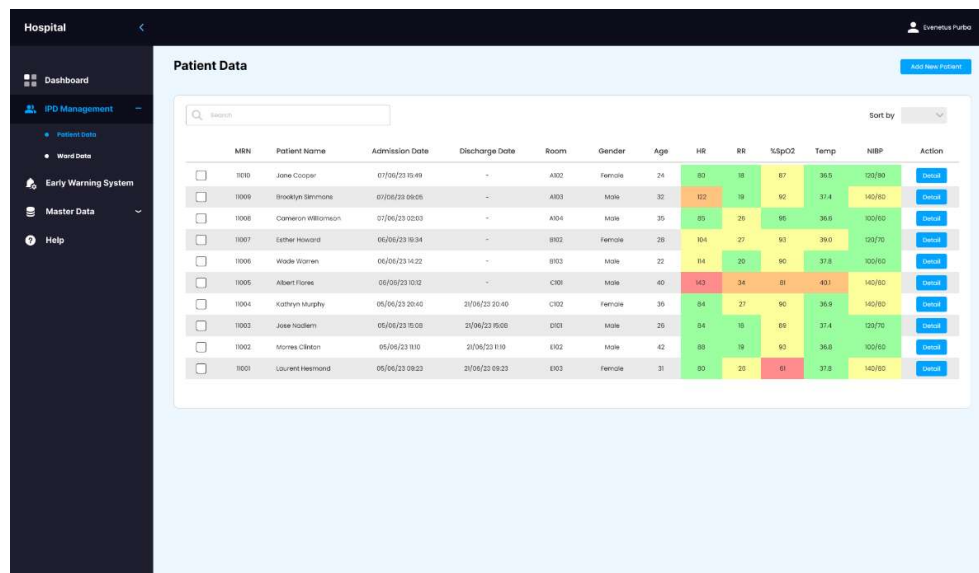
memberikan mengetahui kondisi pasien dan dapat memberikan penanganan terbaik kepada pasien yang sedang dirawat.

IV.3 Fase *Modelling Quick Design*

Pada Fase *Modelling Quick Design* diberikan gambaran jelas tentang rancangan sistem secara keseluruhan sebelum melanjutkan ke tahap implementasi. Pada tahap ini akan dirancang antarmuka pengguna yang akan digunakan. Sehingga struktur dan fungsi yang diharapkan dari sistem akan tergambar.

IV.3.1 Perancangan Desain Antarmuka *Dashboard*

1. Perancangan Desain Halaman *Patient Data*



The screenshot displays a 'Patient Data' dashboard within a 'Hospital' application. The interface includes a sidebar with navigation options like 'Dashboard', 'IPD Management', 'Patient Data', 'Ward Data', 'Early Warning System', 'Master Data', and 'Help'. The main content area features a search bar, a 'Sort by' dropdown, and a table of patient records. Each record includes a checkbox, MRN, Patient Name, Admission Date, Discharge Date, Room, Gender, Age, and vital signs (HR, RR, %SpO2, Temp, NBP) with color-coded status indicators. An 'Action' column with 'Detail' buttons is also present.

MRN	Patient Name	Admission Date	Discharge Date	Room	Gender	Age	HR	RR	%SpO2	Temp	NBP	Action
1010	Jane Cooper	07/06/23 09:49	-	A302	Female	24	83	18	87	36.5	100/80	Detail
1009	Brooklyn Simmons	03/06/23 09:05	-	A303	Male	32	92	19	90	37.4	140/80	Detail
1006	Cameron Williamson	27/06/23 02:03	-	A304	Male	35	85	20	95	36.5	100/80	Detail
1007	Ester Howard	06/06/23 16:34	-	B102	Female	28	104	27	93	39.0	100/70	Detail
1006	Wade Warren	06/06/23 14:22	-	B103	Male	22	114	20	90	37.8	100/60	Detail
1005	Albert Flores	08/06/23 10:12	-	C108	Male	40	103	34	81	43.1	140/80	Detail
1004	Kathryn Murphy	06/06/23 20:40	20/06/23 20:40	C102	Female	35	84	27	90	36.9	140/80	Detail
1003	Jose Nielsen	06/06/23 16:08	20/06/23 16:00	D101	Male	20	84	18	88	37.4	100/70	Detail
1002	Morea Clinton	06/06/23 19:10	20/06/23 19:10	E102	Male	42	88	19	93	36.8	100/60	Detail
1001	Laurent Hammond	06/06/23 09:23	20/06/23 09:23	E103	Female	31	80	20	81	37.8	140/80	Detail

Gambar IV.19 Perancangan Desain Halaman *Patient Data*

2. Perancangan Desain Halaman Data Ruang

No.	Room Name	Class	Room Desc.	Facilities	Room Rates	Available Rooms	Action																		
01	Astrid	II	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, toilet, wash basin	Rp. 125.000	<table border="1"> <tr><td>A10</td><td>A12</td><td>A13</td><td>A15</td><td>A16</td><td>A17</td><td>A18</td><td>A19</td><td>A20</td></tr> <tr><td>A11</td><td>A13</td><td>A14</td><td>A16</td><td>A17</td><td>A18</td><td>A19</td><td>A20</td><td></td></tr> </table>	A10	A12	A13	A15	A16	A17	A18	A19	A20	A11	A13	A14	A16	A17	A18	A19	A20		+
A10	A12	A13	A15	A16	A17	A18	A19	A20																	
A11	A13	A14	A16	A17	A18	A19	A20																		
02	Begonia	II	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, toilet, wash basin	Rp. 125.600	<table border="1"> <tr><td>B10</td><td>B11</td><td>B12</td><td>B13</td><td>B14</td><td>B15</td><td>B16</td><td>B17</td><td>B18</td></tr> <tr><td>B11</td><td>B12</td><td>B13</td><td>B14</td><td>B15</td><td>B16</td><td>B17</td><td>B18</td><td>B19</td></tr> </table>	B10	B11	B12	B13	B14	B15	B16	B17	B18	B11	B12	B13	B14	B15	B16	B17	B18	B19	+
B10	B11	B12	B13	B14	B15	B16	B17	B18																	
B11	B12	B13	B14	B15	B16	B17	B18	B19																	
03	Crocus	I	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 276.000	<table border="1"> <tr><td>C01</td><td>C02</td><td>C03</td><td>C04</td><td>C05</td><td>C06</td><td>C07</td><td>C08</td><td>C09</td></tr> </table>	C01	C02	C03	C04	C05	C06	C07	C08	C09	+									
C01	C02	C03	C04	C05	C06	C07	C08	C09																	
04	Daisy	I	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 250.000	<table border="1"> <tr><td>D01</td><td>D02</td><td>D03</td><td>D04</td><td>D05</td><td>D06</td><td>D07</td><td>D08</td><td>D09</td></tr> </table>	D01	D02	D03	D04	D05	D06	D07	D08	D09	+									
D01	D02	D03	D04	D05	D06	D07	D08	D09																	
05	Edelweiss	I	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 400.000	<table border="1"> <tr><td>E01</td><td>E02</td><td>E03</td><td>E04</td><td>E05</td></tr> </table>	E01	E02	E03	E04	E05	+													
E01	E02	E03	E04	E05																					
06	Fraseria	I	Covid-19 inpatient Room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 400.000	<table border="1"> <tr><td>F01</td><td>F02</td><td>F03</td><td>F04</td><td>F05</td></tr> </table>	F01	F02	F03	F04	F05	+													
F01	F02	F03	F04	F05																					
07	Gardenia	I	Covid-19 ICU	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin, ventilator, wash basin	Rp. 900.000	<table border="1"> <tr><td>G01</td><td>G02</td><td>G03</td><td>G04</td><td>G05</td></tr> </table>	G01	G02	G03	G04	G05	+													
G01	G02	G03	G04	G05																					
08	Hibiscus	I	Covid-19 ICU	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin, ventilator, bedside monitor, wash basin	Rp. 1.000.000	<table border="1"> <tr><td>H01</td><td>H02</td><td>H03</td><td>H04</td><td>H05</td></tr> </table>	H01	H02	H03	H04	H05	+													
H01	H02	H03	H04	H05																					

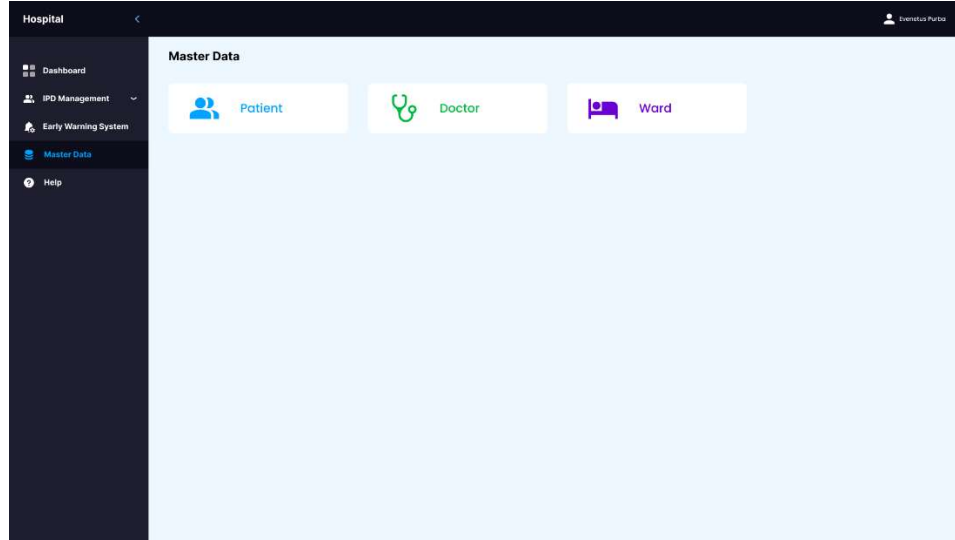
Gambar IV.20 Perancangan Desain Halaman Data Ruang

3. Perancangan Desain Halaman *Early Warning System*

Patient Name	MRN	Room	Patient Handling	Date	Vital Sign
Jane Cooper	11010	A102	<input type="checkbox"/>	10/01/1999	Heart Rate: 150
			<input type="checkbox"/>	10/01/1999	Respiration Rate: 150
			<input type="checkbox"/>	10/01/1999	Temperature: 150
			<input type="checkbox"/>	10/01/1999	Spo2: 150
			<input type="checkbox"/>	10/01/1999	Nibp: 150/80
Jane Cooper	11010	A102	<input type="checkbox"/>	10/01/1999	Heart Rate: 150
			<input type="checkbox"/>	10/01/1999	Respiration Rate: 150
			<input type="checkbox"/>	10/01/1999	Temperature: 150
			<input type="checkbox"/>	10/01/1999	Spo2: 150
			<input type="checkbox"/>	10/01/1999	Nibp: 150/80
Jane Cooper	11010	A102	<input type="checkbox"/>	10/01/1999	Heart Rate: 150
			<input type="checkbox"/>	10/01/1999	Respiration Rate: 150
			<input type="checkbox"/>	10/01/1999	Temperature: 150
			<input type="checkbox"/>	10/01/1999	Spo2: 150
			<input type="checkbox"/>	10/01/1999	Nibp: 150/80

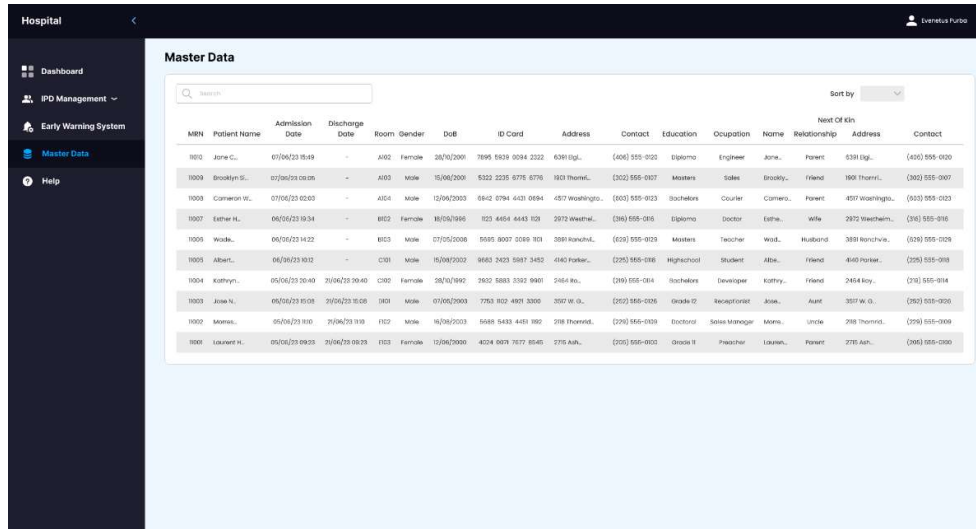
Gambar IV.21 Perancangan Desain Halaman *Early Warning System*

4. Perancangan Desain Halaman Master Data



Gambar IV.22 Perancangan Desain Halaman Master Data

5. Perancangan Desain Halaman Master Data *Patient*



Gambar IV.23 Perancangan Desain Halaman Master Data *Patient*

6. Perancangan Desain Halaman Master Data *Ward*

Master Data

No.	Ward Name	Class	Room Desc.	Facilities	Room Rates	Room Qty
01	Asterid	II	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, toilet, wash basin	Rp. 35.000	20 Room
02	Begonia	II	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, toilet, wash basin	Rp. 35.000	20 Room
03	Crocus	II	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 250.000	10 Room
04	Daisy	II	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, toilet, wash basin	Rp. 250.000	10 Room
05	Edelweiss	I	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, 1 sofa bed, 1 toilet, toilet, wash basin	Rp. 400.000	10 Room
06	Freesia	I	Covid-19 inpatient room	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, 1 sofa bed, 1 toilet, toilet, wash basin	Rp. 400.000	10 Room
07	Gardenia	I	Covid-19 ICU	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, 1 sofa bed, 1 toilet, ventilator, wash basin	Rp. 600.000	5 Room
08	Hibiscus	I	Covid-19 ICU	1 electric bed, 1 side bed table, 1 guest chair, 1 wardrobe, 1 LCD TV, 1 telephone, 1 sofa bed, 1 toilet, ventilator, bedside monitor, wash basin	Rp. 1.000.000	5 Room

Gambar IV.24 Perancangan Desain Halaman Master Data *Ward*

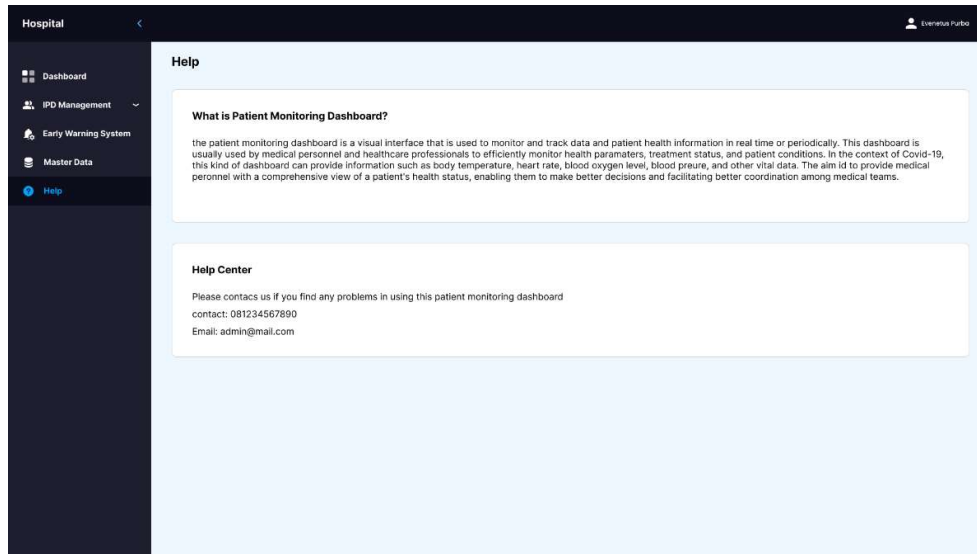
7. Perancangan Desain Halaman Master Data *Doctor*

Master Data

No.	Doctor Name	Gender	Birth	Address	Contact	Specialization
1010	Jane Cooper	Female	07/04/69	6391 Eigh St, Carina	(071) 555-2180	Orthopedics
1009	Brooklyn Simmons	Male	07/06/68	1959 Thornridge Ct, Shalsh	(234) 555-9108	Cardiology
1008	Comeron Williamson	Male	07/04/62	4857 Washington Ave, Manchester	(205) 555-0102	Cardiology
1007	Ester Howard	Female	06/06/67	2872 Westheimer Rd, Santa Ana	(507) 555-9133	Pediatrics
1006	Wade Warren	Male	06/04/66	383F Marchew Dr, Richardson	(205) 555-3225	Orthopedics
1005	Albert Flores	Male	05/04/69	440 Parker Rd, Allentown	(624) 555-0128	Pediatrics
1004	Kathryn Murphy	Female	05/04/69	2484 Royal Ln, Mass	(499) 555-2103	Orthopedics
1003	Jose Hudson	Male	06/04/97	3917 W. Gray St, Mico	(404) 555-2120	Cardiology
1002	Morris Clinton	Male	05/06/87	283 Thornridge Cr, Synouise	(202) 555-9121	Cardiology
1001	Laurent Hayward	Female	05/04/90	275 Ash Cr, San Jose	(603) 555-0523	Pediatrics

Gambar IV.25 Perancangan Desain Halaman Master Data *Doctor*

8. Perancangan Desain Halaman *Help*



Gambar IV.26 Perancangan Desain Halaman *Help*

BAB V IMPLEMENTASI DAN PENGUJIAN

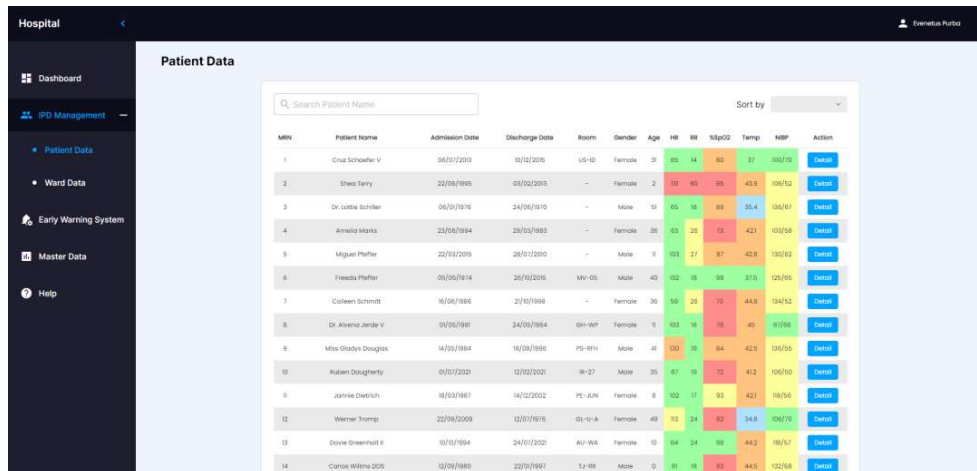
V.1 Implementasi

Pada subbab ini akan dirincikan proses-proses yang dilakukan selama mengembangkan *dashboard*. Proses implementasi dibagi dalam dua iterasi.

V.1.1 Iterasi 1

Setelah melakukan seluruh tahapan dari fase-fase *prototyping process* model, pada iterasi pertama akan difokuskan untuk mengembangkan fitur-fitur pada halaman *patient data*, *ward data*, dan fitur-fitur pada halaman Early Warning System.

1. Halaman *Patient Data*



The screenshot displays a web application interface for a hospital. On the left is a dark sidebar with navigation options: Dashboard, IPD Management, Patient Data (selected), Ward Data, Early Warning System, Master Data, and Help. The main content area is titled 'Patient Data' and features a search bar for patient names and a 'Sort by' dropdown. Below these is a table with 14 rows of patient data. Each row includes an ID, patient name, admission and discharge dates, room number, gender, age, heart rate (HR), respiratory rate (RR), oxygen saturation (SpO2), temperature (Temp), blood pressure (NBP), and an 'Action' button.

ID	Patient Name	Admission Date	Discharge Date	Room	Gender	Age	HR	RR	SpO2	Temp	NBP	Action
1	Chris Schoeder V	06/07/2019	07/02/2019	105-10	Female	31	85	14	92	37	102/70	Detail
2	Shea Terry	22/06/1995	03/02/2019	-	Female	2	101	60	95	43.8	106/70	Detail
3	Dr. Loba Schiller	06/07/1976	24/06/1976	-	Male	31	65	15	93	36.4	105/67	Detail
4	Amelia Marks	22/04/1984	29/03/1983	-	Female	36	63	23	73	42.1	100/58	Detail
5	Miguel Pfeiffer	22/01/2019	26/01/2019	-	Male	11	103	27	97	42.9	102/62	Detail
6	Frieda Pfeiffer	05/05/1974	24/02/2019	101-05	Male	43	102	15	98	37.5	120/85	Detail
7	Cathleen Schmidt	16/06/1986	21/10/1988	-	Female	30	74	20	71	44.8	104/70	Detail
8	Dr. Alvena Jarde V	03/02/1981	24/09/1984	011-01	Female	11	93	18	78	45	97/58	Detail
9	Miss Gladys Douglas	14/09/1984	16/08/1986	101-01	Male	41	102	16	94	42.1	105/75	Detail
10	Ruben Dougherty	09/07/2020	12/02/2020	10-27	Male	35	67	19	72	41.2	106/50	Detail
11	Johnie Diecklich	18/03/1987	14/12/2012	101-04	Female	8	102	17	93	42.1	105/50	Detail
12	Warner Tromp	22/09/2009	12/07/1976	011-11-A	Female	49	113	23	82	34.8	104/70	Detail
13	Doris Greenhall S	10/01/1994	24/07/2020	101-04	Female	10	64	24	98	44.2	105/57	Detail
14	Carole Wilms DDS	12/04/1985	22/01/1997	11-08	Male	5	81	16	83	44.3	122/54	Detail

Gambar V.1 Implementasi Halaman *Patient Data*

Dokter dapat melihat kondisi vital pasien dengan lebih detail pada halaman *Patient Data*. Halaman ini dilengkapi dengan fitur *search patient name* dan *sorting* data. Sehingga dapat memudahkan Dokter dalam mencari data pasien.

2. Halaman *Ward Data*

No.	Room Name	Class	Room Desc.	Facilities	Room Rates	Available Rooms
01	Asteraceae	V	Donec odio justo, sed felicitudin ut, suscipit e, feugiat et, eros. Vestibulum ac est lacina nisi venenatis tristique.	Cirium edule Nutt.	Rp. 3,220,834	01-01 01-02 01-03 01-04 01-05
02	Cuscutaceae	III	Vestibulum rutrum rutrum neque.	Hypericum pulchrum L.	Rp. 2,340,177	02-01 02-02 02-03 02-04
03	Cucurbitaceae	V	Etiam faucibus cursus urna.	Sicyos lasiocephalus Skottsb.	Rp. 1,794,448	03-01 03-02 03-03 03-04 03-05
04	Poaceae	II	Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem.	Leymos innovatus (Beal) Png.	Rp. 480,716	04-01 04-02 04-03 04-04 04-05
05	Anthomycesaceae	V	Cras non velit nec nisi pulvinar quamquam.	Lepitophaphis epidermidis	Rp. 277,038	05-01 05-02 05-03 05-04 05-05

Gambar V.3 Implementasi Halaman *Ward Data*

Halaman *Ward data* adalah halaman yang menampilkan data bangsal dan ruangan-pasien pada bangsal tersebut. Ruangan yang terisi oleh pasien akan berwarna kuning, sedangkan ruangan yang masih kosong akan berwarna hijau.

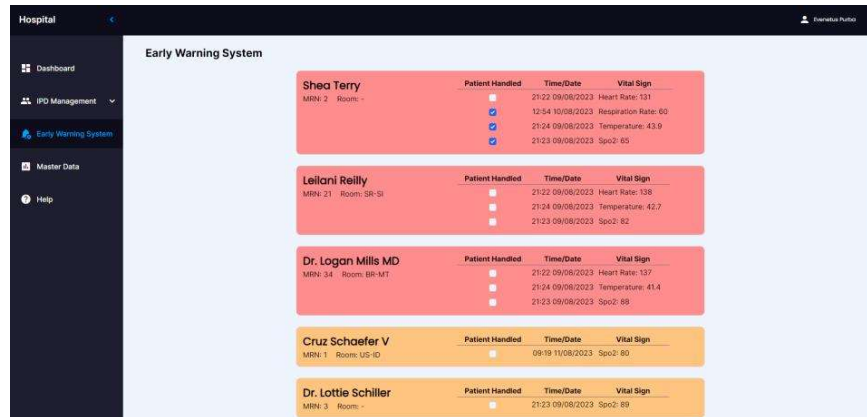
```

<BaseLayout>
  <div className={'styles.title'}>Ward Data</div>
  <div className={'styles.table-bg'}>
    <div className="flex items-center justify-end pt-6">
      <div className="flex items-center gap-3">
        </div>
      </div>
    </div>
    <div className="flex items-center justify-end pt-6">
      <div className="flex items-center gap-3">
        </div>
      </div>
    </div>
    <table border="1">
      <thead>
        <tr>
          <th scope="col" className="py-3">No.</th>
          <th scope="col" className="py-3">Room Name</th>
          <th scope="col" className="py-3">Class</th>
          <th scope="col" className="py-3">Room Desc.</th>
          <th scope="col" className="py-3">Facilities</th>
          <th scope="col" className="py-3">Room Rates</th>
          <th scope="col" className="py-3">Available Rooms</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>01</td>
          <td>Asteraceae</td>
          <td>V</td>
          <td>Donec odio justo, sed felicitudin ut, suscipit e, feugiat et, eros. Vestibulum ac est lacina nisi venenatis tristique.</td>
          <td>Cirium edule Nutt.</td>
          <td>Rp. 3,220,834</td>
          <td>01-01 01-02 01-03 01-04 01-05</td>
        </tr>
        <tr>
          <td>02</td>
          <td>Cuscutaceae</td>
          <td>III</td>
          <td>Vestibulum rutrum rutrum neque.</td>
          <td>Hypericum pulchrum L.</td>
          <td>Rp. 2,340,177</td>
          <td>02-01 02-02 02-03 02-04</td>
        </tr>
        <tr>
          <td>03</td>
          <td>Cucurbitaceae</td>
          <td>V</td>
          <td>Etiam faucibus cursus urna.</td>
          <td>Sicyos lasiocephalus Skottsb.</td>
          <td>Rp. 1,794,448</td>
          <td>03-01 03-02 03-03 03-04 03-05</td>
        </tr>
        <tr>
          <td>04</td>
          <td>Poaceae</td>
          <td>II</td>
          <td>Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem.</td>
          <td>Leymos innovatus (Beal) Png.</td>
          <td>Rp. 480,716</td>
          <td>04-01 04-02 04-03 04-04 04-05</td>
        </tr>
        <tr>
          <td>05</td>
          <td>Anthomycesaceae</td>
          <td>V</td>
          <td>Cras non velit nec nisi pulvinar quamquam.</td>
          <td>Lepitophaphis epidermidis</td>
          <td>Rp. 277,038</td>
          <td>05-01 05-02 05-03 05-04 05-05</td>
        </tr>
      </tbody>
    </table>
  </div>
</BaseLayout>

```

Gambar V.4 Code Halaman *Ward Data*

3. Halaman *Early Warning System* dan Menandai Penanganan Pasien Kondisi Darurat



Gambar V.5 Implementasi Halaman *Early Warning System* dan Menandai Penanganan Pasien Kondisi Darurat

Pada halaman ini, Dokter dapat melihat pasien-pasien dengan kondisi vital yang sedang darurat. Dokter akan mengetahui pasien mana yang memerlukan penanganan dengan segera. Dokter dapat memberikan centang bagi pasien yang sudah ditangani.

```

classLayout()
{
  /* TITLE FRAME */
  <div className={`${styles.title} ${'flex items-center justify-between'}}>
    <<Early Warning System/>
  </div>

  /* <div > */
  {data_ews.length > 0
    ? sortedData.map((item, index) => {
        <div
          className={`${styles['ews-bg']} ${color_indicator_alert(
            item.indicator
          )}`}
        >
          { /* K1P1 */ }
        <div className="w-[300px]">
          </div>
          { /* Kanan */ }
        <div>
          <table>
            <thead>
              </thead>
              <tbody>
                { /* HR */ }
                <tr
                  className={`${item.alert_hr_date ? '' : 'hidden'}}
                  <td className="py-1 text-center">
                    <input
                      type="checkbox"
                      className="cursor-pointer h-4 w-4 rounded border-gray-300 bg-gray-100"
                      checked={
                        item.hr.length > 0 && item.hr[0].patient_check === 1
                      } // SET checked based on patient_check value
                      onChange={() =>
                        handleCheckboxClick(item.id, item.hr[0].id, 'hr', item.hr[0].patient_check)
                      }
                    </input>
                </tr>
                <tr
                  className="w-[150px] py-1 text-center">
                  <td className="py-1 text-center">
                    {item.alert_hr_date}
                </td>
                <td className="w-[150px] px-1 py-1">
                  Heart Rate: {item.alert_patient_hr}
                </td>
                </tr>
                { /* RR */ }
                <tr
                  className={`${item.alert_rr_date ? '' : 'hidden'}}
                </tr>
                { /* Temp */ }
                <tr
                  className={`${item.alert_temp_date ? '' : 'hidden'}}
                </tr>
                { /* SpO2 */ }
                <tr
                  className={`${item.alert_spo2_date ? '' : 'hidden'}}
                </tr>
                { /* NIBP */ }
                <tr
                  className={`${item.alert_nibp_date ? '' : 'hidden'}}
                </tr>
              </tbody>
            </table>
          </div>
        </div>
      )
    } : 'No Alert'
  )
}

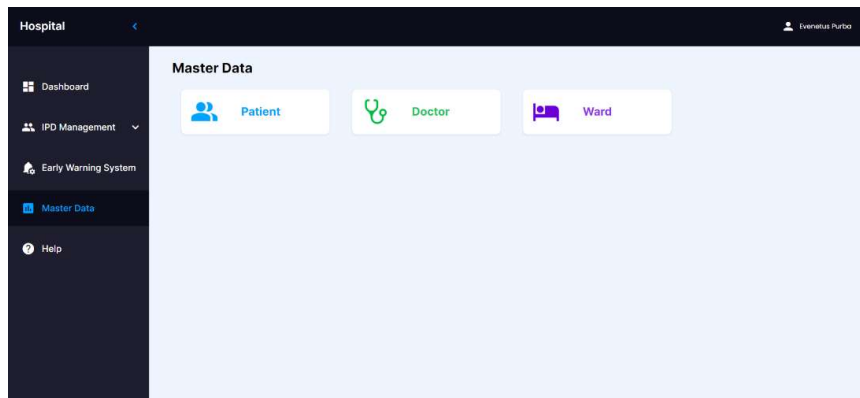
```

Gambar V.6 Code Halaman *Early Warning System* dan Menandai Penanganan Pasien Kondisi Darurat

V.1.2 Iterasi 2

Setelah melakukan seluruh tahapan dari fase-fase *prototyping process* model, pada iterasi kedua pengembangan *dashboard* berfokus pada fitur-fitur halaman master data halaman *help*. Pada halaman master data, terdapat sub-sub halaman seperti master data *patient*, *doctor*, dan master data *ward*.

1. Halaman Master Data



Gambar V.7 Implementasi Halaman Master Data

Halaman Master Data terdiri dari tiga master data. Master data pasien, master data dokter, dan master data bangsal.

```
import Link from 'next/link';
import React from 'react';
import Nav_bar from '@pages/dashboard/nav_bar';
import Side_bar from '@pages/dashboard/side_bar';
import BaseLayout from '@pages/dashboard/baseLayout';
import styles from '../MainContent.module.css';
import MasterDataList from '@pages/dashboard/dummyData/masterDataList';

// Interface MasterData {
//   image: string;
//   name: string;
//   color: string;
//   href: string;
// }
const Master = () => {
  return (
    <BaseLayout
      // title="Master Data"
      // backPageButton={false}
      // contentColor="bg-transparent"
    >
      <Nav_bar />
      <div className={styles.layout}>
        <Side_bar />
        <div className={styles.children}>
          <div className={styles.title}>Master Data</div>
          <div className="mx-auto mt-6 grid w-[1172px] grid-cols-4 gap-10">
            <MasterDataList.map((e, k) => {
              <Link key={k} href={e.href}>
                <div className="flex h-20 items-center gap-10 rounded-lg bg-white p-4 shadow hover:bg-gray-100">
                  <img
                    src={e.image}
                    className="h-[50px] w-[50px]"
                  />
                  <p
                    className="text-xl font-bold ${e.color} capitalize">
                      {e.name}
                    </p>
                </div>
              </Link>
            </>
          </div>
        </div>
      </div>
    </BaseLayout> */
  );
};

export default Master;
```

Gambar V.8 Code Halaman Master Data

2. Halaman Master Data Pasien

MBN	Patient Name	Admission Date	Discharge Date	Room Gender	Birth	ID Card	Address	Contact	Education	Occupation	Name	Relationship	Address	Contact
1	Clara Schaefer's	06/07/2013	10/12/2015	USF ID	Female 05/08/1992	46312572009478	2708 Karolann Ford Suite 215 Bensenville, PA 81363	+989-628-0210	University	Medical Secretary	Mrs. Herla Crain III	Spouse	34257 Lonny Stream North, Glen, AZ 84422	731-319-1545
2	Enea Terry	22/06/1955	03/02/2013	-	Female 10/10/2020	805884605310888	4150 Bayer Street South Albemarle, NJ 20348 4884	+1 208-298-5769	Master	Railroad Conductors	Ermitt McPherson DDS	Child	45505 McCallough Pike Suite 603 Port Roccade, MI 94716-2684	(201) 950-4584
3	Dr. Lottie Schlier	06/01/1978	24/06/1979	-	Male 12/06/1873	805386951009370	4508 Central Harbor Apt. 858 West Marse, AL 80728 9828	(860) 614-0191	High School	Legal Secretary	Dr. El HerLue II	Sibling	8035 Christiansen Village Apt. 026 Wiscasset, SC 96551-3915	763.8272754
4	Amelia Marks	23/08/1994	28/03/1983	-	Female 14/10/1986	403181837801027	8740 Greenholt View Suite 543 Jamboregon, MN 77706	+14583954822	Master	Dental Laboratory Technician	Hilton Harbor	Sibling	546 Hayes Summit North Maitland, FL 29061	603.794.9382

Gambar V.9 Implementasi Halaman Master Data Pasien

Pada halaman master data pasien, Dokter akan ditampilkan oleh data-data identitas pasien dan data kerabat pasien.


```

return (
  <>
    <Nav_bar />
    <div className={styles.layout}>
      <Side_bar />
      <div className={styles['children_master']}>
        {/* Title */}
        <div className={styles.title}>Master Data Patient</div>
        {/* TABLE FRAME */}
        <div className={styles['table-master-bg']}>
          <div className="m-6 flex items-center justify-between pt-6 ">
            {/* SEARCH PATIENT NAME */}
            <div className="relative inline-block">
              <input type="text" />
            </div>
            <div className="flex items-center gap-3">
              <button type="button" />
            </div>
          </div>
          {/* table */}
          <div className="m-6 flex items-center justify-center">
            <table className="w-[1172px] text-center text-gray-500">
              <thead className="bg-[#EAEAEA] text-xs text-gray-700">
                <tr>
                  <th>No</th>
                  <th>Ward Name</th>
                  <th>Class</th>
                  <th>Description</th>
                  <th>Facilities</th>
                  <th>Rates</th>
                </tr>
              </thead>
              <tbody className="text-xs">
                {({searchValue == '' ? patients : searchResults}).map(
                  (item, index) => (
                    <tr>
                      <td>{index}</td>
                      <td>{item.ward_name}</td>
                      <td>{item.class}</td>
                      <td>{item.description}</td>
                      <td>{item.facilities}</td>
                      <td>{item.rates}</td>
                    </tr>
                  )
                )}
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  );
};

```

Gambar V.10 Halaman Master Data Pasien

3. Halaman Master Data Ward

No.	Ward Name	Class	Description	Facilities	Rates
1	Asteraceae	V	Dolor et odio justo, seditionibus, suscipit et, eros. Vestibulum ac est lacina nisi venenatis tristique.	Cirium edule Nutt.	Rp. 3.220,834
2	Cuscutaceae	II	Vestibulum rutrum rutrum neque.	Hypericum pulchrum L.	Rp. 2.340,377
3	Cucurbitaceae	V	Etiam faucibus cursus urna.	Sicyos leiocephalus Skottsb.	Rp. 1.794,448
4	Poleaceae	II	Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem.	Laymus monovatus (Beet) Pilg.	Rp. 420,716
5	Arthropodaceae	V	Cras non velit nec nisi vulputate nonummy.	Lepidoptera epidermidis (Act.) Th. Fr.	Rp. 3.770,38
6	Polemoniaceae	II	Cras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. In blandit ultrices enim.	Phox hopodi Richardson.	Rp. 1.081,385
7	Urticaceae	II	Vivamus in felis eu sapien cursus vestibulum. Proin eu mi.	Urtica Gussfeldt.	Rp. 1.038,054
8	Sphinctroaceae	I	Maecenas tincidunt lacus at velit.	Sphinctria anglica Hyl.	Rp. 3.983,982

Gambar V.11 Implementasi Halaman Master Data Ward

4. Halaman Master Data *Doctor*



No.	Doctor Name	Gender	Birth	Address	Contact	Specialization
1	Rae Rodriguez	Female	19/02/1982	55049 Cassin Pass Apt. 931 Towneshire, OH 80323	910.993.3785	Orthopedics
2	Dorothea Abernathy Sr.	Female	17/04/1973	3108 Howe Plaza New Richtown, OK 91950-7379	+18308904786	Cardiology
3	Prof. Chandler Reilly	Male	29/05/2017	9249 Oshinski Plains Keelinterview, TX 62535- 1913	785.603.0081	Cardiology
4	Miss Baby Dickisson V.	Female	01/07/1972	4595 Aletha Drive Suite 510 Jonathonton, CO 67504- 3896	854-735-8805	Pediatrics
5	Prof. Katharina O'Reilly Jr.	Female	01/01/2015	178 Bogart Street Yeolbury, KY 61277	+1830.580.8054	Cardiology
6	Dr. Norberto Dare	Female	10/02/2003	5919 Johnston Lodge Suite 412 Faeviev, ID 26384- 8916	+1 (651) 507-0137	Pediatrics

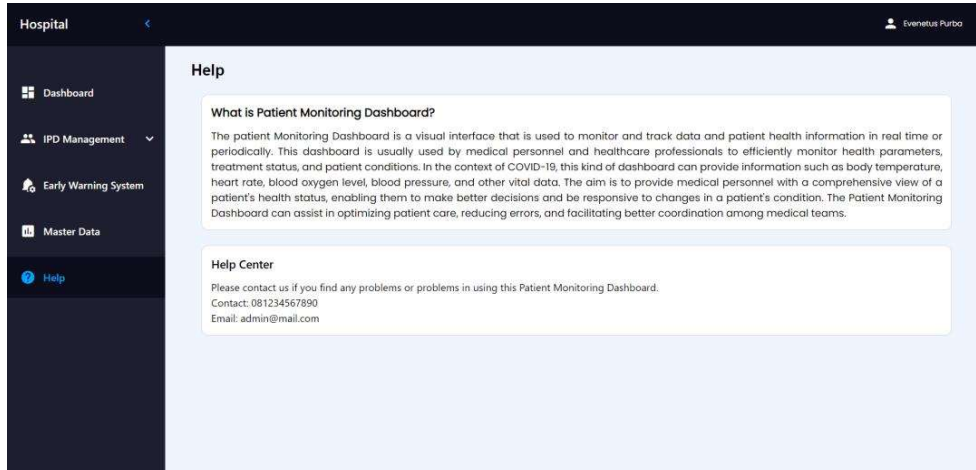
Gambar V.13 Halaman Master Data *Doctor*

Pada halaman master data *doctor*, Dokter akan ditampilkan oleh data-data identitas dokter.

```
return (\n  <BaseLayout>\n    /* Title */\n    <div className={` ${styles.title} `}>Master Data Doctor</div>\n    /* TABLE FRAME */\n    <div className={styles['table-master-bg']}>\n      <div className='mx-6 flex items-center justify-between pt-6 '>\n        /* SEARCH PATIENT NAME */\n        <div className='relative inline-block'> ...\n      </div>\n\n      /* SORT PATIENT DATA */\n      <div className='flex items-center gap-3'> ...\n    </div>\n  </div>\n\n  /* table */\n  <div className='m-6 flex items-center justify-center'>\n    <table className='w-full text-center text-gray-500'>\n      <thead className='bg-[#EAEAEA] text-xs text-gray-700'> ...\n    </thead>\n    <tbody className='text-xs'>\n      {(searchValue == '' ? doctor : searchResults).map(\n        (item, index) => (\n          <tr\n            key={index}\n            className={` ${index % 2 != 0 ? 'bg-[#EAEAEA]' : ''} `}\n            >\n            <td className='py-4 px-4'>{index + 1}</td>\n            <td className='w-[400px] px-1 py-4'>{item.name}</td>\n            <td className='px-3 py-4'>{item.gender}</td>\n            <td className='px-1 py-4'>{item.birth_date}</td>\n            <td className='w-[200px] px-1 py-4'>{item.address}</td>\n            <td className='w-[200px] px-1 py-4'>{item.contact}</td>\n            <td className='w-[400px] px-1 py-4'>\n              {item.specialization}\n            </td>\n          </tr>\n        )\n      )\n    </tbody>\n  </table>\n  </div>\n  <AutoGoToTop />\n</BaseLayout>
```

Gambar V.14 Code Halaman Master Data *Doctor*

5. Halaman *Help*



Gambar V.15 Halaman *Help*

Pada halaman ini, Dokter yang mengalami kendala dan membutuhkan bantuan selama menggunakan *dashboard* akan diarahkan untuk menghubungi kontak dan email yang disediakan.

```
import React from 'react'
import BaseLayout from '@pages/dashboard/baseLayout'
import styles from './MainContent.module.css';

function Help() {
  return (
    <BaseLayout>
      { /* Title */ }
      <div className={` ${styles.title} font-inter`} >Help</div>
      <div className={` ${styles['help-bg']} font-poppins`} >
        <p className="text-lg font-semibold">What is Patient Monitoring Dashboard?</p>
        <p className="">
          The patient Monitoring Dashboard is a visual interface that is used to monitor and track data and pa
        </p>
      </div>
      <div className={` ${styles['help-bg']}`} >
        <p className="text-lg font-semibold">Help Center</p>
        <p className="">
          Please contact us if you find any problems or problems in using this Patient Monitoring Dashboard.
          <br/>Contact: 081234567890
          <br/>Email: admin@mail.com
        </p>
      </div>
    </BaseLayout>
  )
}

export default Help
```

Gambar V.16 Code Halaman *Help*

V.2 Pengujian

Dashboard akan diuji menggunakan metode *System Usability Scale* (SUS). Dengan metode SUS, akan diketahui seberapa jauh tingkat kepuasan pengguna, efektivitas antarmuka, kemudahan penggunaan, serta efisiensi dalam mengakses informasi pasien Covid-19.

Tabel V.1 Daftar Pertanyaan *System Usability Scale*

No.	Daftar Pertanyaan
1	Saya pikir bahwa saya akan lebih sering menggunakan <i>Website Pasien Covid-19 Monitoring Dashboard</i>
2	Saya menemukan bahwa <i>Website Pasien Covid-19 Monitoring Dashboard</i> , tidak harus dibuat serumit ini
3	Saya pikir <i>Website Pasien Covid-19 Monitoring Dashboard</i> mudah untuk digunakan
4	Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan <i>Website Pasien Covid-19 Monitoring Dashboard</i> ini
5	Saya menemukan berbagai fungsi di <i>Website Pasien Covid-19 Monitoring Dashboard</i> ini diintegrasikan dengan baik
6	Saya pikir ada terlalu banyak ketidaksesuaian dalam <i>Website Pasien Covid-19 Monitoring Dashboard</i>
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari <i>Website Pasien Covid-19 Monitoring Dashboard</i> dengan sangat cepat
8	Saya menemukan, <i>Website Pasien Covid-19 Monitoring Dashboard</i> sangat rumit untuk digunakan
9	Saya merasa sangat percaya diri untuk menggunakan <i>Website Pasien Covid-19 Monitoring Dashboard</i> ini
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan <i>Website Pasien Covid-19 Monitoring Dashboard</i>

Adapun hasil dari pengujian *System Usability Scale* yang telah dilakukan kepada aktor akan dihitung dengan rumus yang telah ditentukan untuk mendapatkan Skor *System Usability Scale*.

Tabel V.2 Jawaban *System Usability Scale*

Responden	Jawaban										SUS Score
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
R1	4	3	4	2	4	3	4	2	4	2	70
R2	5	2	5	5	5	1	5	2	5	5	75
R3	5	2	5	2	5	2	5	2	5	2	87
R4	5	1	4	2	4	3	4	1	4	4	75
R5	5	1	5	1	5	1	5	1	5	1	100
Total											81.5

Dari hasil perhitungan maka didapatkan *perspective* pengguna terhadap sistem monitoring Pasien Covid-19. *Acceptability* sistem adalah Acceptable, *Grade Scale* sistem adalah B, dan *Adjective Rating* sistem adalah GOOD.

BAB VI KESIMPULAN DAN SARAN

VI.1 Kesimpulan

Berdasarkan sistem yang telah dikembangkan dan diuji, maka dapat disimpulkan:

1. Proses bisnis sistem *monitoring* pasien Covid-19 di rumah sakit berhasil dirancang. Dengan proses ini, perawat dan dokter dapat mengurangi interaksi langsung dengan pasien Covid-19 namun proses pemantauan pasien tetap berjalan dengan baik.
2. Berdasarkan analisis proses bisnis alur pelayanan pasien Covid-19 di Rumah Sakit yang dilakukan, *dashboard* yang dikembangkan telah sesuai dengan kebutuhan. Hal tersebut juga didukung oleh hasil Skor *System Usability Scale* (SUS) sebesar 81.5. Dari skor tersebut, dapat disimpulkan bahwa *dashboard* memiliki tingkat penerimaan yang tinggi di antara pengguna. Pengguna secara umum merasa nyaman dan mampu menggunakan sistem dengan baik. Hal itu menunjukkan bahwa sistem memiliki potensi untuk diterima dengan baik oleh pengguna dan memenuhi kebutuhan pengguna.
3. Telah dikembangkan *dashboard* Pasien Covid-19 menggunakan *Prototyping Process Model* sebagai metode pengembangan, serta *framework* Nextjs dan Laravel sebagai *tools* pengembangannya. *Dashboard* juga mengimplementasikan API sebagai integrasi data dengan sistem informasi manajemen rumah sakit.

VI.2 Saran

Adapun saran yang dari penelitian ini adalah:

1. Mengoptimalkan fungsi *auto reload page* pada halaman detail pasien. Karena setiap kali halaman tersebut *ter-reload*, terjadi suatu kedipan pada *chart* tanda-tanda vital pasien.
2. Melakukan pengujian menggunakan IoT pendeteksi tanda-tanda vital. Dengan pengujian tersebut, diharapkan pengembang berikutnya akan mengetahui performa sistem yang berbeda dan kendala-kendala baru apa saja yang akan muncul.

DAFTAR PUSTAKA

- Aryanata Andipradana, K. D. (2021). Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum. *Jurnal Algoritma*, 164.
- Bach, B., Freeman, E., Abdul-Rahman, A., Turkay, C., Khan, S., Fan, Y., & Chen, M. (2023). Dashboard Design Patterns. *IEEE Transactions on Visualization and Computer Graphics*, 29(1). <https://doi.org/10.1109/TVCG.2022.3209448>
- Bhardwaj, V., Joshi, R., & Gaur, A. M. (2022). IoT-Based Smart Health Monitoring System for COVID-19. *SN Computer Science*, 3(2). <https://doi.org/10.1007/s42979-022-01015-1>
- Bhatt, B., & Nandu, M. (2021). An Overview of Structural UML Diagrams. *International Research Journal of Engineering and Technology*, 1577–1583. Retrieved from www.irjet.net
- Bol, N., Helberger, N., & Weert, J. C. M. (2018). Differences in mobile health app use: A source of new digital inequalities? *Information Society*, 34(3), 183–193. <https://doi.org/10.1080/01972243.2018.1438550>
- Carter, C., Aedy, H., & Notter, J. (2020). COVID-19 disease: assessment of a critically ill patient. *Clinics in Integrated Care*, 1. <https://doi.org/10.1016/j.intcar.2020.100001>
- Ciccozzi, F., Malavolta, I., & Selic, B. (2019). Execution of UML models: a systematic review of research and practice. *Software and Systems Modeling*, 18(3), 2313–2360. <https://doi.org/10.1007/s10270-018-0675-4>
- Direktorat Jenderal Pelayanan Kesehatan; Direktorat Pelayanan Kesehatan Rujukan. (2021). *Pedoman Pelayanan Rumah Sakit Pada Masa Pandemi Covid-19*. Jakarta: Kementerian Kesehatan Republik Indonesia.
- Doshi, D., Jain, L., & Gala, K. (2021). REVIEW OF THE SPIRAL MODEL AND ITS APPLICATIONS. *International Journal of Engineering Applied Sciences and Technology*, 5(12). <https://doi.org/10.33564/ijeast.2021.v05i12.053>
- Dr. Babasaheb Ambedkar Open University. (2021). *Object Oriented Analysis and Design using UML*. India: Dr. Babasaheb Ambedkar Open University.
- Ekohariadi2021PENERAPAN GITHUB SEBAGAI MEDIA E-LEARNING UNTUK MENGETAHUI KEEFEKTIFAN KOLABORASI PROJECT PADA MATA PELAJARAN PEMROGRAMAN WEB DAN PERANGKAT BERGERAK DI SMK NEGERI 2 SURABAYA *Jurnal IT-EDU15*

- Fahyurisandi, R., & Neforawati, I. (2020). Rancang Bangun Sistem Monitoring Pintu Gudang PT XYZ Berbasis Android Menggunakan Perangkat SIM8001 dan Mikrokontroler AT Mega 328p. *MULTINETICS*, 5(1). <https://doi.org/10.32722/multinetics.v5i1.2793>
- Filipova, O., & Vilão, R. (2018). Software Development From A to Z. Dalam *Software Development From A to Z*. <https://doi.org/10.1007/978-1-4842-3945-2>
- Fu'adi, A., & Prianggono, A. (2022). Analisa dan Perancangan Sistem Informasi Akademik Akademi Komunitas Negeri Pacitan Menggunakan Diagram UML dan EER. *Jurnal Ilmiah Teknologi Informasi Asia*, 16(1), 45-54.
- Hadiprakoso, R. B. (2021). *Rekayasa Perangkat Lunak*. Jakarta: RBH.
- HERI SANTOSO, M. (2019). *REKAYASA PERANGKAT LUNAK*. Medan: Universitas Islam Negeri Sumatera Utara. Diambil kembali dari <http://repository.uinsu.ac.id/9680/1/Diktat.pdf>
- Herwanto, G. B. (2019). Pengelompokan Artefak Dokumen Perangkat Lunak Open Source Dengan Vektor Paragraf. *Pseudocode*, 6(2). <https://doi.org/10.33369/pseudocode.6.2.181-185>
- Internet of things based real-time coronavirus 2019 disease 2022 *International Journal of Electrical and Computer Engineering (IJECE)*6807
- Islam, M. M., Rahaman, A., & Islam, M. R. (2020). Development of Smart Healthcare Monitoring System in IoT Environment. *SN Computer Science*, 1(3). <https://doi.org/10.1007/s42979-020-00195-y>
- Kementerian Kesehatan Malaysia. (2020, Oktober 2). <https://covid-19.moh.gov.my/>. Diambil kembali dari <https://www.moh.gov.my/>: https://covid-19.moh.gov.my/garis-panduan/garis-panduan-kkm/Annex_2d_GUIDELINES_ON_THE_MX_OF_SUSPECTED_COVID19_PATIENT_IN_THE_ETD_2.10.20.pdf
- KH, R. (2021). *Asosiasi RS Curhat Penagihan Perawatan Pasien Covid-19 Rumit!* CNBC Indonesia.
- Khorib, P. (2021). Percabangan Dalam Perulangan. *Jurnal Komputer*, 021.
- Kulesza, R., de Sousa, M. F., Moura De Araújo, M. L., de Araújo, C. P., & Macedo Filho, A. (2020). Evolution of web systems architectures: A roadmap. Dalam *Special Topics in Multimedia, IoT and Web Technologies*. https://doi.org/10.1007/978-3-030-35102-1_1
- Kumar, K., Kumar, N., & Shah, R. (2020, January 1). Role of IoT to avoid spreading of COVID-19. *International Journal of Intelligent Networks*. KeAi Communications Co. <https://doi.org/10.1016/j.ijin.2020.05.002>

- Kurniawan, E., Nofriadi, N., & Nata, A. (2022). PENERAPAN SYSTEM USABILITY SCALE (SUS) DALAM PENGUKURAN KEBERGUNAAN WEBSITE PROGRAM STUDI DI STMIK ROYAL. *JOURNAL OF SCIENCE AND SOCIAL RESEARCH*, 5(1). <https://doi.org/10.54314/jssr.v5i1.817>
- Mariko, S. (2019). Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus. *Jurnal Inovasi Teknologi Pendidikan*, 6(1). <https://doi.org/10.21831/jitp.v6i1.22280>
- Meta Platforms, Inc. (2023). *Testing*. Retrieved from [reactjs.org](https://reactjs.org/docs/testing.html): <https://reactjs.org/docs/testing.html>
- Narayan, D. (2021). STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES. *EPRA International Journal of Multidisciplinary Research*, 3.
- Noonpakdee, W., Khunkornsiri, T., Phothichai, A., & Danaisawat, K. (2018). A framework for analyzing and developing dashboard templates for small and medium enterprises. In *2018 5th International Conference on Industrial Engineering and Applications, ICIEA 2018* (pp. 479–483). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/IEA.2018.8387148>
- Ozkaya, M., & Erata, F. (2020). A survey on the practical use of UML for different software architecture viewpoints. *Information and Software Technology*, 121, 106275.
- Patil, H. R., & Garge, D. S. (2018). Patient Monitoring System. *International Journal of Advance Research in Science and Engineering*, 26.
- Pengembangan Aplikasi Android E-Initiative Patriot Pangan sebagai Platform CrowdfundingKebutuhan Balita Menggunakan REST API2020*Jurnal Ilmu Komputer Agri-Informatika*87
- Pérez-Castillo, R., & Piattini, M. (2022). Design of classical-quantum systems with UML. *Computing*, 104(11), 2375-2403.
- 2021*Persi: Pasien Terus Berdatangan, Rumah Sakit Kewalahan*Jakarta SelatanBeritasatu.com
- PRESSMAN, R. S., & MAXIM, B. R. (2020). *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH*. New York: McGraw-Hill Education.

- Prihandoyo, M. T. (2018). Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 126-129.
- Putra, D. W. T., & Andriani, R. (2019). Unified modelling language (uml) dalam perancangan sistem informasi permohonan pembayaran restitusi sppd. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 7(1), 32-39.
- Raharjo, A. (2023). Kemenkes: 40 persen proporsi kasus covid-19 Saat Ini subvarian eg.2 dan eg.5. Retrieved from <https://news.republika.co.id/berita/rzqtvy436/kemenkes-40-persen-proporsi-kasus-covid19-saat-ini-subvarian-eg2-dan-eg5>
- Ramdhan, N. A., & Nufriana, D. A. (2019). Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Oline Berbasis WEB. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, 1(02). <https://doi.org/10.46772/intech.v1i02.75>
- REVIEW OF THE SPIRAL MODEL AND ITS2021 *International Journal of Engineering Applied Sciences and Technology* 311
- Reza, F., & Putra, A. D. (2021). SISTEM INFORMASI E-SMILE (ELEKTRONIC SERVICE MOBILE) (STUDI KASUS: DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KABUPATEN TULANG BAWANG). *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 2(3).
- Saed Novendri, M., Saputra, A., Firman, C. E., Manajemen Informatika, J., Dumai, A., Informatika, J. T., Dumai, S., Informatika, J. M., Karya, J. U., Batrem, B., & Kode, D.-. (2019). APLIKASI INVENTARIS BARANG PADA MTS NURUL ISLAM DUMAI MENGGUNAKAN PHP DAN MYSQL. *Lentera Dumai*, 10(2).
- Sama, H., & Hartanto, E. (2021). Studi Deskriptif Evolusi Website Dari Html1 Sampai Html5 Dan Pengaruhnya Terhadap Perancangan Dan Pengembangan Website. *Conference on Management, Business, Innovation, Education and Social Sciences (CoMBInES)*, 1(1).
- Setiawan, A. A. (2019). Rancang Bangun Aplikasi Unsrat E-Catalog. *Jurnal Teknik Informatika*, 14(4).
- Setiawati, S. (2023, Agustus 09). Virus 'Eris' Masuk ke RI, Ini Bedanya dengan Delta & Omicron. Diambil kembali dari cnbcindonesia: <https://www.cnbcindonesia.com/research/20230809121139-128-461429/virus-eris-masuk-ke-ri-ini-bedanya-dengan-delta-omicron>

- Shi, C., & Ge, H. (2020, January). The application of uml in the modeling of automation laboratory management system. In *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* (pp. 816-819). IEEE.
- Shon Hadji, M. T. (2019). IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG). *KONFERENSI ILMIAH MAHASISWA UNISSULA (KIMU) 2*, 33-34.
- Software Development A Practical Approach!* 2020 Norway Hans-Petter Halvorsen
a Blog about Technology
- Storey, M.-A., & Treude, C. (2019). Software Engineering Dashboards: Types, Risks, and Future. In *Rethinking Productivity in Software Engineering* (pp. 179–190). Apress. https://doi.org/10.1007/978-1-4842-4221-6_16
- STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES 2021 *EPRA International Journal of Multidisciplinary Research (IJMR)* 1-3
- Vedaei, S. S., Fotovvat, A., Mohebbian, M. R., Rahman, G. M. E., Wahid, K. A., Babyn, P., ... Sami, R. (2020). COVID-SAFE: An IoT-based system for automated health monitoring and surveillance in post-pandemic life. *IEEE Access*, 8, 188538–188551. <https://doi.org/10.1109/ACCESS.2020.3030194>
- Welda, W., Putra, D. M. D. U., & Dirgayusari, A. M. (2020). Usability Testing Website Dengan Menggunakan Metode System Usability Scale (Sus)s. *International Journal of Natural Science and Engineering*, 4(3). <https://doi.org/10.23887/ijnse.v4i2.28864>
- Yakubu, O., & Wereko, E. (2021). Internet of things based vital signs monitoring system: A prototype validity test. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(2), 962–972. <https://doi.org/10.11591/ijeecs.v23.i2.pp962-972>

LAMPIRAN

Lampiran A. Formulir *System Usability Scale*

System Usability Scale of Patient Monitoring Dashboard

AsistenManajemen Keperawatan: Heteroskaidati

Perkenalkan kami dari kelompok Tugas Akhir Dashboard Monitoring Pasien Covid-19 yang beranggotakan Alvin, Rendiya Ardi, Muzand Nash Fadlan, dan Ewelita Wicoraper. Kami ingin menguji apakah Anda setuju untuk memberikan tanggapan terhadap aplikasi Website Patient Covid-19 Monitoring Dashboard yang akan kami gunakan.

Website Patient Covid-19 Monitoring Dashboard adalah website yang dirancang untuk membantu petugas kesehatan (NAKES) dalam memantau pasien Covid-19 yang tinggal khususnya di Rumah Sakit. Aplikasi website ini akan terintegrasi dengan IoT (sebagai pemantauan Tanda-tanda vital pasien) dan terhubung dengan Sistem Informasi Rumah Sakit (SIMRS). Namun karena keterbatasan kemampuan kami, kami menggunakan aplikasi Python sebagai simulasi integrasi SIMRS dan IoT.

Maka dari itu kami harap, apabila Anda bersedia untuk memberikan tanggapan terhadap aplikasi kami dengan mengisi kuesioner System Usability Scale berikut.

Kurang lebihnya mohon maaf dan Terima kasih.

Link website:
rendiyya2001@gmail.com (silahkan kirim email)
 TTD: asipraban

*Menunjukkan pertanyaan yang wajib diisi

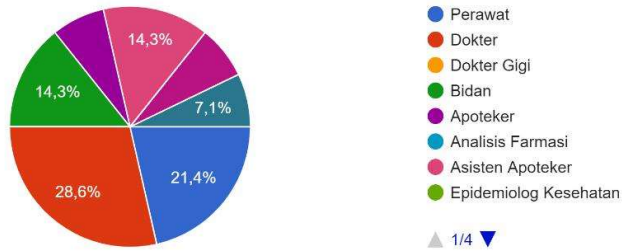
Nama Lengkap *

Profesi:

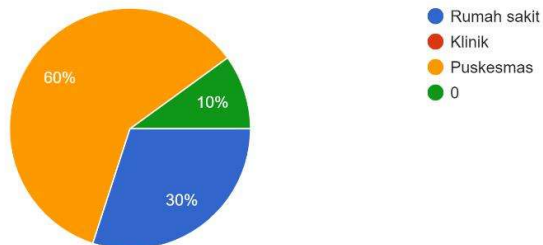
Perawat
 Dokter
 Dokter Gigi
 Bidan
 Apoteker
 Analisis Farmasi
 Asisten Apoteker
 Epidemiolog Kesehatan

Lampiran B. Jawaban *System Usability Scale* Responden

Profesi
14 jawaban

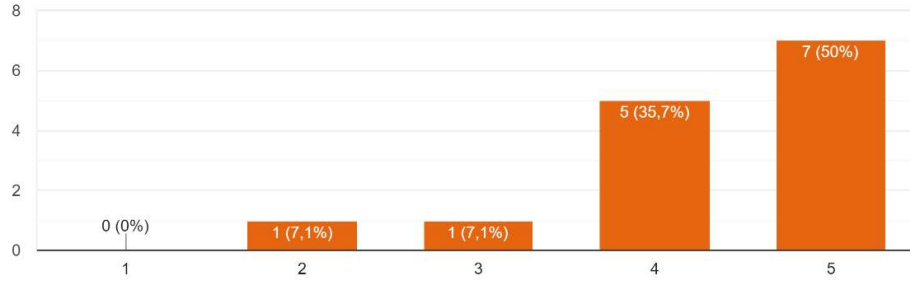


Tempat Kerja
10 jawaban



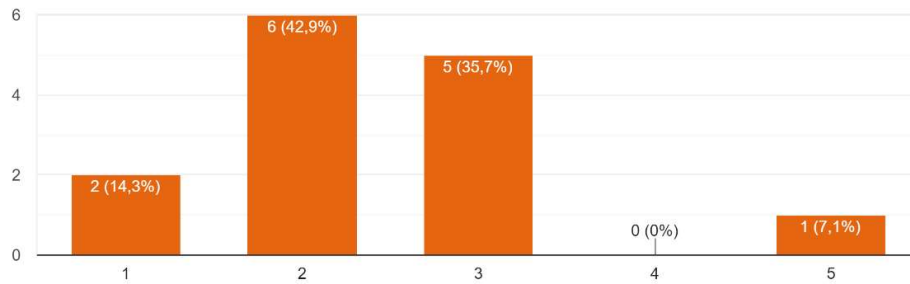
Saya pikir bahwa saya akan lebih sering menggunakan Website Pasien Covid-19 Monitoring Dashboard

14 jawaban



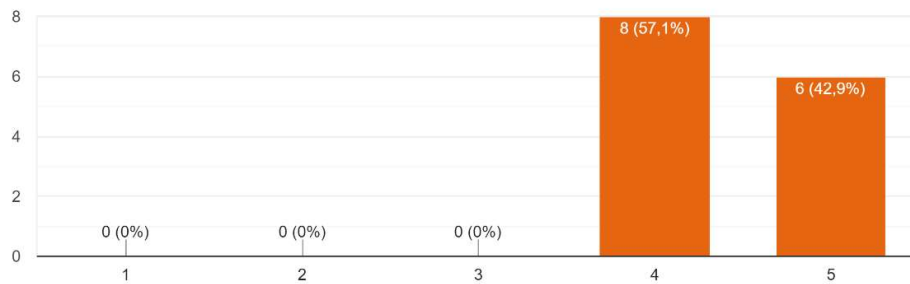
Saya menemukan bahwa Website Pasien Covid-19 Monitoring Dashboard, tidak harus dibuat serumit ini

14 jawaban



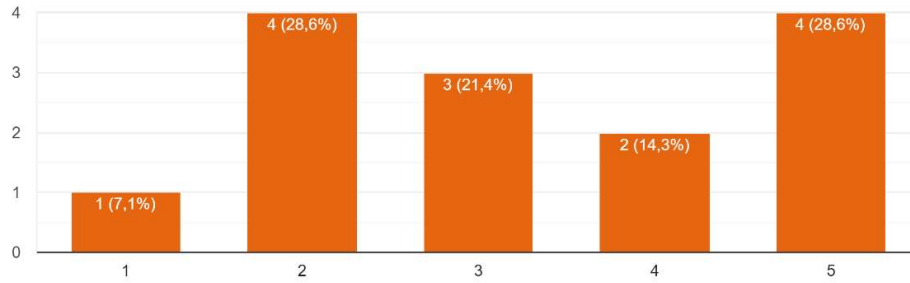
Saya pikir Website Pasien Covid-19 Monitoring Dashboard mudah untuk digunakan

14 jawaban



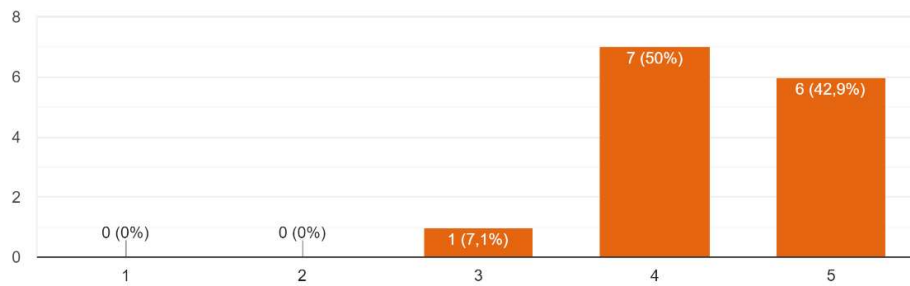
Saya pikir bahwa saya akan membutuhkan bantuan dari orang teknis untuk dapat menggunakan Website Pasien Covid-19 Monitoring Dashboard ini

14 jawaban



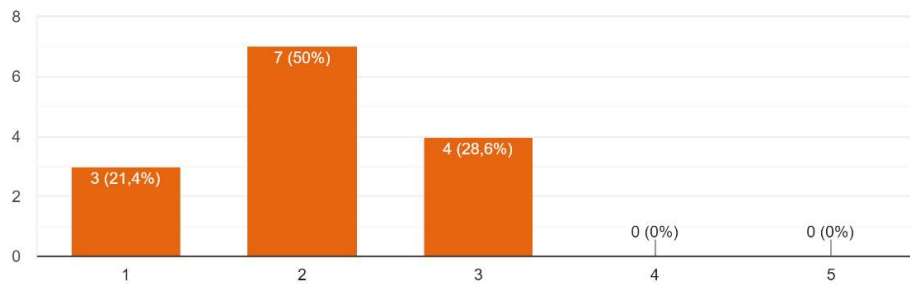
Saya menemukan berbagai fungsi di Website Pasien Covid-19 Monitoring Dashboard ini diintegrasikan dengan baik

14 jawaban



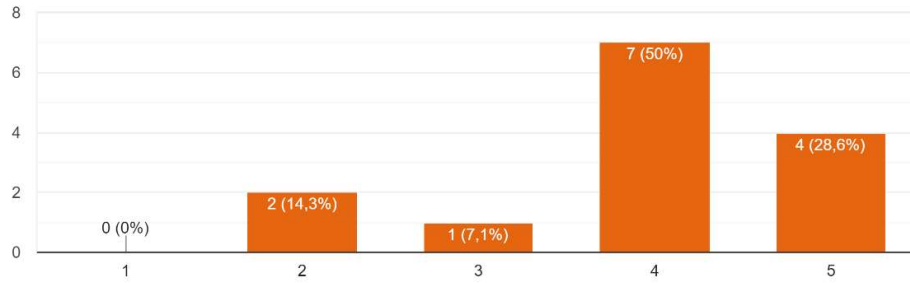
Saya pikir ada terlalu banyak ketidaksesuaian dalam Website Pasien Covid-19 Monitoring Dashboard

14 jawaban



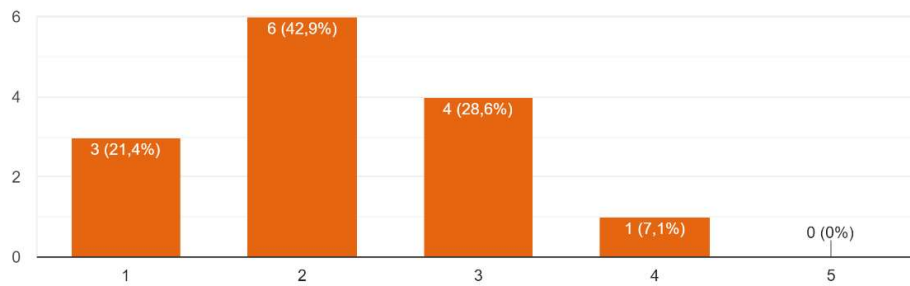
Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari Website Pasien Covid-19 Monitoring Dashboard dengan sangat cepat

14 jawaban



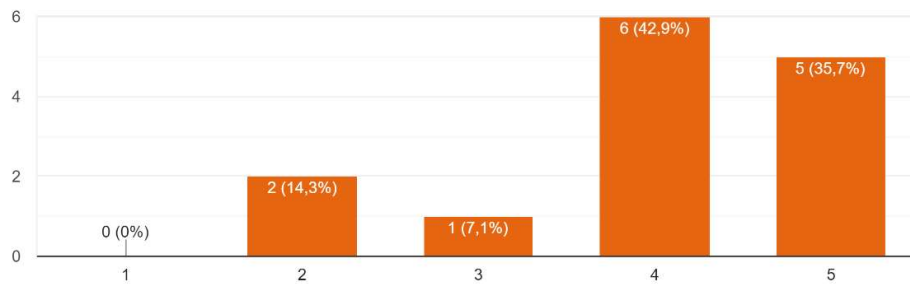
Saya menemukan, Website Pasien Covid-19 Monitoring Dashboard sangat rumit untuk digunakan

14 jawaban



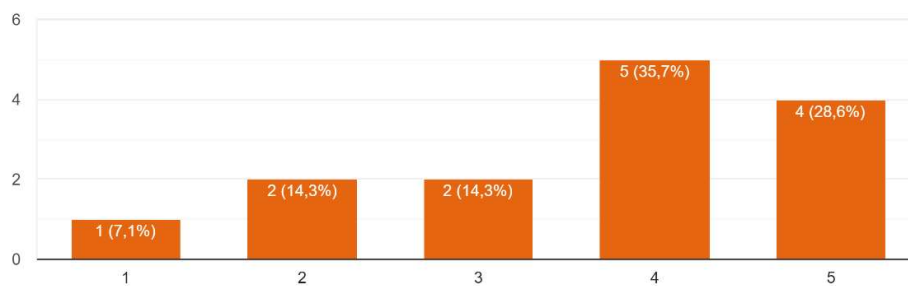
Saya merasa sangat percaya diri untuk menggunakan Website Pasien Covid-19 Monitoring Dashboard ini

14 jawaban



Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan Website Pasien Covid-19 Monitoring Dashboard

14 jawaban



Masukan lain untuk Website Pasien Covid-19 Monitoring Dashboard

8 jawaban

Fiturnya lebih dilengkapi lagi, terkait dengan monitoring pasien selain monitoring tanda tanda vital

Sebaiknya ditampilkan juga dokter penanggungjawab masing2 pasien
Trus pada tampilan penanganan pasien , sebaiknya terapinya dimunculkan juga

Bisa ditambahkan untuk pediatric dan maternal

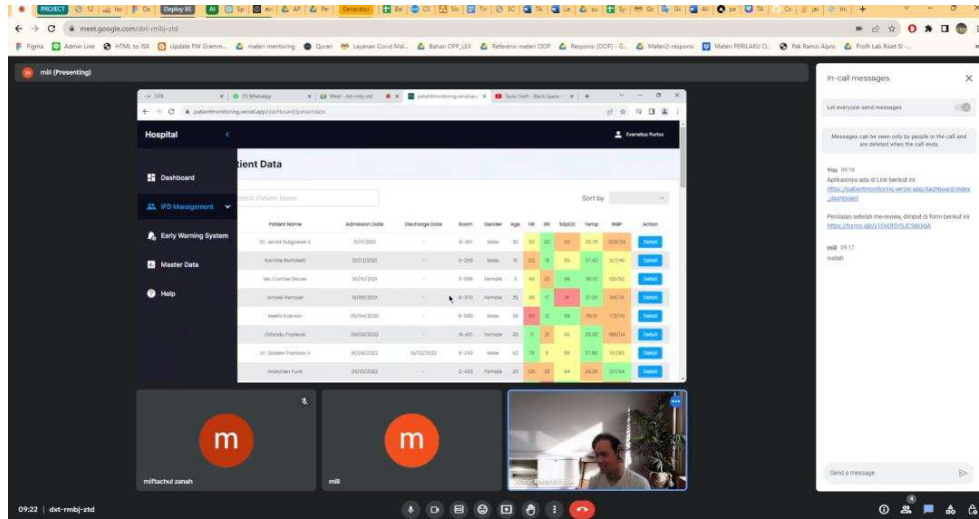
Di cobakan dulu ke pasien

Dapat terus dikembangkan

Semangat, mohon menggunakan bahasa indonesia saja agar kami mudah memahami, trima ksh

Sebaiknya menggunakan bahasa Indonesia agar lebih mudah dipahami pengguna, Terimakasih

Agar memudahkan alangkah baiknya ada pengaturan bahasa Indonesia



Lampiran D. Pelaksanaan *Card Sorting*

