

**PERANCANGAN APLIKASI PENJADWALAN *SHIFT*
PERAWAT MENGGUNAKAN ALGORITMA
LATE ACCEPTANCE HILL CLIMBING HYPER-HEURISTIC
(STUDI KASUS RSU. WULAN WINDY)**

SKRIPSI

Oleh:

RAMA ARVIANSYAH HUTASUHUT

NIM. 2044133



**JENJANG PENDIDIKAN STRATA-1
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME
MEDAN
2024**

LEMBAR PENGESAHAN

**PERANCANGAN APLIKASI PENJADWALAN SHIFT PERAWAT
MENGUNAKAN ALGORITMA *LATE ACCEPTANCE HILL*
CLIMBING HYPER-HEURISTIC
(STUDI KASUS RSU. WULAN WINDY)**

SKRIPSI

Diajukan Untuk Melengkapi Persyaratan Guna

Mendapat Gelar Sarjana Strata Satu

Program Studi Teknik Informatika

Oleh:

RAMA ARVIANSYAH HUTASUHUT

NIM 2044123

Disetujui Oleh:

Dosen Pembimbing I



Dr. Huliman, M.Kom

Dosen Pembimbing II



Devi, S.kom., S.A., M.M

Medan, 25 Juli 2024 Diketahui

dan Disahkan Oleh:

Ketua STMIK TIME

Ketua Program Studi



Dr. Edi Wijaya, S.Kom., M.Kom., M.M

Robet, M.Kom

Kepada

Yth Ketua STMIK TIME

Dr. Edi Wijaya, S.Kom., M.Kom., M.M
di Medan

Hal : Surat Pernyataan Bebas Plagiat

Saya yang bertanda tangan di bawah ini :

Nama : Rama Arviansyah Hutasuhut
NIM : 2044133
Program Studi : Teknik Informatika
Angkatan : 2020
Judul Skripsi : Perancangan Aplikasi Penjadwalan Shift Perawat
Menggunakan Algoritma *Late Acceptance Hill Climbing*
Hyper-Heuristic (Studi Kasus RSUD. Wulan Windy)

Dengan ini menyatakan bahwa skripsi yang Saya tulis adalah karya ilmiah asli dari pikiran saya sendiri.

Apabila kemudian hari terbukti bahwa skripsi tersebut merupakan Plagiat, maka saya bersedia gelar dan ijazah yang saya peroleh ditinjau kembali.

Demikian surat pernyataan ini Saya perbuat dengan pikiran yang sehat, tanpa tekanan dari siapapun.

Diketahui oleh

Pembantu Ketua I



Hendri M.Kom.

Medan, 29 Juli 2024

Yang membuat pernyataan



Rama Arviansyah Hutasuhut

ABSTRAK

Penjadwalan merupakan kegiatan penting dalam mengatur waktu kerja untuk mencapai efisiensi dan optimalisasi pekerjaan. Penelitian ini membahas penjadwalan perawat, yang didefinisikan sebagai individu dengan pengetahuan dan keterampilan dalam memberikan asuhan keperawatan. Tantangan dalam penjadwalan perawat, yang dikenal sebagai Nurse Scheduling Problem (NSP), adalah membuat jadwal yang adil dan memenuhi berbagai batasan penjadwalan, seperti jumlah minimal perawat per shift dan jumlah maksimal libur per perawat. Penelitian ini menggunakan algoritma Late Acceptance Hill Climbing (LAHC) untuk optimasi penjadwalan, yang merupakan pengembangan dari metode hill climbing. LAHC dipilih karena kemudahannya dalam implementasi dan kemampuannya untuk mengakomodasi lebih banyak kandidat solusi. Hasil penelitian menunjukkan bahwa algoritma LAHC efektif dalam menghasilkan jadwal shift perawat yang adil di RSUD Wulan Windy, dengan aplikasi berbasis web yang mencakup berbagai parameter penjadwalan.

Kata kunci: Penjadwalan perawat, *Web*, Algoritma LAHC, *Hyper-Heuristic*

Abstract

Scheduling is a crucial activity in organizing work time to achieve efficiency and optimization of tasks. This research discusses nurse scheduling, defined as individuals with the knowledge and skills in providing nursing care. The challenge in nurse scheduling, known as the Nurse Scheduling Problem (NSP), is to create fair schedules that meet various scheduling constraints, such as the minimum number of nurses per shift and the maximum number of days off per nurse. This study uses the Late Acceptance Hill Climbing (LAHC) algorithm for scheduling optimization, which is an advancement of the hill climbing method. LAHC was chosen for its ease of implementation and its ability to accommodate more candidate solutions. The research results show that the LAHC algorithm is effective in producing fair nurse shift schedules at Wulan Windy General Hospital, with a web-based application encompassing various scheduling parameters.

Keywords: Nurse Scheduling, Web, Algorithm LAHC, Hyper-Heuristic

KATA PENGANTAR

Ucapan puji dan syukur kepada hadirat Tuhan Yang Maha Esa atas segala berkat, hikmat, dan pertolongan-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini tepat pada waktunya.

Tugas Akhir ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Program Strata I pada STMIK TIME. Penulis berharap agar hasil dari penelitian yang dilakukan ini dapat membawa banyak manfaat bagi berbagai pihak.

Dalam menyelesaikan tugas akhir ini, penulis banyak mendapat bantuan dari berbagai pihak. Oleh karena itu, melalui kesempatan ini penulis ingin menyampaikan terima kasih kepada:

1. Bapak Dr. Huliman, M.Kom., selaku Pembimbing I yang telah membimbing penulis selama mengerjakan skripsi ini.
2. Ibu Devi, S.KOM., S.AB., M.M, selaku Pembimbing II yang telah membimbing penulis selama mengerjakan skripsi ini.
3. Bapak Simon Kangga Lee, selaku Ketua Yayasan STMIK TIME Medan.
4. Bapak Dr. Edi Wijaya, S.Kom., M.Kom., M.M, selaku Ketua STMIK TIME Medan.
5. Bapak Robet, M.Kom., selaku Ketua Program Studi Teknik Informatika STMIK TIME Medan.
6. Seluruh Dosen STMIK TIME Medan yang telah mendidik dan membimbing penulis selama masa perkuliahan.
7. Teristimewa kepada orang tua dan keluarga tercinta, yang telah memberikan dukungan baik moral maupun materi kepada penulis selama ini.
8. Amelia selaku kekasih saya yang terus memberikan dukungan dengan tulus untuk berjuang menyelesaikan skripsi ini hingga tuntas.
9. Semua yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak dapat disebutkan satu persatu

Penulis menyadari dalam proses penyelesaian skripsi yang berjudul “Perancangan Aplikasi Penjadwalan *Shift* Perawat Menggunakan Algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* (Studi Kasus RSUD. Wulan Windy)” masih jauh dari sempurna, mengingat keterbatasan kemampuan yang penulis miliki dan terbatasnya waktu yang digunakan. Untuk itu, penulis mengharapkan saran, kritikan, dan masukan yang membangun demi perkembangan dan kemajuan penelitian selanjutnya. Semoga penulisan skripsi ini dapat bermanfaat bagi pihak yang berkepentingan.

Akhir kata penulis berharap agar laporan penulisan Tugas Akhir ini dapat berguna dan bermanfaat bagi pembaca.

Medan, 29 Juli 2024

Penulis,

Rama Arviansyah Hutasuhut

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat Penelitian.....	4
1.5 Metodologi Penelitian.....	5
1.6 Analisis Sistem.....	7
1.7 Perancangan Sistem.....	7
1.8 Sistematika Penulisan	8
BAB II TINJAUAN PUSTAKA.....	10
2.1 Konsep Sistem Informasi.....	10
2.1.1 Sistem.....	10
2.1.2 Informasi.....	13
2.1.3 Sistem Informasi	15
2.2 Penjadwalan	17
2.3 Algoritma <i>Hill Climbing</i>	18
2.4 Algoritma <i>Late Acceptance</i>	20
2.5 Algoritma <i>Hyper-Heuristic</i>	22
2.6 <i>Use Case Diagram</i>	24
2.7 <i>Website</i>	25
2.8 Basis Data	28
2.9 <i>Entity Relationship Diagram</i>	31
BAB III METODE PENELITIAN	33
3.1 Analisis Masalah	33
3.1.1 Analisis Proses Yang Berjalan Sesuai di Lapangan	33

3.1.2	Analisis Algoritma Yang Digunakan	35
3.1.3	Analisis Sistem Usulan.....	45
3.2	Perancangan	53
3.2.1	Perancangan Tampilan	53
3.2.2	Perancangan Basis Data.....	82
BAB IV HASIL DAN PEMBAHASAN.....		87
4.1	Hasil.....	87
4.2	Pembahasan.....	106
BAB V KESIMPULAN DAN SARAN.....		109
5.1	Kesimpulan	109
5.2	Saran	109
DAFTAR PUSTAKA		111

DAFTAR TABEL

Tabel 1.1 Jadwal Penelitian	5
Tabel 2.1 Simbol <i>Use Case Diagram</i> [14]	24
Tabel 2.2 Simbol <i>Entity Relationship Diagram</i> [14]	31
Tabel 3.1 Skenario <i>Use Case</i> : Mengelola Data Waktu	47
Tabel 3.2 Skenario <i>Use Case</i> : Mengelola Data Hari.....	47
Tabel 3.3 Skenario <i>Use Case</i> : Mengelola Data Waktu Berhalangan	48
Tabel 3.4 Skenario <i>Use Case</i> : Mengelola Data Perawat.....	49
Tabel 3.5 Skenario <i>Use Case</i> : Mengelola Data Pasien	49
Tabel 3.6 Skenario <i>Use Case</i> : Mengelola Data Perawat Pengurus Pasien	50
Tabel 3.7 Skenario <i>Use Case</i> : Mengelola Data Ruangan.....	51
Tabel 3.8 Skenario <i>Use Case</i> : Melakukan <i>Generate</i> Jadwal.....	51
Tabel 3.9 Skenario <i>Use Case</i> : Mengelola Data <i>User</i>	52
Tabel 3.10 Struktur Tabel Pasiens.....	82
Tabel 3.11 Struktur Tabel Days	83
Tabel 3.12 Struktur Tabel Perawats	83
Tabel 3.13 Struktur Tabel Migrations	83
Tabel 3.14 Struktur Tabel Rooms	83
Tabel 3.15 Struktur Tabel Schedules.....	84
Tabel 3.16 Struktur Tabel Settings.....	84
Tabel 3.17 Struktur Tabel Perawatans.....	85
Tabel 3.18 Struktur Tabel Timedays	85
Tabel 3.19 Struktur Tabel Times.....	85
Tabel 3.20 Struktur Tabel Time_not_available.....	86
Tabel 3.21 Struktur Tabel Users	86

DAFTAR GAMBAR

Gambar 2.1 <i>Framework Algoritma Hyper-Heuristic</i>	23
Gambar 3.1 <i>Use Case Diagram</i> Sistem Usulan	46
Gambar 3.2 Rancangan Tampilan <i>Login</i>	54
Gambar 3.3 Rancangan Tampilan <i>Dashboard</i>	55
Gambar 3.4 Rancangan Tampilan Kelola Data Waktu	56
Gambar 3.5 Rancangan Tampilan Tambah Data Waktu	57
Gambar 3.6 Rancangan Tampilan Ubah Data Waktu	58
Gambar 3.7 Rancangan Tampilan Kelola Data Hari	59
Gambar 3.8 Rancangan Tampilan Tambah Data Hari	60
Gambar 3.9 Rancangan Tampilan Ubah Data Hari	61
Gambar 3.10 Rancangan Tampilan Kelola Data Waktu Berhalangan	62
Gambar 3.11 Rancangan Tampilan Tambah Data Waktu Berhalangan	63
Gambar 3.12 Rancangan Tampilan Ubah Data Waktu Berhalangan	64
Gambar 3.13 Rancangan Tampilan Kelola Data Perawat	65
Gambar 3.14 Rancangan Tampilan Tambah Data Perawat	66
Gambar 3.15 Rancangan Tampilan Ubah Data Perawat	67
Gambar 3.16 Rancangan Tampilan Kelola Data Pasien	68
Gambar 3.17 Rancangan Tampilan Tambah Data Pasien	69
Gambar 3.18 Rancangan Tampilan Ubah Data Pasien	70
Gambar 3.19 Rancangan Tampilan Kelola Data Perawat Pengurus Pasien	71
Gambar 3.20 Rancangan Tampilan Tambah Data Perawat Pengurus Pasien	72
Gambar 3.21 Rancangan Tampilan Ubah Data Perawat Pengurus Pasien	73
Gambar 3.22 Rancangan Tampilan Kelola Data Ruangan	74
Gambar 3.23 Rancangan Tampilan Tambah Data Ruangan	75
Gambar 3.24 Rancangan Tampilan Ubah Data Ruangan	76
Gambar 3.25 Rancangan Tampilan <i>Form Generate Jadwal</i>	77
Gambar 3.26 Rancangan Tampilan Hasil <i>Generate Optimasi Jadwal</i>	78
Gambar 3.27 Rancangan Tampilan Kelola Data <i>User Admin</i>	79

Gambar 3.28 Rancangan Tampilan Tambah Data <i>User Admin</i>	80
Gambar 3.29 Rancangan Tampilan Ubah Data <i>User Admin</i>	81
Gambar 3.30 ERD Sistem Usulan.....	82
Gambar 4.1 Tampilan <i>Login</i>	87
Gambar 4.2 Tampilan <i>Dashboard</i>	88
Gambar 4.3 Tampilan Kelola Data Waktu	89
Gambar 4.4 Tampilan Tambah Data Waktu	89
Gambar 4.5 Tampilan Ubah Data Waktu	90
Gambar 4.6 Tampilan Kelola Data Hari.....	91
Gambar 4.7 Tampilan Tambah Data Hari	91
Gambar 4.8 Tampilan Ubah Data Hari.....	92
Gambar 4.9 Tampilan Kelola Data Waktu Berhalangan.....	93
Gambar 4.10 Tampilan Tambah Data Waktu Berhalangan.....	93
Gambar 4.11 Tampilan Ubah Data Waktu Berhalangan.....	94
Gambar 4.12 Tampilan Kelola Data Perawat	95
Gambar 4.13 Tampilan Tambah Data Perawat.....	95
Gambar 4.14 Tampilan Ubah Data Perawat	96
Gambar 4.15 Tampilan Kelola Data Pasien.....	97
Gambar 4.16 Tampilan Tambah Data Pasien	97
Gambar 4.17 Tampilan Ubah Data Pasien	98
Gambar 4.18 Tampilan Kelola Data Perawat Pengurus Pasien	99
Gambar 4.19 Tampilan Tambah Data Perawat Pengurus Pasien.....	99
Gambar 4.20 Tampilan Ubah Data Perawat Pengurus Pasien.....	100
Gambar 4.21 Tampilan Kelola Data Ruangan.....	101
Gambar 4.22 Tampilan Tambah Data Ruangan.....	101
Gambar 4.23 Tampilan Ubah Data Ruangan.....	102
Gambar 4.24 Tampilan <i>Form Generate Jadwal</i>	103
Gambar 4.25 Tampilan Hasil <i>Generate Optimasi Jadwal</i>	103
Gambar 4.26 Tampilan Hasil <i>Export File Excel</i> Hasil Generate Optimasi Jadwal	104
Gambar 4.27 Tampilan Kelola Data <i>User Admin</i>	105
Gambar 4.28 Tampilan Tambah Data <i>User Admin</i>	105

Gambar 4.29 Tampilan Ubah Data <i>User</i> Admin	106
--	-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penjadwalan merupakan suatu kegiatan yang sangat erat kaitannya dengan suatu pekerjaan . Penjadwalan diperlukan untuk mengatur waktu kerja , sehingga menghasilkan sebuah jadwal yang efisien, dengan harapan dapat menghasilkan pekerjaan yang optimal. Pada penelitian kali ini akan dibahas penjadwalan perawat [1].

Perawat didefinisikan sebagai seseorang yang memiliki pengetahuan, keterampilan dan kewenangan untuk memberikan asuhan keperawatan pada orang lain berdasarkan ilmu dan kiat yang dimilikinya dalam batas-batas kewenangan yang dimilikinya (UU RI No. 23 tahun 1992). Dampak dari panjangnya jam kerja perawat serta tugas yang banyak dikhawatirkan akan mempengaruhi kinerja , kondisi fisik maupun kehidupan sosial. Salah satu cara yang dapat dilakukan pihak manajemen rumah sakit adalah membuat kebijakan berupa penjadwalan kerja perawat secara adil kepada seluruh perawat yang tersedia.

Penjadwalan daftar jaga perawat merupakan sebuah proses penjadwalan yang mengatur waktu kerja perawat di sebuah rumah sakit. Permasalahan penjadwalan perawat atau dikenal dengan *Nurse Scheduling Problem* (NSP) menjadi hal yang menantang karena dengan jumlah perawat yang relatif terbatas disbanding banyaknya *shift* kerja ,penjadwalan dituntut untuk mendapatkan jadwal dengan beban kerja seadil mungkin untuk setiap perawat serta memenuhi

batasan-batasan penjadwalan yang ada. Pemenuhan semua batasan penjadwalan seringkali terhambat ketika satu batasan terpenuhi, namun ternyata batasan lain terlanggar. Misalnya batasan jumlah maksimal libur per perawat mungkin terlanggar Ketika batasan jumlah minimal perawat dalam satu *shift* kerja terpenuhi [2].

LAHC merupakan algoritma yang digunakan untuk optimasi. LAHC merupakan sebuah metode pencarian *local search*. Metode LAHC merupakan pengembangan dari metode *hill climbing* . yang sudah ada sebelumnya. Perbedaan antara LAHC dan *hill climbing* biasa yaitu pada titik penerimaan kandidat solusinya. Jika pada *hill climbing* biasa, kandidat solusi akan diterima sebagai solusi baru apabila kandidat solusi tersebut lebih baik dari solusi sebelumnya, sedangkan jika pada LAHC, kandidat solusi akan diterima sebagai solusi baru apabila solusi tersebut lebih baik dari pada iterasi sebelumnya. Oleh karena itu, LAHC dapat mencakup kandidat solusi yang lebih luas dari pada *hill climbing* biasa. Metode pencarian yang juga sering digunakan selain LAHC yaitu *Simulated Annealing* (SA), *Thresold Accepting* (TA) ,*the Great Deluge Algorithm* (GDA) , dan *Constraint Programming* (CP) . Metode LAHC dipilih oleh penulis karena metode ini cukup mudah untuk diimplementasikan, *simple* dan cukup efektif untuk menyelesaikan masalah pencarian penjadwalan. Jika dibandingkan dengan metode pencarian tersebut, LAHC juga memiliki keunggulan tersendiri yaitu tidak adanya *cooling schedule* untuk penerapannya [3].

Pentingnya penggunaan algoritma LAHC didukung dengan adanya penelitian-penelitian sebelumnya terkait dengan pembuatan jadwal yang bisa dijadikan referensi dalam pembuatan tugas akhir ini. diantaranya adalah penelitian

dengan judul “ *PERANCANGAN APLIKASI PERAWAT DENGAN ALGORITMA LATE ACCEPTANCE HILL CLIMBING HYPER-HEURISTIC MENGGUNAKAN BENCHMARK DATASET DARI RUMAH SAKIT NORWEGIA*” Rizal risnanda 2020 [4]. Pada penelitian ini variabel yang digunakan dalam proses penjadwalan adalah hari, waktu, informasi jaga, dokter, ruangan, waktu ketersediaan dan ketidaksediaan jaga perawat.

Berdasarkan latar belakang di atas, peneliti tertarik untuk melakukan penelitian mengenai penjadwalan dengan penerapan algoritma *late acceptance hill climbing* (LAHC) dan mengambil judul “**Perancangan Aplikasi Penjadwalan Shift Perawat Menggunakan Algoritma Late Acceptance Hill Climbing Hyper-Heuristic (Studi Kasus di RSUD Wulan Windy)**”.

1.2 Rumusan Masalah

Sesuai uraian latar belakang masalah tersebut, dirumuskan permasalahan pada penelitian ini, yaitu:

1. Bagaimana membangun aplikasi penjadwalan *shift* perawat pada RSUD Wulan Windy berbasis *web*?
2. Bagaimana penerapan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* (LAHC) untuk menyelesaikan penjadwalan *shift* perawat?

1.3 Batasan Masalah

Berikut ini akan dilakukan pembatasan masalah penelitian dengan tujuan agar penelitian dapat diselesaikan tepat pada waktu yang ditentukan antara lain:

1. Penjadwalan dibuat untuk periode satu bulan (akumulasi 30 hari).

2. Data yang digunakan berasal dari RSUD. Wulan Windy yang berlokasi di Kecamatan Medan Marelan, Kota Medan.
3. Penjadwalan yang dibuat hanya 1 bulan, karena sifatnya masih *prototype*.
4. Penerapan algoritma dibangun menggunakan bahasa pemrograman PHP, HTML, dan CSS.
5. Data perawat yang disusun bertugas di ruang rawat inap dan dibagi ke grup 3A, 3B, 4A, 4B, 5A, 5B. Masing-masing grup lantai terdiri dari 14 sampai 21 perawat.
6. Penjadwalan perawat terbagi 3 *shift* pagi, sore, dan malam.
7. Sistem yang dibangun berbasis web.
8. Hasil *generate* penjadwalan berupa jadwal *shift* perawat untuk periode satu minggu.

1.4 Tujuan dan Manfaat Penelitian

Tujuan yang diharapkan dalam penelitian ini adalah:

1. Untuk membangun aplikasi penjadwalan *shift* perawat pada RSUD. Wulan Windi berbasis *web*.
2. Untuk menerapkan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* (LAHC) untuk menyelesaikan penjadwalan *shift* perawat.

Manfaat yang diharapkan dalam penelitian ini adalah:

1. Manfaat Teoritis:
 - a. Penelitian ini dapat menambah wawasan terkait penerapan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* terhadap optimasi penjadwalan *shift* perawat.

2. Bahan dan Alat

Bahan dan alat penelitian dibagi menjadi dua (2) yaitu dari sisi *hardware* (perangkat keras) dan *software* (perangkat lunak).

Hardware yang digunakan untuk melakukan penelitian yaitu sebuah laptop dengan spesifikasi:

- a. Processor Intel® Core™ i5-8300H CPU @ 2,30GHz
- b. RAM 8GB.
- c. Perangkat *keyboard*.
- d. Perangkat *touchpad*.
- e. *Display* dengan resolusi 1920 x 1080 pixel.

Software yang digunakan untuk melaksanakan penelitian, yaitu:

- a. Sistem Operasi Windows 11 64 Bit.
- b. XAMPP Control Panel.
- c. Sublime Text 3.

3. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini antara lain sebagai berikut:

- a. Metode Observasi

Melakukan observasi secara langsung yaitu mendatangi lokus penelitian yaitu RSUD. Wulan Windy untuk mengumpulkan informasi data hari, waktu, informasi jaga, dokter, ruangan, waktu ketersediaan dan ketidaksediaan jaga perawat.

b. Metode Wawancara

Melakukan wawancara secara langsung terhadap kepala unit RSUD. Wulan Windy untuk mendapatkan informasi data tentang sistem penjadwalan *shift* perawat, yang dilakukan secara terstruktur.

c. Metode Studi Pustaka

Mengumpulkan data teori melalui jurnal, media cetak, ataupun sumber-sumber referensi dari internet.

1.6 Analisis Sistem

Analisis sistem pada penelitian ini terbagi menjadi 3 tahapan proses, yaitu:

1. Analisis proses yang berjalan sesuai di lapangan yaitu menguraikan sistem berjalan yang digunakan dalam menentukan penjadwalan jaga perawat.
2. Analisis algoritma yang digunakan berisikan tahapan-tahapan yang berupa penguraian cara kerja algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* dalam melakukan penjadwalan jaga perawat.
3. Analisis sistem usulan yaitu menggambarkan sistem usulan yang akan dibangun beserta fitur-fitur yang tersedia yang dimodelkan dengan menggunakan *tools* pemodelan sistem *Use Case Diagram*.

1.7 Perancangan Sistem

Tahapan perancangan sistem yang dilakukan pada penelitian ini adalah berupa perancangan tampilan aplikasi dengan menggunakan *software* Balsamiq Mockup 3 dan perancangan basis data dengan menggunakan *Entity Relationship Diagram* (ERD).

BAB V KESIMPULAN DAN SARAN

Bagian ini berisikan mengenai kesimpulan yang diperoleh setelah melakukan penelitian dengan proses analisa terhadap teori dan hasil pengujian. Selain itu, juga diberikan saran yang berisikan hal-hal yang belum berhasil diselesaikan sehingga dapat dilanjutkan pada penelitian berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

Secara umum, konsep dari sistem informasi disusun oleh 2 bagian yaitu sistem dan informasi.

2.1.1 Sistem

Sistem adalah suatu kumpulan dari prosedur-prosedur, saling berhubungan dan membentuk jaringan kerja untuk mencapai tujuan tertentu. Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu [5].

Sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai tujuan dari sistem tersebut [6].

Sistem adalah sekumpulan unsur/elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan [7].

Suatu sistem untuk mencapai tujuan tertentu harus memiliki sifat-sifat dasar yang menjadikannya suatu karakter. Suatu sistem untuk mencapai tujuan tertentu harus memiliki sifat-sifat dasar yang menjadikannya suatu karakter. Karakteristik suatu sistem terdiri dari komponen-komponen (*components*), batas

sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objectives*) atau tujuan (*goal*) [5].

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu sub sistem. Setiap sub sistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem, dapat bersifat menguntungkan maupun merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu sub sistem dengan sub sistem lainnya. Keluaran (*output*) dari satu sub sistem akan menjadi masukan (*input*) untuk sub sistem yang lainnya dengan melalui penghubung.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikan menjadi keluaran yang berguna dan sisi pembuangan. Keluaran dapat merupakan masukan untuk sub sistem yang lain atau kepada super sistem.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem pasti mempunyai sasaran (*objectives*) atau tujuan (*goal*), kalau tidak mempunyai sasaran maka operasi sistem tidak akan ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

Berdasarkan informasi dari berbagai sumber tersebut, dapat disimpulkan bahwa sistem adalah suatu entitas terorganisir yang terdiri dari berbagai komponen atau unsur yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu. Sistem memiliki karakteristik dasar yang mencakup komponen sistem, batas sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolah sistem, dan sasaran sistem.

2.1.2 Informasi

Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang [8].

Informasi adalah data yang telah diproses, diorganisasikan atau diringkas menjadi sesuatu yang berguna [6].

Informasi adalah aset yang sangat vital dalam sebuah organisasi dan wajib diproteksi karena informasi merupakan aset yang berharga bagi organisasi [9].

Informasi harus memiliki nilai jika dapat mengubah tindakan yang diambil. Perubahan ini penting untuk dipahami karena data sebenarnya tidak memiliki nilai dalam pengambilan keputusan, hanya informasi yang memiliki nilai, dalam arti informasi memudahkan manajer dalam mengambil keputusan. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data nyata akan menggambarkan suatu kejadian-kejadian dan kesatuan nyata [8].

Data yang diolah untuk menghasilkan informasi menggunakan model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, yang berarti menghasilkan suatu keputusan dan melakukan tindakan yang lain akan membuat sejumlah data kembali. Data tersebut akan di *input*, diproses kembali lewat suatu model yang disebut dengan siklus informasi (*information cycle*). Siklus ini juga disebut dengan siklus pengelolaan data (*data processing cycles*). Kualitas informasi terdiri dari 3 hal yaitu [5]:

1. Informasi harus akurat (*accurate*) berarti informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.
2. Tepat pada waktunya (*time lines*) merupakan informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah asing tidak akan memiliki nilai lagi.
3. Relevan (*relevance*) berarti informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

Berdasarkan informasi dari berbagai sumber tersebut, dapat disimpulkan bahwa informasi adalah hasil pengolahan data yang telah diproses, diorganisir, atau diringkas sehingga menjadi sesuatu yang memiliki nilai dan kegunaan bagi penerimanya. Informasi memiliki peran penting sebagai dasar dalam pengambilan keputusan, baik secara langsung pada saat itu juga maupun secara tidak langsung pada masa mendatang.

Lebih lanjut, informasi dianggap sebagai aset vital dalam sebuah organisasi dan perlu diproteksi karena nilainya yang berharga. Proses transformasi data menjadi informasi melibatkan suatu model proses, yang sering disebut sebagai siklus informasi atau siklus pengelolaan data. Data yang diolah melalui siklus ini menjadi informasi, yang kemudian diterima oleh penerima, menghasilkan keputusan, dan memicu tindakan yang dapat menghasilkan sejumlah data kembali.

Kualitas informasi juga menjadi hal krusial, dengan tiga elemen utama, yaitu keakuratan (*accurate*), ketepatan waktu (*time lines*), dan relevansi (*relevance*). Informasi harus bebas dari kesalahan dan tidak menyesatkan (akurat),

harus disampaikan pada waktu yang tepat tanpa terlambat (tepat pada waktunya), dan harus memiliki manfaat atau relevansi bagi pemakainya.

Dengan demikian, informasi bukan hanya sekadar data yang diolah, tetapi merupakan output yang bernilai tinggi, mampu membantu pengambilan keputusan, dan memiliki dampak yang signifikan pada tindakan organisasi. Oleh karena itu, perlindungan dan manajemen kualitas informasi menjadi suatu keharusan dalam lingkungan organisasi modern.

2.1.3 Sistem Informasi

Sistem informasi merupakan sekumpulan *hardware, software, brainware*, prosedur, dan atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna untuk memecahkan masalah dan pengambilan keputusan. Sistem informasi adalah kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan *output* baik dalam bentuk gambar, suara maupun tulisan [5].

Sistem informasi adalah sebuah sistem yang terdiri dari entitas-entitas yang saling berhubungan dan bekerjasama untuk mengumpulkan, mengolah, menyimpan, mengelola, dan mendistribusikan informasi guna mendukung pengambilan keputusan, pengawasa, analisis, dan pengelolaan suatu organisasi atau bisnis [9].

Sistem informasi adalah suatu sistem yang menyediakan informasi untuk mengambil keputusan dan juga menjalankan suatu bisnis yang ada di perusahaan [10].

Sistem informasi juga dapat dikatakan sebagai kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Pendapat tersebut mengemukakan, bahwa sistem informasi merupakan kumpulan kegiatan yang diintegrasikan antara program kerja, informasi ke dalam suatu *server database* sehingga keinginan suatu organisasi dalam mencapai tujuan bisa terwujud [5].

Komponen sistem informasi terdiri dari [5]:

1. *Hardware* (perangkat keras), terdiri dari komputer, printer dan jaringan.
2. *Software*, kumpulan perintah yang ditulis dengan aturan untuk memerintah komputer melaksanakan tugas tertentu.
3. *Data*, merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi
4. *Manusia*, yang terlibat dalam komponen seperti operator dan pimpinan.
5. *Prosedur*, dokumentasi proses sistem buku penuntun operasional (aplikasi) dan teknis.

Sistem Informasi memiliki tujuan di antaranya adalah suatu sistem informasi yang baik haruslah relevan agar dapat memperoleh hasil yang bagus dalam menentukan suatu keputusan dalam sebuah kelompok. Seluruh komponen sistem informasi dan data-data, harus memberikan hasil yang bermanfaat sekurang-kurangnya sebanyak pengeluaran yang telah dilakukan [11].

Peranan sistem informasi yang sering kita jumpai dalam kehidupan sehari-hari, contoh di antaranya yaitu [11]:

1. Sistem akademis berbasis *website*, yang dapat mempermudah kita untuk mengakses data terkait akademisi.

2. Sistem pemerintahan berbasis *website*, atau *e-goverment*, yang dapat memberikan layanan pemerintahan melalui internet.
3. Sistem pertukaran data digital, dengan menggunakan komputer kita dapat bertukar data secara cepat.
4. Sistem biometrik yaitu sistem yang dikembangkan untuk keamanan informasi dengan menggunakan pola wajah dan sidik jari agar tidak sembarang orang melakukan akses.

Berdasarkan informasi dari berbagai sumber tersebut, dapat disimpulkan bahwa sistem informasi adalah suatu kumpulan komponen yang terintegrasi, termasuk *hardware*, *software*, *brainware*, prosedur, dan aturan, yang bekerja bersama untuk mengolah data menjadi informasi yang bermanfaat. Sistem informasi tidak hanya mencakup teknologi, tetapi juga melibatkan manusia dan prosedur dalam rangka mencapai tujuan suatu organisasi atau bisnis.

2.2 Penjadwalan

Jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau rencana kegiatan dengan pembagian waktu pelaksanaan terperinci, sedangkan penjadwalan adalah proses, cara, pembuatan menjadwalkan atau memasukkan dalam jadwal. Berdasarkan Kamus Besar Bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja. Jadwal juga didefinisikan sebagai daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan penjadwalan merupakan proses penyusunan daftar

pekerjaan yang akan dilakukan untuk mencapai atau mewujudkan suatu tujuan tertentu yang juga memuat tabel waktu pelaksanaan [12].

Penjadwalan adalah pengalokasian aktivitas ke dalam slot waktu yang ditentukan sehingga memudahkan proses pengaturan. Penjadwalan perawat merupakan sebuah proses penjadwalan yang mengatur waktu kerja perawat di sebuah rumah sakit. Proses pembuatan jadwal daftar jaga perawat berkaitan dengan beberapa komponen berikut yaitu jumlah perawat, ruang perawatan pasien, hari kerja dan *shift*. Proses pembuatan jadwal daftar jaga perawat yang harus dipenuhi sehingga mendapatkan jadwal yang akurat dan meningkatkan kinerja serta mutu pelayanan rumah sakit [13].

2.3 Algoritma *Hill Climbing*

Hill Climbing merupakan salah satu contoh dari *local search*. Cara kerja *Hill Climbing* yaitu mencari solusi yang lebih baik pada setiap solusi *neighbor*. Pencarian solusi dengan menggunakan *Hill Climbing* akan berhenti ketika tidak ditemukannya kandidat solusi pada *neighbor* yang lebih baik. Kelebihan *Hill Climbing* yaitu [14]:

1. *Hill Climbing* adalah algoritma sederhana dari intuitif yang mudah dipahami dan diterapkan.
2. Ini dapat digunakan dalam berbagai masalah optimasi, termasuk yang memiliki ruang pencarian yang besar dan keterbatasan yang kompleks.
3. *Hill Climbing* sering sangat efisien dalam menemukan *local optima*, menjadikannya pilihan yang baik untuk masalah dimana solusi yang baik dibutuhkan dengan cepat.

4. *Hill climbing* dapat dengan mudah dimodifikasi dan diperluas untuk mencakup heuristik atau keterbatasan tambahan.

Kelemahan *Hill Climbing* yaitu [14]:

1. *Hill Climbing* dapat terjebak dalam *local optima*, yang berarti bahwa itu mungkin tidak menemukan optimum global dari masalah.
2. Algoritma ini sensitif terhadap pilihan solusi awal, dan solusi awal yang buruk dapat mengakibatkan solusi akhir yang buruk.
3. *Hill Climbing* tidak mengeksplorasi ruang pencarian dengan sangat menyeluruh, yang dapat membatasi kemampuan untuk menemukan solusi yang lebih baik.
4. Ini mungkin kurang efektif dari pada algoritma optimasi lainnya, seperti algoritma genetik atau simulasi analisis, untuk jenis masalah tertentu.

Cara kerja algoritma *Hill Climbing* dapat dijelaskan sebagai berikut [14]:

1. Inisialisasi: Algoritma *Hill Climbing* memulai pencarian dari sebuah solusi awal yang dipilih secara acak atau diberikan.
2. Evaluasi: Setelah mendapatkan solusi awal, algoritma mengevaluasi kualitas solusi tersebut dengan menggunakan suatu fungsi evaluasi atau fungsi tujuan (*objective function*). Fungsi evaluasi ini memberikan skor atau nilai terhadap setiap solusi, menunjukkan seberapa baik atau buruk solusi tersebut dalam menyelesaikan masalah.
3. Generasi Tetangga: Setelah mengevaluasi solusi awal, algoritma mencoba untuk memperbaiki solusi tersebut dengan melakukan perubahan kecil pada solusi tersebut. Perubahan ini menghasilkan solusi alternatif yang disebut sebagai "tetangga" dari solusi awal.

4. Seleksi Tetangga Terbaik: Dari sekumpulan solusi tetangga yang dihasilkan, algoritma memilih solusi tetangga yang memiliki nilai evaluasi terbaik (misalnya, nilai fungsi evaluasi yang paling tinggi dalam kasus masalah optimisasi maksimisasi, atau nilai yang paling rendah dalam kasus optimisasi minimisasi).
5. Iterasi: Proses evaluasi, generasi tetangga, dan seleksi tetangga terbaik dilakukan secara berulang-ulang (iteratif) hingga suatu kondisi berhenti terpenuhi. Kondisi berhenti ini dapat berupa mencapai batasan jumlah iterasi, mencapai solusi yang memenuhi kriteria tertentu, atau tidak ada lagi solusi tetangga yang lebih baik dari solusi saat ini.
6. Konvergensi: Algoritma berhenti ketika tidak ada lagi perubahan yang dapat meningkatkan solusi, dan solusi yang ditemukan dianggap sebagai solusi terbaik atau solusi yang memenuhi kriteria yang diinginkan.

2.4 Algoritma *Late Acceptance*

Late Acceptance adalah sebuah pendekatan baru mengenai teknik *meta-heuristic* dengan tujuan umum. Algoritma ini diusulkan oleh Burke dan Bykov dalam penelitiannya mengenai permasalahan penjadwalan ujian. Meskipun saat ini penelitian mengenai penggunaan algoritma *Late Acceptance* masih dalam tahap awal, metode ini terbukti telah menunjukkan performa yang menjanjikan. Algoritma *Late Acceptance* termasuk dalam kelompok teknik pencarian iteratif tetapi menggunakan mekanisme penerimaan yang lebih canggih [1].

Hal ini tidak lazim digunakan pada sebagian besar metode pencarian *meta-heuristic* (contohnya pada *Hill-Climbing* dan *Late acceptance*), dimana pada

masing-masing iterasi, kandidat solusi baru yang dihasilkan akan dibandingkan dengan solusi yang sudah ada saat ini. Namun sebaliknya, pada algoritma *Late Acceptance*, ide utamanya adalah membandingkan kandidat solusi dengan solusi “saat ini” pada beberapa iterasi sebelumnya. Sejalan dengan itu, setiap solusi saat ini tidak langsung digunakan sebagai perbandingan pada saat itu juga, tetapi digunakan pada beberapa iterasi mendatang [1].

Dalam perbandingan inilah yang kemudian menginspirasi lahirnya sebuah metode *meta-heuristic* baru, yang kemudian juga memungkinkan penggunaan aturan *greedy acceptance* yang paling sederhana. Versi terakhir dari algoritma *Late Acceptance* yaitu membandingkan fungsi biaya dari kandidat dengan kompetitor “sebelumnya” dan hanya kandidat dengan biaya yang lebih baik (atau sama) yang diterima. Berikut adalah cara kerja algoritma *Late Acceptance* [1]:

1. Inisialisasi: Algoritma dimulai dengan menginisialisasi solusi awal secara acak atau dengan metode lain yang sesuai dengan masalah yang dihadapi.
2. Penilaian: Solusi awal dievaluasi menggunakan fungsi tujuan (*objective function*) untuk mendapatkan nilai evaluasi atau skor. Nilai ini mencerminkan kualitas solusi dalam menyelesaikan masalah yang diberikan.
3. Penyesuaian: Setelah mendapatkan nilai evaluasi untuk solusi awal, algoritma akan memperhitungkan sejarah nilai-nilai solusi dalam rentang waktu tertentu yang disebut sebagai jendela waktu (*window*). Algoritma akan mempertimbangkan solusi yang diterima lebih belakangan jika nilai solusi tersebut lebih baik daripada solusi yang ada dalam jendela waktu tersebut.
4. Generasi Solusi Tetangga: Algoritma akan menghasilkan solusi tetangga dari solusi saat ini dengan melakukan modifikasi kecil pada solusi tersebut.

Modifikasi ini dapat berupa perubahan pada satu atau beberapa komponen solusi, tergantung pada jenis masalah yang dihadapi.

5. Evaluasi Solusi Tetangga: Solusi tetangga dievaluasi menggunakan fungsi tujuan yang sama untuk mendapatkan nilai evaluasi atau skor.
6. Penentuan Solusi Baru: Algoritma *Late Acceptance* mempertimbangkan nilai solusi tetangga baru dan membandingkannya dengan nilai solusi yang ada dalam jendela waktu. Jika nilai solusi tetangga baru lebih baik daripada nilai solusi yang ada dalam jendela waktu, solusi tersebut diterima sebagai solusi saat ini. Jika tidak, solusi saat ini tetap dipertahankan.
7. Iterasi: Proses evaluasi, generasi solusi tetangga, dan penentuan solusi baru dilakukan secara berulang-ulang hingga kriteria berhenti terpenuhi. Kriteria berhenti ini dapat berupa mencapai batasan jumlah iterasi, mencapai solusi yang memenuhi kriteria tertentu, atau tidak adanya peningkatan yang signifikan dalam solusi selama beberapa iterasi.
8. Konvergensi: Algoritma berhenti ketika tidak ada lagi perubahan yang signifikan dalam solusi, dan solusi yang ditemukan dianggap sebagai solusi optimal atau solusi yang memenuhi kriteria yang diinginkan.

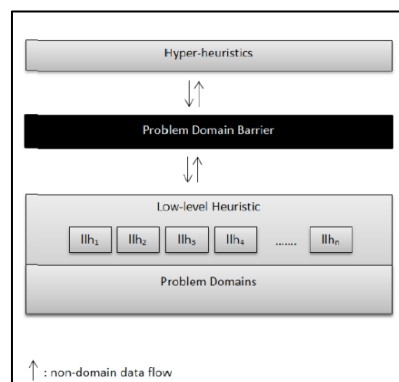
2.5 Algoritma *Hyper-Heuristic*

Ide dasar dari algoritma *Hyper-Heuristic* adalah gagasan untuk mengembangkan sebuah algoritma yang lebih umum yang bertentangan dengan pendekatan *meta-heuristic* standar. Umumnya, kelemahan dari algoritma *meta-heuristic* adalah algoritma memerlukan parameter tuning yang intensif dan memerlukan pengetahuan problem domain yang spesifik. Sehingga, untuk

problem instance yang berbeda diperlukan parameter *tuning* yang berbeda pula. Hal ini pada akhirnya akan berpengaruh pada performa algoritma. Artinya, satu *problem instance* dapat mempunyai performa algoritma yang baik namun *problem instance* lainnya dapat mempunyai performa algoritma yang sangat buruk [2].

Oleh karena itulah, dilakukan pendekatan dengan menggunakan algoritma *Hyper-Heuristic*. *Hyper-Heuristic* didefinisikan sebagai kumpulan pendekatan yang bertujuan untuk mengotomasi proses. Otomasi yang dilakukan tersebut biasanya dilakukan melalui metode *machine learning*. Tujuannya adalah memilih dan mengombinasikan *heuristic* yang sederhana atau menghasilkan *heuristic* baru dengan komponen *heuristic* yang sudah ada, sehingga dapat menyelesaikan permasalahan pencarian komputasi yang sangat sulit dilakukan secara manual [2].

Gambar 2.1 menunjukkan *framework* algoritma *Hyper-Heuristic* [2].



Gambar 2.1 Framework Algoritma *Hyper-Heuristic*

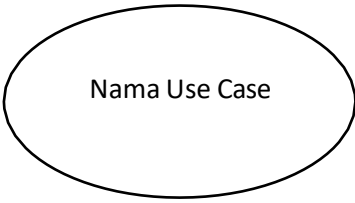
Pada *framework* ini digambarkan bahwa algoritma *Hyper-Heuristic* menggunakan *problem domain barrier* yang mengakibatkan strategi *Hyper-Heuristic* tidak bersinggungan langsung dengan *solution space*, sebagaimana yang biasa terjadi pada pendekatan *meta-heuristic*. Sebaliknya, algoritma *Hyper-Heuristic* hanya bersinggungan dengan *heuristic* pada *level* yang lebih rendah. Algoritma *Hyper-Heuristic* bekerja di atas *search space* berupa *heuristic* pada

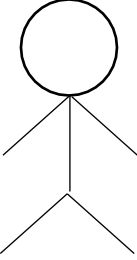
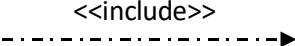
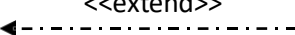
level yang lebih rendah, sedangkan *meta-heuristic* bekerja di atas *search space* berupa *solution space*. Dengan adanya *search space* yang lebih tinggi ini, algoritma *hyper-heuristic* tidak bergantung pada *problem domain* tertentu saja sehingga tidak diperlukan *parameter tuning* secara manual untuk setiap *problem domain*. Hal ini dikarenakan algoritma *Hyper-Heuristic* sudah mampu melakukan mekanisme *parameter tuning* secara otomatis [2].

2.6 Use Case Diagram

Use Case Diagram adalah salah satu alat pemodelan sistem yang merupakan komponen dari *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem. Penggunaan Diagram *Use Case* dapat memberikan gambaran yang efektif dalam menjelaskan konteks suatu sistem dengan jelas, sehingga memperlihatkan batasan sistem tersebut dengan jelas. Terdapat beberapa simbol-simbol dari *Use Case Diagram* seperti yang ditunjukkan pada Tabel 2.1 berikut [15].

Tabel 2.1 Simbol Use Case Diagram [15]

Nama	Simbol	Deskripsi
<i>Use Case</i>		Simbol yang menggambarkan fungsi dari sistem umumnya diekspresikan dalam bentuk kata kerja di awal setiap frasa.

Aktor		Simbol yang mewakili pengguna yang terlibat dalam <i>Use Case</i> biasanya menyatakan orang-orang yang berinteraksi dengan sistem.
Asosiasi		Simbol yang menghubungkan antara <i>Use Case</i> dan aktor menggambarkan komunikasi antara keduanya. Asosiasi digunakan untuk menggambarkan hubungan statis antara elemen-elemen tersebut.
<i>Include</i>		Simbol ini menggambarkan bahwa <i>Use Case</i> secara keseluruhan merupakan bagian dari fungsionalitas <i>Use Case</i> lainnya.
<i>Extend</i>		Simbol yang serupa dengan " <i>include</i> ", namun menunjukkan bahwa sebuah <i>Use Case</i> tambahan dapat berdiri sendiri tanpa membutuhkan <i>Use Case</i> yang ditambahkan.
Sistem		Simbol yang membatasi bahwa setiap interaksi dengan simbol lain harus berada di dalamnya.

2.7 Website

Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh seluruh siapapun yang terkoneksi jaringan internet. Jenis kategori *website* antara lain [16]:

1. *Web statis* merupakan *website* yang mempunyai halaman yang tidak berubah. Perubahan suatu halaman dilakukan secara manual dengan meng-*edit code* yang menjadi stuktur dari *website* tersebut.
2. *Web dinamis* merupakan *website* yang secara terstruktur diperuntukan untuk di-*update* sesering mungkin. Biasanya disediakan halaman *back-end* untuk melakukan perubahan konten dari *website* tersebut. Contohnya : *web portal*, *web berita*, dan lain-lain.
3. *Web interaktif* merupakan *website* yang berinteraksi antara penggunanya, biasanya berupa forum diskusi maupun *blog* dengan adanya moderator sebagai pengatur alur diskusi.

Santi adalah pembuatan aplikasi program dengan bahasa skrip yang akan menghasilkan sebuah aplikasi yang diakses pada *web browser*. Ada beberapa istilah yang sering digunakan pada Santi ini diantaranya yaitu [16]:

1. Internet

Internet berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan. Sistem komputer terhubung secara global dan menggunakan TCP/IP sebagai protokol. Melalui internet pertukaran informasi dapat dilakukan tanpa batas asalkan kedua pihak terhubung ke dalam jaringan yang sama.

2. *World Wide Web (WWW)*

WWW merupakan kumpulan *web server* di seluruh dunia yang dapat menyediakan data dan informasi untuk dapat digunakan secara massal.

3. *Website*

Website merupakan istilah yang merujuk kepada suatu nama halaman *web* yang dapat diakses jika terkoneksi dengan internet. Untuk mengakses sebuah *website*, selain terkoneksi dengan internet, maka *user* harus membukanya dalam sebuah aplikasi *web browser*.

4. *Web Server*

Web server merupakan perangkat lunak yang digunakan untuk menerima permintaan melalui protokol HTTP atau HTTPS dari *client*, kemudian dikirimkan kembali dalam bentuk halaman-halaman *web*. Contoh *web server*: Xampp, Apache2Triad, dan lain-lain.

5. *Universal Resource Locator (URL)*

URL merupakan suatu alamat yang menunjukkan sebuah halaman tertentu internet. Contoh URL adalah: <http://www.google.com>.

6. *Hypertext Transfer Protocol (HTTP)*

HTTP merupakan bagian dari URL digunakan untuk mengidentifikasi lokasi *web*, dan digunakan dalam protokol HTML.

7. *Domain Name System (DNS)*

DNS merupakan sistem database terdistribusi yang tidak banyak dipengaruhi oleh bertambahnya *database*. DNS menjamin informasi *host* terbaru akan disebarkan ke jaringan bila diperlukan.

8. *Internet Protocol (IP)*

IP merupakan protokol yang digunakan dalam internet, secara teknis bermakna suatu bentuk pengisian dan pengalamatan data dan informasi yang akan dikirim melalui internet.

9. *Hyperlink*

Hyperlink atau disebut *link* merupakan sebuah *tools* yang berperan dalam mempopulerkan penggunaan internet. *hyperlink* dapat mereferensikan sebuah teks atau gambar ke alamat lain di internet.

10. *Web browser*

Web browser mudah digunakan, yang diperlukan hanyalah harus memiliki alamat *web* yang akan dibuka. Alamat ini biasa disebut dengan *Uniform Resource Locator* (URL). Jika menggunakan sistem operasi Windows terdapat program *web browser* bawaan, yaitu *Internet Explorer*. Terdapat banyak program alternatif *web browser* yang sebagian besar bersifat gratis, seperti *Firefox*, *Opera*, *Safari*, *Chrome* dan seterusnya.

2.8 Basis Data

Data adalah fakta-fakta yang menggambarkan suatu kejadian yang sebenarnya pada waktu tertentu. Data didapatkan dari suatu kejadian yang benar-benar terjadi, misalnya dari transaksi penjualan, pembelian, dan sebagainya. Data identik dengan bukti transaksi yang terjadi di suatu perusahaan seperti kwitansi, faktur, formulir dan lain-lain. Data yang telah diproses kemudian dapat menghasilkan informasi berupa laporan, seperti laporan keuangan, laporan penjualan, dan sebagainya [17].

Basis data merupakan kumpulan informasi yang ada selama periode waktu yang lama, seringkali bertahun-tahun. Basis data merupakan hal yang sangat penting untuk semua bisnis. Basis data berada di belakang layar perusahaan besar maupun perusahaan kecil. Perusahaan menyimpan setiap data-data penting mereka ke dalam basis data. Kekuatan basis data berasal dari pengetahuan dan

teknologi yang telah berkembang dan diwujudkan dalam perangkat lunak khusus yang disebut sistem manajemen basis data, atau *Database Management System* (DBMS), atau disebut juga Sistem Basis Data [17].

DBMS adalah alat yang ampuh untuk membuat dan mengelola jumlah data yang besar secara efisien dan memungkinkannya bertahan dalam jangka waktu yang lama dengan aman. DBMS adalah kumpulan data yang saling terkait dan seperangkat program untuk mengakses data tersebut. Pengumpulan data, biasanya disebut sebagai basis data, berisi informasi yang relevan dengan suatu perusahaan. Tujuan utama DBMS adalah menyediakan cara untuk menyimpan dan mengambil informasi basis data yang nyaman dan efisien. Sistem basis data dirancang untuk mengelola banyak informasi. Manajemen data melibatkan struktur pendefinisian untuk penyimpanan informasi dan menyediakan mekanisme untuk manipulasi informasi. Selain itu, sistem basis data harus memastikan keamanan informasi yang disimpan, meskipun sistem macet atau upaya akses yang tidak sah. Jika data akan dibagikan di antara beberapa pengguna, sistem harus menghindari kemungkinan hasil yang tidak normal. DBMS diharapkan untuk [17]:

1. Memungkinkan pengguna untuk membuat basis data baru dan menentukan skema basis data.
2. Memberi pengguna kemampuan untuk meminta data dan memodifikasi data, menggunakan bahasa *query*.
3. Mendukung penyimpanan data dalam jumlah yang sangat besar dan banyak dalam jangka waktu yang lama.
4. Memungkinkan akses yang efisien ke data untuk permintaan dan modifikasi basis data.

5. Mendukung pemulihan basis data dalam menghadapi kegagalan, banyak kesalahan, atau penyalahgunaan yang disengaja.
6. Kontrol akses ke data dari banyak pengguna sekaligus.

Dalam penerapannya, terdapat beberapa jenis *software* DBMS yang sering diaplikasikan untuk mengelola *database* perusahaan yaitu diantaranya [17]:

1. *MySQL*

MySQL banyak digunakan di perusahaan karena memang tersedia secara gratis. Sehingga aplikasi ini cocok digunakan untuk bisnis-bisnis yang sedang berkembang. Meskipun tidak berbayar, namun tingkat keamanannya cukup baik dengan kecepatan akses data yang selalu stabil. Akan tetapi *MySQL* kurang kompatibel dengan bahasa pemrograman *Foxpro*, *Visual Basic* (VB) dan *Delphi* serta kurang mampu menangani data yang jumlahnya terlalu besar.

2. *Oracle*

Oracle merupakan perangkat lunak DBMS yang bagus dan berbayar. *Oracle* memiliki beragam fitur yang dapat memenuhi tuntutan fleksibilitas perusahaan besar. Bahkan *oracle* juga memiliki pemrosesan transaksi dengan performa yang sangat tinggi. Untuk memenuhi kriteria seperti pada pengertian DBMS, *oracle* tidak perlu diragukan lagi dalam hal keamanan.

3. *Microsoft SQL Server*

Selain *Oracle*, *Microsoft SQL Server* juga cocok diaplikasikan pada sistem jaringan komputer perusahaan-perusahaan besar karena memiliki kemampuan mengelola data yang besar. *Microsoft SQL Server* memiliki sistem pengamanan data yang baik dan memiliki fitur *back up*, *recovery* dan

rollback data. Namun, perangkat ini hanya bisa berjalan pada sistem operasi *Windows* saja.

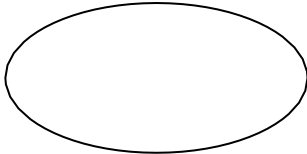
4. *Firebird*

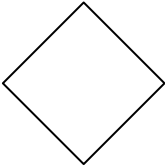
Perangkat lunak DBMS lainnya adalah *Firebird* sebagai sistem manajemen basis data yang relasional. *Firebird* menawarkan fitur yang sesuai dengan standar *SQL-2003* dan *ANSI SQL-99* serta dapat bekerja pada sistem operasi *Windows* dan *Linux*.

2.9 *Entity Relationship Diagram*

Entity Relationship Diagram merupakan teknik pendekatan *top-bottom* untuk desain *database*, yang dilakukan dengan proses berupa identifikasi informasi penting kemudian menggambarannya dalam sebuah model. *Entity Relationship Diagram* adalah model yang berguna membantu dalam memahami data dan bagaimana data tersebut digunakan dalam sebuah perusahaan. Terdapat beberapa simbol-simbol dari *Use Case Diagram* seperti yang ditunjukkan pada Tabel 2.2 berikut [15].

Tabel 2.2 Simbol *Entity Relationship Diagram* [14]

Nama	Simbol	Deskripsi
Entitas		Kumpulan dari objek yang dapat diidentifikasi secara unik biasanya merupakan nama dari tabel di dalam rancangan basis data.
Atribut		Gambaran karakteristik dari sebuah entitas yang biasanya dianggap sebagai nama kolom di dalam rancangan basis data.

Nama	Simbol	Deskripsi
Hubungan		Menyatakan jumlah himpunan relasi antar entitas.
Garis		Penghubung entitas, atribut dan hubungan.

BAB III

METODE PENELITIAN

3.1 Analisis Masalah

Sebelum memulai proses pengembangan sistem optimasi penjadwalan *shift* perawat menggunakan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* (LAHC), langkah pertama yang penting adalah melakukan analisis mendalam terhadap masalah yang akan diselesaikan. Dalam konteks ini, masalah yang dihadapi adalah efisiensi penjadwalan *shift* perawat di RSUD. Wulan Windy. Proses analisis masalah dibagi menjadi 3 yaitu analisis proses yang berjalan sesuai di lapangan, analisis algoritma yang digunakan, dan analisis sistem usulan.

3.1.1 Analisis Proses Yang Berjalan Sesuai di Lapangan

Analisis proses yang berjalan sesuai di lapangan yaitu menguraikan sistem berjalan yang digunakan dalam menentukan penjadwalan jaga perawat di RSUD. Wulan Windy. Berikut ini adalah sistem berjalan yang digunakan antara lain:

1. Pengumpulan Data Ketersediaan Perawat.

Pada tahap awal, manajer jadwal atau tim pengelola penjadwalan akan mengumpulkan data ketersediaan perawat. Ini mungkin melibatkan pengisian formulir oleh perawat yang mencantumkan ketersediaan mereka untuk *shift-shift* tertentu, baik secara manual atau melalui pengumpulan informasi yang dilakukan oleh manajer atau koordinator jadwal.

2. Perencanaan Kebutuhan Perawat.

Setelah data ketersediaan perawat terkumpul, manajer jadwal akan melakukan perencanaan kebutuhan perawat untuk periode tertentu. Ini biasanya didasarkan pada estimasi jumlah perawat yang diperlukan pada setiap *shift*, yang bisa dipengaruhi oleh faktor-faktor seperti volume pasien, kompleksitas kasus, dan kebijakan rumah sakit.

3. Penjadwalan Manual Menggunakan Kertas.

Dalam sistem konvensional ini, penjadwalan *shift* perawat umumnya dilakukan secara manual menggunakan kertas. Manajer jadwal atau koordinator jadwal akan menggunakan lembar kerja atau tabel jadwal untuk mencocokkan ketersediaan perawat dengan kebutuhan *shift* yang telah direncanakan.

4. Penetapan Jadwal Awal.

Setelah pencocokan dilakukan, jadwal awal akan ditetapkan dengan menempatkan perawat yang tersedia pada *shift* yang sesuai dengan ketersediaan mereka. Proses ini dapat melibatkan pemecahan konflik jadwal secara manual oleh manajer jadwal.

5. Komunikasi Jadwal Kepada Perawat.

Setelah jadwal awal ditetapkan, informasi jadwal tersebut akan dikomunikasikan kepada perawat terkait. Ini bisa dilakukan melalui pengumuman di papan pengumuman, pengiriman *email*, atau distribusi lembar jadwal cetak kepada perawat.

6. Pemantauan dan Penyesuaian Manual.

Selama periode jadwal berlangsung, manajer jadwal atau koordinator jadwal akan terus memantau ketersediaan perawat dan kebutuhan *shift*. Jika terjadi perubahan atau masalah, penyesuaian jadwal akan dilakukan secara manual, yang melibatkan komunikasi langsung dengan perawat terkait.

Oleh karena sistem penjadwalan *shift* perawat di RSUD. Wulan Windy masih menggunakan pendekatan konvensional yang melibatkan proses manual dengan menggunakan kertas, maka perlu dibangun sebuah sistem penjadwalan *shift* perawat berbasis teknologi untuk meningkatkan efisiensi dan akurasi dalam manajemen jadwal. Sistem ini akan menggantikan proses manual dengan alat yang memudahkan pengelolaan jadwal perawat secara efisien. Dengan memanfaatkan teknologi, sistem akan menggunakan algoritma optimasi untuk menentukan penjadwalan yang optimal berdasarkan ketersediaan perawat dan kebutuhan jadwal. Algoritma yang digunakan pada penelitian ini adalah *Late Acceptance Hill Climbing Hyper-Heuristic*.

3.1.2 Analisis Algoritma Yang Digunakan

Analisis algoritma yang digunakan berisikan tahapan-tahapan yang berupa penguraian cara kerja algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* dalam melakukan penjadwalan jaga perawat. Ada beberapa tahapan saat melakukan *generate* jadwal dengan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic*, yaitu:

1. Inisialisasi.

Pada tahap ini sistem mengambil semua data yang terdapat pada perawat, pasien, perawat pengurus pasien, hari, jam, kelas dan ruang. Data yang digunakan adalah data semester genap tahun perawatan 2018-2019. Disini penulis menggunakan parameter algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* yaitu:

Jumlah populasi = 5.

Probabilitas *crossover* = 0,70.

Probabilitas mutasi = 0,40.

Jumlah Generasi = 10.000.

Untuk pembentukan kromosom awal dilakukan secara acak sebanyak jumlah populasi parameter algoritma *Late Acceptance Hill Climbing Hyper-Heuristic*. Setelah melakukan pengacakan diambil 5 sampel tiap kromosom dan didapatkan jadwal sebagai berikut :

Kromosom 0: [Perawat 1:Senin:09.00-11.00:Rudi]

[Perawat 2:Rabu:17.00-19.00:Kevin]

[Perawat 3:Kamis :19.00-21.00:Santi]

[Perawat 4:Selasa:09.00-11.00:Mega]

[Perawat 1:Senin:09.00-11.00:Adela]

Kromosom 1: [Perawat 3:Rabu:19.00-21.00:Adeline]

[Perawat 4:Senin:19.00-21.00: Santi]

[Perawat 1:Kamis:09.00-11.00: Rudi]

[Perawat 1:Selasa:17.00-19.00: Kevin]

[Perawat 2:Jumat:17.00-19.00: Mega]

Kromosom 2: [Perawat 2:Jumat:19.00-21.00: Kevin]

[Perawat 1:Selasa:19.00-21.00:Mega]

[Perawat 1:Kamis:17.00-19.00:Grisella]

[Perawat 3: Senin:19.00-21.00:Simizu]

[Perawat 4:Rabu:09.00-11.00:Gloria]

Kromosom 3: [Perawat 2:Jumat:19.00-21.00: Kevin]

[Perawat 1:Selasa:19.00-21.00:Mega]

[Perawat 1:Kamis:17.00-19.00:Grisella]

[Perawat 3: Senin:19.00-21.00:Simizu]

[Perawat 4:Rabu:09.00-11.00:Gloria]

Kromosom 4: [Perawat 4:Rabu:09.00-11.00:Simizu]

[Perawat 3:Selasa:09.00-11.00:Santi]

[Perawat 1:Jumat:17.00-19.00:Rudi]

[Perawat 2:Kamis:19.00-21.00:Grisella]

[Perawat 1:Rabu:09.00-11.00: Adela]

2. Penilaian.

Pada tahapan ini setiap kromosom dihitung nilai *fitness*nya. Untuk menentukan nilai *fitness* maka digunakan persamaan :

$$Fitness = (1 / 1 + pinalti).$$

Dimana pinalti pada program ini adalah:

- a. Bentrok jadwal pasien wajib perawatan dengan jadwal perawat.
- b. Bentrok jadwal pasien wajib dengan jadwal perawatan.
- c. Bentrok hari, jam, dan ruang yang sama antar perawatan.
- d. Bentrok hari dan jam pada 2 perawatan atau lebih.

Untuk setiap *array* memiliki jumlah maksimum pinalti sebanyak 4. Jadi untuk tiap kromosom yang memiliki sebanyak 5 array maka kemungkinan terburuk pinalti tiap kromosom terdapat 20 pinalti.

Pada tiap sampel, terdapat maka didapat nilai pinalti dan *fitness* :

- a. Kromosom 0. pinalti = 7

$$\begin{aligned} Fitness &= (1/ 1+8) \\ &= 0,125 \end{aligned}$$

- b. Kromosom 1. pinalti = 13

$$\begin{aligned} Fitness &= (1/ 1+13) \\ &= 0,0714 \end{aligned}$$

- c. Kromosom 2. pinalti = 11

$$\begin{aligned} Fitness &= (1/ 1+11) \\ &= 0,083 \end{aligned}$$

- d. Kromosom 3. pinalti = 5

$$\begin{aligned} Fitness &= (1/ 1+5) \\ &= 0,33 \end{aligned}$$

- e. Kromosom 4. pinalti = 9

$$\begin{aligned} Fitness &= (1/ 1+9) \\ &= 0,1 \end{aligned}$$

Dari data tersebut, dapat disimpulkan bahwa kromosom 3 merupakan kromosom terbaik diantara 5 kromosom karena memiliki nilai *fitness* tertinggi.

3. Penyesuaian.

Pada tahapan ini, dipilihlah kromosom mana yang akan disilangkan dengan kromosom lain. Maka akan terjadi perubahan nilai *array* dari suatu komponen. Penulis menggunakan metode *rank selection* untuk seleksi. Untuk melakukan seleksi, pertama ditentukan ranking tiap kromosom berdasarkan nilai *fitness*, maka didapatkan peringkat :

Kromosom 3 = 0.33

Kromosom 1 = 0.125

Kromosom 4 = 0.1

Kromosom 2 = 0.083

Kromosom 1 = 0.0714

Dikarenakan kromosom 3 mendapatkan peringkat terbaik, maka nilai peringkat sebanyak jumlah populasi yang ada, yaitu 5. Dan untuk nilai peringkat populasi dibawahnya akan bernilai (n-1). Maka didapatkanlah nilai peringkatnya:

- a. Kromosom 3 = 5
- b. Kromosom 0 = 4
- c. Kromosom 4 = 3
- d. Kromosom 2 = 2
- e. Kromosom 1 = 1

Lalu total nilai ranking seluruh kromosom ditambahkan, maka didapatkanlah seluruh nilai ranking $(5+4+3+2+1) = 15$.

Setelah itu dibuatlah kumulatif dari seluruh nilai ranking yang didapatkan, maka didapatkan kumulatif :

- a. Kromosom 3 = 0-4

- b. Kromosom 0 = 5-9
- c. Kromosom 4 = 10-12
- d. Kromosom 2 = 13-14
- e. Kromosom 1 = 15

Setelah itu dibuatlah *roulette* bilangan acak dari 0-15. *Roulette* dilakukan sebanyak nilai populasi yaitu 5. *Roulette* dilakukan untuk menyeleksi kromosom mana yang akan di *crossover* dan dipilih pasangan kromosom yang terpilih dari seleksi. Hasil dari *roulette* tersebut adalah:

- a. Kromosom 3 = 3
- b. Kromosom 0 = 6
- c. Kromosom 4 = 11
- d. Kromosom 2 = 4
- e. Kromosom 1 = 7

Dari hasil *roulette* tersebut, maka hasil seleksi dari kromosom kromosom tersebut:

- a. [Kromosom 3] X [Kromosom 3]
- b. [Kromosom 0] X [Kromosom 0]
- c. [Kromosom 4] X [Kromosom 4]
- d. [Kromosom 2] X [Kromosom 3]
- e. [Kromosom 1] X [Kromosom 0]

Maka kromosom yang akan melalui proses *crossover* adalah kromosom 2 dan kromosom 1.

4. Generasi dan Evaluasi Solusi Tetangga.

Pada tahapan ini, kromosom yang terpilih dari seleksi akan mengalami proses *crossover*. Maka nilai tiap *array* dari kromosom 2 akan diganti dengan nilai kromosom 3. Begitu juga nilai *array* dari kromosom 1 akan diganti dengan nilai *array* kromosom 0. Metode yang digunakan oleh penulis adalah 2 *cut point*. Artinya diharapkan setidaknya ada 2 nilai yang berubah dari suatu *array*. Maka dipilihlah secara acak 2 nilai *array* yang berubah. Di penelitian ini, *crossover* dilakukan dengan membangkitkan bilangan persentase dari 0% - 100%. Untuk jumlah *roulette* sebanyak jumlah populasi, jadi *roulette* berjumlah 5. Disini penulis menggunakan probabilitas *crossover* 0.70 yang dipersentasekan bernilai 70%. Jadi, jika hasil dari *roulette* < 70%, maka nilai dari suatu *array* pada kromosom tersebut mengalami *crossover*.

Untuk proses persilangan kromosom 2:

Kromosom 2 : [0] hasil *roulette* = 43 %

[1] hasil *roulette* = 53 %

[2] hasil *roulette* = 69 %

[3] hasil *roulette* = 80 %

[4] hasil *roulette* = 72 %

Maka nilai array 0,1,2 pada kromosom 2 akan disilangkan dengan nilai array kromosom 3 dengan 2 *point cut* secara acak.

[Kromosom 2] X [Kromosom 3].

Maka hasil dari kromosom yang baru adalah:

Kromosom 2: [Perawat 2:Selasa:19.00-21.00: Adela]

[Perawat 1: Senin:19.00-21.00: Gloria]

[Perawat 1:Kamis :19.00-21.00:Santi]

[Perawat 3:Senin:19.00-21.00: Simizu]

[Perawat 4:Rabu:09.00-11.00:Gloria]

Untuk proses persilangan kromosom 1:

Kromosom 1 : [0] hasil *roulette* = 73 %

[1] hasil *roulette* = 33 %

[2] hasil *roulette* = 19 %

[3] hasil *roulette* = 99 %

[4] hasil *roulette* = 89 %

Maka nilai array 1,2 pada kromosom 1 akan disilangkan dengan nilai array kromosom 0 dengan 2 *point cut* secara acak.

[Kromosom 1] X [Kromosom 0].

Maka hasil dari kromosom yang baru adalah :

Kromosom 1: [Perawat 3: Senin:11.00-13.00:Simizu]

[Perawat 4:Senin:17.00-19.00: Kevin]

[Perawat 1:Kamis:09.00-11.00:Rudi]

[Perawat 1:Selasa : 17.00-19.00:Gloria]

[Perawat 2:Jumat:17.00-19.00:Mega]

5. Penentuan Solusi Baru, Iterasi, dan Konvergensi.

Pada tahap ini, nilai *array* dari suatu kromosom akan diganti dengan nilai yang lain yang nilai barunya. Pemilihan *array* dipilih secara acak. Begitu juga nilai barunya didapatkan secara acak. Pertama bangkitkan nilai persentase acak dari 0% - 100% sebagai *roulette*. *Roulette* dilakukan sebanyak jumlah populasi, yaitu 5. Penulis menggunakan probabilitas mutasi sebanyak 40%. Artinya jika nilai persentase acak < 40%, maka kromosom terjadi mutasi. Jika

suatu kromosom terjadi mutasi. Maka setelah itu bangkitkan bilangan acak dari 0 sampai jumlah *array* – 1. Maka untuk penelitian ini karena jumlah *array* sebanyak 5. Maka angka random yang dibangkitkan 0 – 4. Lalu lakukan *roulette* pada bilangan 0 – 4. Hasil dari *roulette* adalah nilai *array* yang akan berubah. Hasil dari *roulette* tiap kromosom adalah :

Kromosom 0 : hasil *roulette* = 83 %

Kromosom 1 : hasil *roulette* = 36 %

Kromosom 2 : hasil *roulette* = 64 %

Kromosom 3 : hasil *roulette* = 76 %

Kromosom 4 : hasil *roulette* = 92 %

Dari data tersebut, hanya kromosom 1 yang mengalami mutasi, sementara kromosom lain tidak terjadi mutasi.

Maka setelah itu bangkitkan bilangan acak dari 0 – 4 untuk memilih *array* yang terjadi mutasi. Hasil dari *roulette* adalah : 3.

Maka kromosom 1 *array* 3 mengalami mutasi.

Nilai dari *array* 3 didapat dari pengacakan data hari, jam, ruang yang terdapat pada *database*. Maka nilai kromosom yang baru adalah:

Kromosom 1: [Perawat 3: Senin:11.00-13.00:Simizu]

[Perawat 4:Senin:17.00-19.00: Adela]

[Perawat 1:Kamis:09.00-11.00:Rudi]

[Perawat 1:Selasa:17.00-19.00:Gloria]

[Perawat 2:Jumat:17.00-19.00:Mega]

Maka hasil kromosom yang baru adalah:

Kromosom 0: [Perawat 1: Senin:09.00-11.00:Rudi]

[Perawat 2:Rabu:17.00-19.00: Adela]

[Perawat 3:Kamis:19.00-21.00:Santi]

[Perawat 4:Selasa:09.00-11.00:Mega]

Kromosom 1: [Perawat 3: Senin:09.00-11.00:Simizu]

[Perawat 4:Senin:17.00-19.00:Adela]

[Perawat 1:Kamis:09.00-11.00:Algoritma &Pemrograman]

[Perawat 1:Selasa:17.00-19.00:Gloria]

[Perawat 2:Jumat:17.00-19.00:Mega]

Kromosom 2: [Perawat 2: Selasa:19.00-21.00:Adela]

[Perawat 1:Senin:19.00-21.00: Gloria]

[Perawat 1:Kamis:19.00-21.00:Santi]

[Perawat 3:Senin:17.00-19.00:Jaringan Komputer 1]

[Perawat 4:Rabu:09.00-11.00: Jaringan Komputer 2]

Kromosom 3: [Perawat 1: Selasa:17.00-19.00:Adela]

[Perawat 4:Senin:09.00-11.00:Gloria]

[Perawat 1:Kamis:19.00-21.00:Santi]

[Perawat 2:Jumat:09.00-11.00:Grisella]

[Perawat 3:Rabu:09.00-11.00: Rudi]

Kromosom 4: [Perawat 4: Rabu:09.00-11.00:Simizu]

[Perawat 3:Selasa:09.00-11.00:Kevin]

[Perawat 1:Jumat:17.00-19.00:Rudi]

[Perawat 2:Kamis:19.00-21.00:Grisella]

[Perawat 1:Rabu:09.00-11.00:Santi]

Dilakukanlah perhitungan kembali nilai *fitness*

- a. Kromosom 0. pinalti = 7

$$Fitness = (1/ 1+8)$$

$$= 0,125$$

- b. Kromosom 1. pinalti = 10

$$Fitness = (1/ 1+10)$$

$$= 0,091$$

- c. Kromosom 2. pinalti = 8

$$Fitness = (1/ 1+8)$$

$$= 0,111$$

- d. Kromosom 3. pinalti = 5

$$Fitness = (1/ 1+5)$$

$$= 0,33$$

- e. Kromosom 4. pinalti = 9

$$Fitness = (1/ 1+9)$$

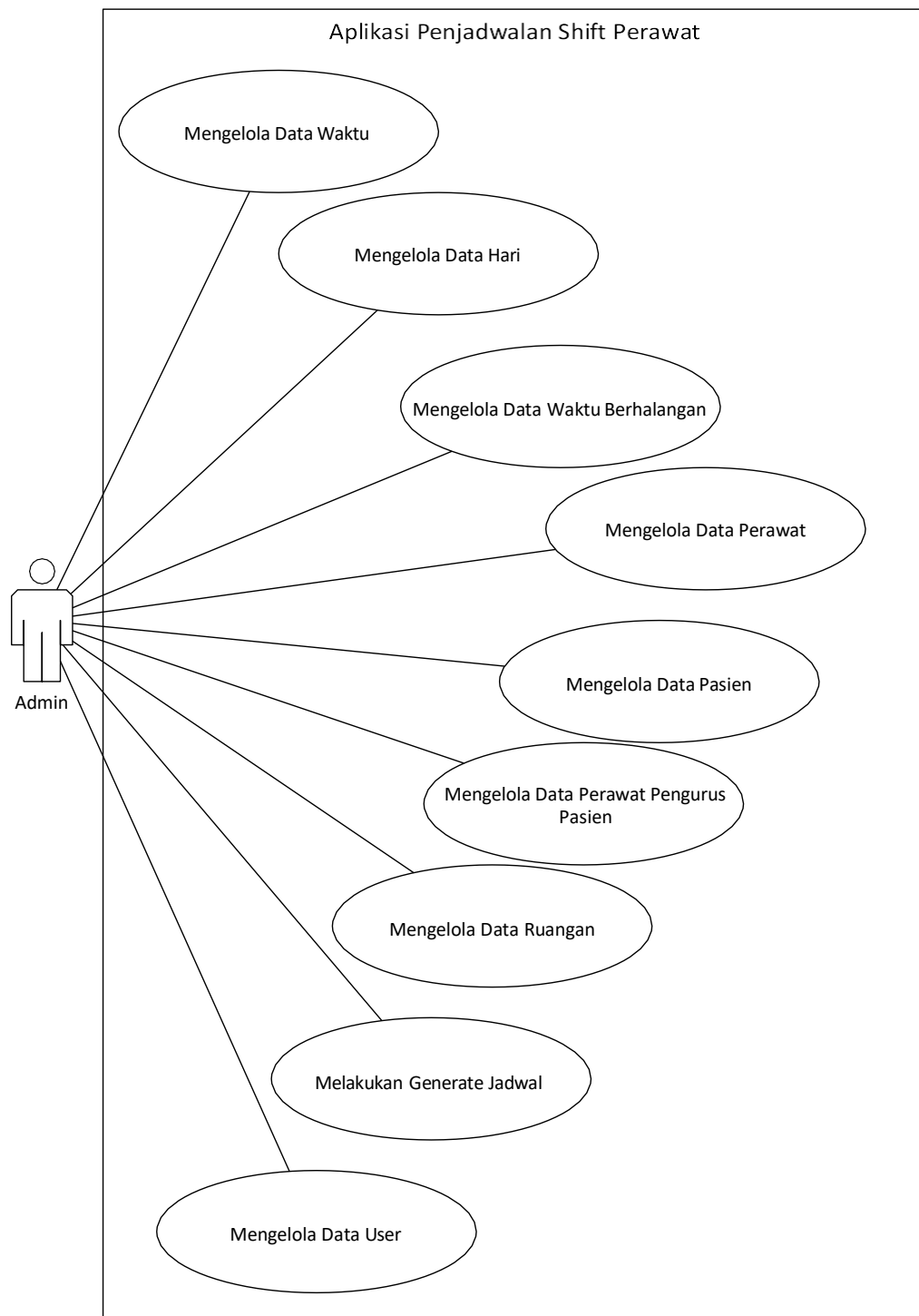
$$= 0,1$$

Maka selesailah proses untuk generasi pertama. Karena nilai *fitness* kromosom 3 memiliki nilai *fitness* tertinggi, maka kromosom 3 merupakan jadwal yang mendekati optimum. Proses tersebut akan diulang sampai mencapai generasi yang diinginkan.

3.1.3 Analisis Sistem Usulan

Analisis sistem usulan adalah proses evaluasi dan penelaahan terhadap sistem yang diusulkan sebagai solusi atas suatu permasalahan atau kebutuhan tertentu. Dalam analisis ini, berbagai aspek dari sistem yang diusulkan dievaluasi,

termasuk fungsionalitas, keandalan, keamanan, kinerja, dan kelayakan ekonomi. Pemodelan sistem usulan pada penelitian menguraikan kebutuhan fungsional pada sistem usulan yang dimodelkan dengan menggunakan *Use Case Diagram*.



Gambar 3.1 Use Case Diagram Sistem Usulan

Seluruh *Use Case* yang ada pada admin harus melalui proses *login* agar dapat berinteraksi dengan *Use Case* tersebut. Tahap berikutnya adalah membuat skenario (*flow of event*) yang menggambarkan urutan interaksi aktor dengan sistem dari *Use Case*. Berikut adalah tabel-tabel interaksi dari masing-masing *Use Case*.

Tabel 3.1 Skenario *Use Case*: Mengelola Data Waktu

Mengelola Data Klasifikasi		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data waktu yakni melihat, menambah, mengubah dan menghapus data waktu		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data waktu		Normal Flow Event
	2. Menampilkan halaman pengelolaan data waktu	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data waktu, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data waktu, dan tombol hapus untuk menghapus data waktu		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.2 Skenario *Use Case*: Mengelola Data Hari

Mengelola Data Hari		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data hari yakni melihat, menambah, mengubah dan menghapus data hari		Deskripsi
Aksi Aktor	Respon Sistem	

1. Masuk ke halaman kelola data hari		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data hari	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data hari, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data hari, dan tombol hapus untuk menghapus data hari		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.3 Skenario Use Case: Mengelola Data Waktu Berhalangan

Mengelola Data Waktu Berhalang		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data waktu berhalangan yakni melihat, menambah, mengubah dan menghapus data waktu berhalangan		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data waktu berhalangan		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data waktu berhalangan	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data waktu berhalangan, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data waktu berhalangan, dan tombol hapus untuk menghapus data waktu		

berhalangan		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.4 Skenario Use Case: Mengelola Data Perawat

Mengelola Data Perawat		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data perawat yakni melihat, menambah, mengubah dan menghapus data perawat		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data perawat		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data perawat	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data perawat, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data perawat, dan tombol hapus untuk menghapus data perawat		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.5 Skenario Use Case: Mengelola Data Pasien

Mengelola Data Pasien		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data pasien yakni melihat, menambah, mengubah dan menghapus data pasien		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data pasien		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data pasien	
3. Memilih tombol yang		

diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data pasien, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data pasien, dan tombol hapus untuk menghapus data pasien		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.6 Skenario Use Case: Mengelola Data Perawat Pengurus Pasien

Mengelola Data Perawat Pengurus Pasien		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data perawat pengurus pasien yakni melihat, menambah, mengubah dan menghapus data perawat pengurus pasien		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data perawat pengurus pasien		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data perawat pengurus pasien	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data perawat pengurus pasien, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data perawat pengurus pasien, dan tombol hapus untuk menghapus data perawat pengurus pasien		
	4. Menampilkan halaman	

	dan aksi sesuai dengan tombol yang dipilih aktor	
--	--	--

Tabel 3.7 Skenario Use Case: Mengelola Data Ruangan

Mengelola Data Ruangan		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data ruangan yakni melihat, menambah, mengubah dan menghapus data ruangan		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data ruangan		<i>Normal Flow Event</i>
	2. Menampilkan halaman pengelolaan data ruangan	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data ruangan, tombol ubah untuk menampilkan halaman <i>form</i> perubahan data ruangan, dan tombol hapus untuk menghapus data ruangan		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

Tabel 3.8 Skenario Use Case: Melakukan Generate Jadwal

Melakukan Generate Jadwal		Nama Use Case
Admin		Aktor
Melakukan <i>generate</i> jadwal dengan algoritma genetika		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman <i>generate</i> jadwal		<i>Normal Flow Event</i>
	2. Menampilkan halaman berupa <i>form</i> untuk <i>input</i> data seperti tahun perawatan, semester, nilai pembangkitan kromosom,	

	maksimal nilai generasi, nilai <i>crossover</i> , dan nilai mutasi	
3. Melakukan pengisian data dan menekan tombol <i>generate</i>		
	4. Menampilkan hasil <i>generate</i> jadwal yang diolah dengan algoritma <i>Late Acceptance Hill Climbing Hyper-Heuristic</i>	
5. Menekan tombol <i>export</i> Excel data ini untuk menampilkan laporan Excel data jadwal perawat		
	6. Mengunduh data Excel dengan informasi jadwal perawat sesuai dengan pengolahan algoritma <i>Late Acceptance Hill Climbing Hyper-Heuristic</i>	

Tabel 3.9 Skenario *Use Case*: Mengelola Data User

Mengelola Data User		Nama Use Case
Admin		Aktor
Melakukan pengelolaan data <i>user</i> yakni melihat, menambah, mengubah dan menghapus data <i>user</i>		Deskripsi
Aksi Aktor	Respon Sistem	
1. Masuk ke halaman kelola data <i>user</i>		Normal Flow Event
	2. Menampilkan halaman pengelolaan data <i>user</i>	
3. Memilih tombol yang diinginkan yaitu tombol tambah data untuk menampilkan halaman <i>form</i> penambahan data <i>user</i> , tombol ubah untuk menampilkan halaman		

<i>form</i> perubahan data <i>user</i>, dan tombol hapus untuk menghapus data <i>user</i>. Jika ingin melakukan penambahan data <i>user</i>, maka cukup lakukan pengisian <i>form</i> dan kemudian tekan tombol simpan. Jika ingin melakukan perubahan data <i>user</i>, maka cukup lakukan perubahan pada <i>form</i> yang telah terisi dan menekan tombol simpan. Penghapusan data <i>user</i> dapat dilakukan dan data yang telah dihapus tentunya tidak akan dapat dipulihkan lagi.		
	4. Menampilkan halaman dan aksi sesuai dengan tombol yang dipilih aktor	

3.2 Perancangan

Pada bagian ini, akan dilakukan perancangan *user interface* serta basis data dari sistem usulan sehingga dapat diketahui jelas sistem usulan yang akan dibangun.

3.2.1 Perancangan Tampilan

Perancangan tampilan dibuat dengan menggunakan *software* Balsamiq Mockup 3. Berikut ini adalah perancangan tampilan *interplace* aplikasi penjadwalan *shift* perawat menggunakan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* pada RSU. Wulan Windy antara lain:

1. Rancangan Tampilan *Login*

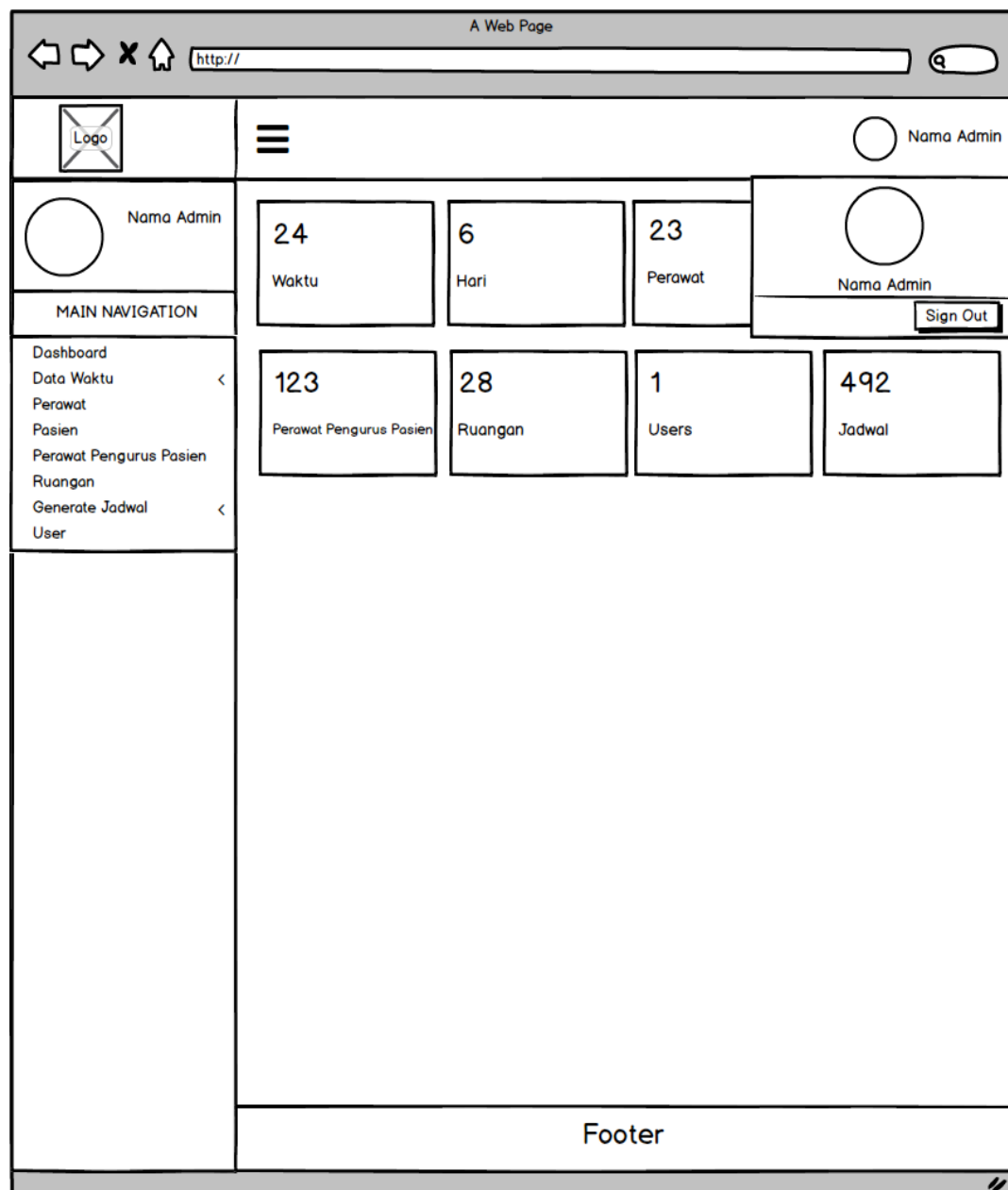
Tampilan *login* berisikan sebuah *form* yang harus diisi oleh pengguna agar dapat masuk ke dalam aplikasi dan melakukan pengelolaan data-data master agar dapat menggunakan aplikasi. Tampilan ini sebagai autentikasi awal untuk memastikan pengguna yang berwenang yang dapat masuk dan menggunakan aplikasi.

The image shows a hand-drawn wireframe of a login page. At the top, there is a browser header with the title "A Web Page" and navigation icons (back, forward, stop, home) and a search bar. The main content area is titled "Aplikasi Penjadwalan Shift Perawat". Below the title is a "Sign In" box. Inside this box, there are two input fields: "Email" and "Password". To the right of the "Password" field is a "Sign In" button.

Gambar 3.2 Rancangan Tampilan *Login*

2. Rancangan Tampilan *Dashboard*

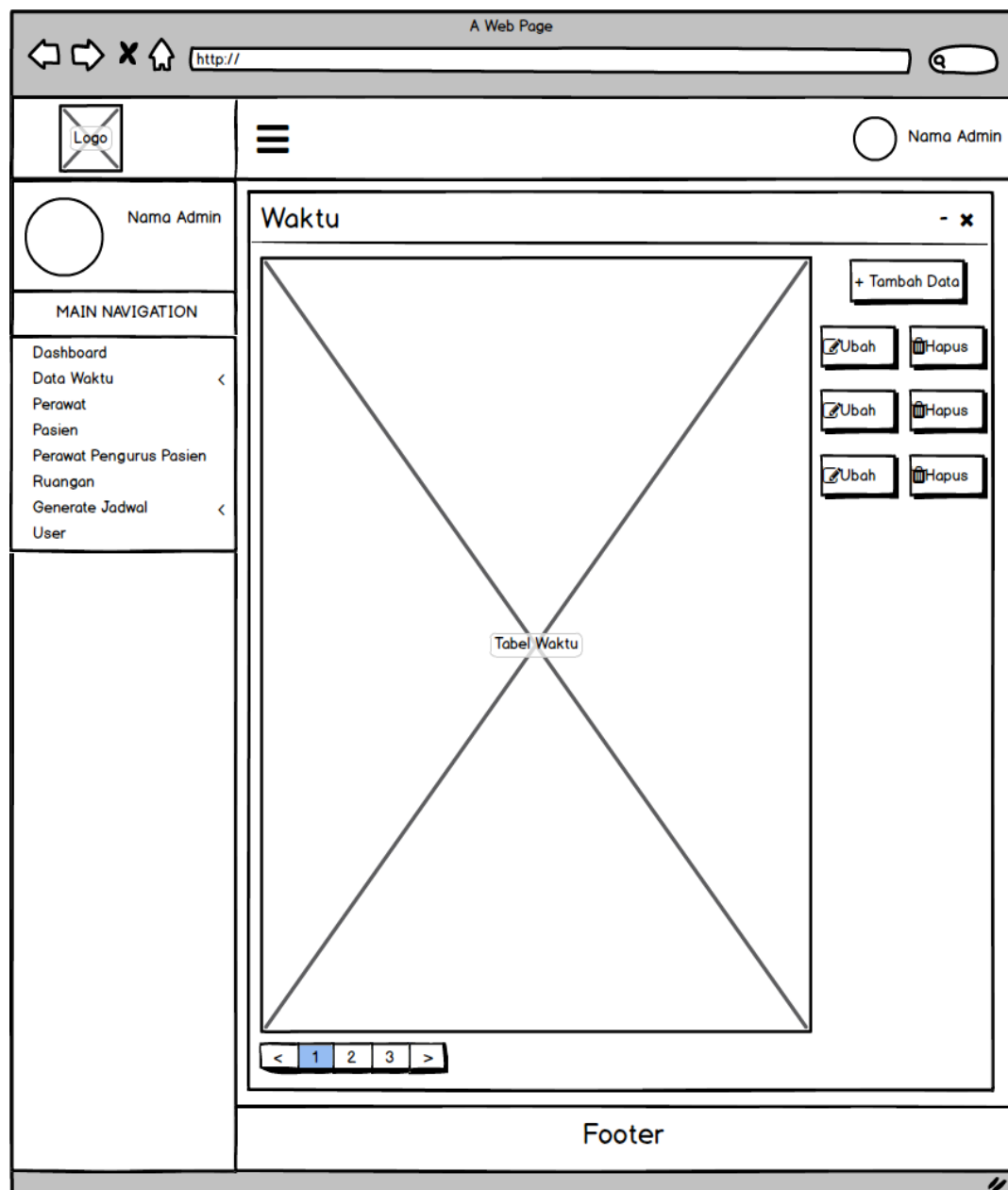
Tampilan *dashboard* berisikan informasi jumlah data-data master yang tersimpan pada aplikasi. Pada tampilan ini tepatnya di sebelah kiri juga disediakan menu-menu navigasi yang mengarahkan pengguna ke tampilan-tampilan pengelolaan data-data tersendiri. Selain itu, di sebelah sudut kanan terdapat adanya tombol *Sign Out* untuk keluar dari aplikasi.



Gambar 3.3 Rancangan Tampilan *Dashboard*

3. Rancangan Tampilan Kelola Data Waktu

Tampilan kelola data waktu merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data waktu. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data waktu.



Gambar 3.4 Rancangan Tampilan Kelola Data Waktu

4. Rancangan Tampilan Tambah Data Waktu

Tampilan tambah data waktu merupakan tampilan yang berisikan *form* untuk melakukan penambahan data waktu. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data waktu ke *database*.

The wireframe illustrates the 'Tambah Waktu' (Add Time) interface. It features a top navigation bar with a logo, a menu icon, and a user profile labeled 'Nama Admin'. Below this is a sidebar with a 'MAIN NAVIGATION' section containing links to 'Dashboard', 'Data Waktu', 'Perawat', 'Pasien', 'Perawat Pengurus Pasien', 'Ruangan', 'Generate Jadwal', and 'User'. The main content area is titled 'Tambah Waktu' and includes a form with the following elements:

- Kode Waktu:** A text input field with the placeholder 'Masukkan Kode Waktu'.
- Waktu Mulai:** A time selection input field showing '09:45'.
- Waktu Akhir:** A time selection input field showing '09:45'.
- Shift:** A dropdown menu labeled 'Pilih Shift'.
- Simpan:** A button to save the data.

The page concludes with a 'Footer' section at the bottom.

Gambar 3.5 Rancangan Tampilan Tambah Data Waktu

5. Rancangan Tampilan Ubah Data Waktu

Tampilan ubah data waktu merupakan tampilan yang berisikan *form* untuk melakukan perubahan data waktu. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data waktu di *database*.

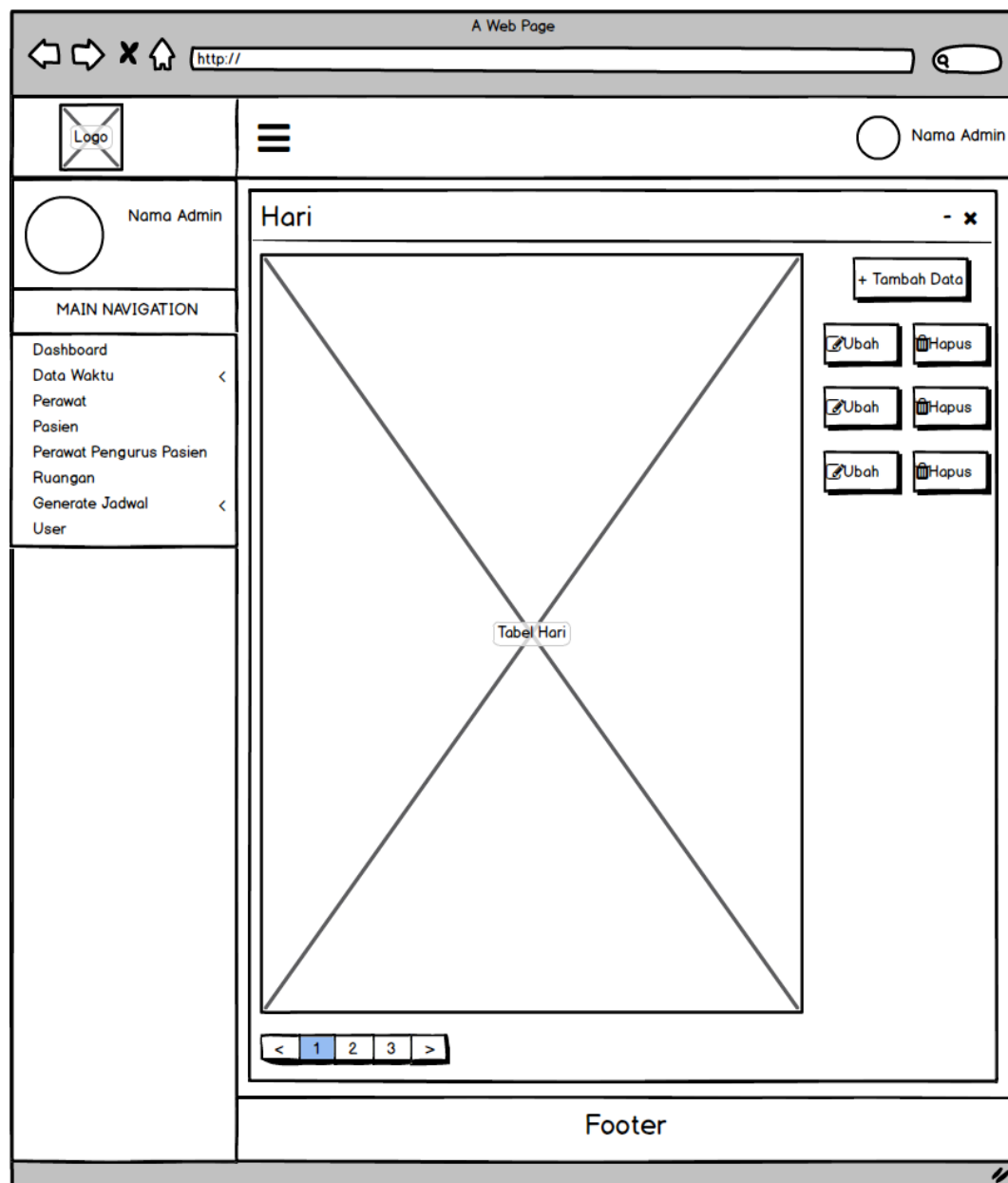
The image shows a web application interface for updating shift data. The interface is divided into several sections:

- Browser Window:** The top part of the interface shows a browser window with a title bar "A Web Page" and a URL bar containing "http://".
- Top Navigation Bar:** This bar contains a logo on the left, a hamburger menu icon in the center, and a user profile icon with the text "Nama Admin" on the right.
- Main Navigation Sidebar:** On the left side, there is a sidebar titled "MAIN NAVIGATION" with a list of links: "Dashboard", "Data Waktu", "Perawat", "Pasien", "Perawat Pengurus Pasien", "Ruangan", "Generate Jadwal", and "User".
- Main Content Area:** The central part of the interface is titled "Ubah Waktu" (Change Time). It contains a form with the following fields:
 - Kode Waktu:** A text input field containing the value "x".
 - Waktu Mulai:** A time input field containing the value "08:00".
 - Waktu Akhir:** A time input field containing the value "09:50".
 - Shift:** A dropdown menu currently showing the value "2".
 - Simpan:** A button labeled "Simpan" (Save) at the bottom of the form.
- Footer:** At the bottom of the page, there is a footer section labeled "Footer".

Gambar 3.6 Rancangan Tampilan Ubah Data Waktu

6. Rancangan Tampilan Kelola Data Hari

Tampilan kelola data hari merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data hari. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data hari.



Gambar 3.7 Rancangan Tampilan Kelola Data Hari

7. Rancangan Tampilan Tambah Data Hari

Tampilan tambah data hari merupakan tampilan yang berisikan *form* untuk melakukan penambahan data hari. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data hari ke *database*.

The wireframe illustrates the 'Tambah Hari' (Add Day) interface. It features a standard web layout with a header, a sidebar, and a main content area. The header includes a logo, a navigation menu icon, and the user's name 'Nama Admin'. The sidebar, titled 'MAIN NAVIGATION', lists various system functions. The main content area is dedicated to adding a new day, with fields for 'Kode Hari' and 'Nama', and a 'Simpan' button to save the data. The footer is also present at the bottom of the page.

Gambar 3.8 Rancangan Tampilan Tambah Data Hari

8. Rancangan Tampilan Ubah Data Hari

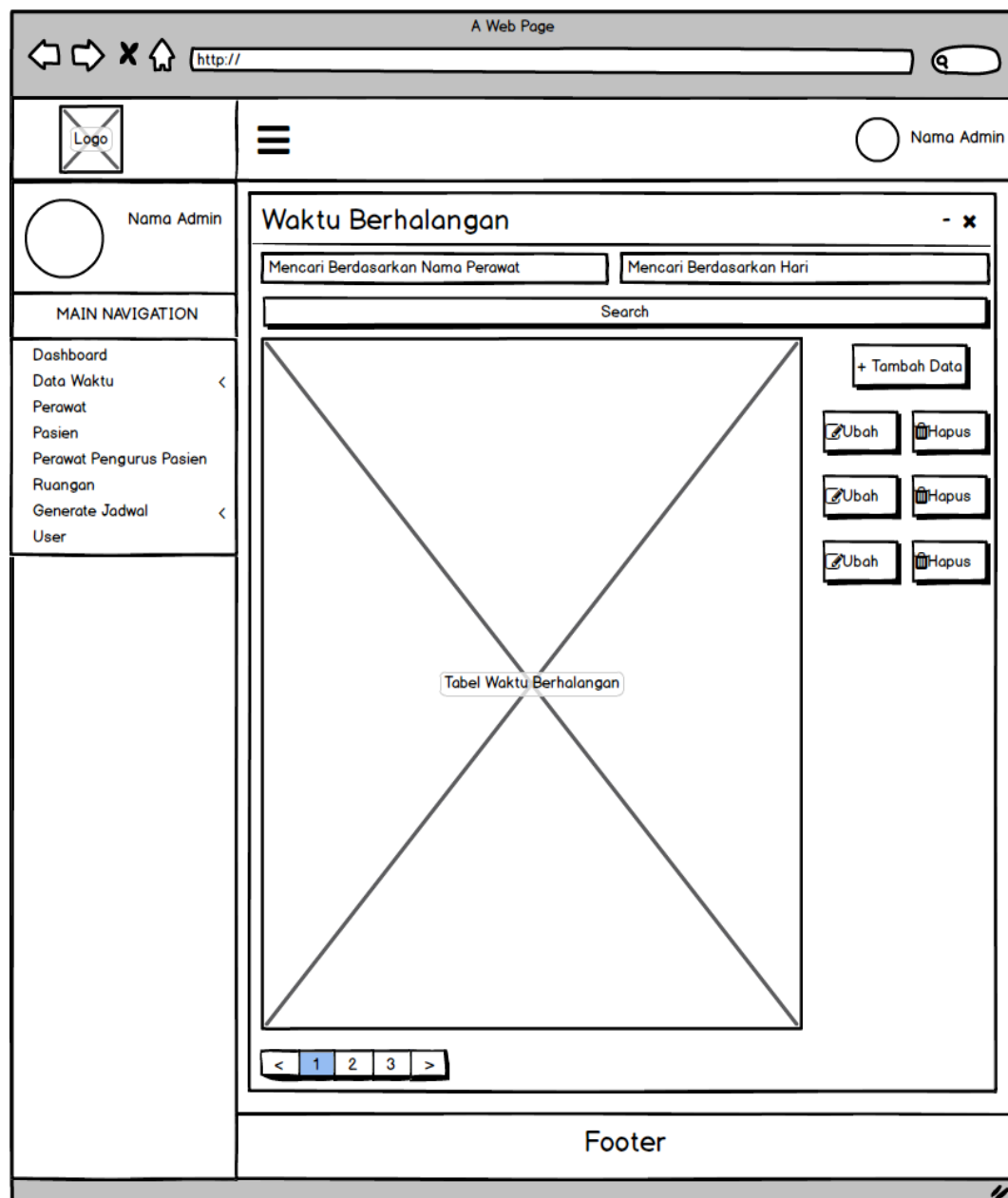
Tampilan ubah data hari merupakan tampilan yang berisikan *form* untuk melakukan perubahan data hari. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data hari di *database*.

The wireframe illustrates the 'Ubah Hari' (Change Day) interface. It is structured with a top header, a left sidebar, a main content area, and a footer. The top header includes a logo, a menu icon, and the user name 'Nama Admin'. The sidebar lists navigation options under 'MAIN NAVIGATION'. The main content area is titled 'Ubah Hari' and contains two input fields: 'Kode Hari' (containing '1') and 'Nama' (containing 'Senin'). A 'Simpan' (Save) button is positioned below the 'Nama' field. The footer section is labeled 'Footer'.

Gambar 3.9 Rancangan Tampilan Ubah Data Hari

9. Rancangan Tampilan Kelola Data Waktu Berhalangan

Tampilan kelola data waktu berhalangan merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data waktu berhalangan. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data waktu berhalangan.



Gambar 3.10 Rancangan Tampilan Kelola Data Waktu Berhalangan

10. Rancangan Tampilan Tambah Data Waktu Berhalangan

Tampilan tambah data waktu berhalangan merupakan tampilan yang berisikan *form* untuk melakukan penambahan data waktu berhalangan. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data waktu berhalangan ke *database*.

The wireframe illustrates the 'Tambah Waktu Berhalangan' (Add Absence Time) interface. It features a standard web layout with a header, a sidebar, a main content area, and a footer. The header includes a logo, a navigation menu icon, and the user's name 'Nama Admin'. The sidebar, titled 'MAIN NAVIGATION', lists various system functions. The main content area is dedicated to adding absence data, with fields for selecting a nurse ('Perawat'), day ('Hari'), and time ('Waktu'), and a 'Simpan' (Save) button. The footer is a simple horizontal bar at the bottom.

Gambar 3.11 Rancangan Tampilan Tambah Data Waktu Berhalangan

11. Rancangan Tampilan Ubah Data Waktu Berhalangan

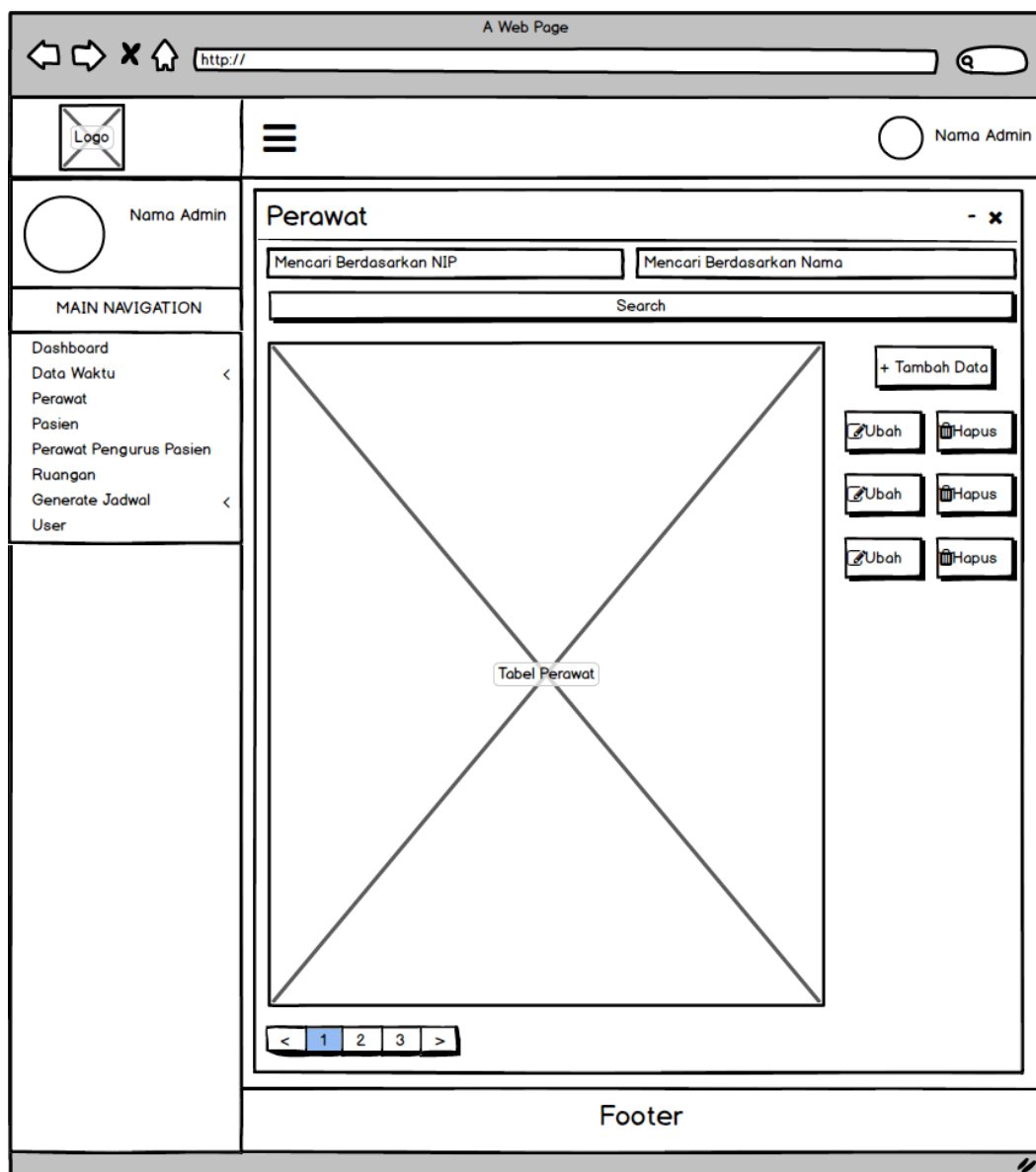
Tampilan ubah data waktu berhalangan merupakan tampilan yang berisikan *form* untuk melakukan perubahan data waktu berhalangan. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data waktu berhalangan di *database*.

The wireframe illustrates the 'Ubah Waktu Berhalangan' (Change Shift Time) interface. It features a standard web browser window at the top. The header area includes a logo placeholder, a navigation menu icon, and the user's name 'Nama Admin'. The left sidebar, titled 'MAIN NAVIGATION', lists various system functions. The central content area is dedicated to the shift change form, which includes three dropdown menus for selecting the nurse ('Perawat'), day ('Hari'), and time ('Waktu'), each currently showing 'xxx'. A 'Simpan' (Save) button is positioned below the dropdowns. The footer of the page is labeled 'Footer'.

Gambar 3.12 Rancangan Tampilan Ubah Data Waktu Berhalangan

12. Rancangan Tampilan Kelola Data Perawat

Tampilan kelola data perawat merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data perawat. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data perawat. Selain itu terdapat adanya *form* isian pencarian nama



Gambar 3.13 Rancangan Tampilan Kelola Data Perawat

13. Rancangan Tampilan Tambah Data Perawat

Tampilan tambah data perawat merupakan tampilan yang berisikan *form* untuk melakukan penambahan data perawat. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data perawat ke *database*.

The wireframe illustrates the 'Tambah Perawat' (Add Nurse) form within a web application. The layout is as follows:

- Browser Window:** The top section shows a browser interface with navigation icons, a search bar, and the address 'http://'. The page title is 'A Web Page'.
- Header:** Contains a 'Logo' placeholder, a hamburger menu icon, and a user profile section with a circular icon and the text 'Nama Admin'.
- MAIN NAVIGATION:** A sidebar menu on the left lists the following options: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The 'Perawat' option is currently selected.
- Tambah Perawat Form:** The main content area features a form titled 'Tambah Perawat' with a close button. The form includes:
 - Kode Perawat:** A text input field with the placeholder 'Masukkan Kode Perawat'.
 - NIP:** A text input field with the placeholder 'Masukkan NIP'.
 - Nama:** A text input field with the placeholder 'Masukkan Nama'.
 - Email:** A text input field with the placeholder 'Masukkan Email'.
 - Simpan:** A button to save the entered data.
- Footer:** A horizontal bar at the bottom of the page labeled 'Footer'.

Gambar 3.14 Rancangan Tampilan Tambah Data Perawat

14. Rancangan Tampilan Ubah Data Perawat

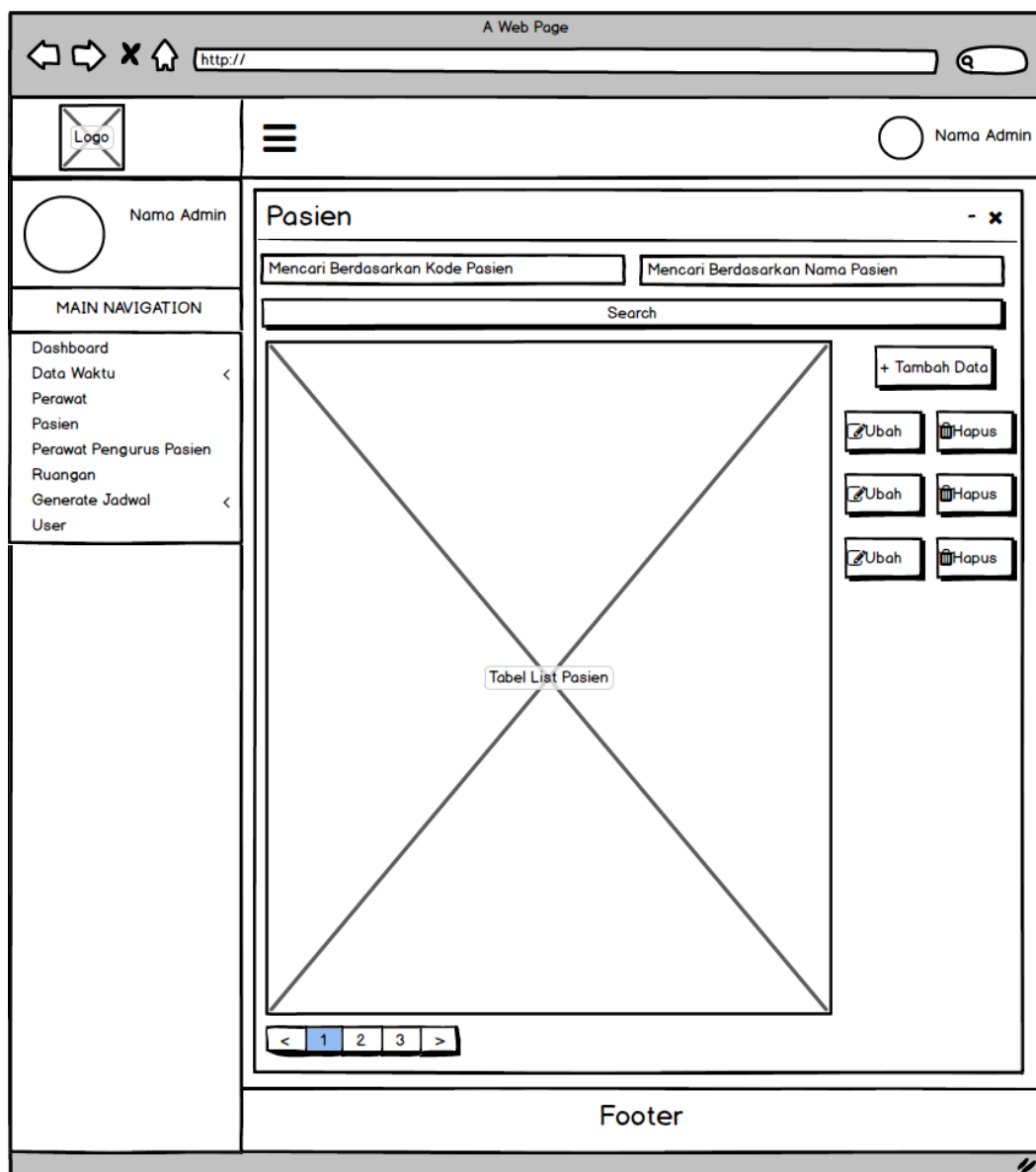
Tampilan ubah data perawat merupakan tampilan yang berisikan *form* untuk melakukan perubahan data perawat. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data perawat di *database*.

The wireframe illustrates the 'Ubah Perawat' (Edit Nurse) interface. It features a top navigation bar with a logo, a hamburger menu icon, and a user profile section labeled 'Nama Admin'. Below the navigation bar, the main content area is divided into a left sidebar labeled 'MAIN NAVIGATION' and a central form area. The sidebar lists various system functions: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The central form area is titled 'Ubah Perawat' and contains four input fields: 'Kode Perawat', 'NIP', 'Nama', and 'Email', each with placeholder text 'xxxxx'. A 'Simpan' (Save) button is positioned at the bottom of the form. The footer of the page is labeled 'Footer'.

Gambar 3.15 Rancangan Tampilan Ubah Data Perawat

15. Rancangan Tampilan Kelola Data Pasien

Tampilan kelola data pasien merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data pasien. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data pasien. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 3.16 Rancangan Tampilan Kelola Data Pasien

16. Rancangan Tampilan Tambah Data Pasien

Tampilan tambah data pasien merupakan tampilan yang berisikan *form* untuk melakukan penambahan data pasien. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data pasien ke *database*.

The wireframe illustrates the 'Tambah Pasien' (Add Patient) web form. It is presented within a browser window titled 'A Web Page'. The interface features a header with a 'Logo' placeholder and a user profile section labeled 'Nama Admin'. A sidebar on the left, titled 'MAIN NAVIGATION', lists various system functions: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The main content area contains a form titled 'Tambah Pasien' with the following fields:

- Kode Pasien:** A text input field with the placeholder 'Masukkan Kode Pasien'.
- Nama Pasien:** A text input field with the placeholder 'Masukkan Nama Pasien'.
- Jumlah Rawat Inap:** A dropdown menu with the placeholder 'Pilih Jumlah Rawat Inap'.
- Tingkatan:** A dropdown menu with the placeholder 'Pilih Tingkatan'.
- Jenis:** A dropdown menu with the placeholder 'Pilih Jenis'.

A 'Simpan' (Save) button is located at the bottom of the form.

Gambar 3.17 Rancangan Tampilan Tambah Data Pasien

17. Rancangan Tampilan Ubah Data Pasien

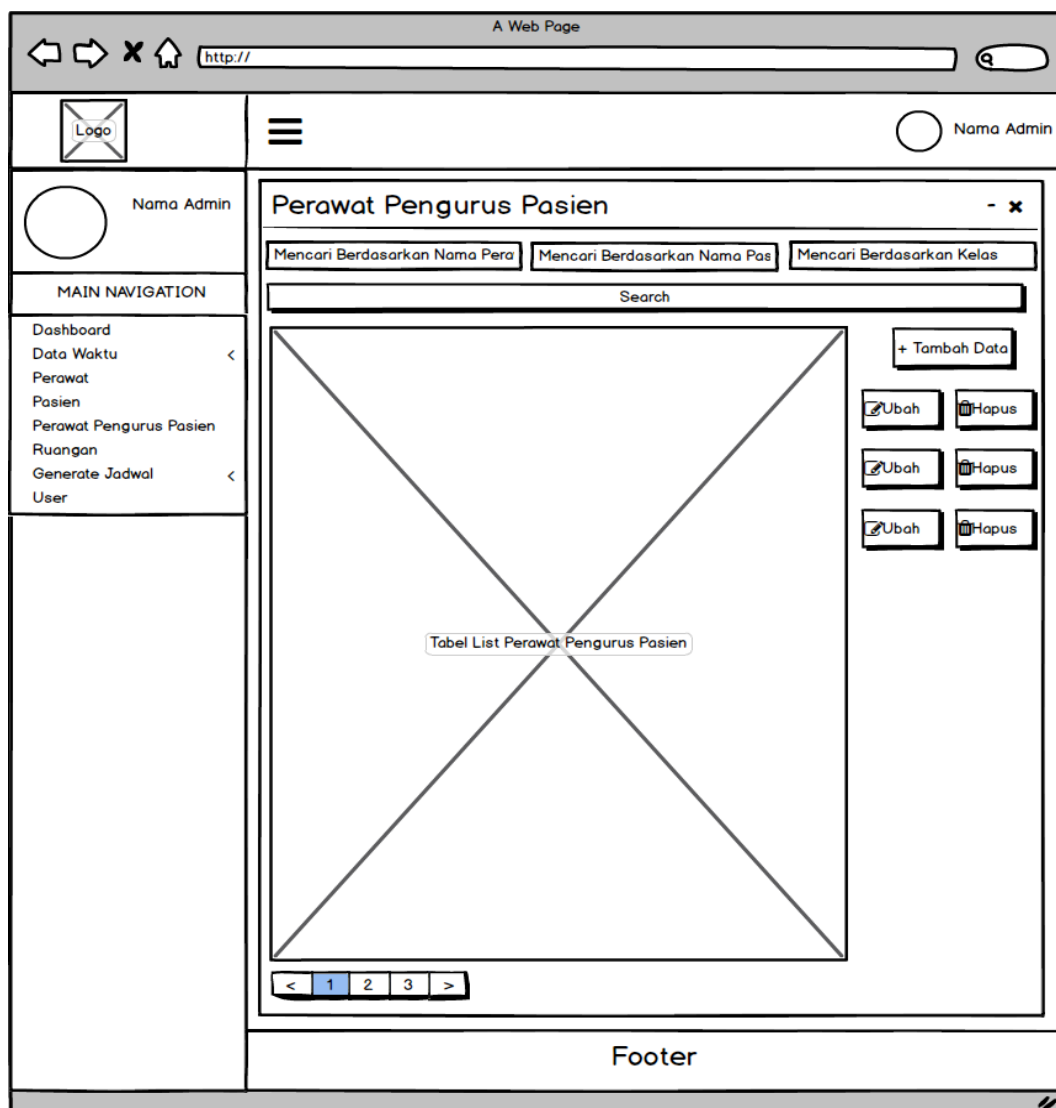
Tampilan ubah data pasien merupakan tampilan yang berisikan *form* untuk melakukan perubahan data pasien. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data pasien di *database*.

The diagram illustrates a web application interface for updating patient data. It features a browser window at the top with a URL bar. Below the browser is a header section containing a logo, a hamburger menu, and a user profile section labeled "Nama Admin". A left sidebar contains a "MAIN NAVIGATION" menu with items: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The main content area is titled "Ubah Pasien" and contains five input fields: "Kode Pasien" (text), "Nama Pasien" (text), "Jumlah Rawat Inap" (number), "Tingkatan" (dropdown), and "Jenis" (dropdown). A "Simpan" button is located at the bottom of the form. The footer area is labeled "Footer".

Gambar 3.18 Rancangan Tampilan Ubah Data Pasien

18. Rancangan Tampilan Kelola Data Perawat Pengurus Pasien

Tampilan kelola data perawat pengurus pasien merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data perawat pengurus pasien. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data perawat pengurus pasien. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 3.19 Rancangan Tampilan Kelola Data Perawat Pengurus Pasien

19. Rancangan Tampilan Tambah Data Perawat Pengurus Pasien

Tampilan tambah data perawat pengurus pasien merupakan tampilan yang berisikan *form* untuk melakukan penambahan data perawat pengurus pasien. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data perawat pengurus pasien ke *database*.

The wireframe illustrates the 'Tambah Perawat Pengurus Pasien' interface. It features a top navigation bar with a logo, a menu icon, and a user profile labeled 'Nama Admin'. A vertical sidebar on the left provides 'MAIN NAVIGATION' with links to various system components, with 'Perawat Pengurus Pasien' currently selected. The main workspace contains a form titled 'Tambah Perawat Pengurus Pasien' with the following elements:

- Perawat:** A dropdown menu with the placeholder 'Pilih Perawat'.
- Pasien:** A dropdown menu with the placeholder 'Pilih Pasien'.
- Kelas:** A text input field with the placeholder 'Masukkan Kelas'.
- Tahun Perawatan:** A text input field with the placeholder 'Masukkan Tahun Perawatan'.
- Simpan:** A button to submit the form data.

The bottom of the page includes a 'Footer' section.

Gambar 3.20 Rancangan Tampilan Tambah Data Perawat Pengurus Pasien

20. Rancangan Tampilan Ubah Data Perawat Pengurus Pasien

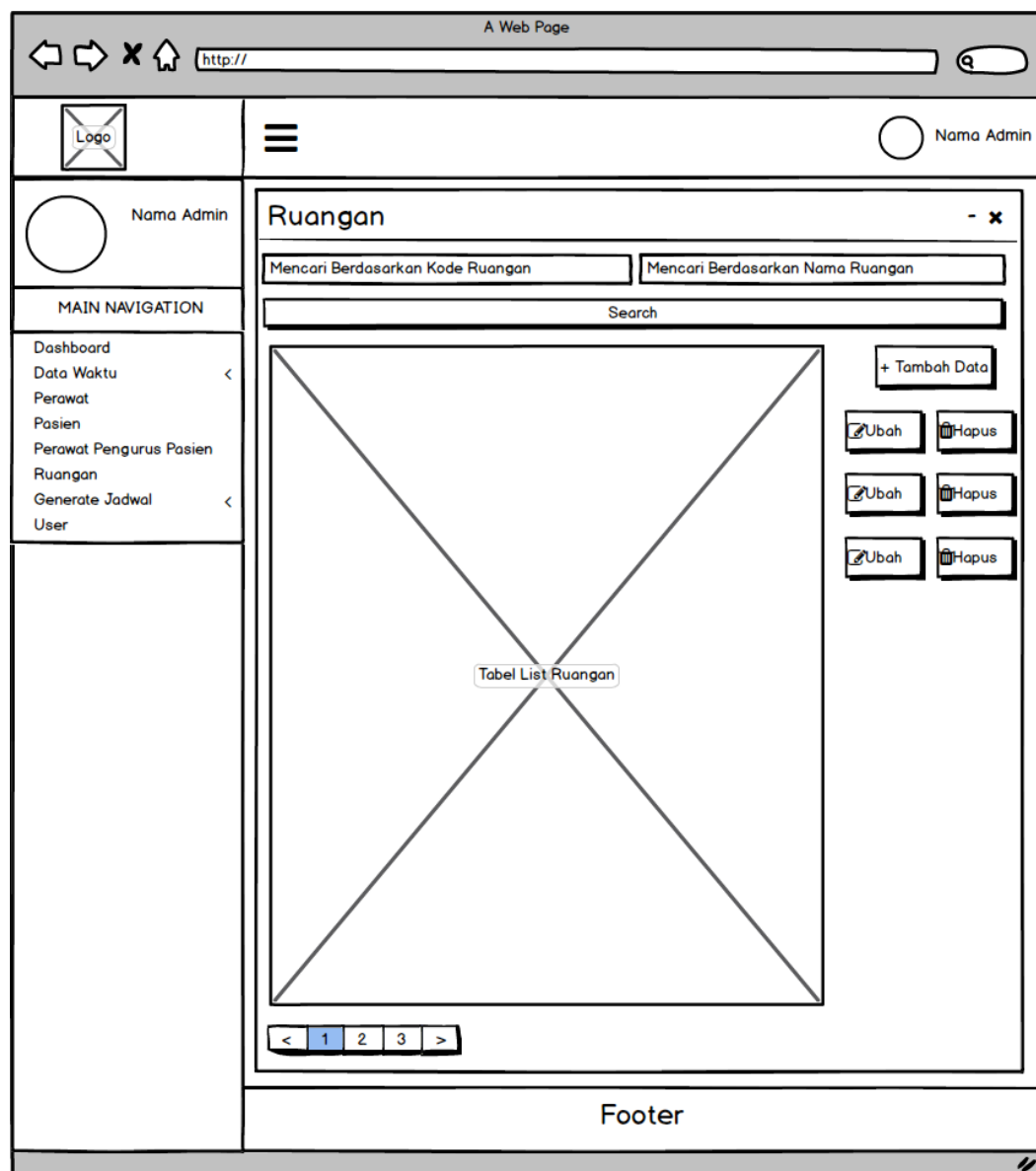
Tampilan ubah data perawat pengurus pasien merupakan tampilan yang berisikan *form* untuk melakukan perubahan data perawat pengurus pasien. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data perawat pengurus pasien di *database*.

The screenshot shows a web application interface for updating patient nurse data. The interface is displayed within a browser window titled "A Web Page". The browser's address bar shows "http://". The application has a header bar with a "Logo" icon on the left and a user profile icon labeled "Nama Admin" on the right. Below the header is a sidebar with a "MAIN NAVIGATION" menu. The menu items are: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The main content area is titled "Ubah Perawat Pengurus Pasien" and contains a form with the following fields: "Perawat" (dropdown menu with "xxx" selected), "Pasien" (dropdown menu with "xxx" selected), "Kelas" (text input field with "SC004" entered), and "Tahun Perawatan" (text input field with "xxxx" entered). A "Simpan" button is located at the bottom of the form. The footer area is labeled "Footer".

Gambar 3.21 Rancangan Tampilan Ubah Data Perawat Pengurus Pasien

21. Rancangan Tampilan Kelola Data Ruangan

Tampilan kelola data ruangan merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data ruangan. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data ruangan. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 3.22 Rancangan Tampilan Kelola Data Ruangan

22. Rancangan Tampilan Tambah Data Ruangan

Tampilan tambah data ruangan merupakan tampilan yang berisikan *form* untuk melakukan penambahan data ruangan. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data ruangan ke *database*.

The wireframe illustrates the 'Tambah Ruangan' (Add Room) interface. It features a standard web layout with a header, a sidebar, a main content area, and a footer. The header includes a logo, a navigation menu icon, and the user's name 'Nama Admin'. The sidebar, titled 'MAIN NAVIGATION', lists various system functions. The main content area is dedicated to adding a new room, with fields for 'Kode Ruangan', 'Nama Ruangan', 'Kapasitas', and 'Jenis'. A 'Simpan' button is provided to save the data. The footer area is labeled 'Footer'.

Gambar 3.23 Rancangan Tampilan Tambah Data Ruangan

23. Rancangan Tampilan Ubah Data Ruangan

Tampilan ubah data ruangan merupakan tampilan yang berisikan *form* untuk melakukan perubahan data ruangan. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data ruangan di *database*.

The wireframe illustrates the 'Ubah Ruangan' (Change Room) interface. It features a top header with a logo, a navigation menu icon, and a user profile labeled 'Nama Admin'. A left sidebar provides 'MAIN NAVIGATION' with links to Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The main content area is titled 'Ubah Ruangan' and contains a form with the following fields: 'Kode Ruangan' (xxx), 'Nama Ruangan' (xxx), 'Kapasitas' (xxx), and 'Jenis' (Umum). A 'Simpan' button is located below the form. The footer area is labeled 'Footer'.

Gambar 3.24 Rancangan Tampilan Ubah Data Ruangan

24. Rancangan Tampilan *Form Generate Jadwal*

Tampilan *form generate* jadwal merupakan tampilan yang berisikan *form* untuk melakukan *generate* jadwal *shift* perawat. Pada tampilan ini terdapat beberapa isian *form* yang harus diisi seperti tahun perawatan, semester, nilai pembangkitan kromosom, maksimal nilai generasi, nilai *crossover*, dan nilai mutasi. Setelah pengaturan dilakukan, maka tekan tombol *generate* untuk melakukan proses *generate* jadwal dengan algoritma LAHC.

The wireframe illustrates the 'Generate Algorithm' form within a web application. The form is titled 'Generate Algorithm' and contains the following elements:

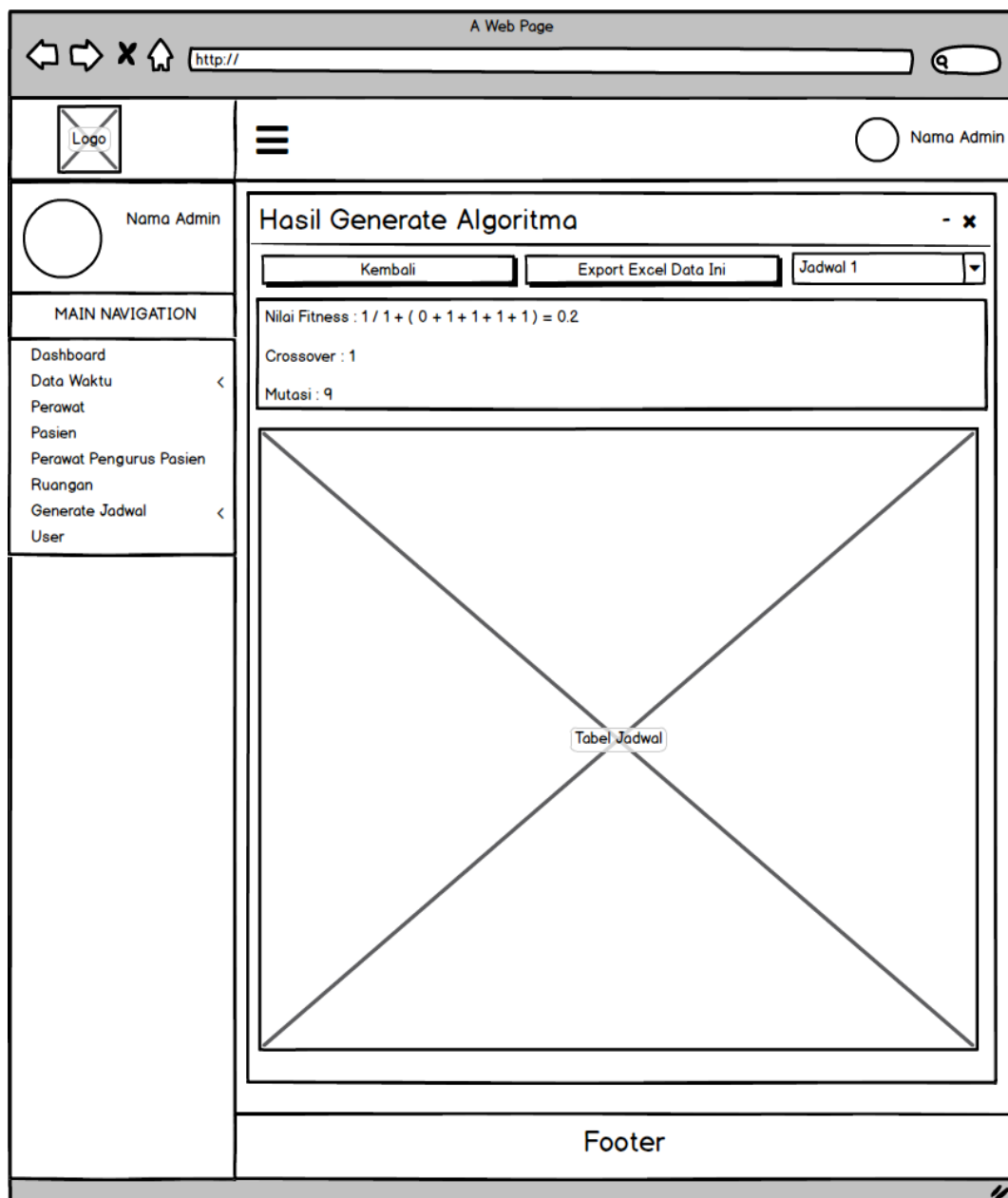
- Form Fields:**
 - Pilih Tahun Perawatan
 - Pilih Tingkatan
 - Masukkan Nilai Pembangkitan Kromosom
 - Masukkan Maksimal Nilai Generasi
 - Masukkan Nilai Crossover
 - Masukkan Nilai Mutasi
- Generate Button:** A button labeled 'Generate' is positioned at the bottom of the form.

The surrounding interface includes a sidebar with 'MAIN NAVIGATION' items: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top navigation bar features a logo, a hamburger menu, and a user profile area labeled 'Nama Admin'. The bottom of the page has a 'Footer' section.

Gambar 3.25 Rancangan Tampilan *Form Generate Jadwal*

25. Rancangan Tampilan Hasil *Generate* Optimasi Jadwal

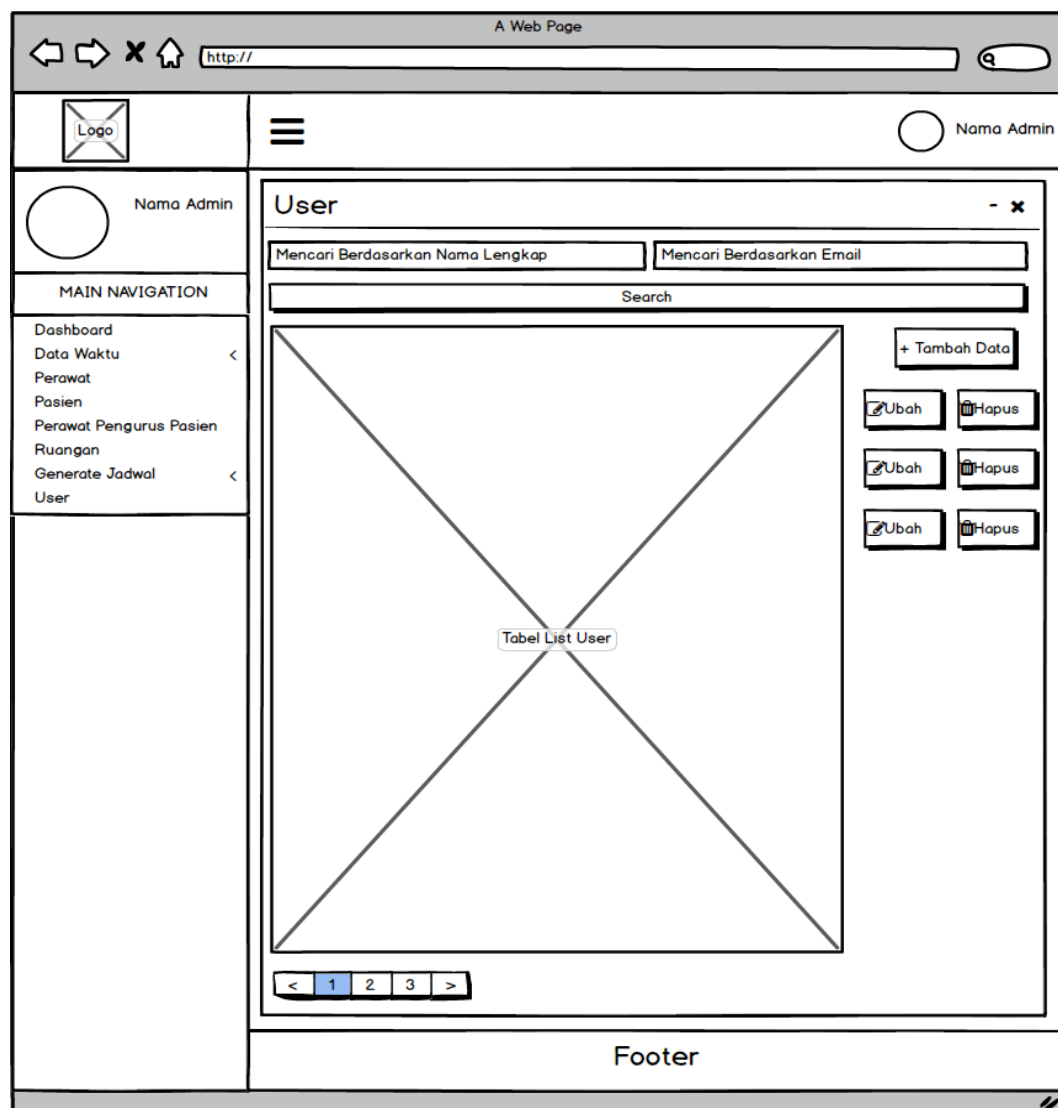
Tampilan hasil *generate* optimasi jadwal merupakan tampilan yang berisikan informasi hasil penjadwalan *shift* perawat yang di-*generate* berdasarkan algoritma LAHC. Pada tampilan ini juga terdapat beberapa tombol seperti tombol kembali, tombol *export* Excel data ini untuk mengunduh hasil *generate* jadwal dalam bentuk Excel dan pilihan jadwal-jadwal lainnya.



Gambar 3.26 Rancangan Tampilan Hasil *Generate* Optimasi Jadwal

26. Rancangan Tampilan Kelola Data *User Admin*

Tampilan kelola data *user admin* merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data *user admin*. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data *user admin*. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 3.27 Rancangan Tampilan Kelola Data *User Admin*

27. Rancangan Tampilan Tambah Data *User Admin*

Tampilan tambah data *user admin* merupakan tampilan yang berisikan *form* untuk melakukan penambahan data *user admin*. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data *user admin* ke *database*.

The wireframe illustrates the 'Tambah User Admin' interface. It features a standard web layout with a header, a sidebar, a main content area, and a footer. The header includes a logo, a navigation menu icon, and a user profile placeholder labeled 'Nama Admin'. The sidebar, titled 'MAIN NAVIGATION', lists various system functions. The main content area is dedicated to adding a new user, with fields for full name, email, password, and password confirmation, followed by a 'Simpan' (Save) button. The footer is a simple horizontal bar at the bottom.

Gambar 3.28 Rancangan Tampilan Tambah Data *User Admin*

28. Rancangan Tampilan Ubah Data *User Admin*

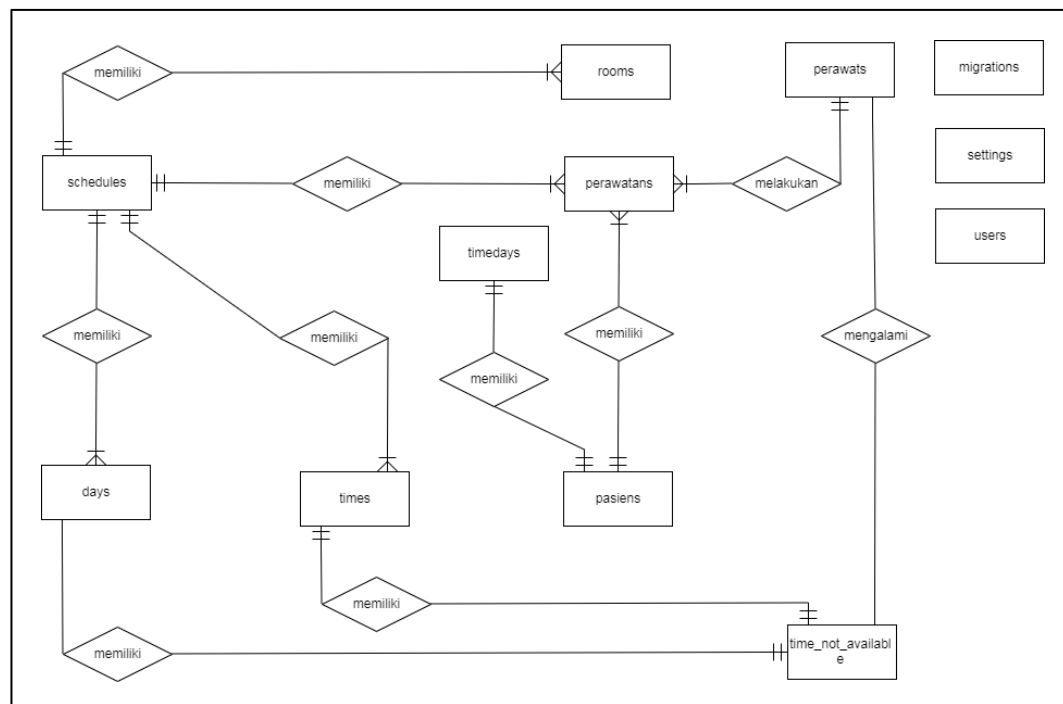
Tampilan ubah data *user admin* merupakan tampilan yang berisikan *form* untuk melakukan perubahan data *user admin*. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data *user admin* di *database*.

The wireframe illustrates the 'Ubah User Admin' interface. It features a top navigation bar with a logo, a menu icon, and a user profile placeholder labeled 'Nama Admin'. A left sidebar provides 'MAIN NAVIGATION' with links to various system components. The central area contains a form titled 'Ubah User Admin' with fields for 'Nama Lengkap', 'Email', 'Password', and 'Password Konfirmasi', each with a placeholder value or instruction. A 'Simpan' button is positioned below the form. The page concludes with a 'Footer' section at the bottom.

Gambar 3.29 Rancangan Tampilan Ubah Data *User Admin*

3.2.2 Perancangan Basis Data

Perancangan basis data merupakan tahap krusial dalam siklus pengembangan perangkat lunak yang memastikan bahwa struktur basis data mendukung kebutuhan aplikasi secara optimal. Perancangan basis data pada penelitian ini menggunakan *Entity Relationship Diagram* (ERD) seperti yang ditunjukkan pada Gambar 3.30.



Gambar 3.30 ERD Sistem Usulan

Setelah ERD dirancang maka berdasarkan *entity* dan *attribute* akan dibuat struktur tabel *basis data* antara lain:

Tabel 3.10 Struktur Tabel Pasiens

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
code_pasiens	varchar(255)	Tidak		
name	varchar(255)	Tidak		
jumlah_rawat_inap	varchar(255)	Tidak		

agama	varchar(255)	Tidak		
type	varchar(255)	Tidak		
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.11 Struktur Tabel Days

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
code_days	varchar(255)	Tidak		
name_day	varchar(255)	Tidak		
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.12 Struktur Tabel Perawats

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
code_perawats	varchar(255)	Tidak		
nip	varchar(255)	Tidak		
name	varchar(255)	Tidak		
email	varchar(255)	Tidak		
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.13 Struktur Tabel Migrations

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
migration	varchar(255)	Tidak		
batch	int(11)	Tidak		

Tabel 3.14 Struktur Tabel Rooms

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		

code_rooms	varchar(255)	Tidak		
name	varchar(255)	Tidak		
capacity	varchar(255)	Tidak		
type	varchar(255)	Tidak		
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.15 Struktur Tabel Schedules

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
perawatans_id (<i>Kunci Foreign</i>)	int(10)	Tidak		teachs -> id
days_id (<i>Kunci Foreign</i>)	int(10)	Tidak		days -> id
times_id (<i>Kunci Foreign</i>)	int(10)	Tidak		times -> id
rooms_id (<i>Kunci Foreign</i>)	int(10)	Tidak		rooms -> id
type	int(10)	Ya	<i>NULL</i>	
value	double(8,2)	Ya	<i>NULL</i>	
value_process	varchar(255)	Ya	<i>NULL</i>	
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.16 Struktur Tabel Settings

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
key	varchar(255)	Tidak		
name	varchar(255)	Ya	<i>NULL</i>	
value	varchar(255)	Ya	<i>NULL</i>	
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.17 Struktur Tabel Perawatans

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
pasiens_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	courses -> id
perawats_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	lecturers -> id
class_room	varchar(255)	Tidak		
year	varchar(255)	Tidak		
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.18 Struktur Tabel Timedays

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
code_timedays	varchar(255)	Tidak		
days_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	days -> id
times_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	times -> id
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.19 Struktur Tabel Times

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
code_times	varchar(255)	Tidak		
time_begin	varchar(255)	Tidak		
time_finish	varchar(255)	Tidak		
range	varchar(255)	Tidak		
shift	int(11)	Ya	<i>NULL</i>	
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.20 Struktur Tabel Time_not_available

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
perawatas_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	lecturers -> id
days_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	days -> id
times_id (<i>Kunci Foreign</i>)	int(10)	Ya	<i>NULL</i>	times -> id
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

Tabel 3.21 Struktur Tabel Users

Kolom	Jenis	Kosong	Bawaan	Tautan ke
id (<i>Kunci Utama</i>)	int(10)	Tidak		
name	varchar(255)	Tidak		
email	varchar(255)	Tidak		
password	varchar(255)	Tidak		
remember_token	varchar(100)	Ya	<i>NULL</i>	
created_at	timestamp	Ya	<i>NULL</i>	
updated_at	timestamp	Ya	<i>NULL</i>	

BAB IV

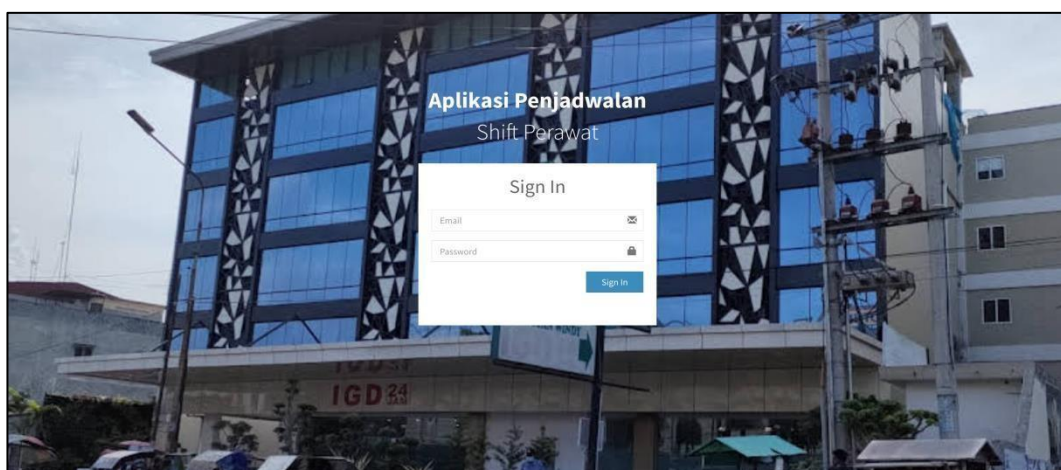
HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil dari penelitian ini berupa dibangunnya sebuah sistem optimasi penjadwalan *shift* perawat menggunakan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* pada RSU. Wulan Windy. Berikut ini adalah perancangan tampilan *interplace* aplikasi optimasi penjadwalan *shift* perawat yang telah selesai dibangun antara lain:

1. Tampilan *Login*

Tampilan pada gambar 4.1 berisikan sebuah *form* yang harus diisi oleh pengguna agar dapat masuk ke dalam aplikasi dan melakukan pengelolaan data-data master agar dapat menggunakan aplikasi. Tampilan ini sebagai autentikasi awal untuk memastikan pengguna yang berwenang yang dapat masuk dan menggunakan aplikasi.



Gambar 4.1 Tampilan *Login*

2. Tampilan *Dashboard*

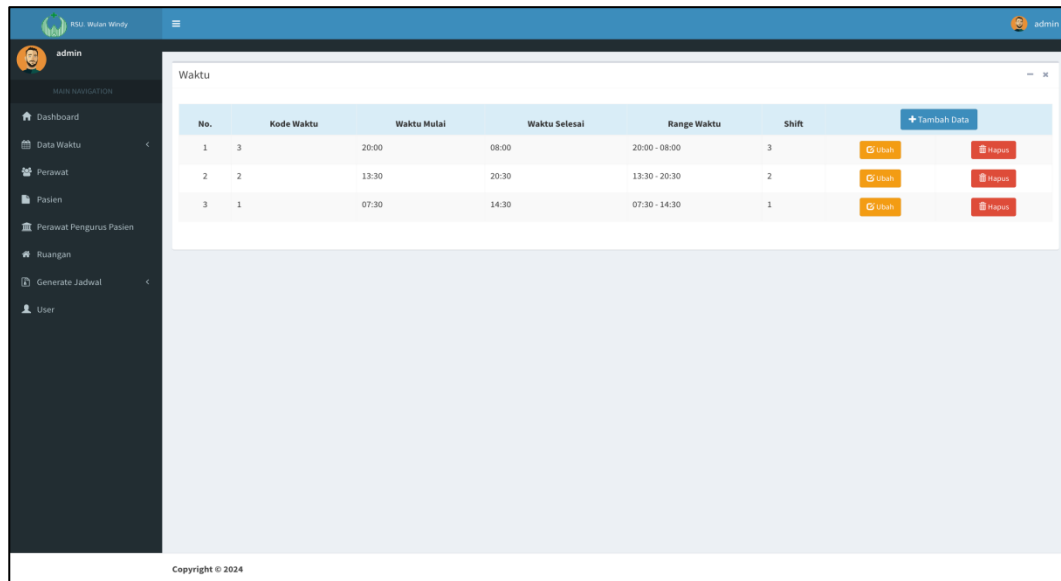
Tampilan *dashboard* pada gambar 4.2 berisikan informasi jumlah data-data master yang tersimpan pada aplikasi. Pada tampilan ini tepatnya di sebelah kiri juga disediakan menu-menu navigasi yang mengarahkan pengguna ke tampilan-tampilan pengelolaan data-data tersendiri. Selain itu, di sebelah sudut kanan terdapat adanya tombol *Sign Out* untuk keluar dari aplikasi.



Gambar 4.2 Tampilan *Dashboard*

3. Tampilan Kelola Data Waktu

Tampilan pada gambar 4.3 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data waktu. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data waktu.

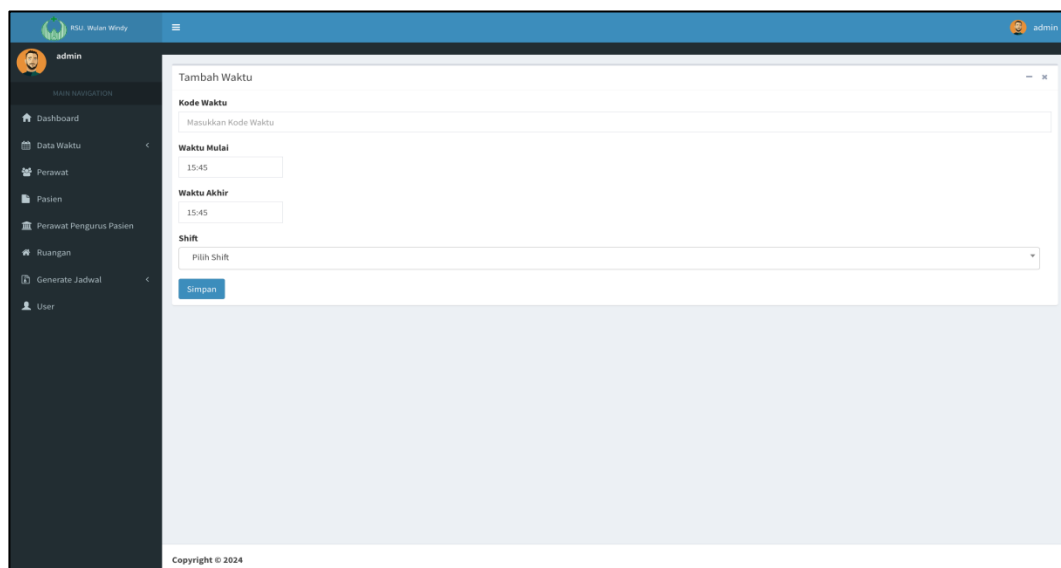


No.	Kode Waktu	Waktu Mulai	Waktu Selesai	Range Waktu	Shift	
1	3	20:00	08:00	20:00 - 08:00	3	Ubah Hapus
2	2	13:30	20:30	13:30 - 20:30	2	Ubah Hapus
3	1	07:30	14:30	07:30 - 14:30	1	Ubah Hapus

Gambar 4.3 Tampilan Kelola Data Waktu

4. Tampilan Tambah Data Waktu

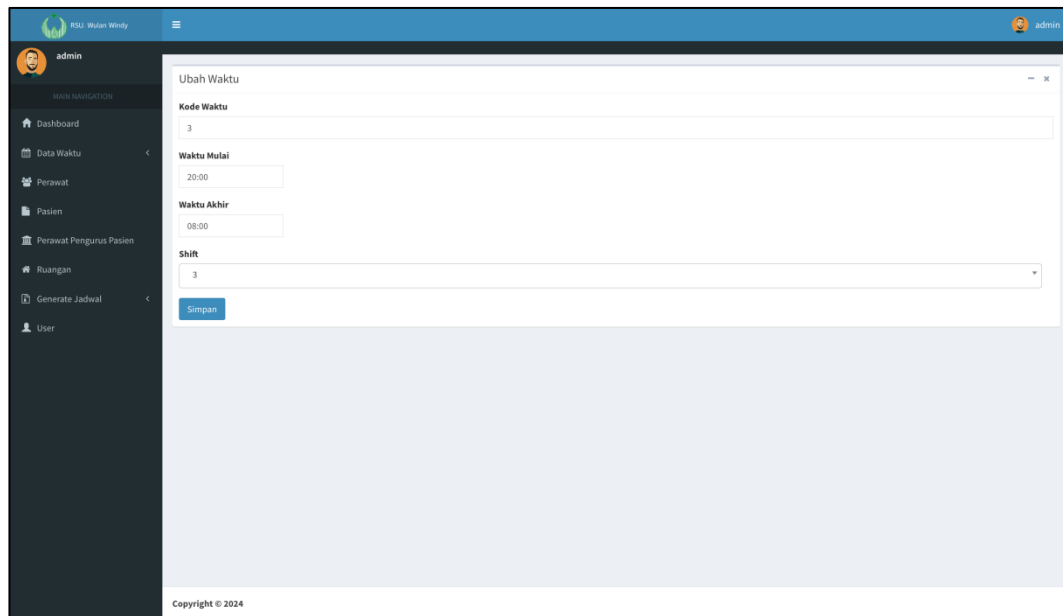
Tampilan pada gambar 4.4 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data waktu. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data waktu ke *database*.



Gambar 4.4 Tampilan Tambah Data Waktu

5. Tampilan Ubah Data Waktu

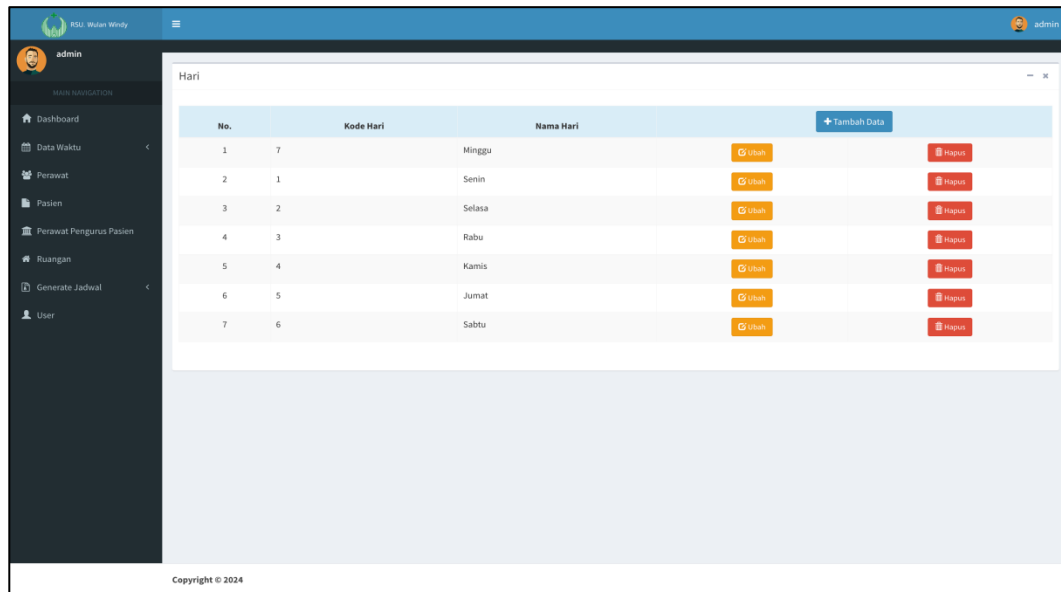
Tampilan pada gambar 4.5 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data waktu. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data waktu di *database*.



Gambar 4.5 Tampilan Ubah Data Waktu

6. Tampilan Kelola Data Hari

Tampilan pada gambar 4.6 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data hari. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data hari.



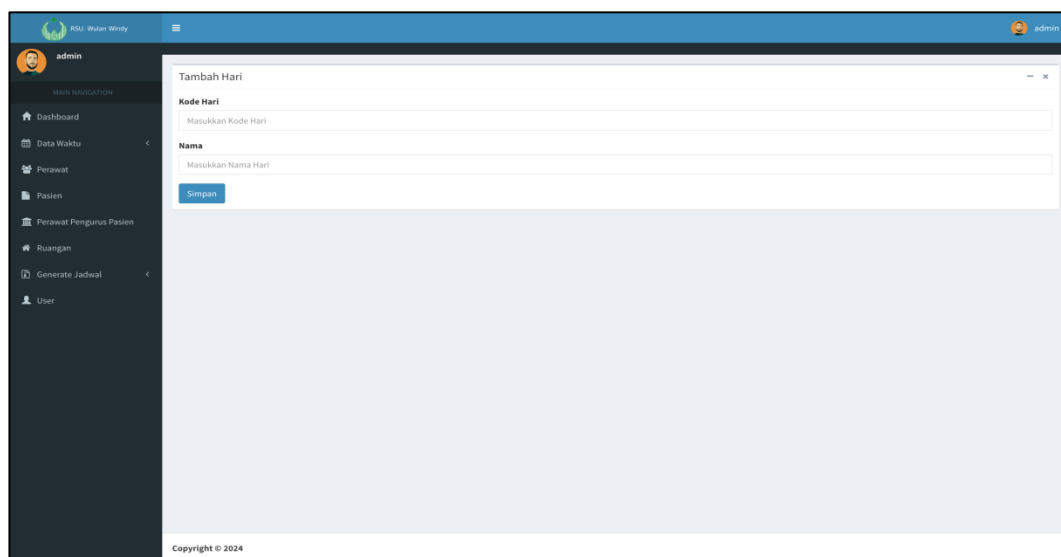
The screenshot shows a web application interface for managing days. On the left is a dark sidebar with a navigation menu including Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The main content area is titled 'Hari' and contains a table with 7 rows. Each row has columns for 'No.', 'Kode Hari', and 'Nama Hari'. To the right of each row are two buttons: a yellow 'Ubah' (Edit) button and a red 'Hapus' (Delete) button. A '+ Tambah Data' button is located at the top right of the table. The footer of the application shows 'Copyright © 2024'.

No.	Kode Hari	Nama Hari	Ubah	Hapus
1	7	Minggu	Ubah	Hapus
2	1	Senin	Ubah	Hapus
3	2	Selasa	Ubah	Hapus
4	3	Rabu	Ubah	Hapus
5	4	Kamis	Ubah	Hapus
6	5	Jumat	Ubah	Hapus
7	6	Sabtu	Ubah	Hapus

Gambar 4.6 Tampilan Kelola Data Hari

7. Tampilan Tambah Data Hari

Tampilan pada gambar 4.7 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data hari. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data hari ke *database*.

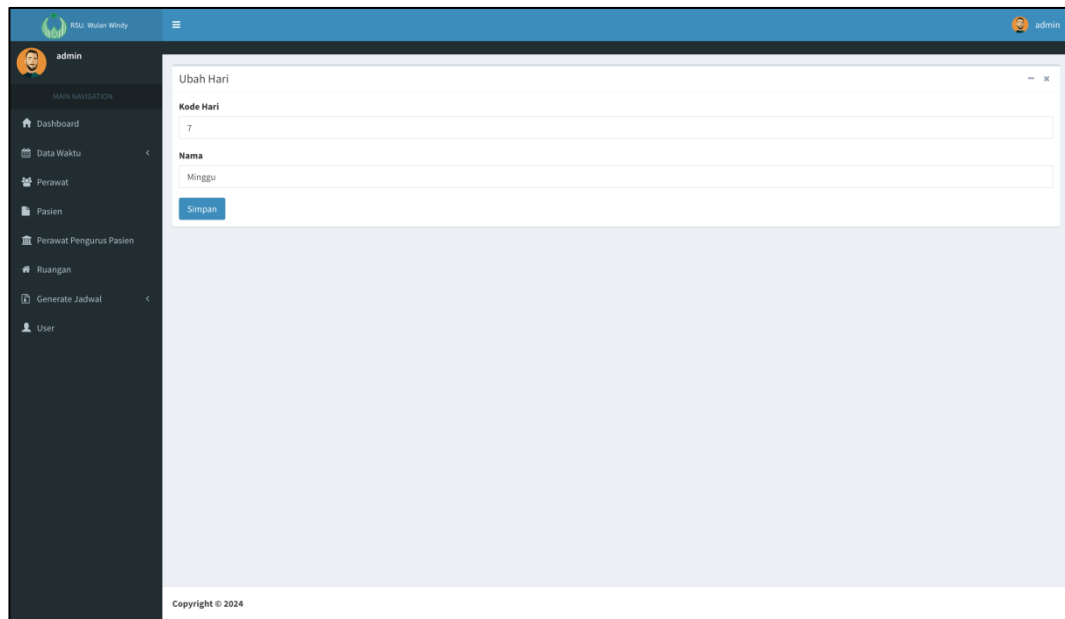


The screenshot shows the 'Tambah Hari' (Add Day) form. It has two input fields: 'Kode Hari' with the placeholder text 'Masukkan Kode Hari' and 'Nama' with the placeholder text 'Masukkan Nama Hari'. Below the input fields is a blue 'Simpan' (Save) button. The sidebar and footer are consistent with the previous screenshot, showing 'Copyright © 2024'.

Gambar 4.7 Tampilan Tambah Data Hari

8. Tampilan Ubah Data Hari

Tampilan pada gambar 4.8 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data hari. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data hari di *database*.

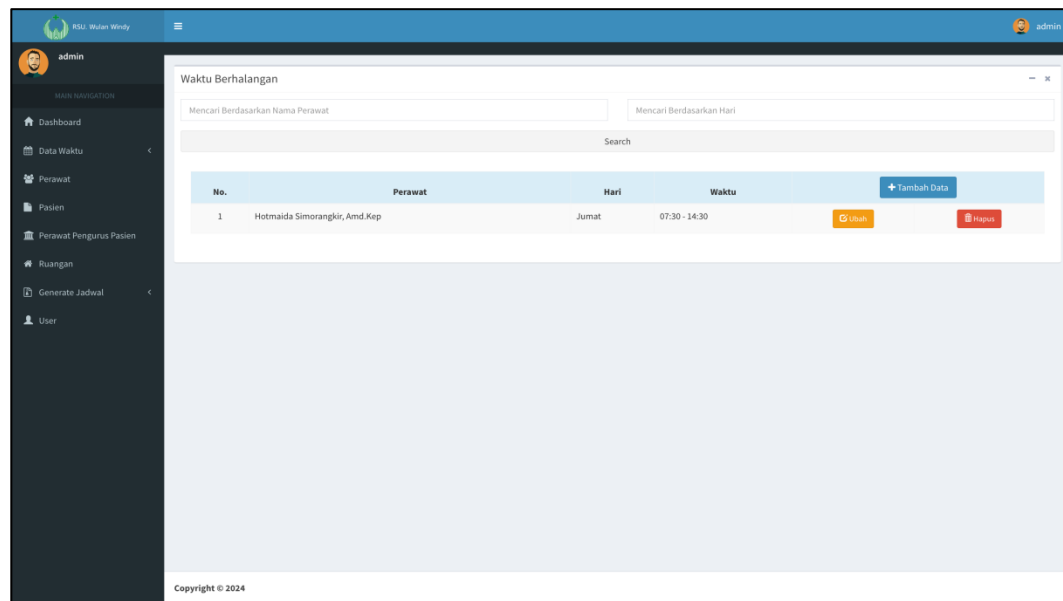


The screenshot shows a web application interface for changing a day. The main content area is titled 'Ubah Hari'. It contains two input fields: 'Kode Hari' with the value '7' and 'Nama' with the value 'Minggu'. Below the 'Nama' field is a blue button labeled 'Simpan'. The interface includes a sidebar menu on the left with options like Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top bar shows the user 'admin' and the application name 'RDU: Wulan Windy'.

Gambar 4.8 Tampilan Ubah Data Hari

9. Tampilan Kelola Data Waktu Berhalangan

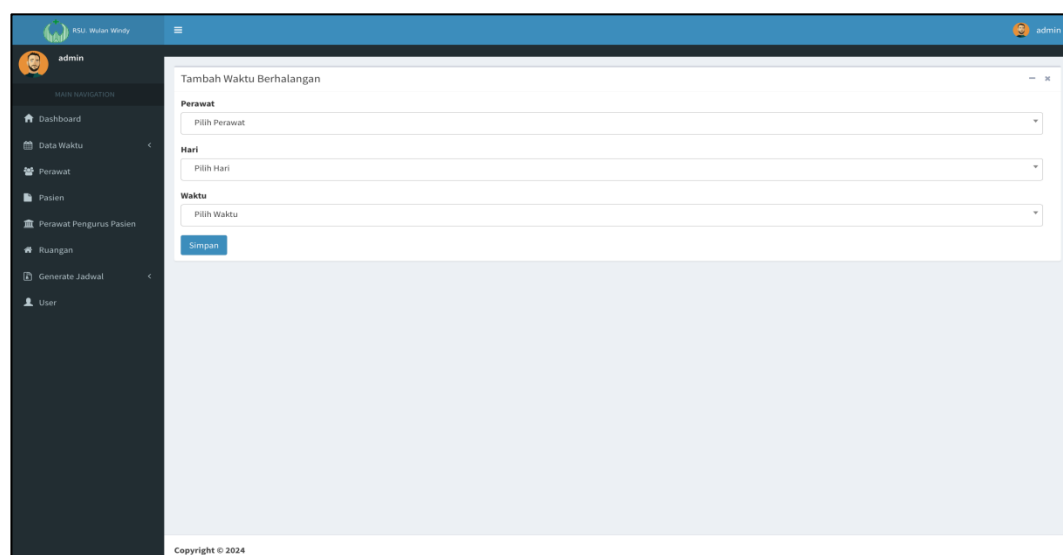
Tampilan pada gambar 4.9 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data waktu berhalangan. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data waktu berhalangan.



Gambar 4.9 Tampilan Kelola Data Waktu Berhalangan

10. Tampilan Tambah Data Waktu Berhalangan

Tampilan pada gambar 4.10 merupakan tampilan berisikan *form* untuk melakukan penambahan data waktu berhalangan. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data waktu berhalangan ke *database*.



Gambar 4.10 Tampilan Tambah Data Waktu Berhalangan

11. Tampilan Ubah Data Waktu Berhalangan

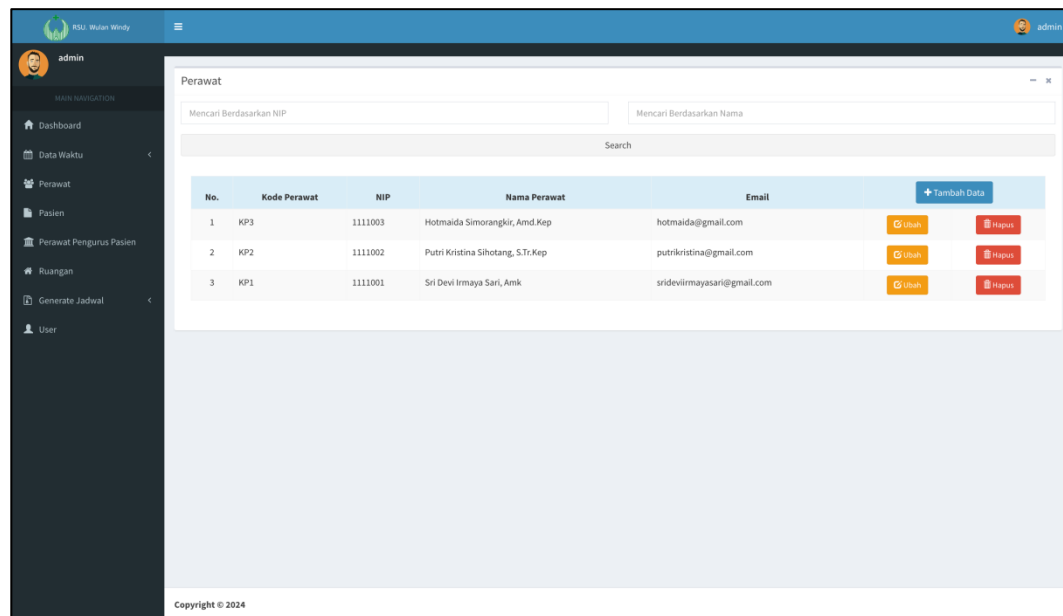
Tampilan pada gambar 4.11 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data waktu berhalangan. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data waktu berhalangan di *database*.

The screenshot shows a web application interface for updating a doctor's unavailable time. The title of the form is "Ubah Waktu Tidak Bersedia Dosen". It features three input fields: "Perawat" with the value "Hotmaida Simorangkir, Amd.Kep", "Hari" with the value "Jumat", and "Waktu" with the value "07:30 - 14:30". A blue "Simpan" button is located below the "Waktu" field. The left sidebar contains a navigation menu with items like Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top bar shows the user is logged in as "admin".

Gambar 4.11 Tampilan Ubah Data Waktu Berhalangan

12. Tampilan Kelola Data Perawat

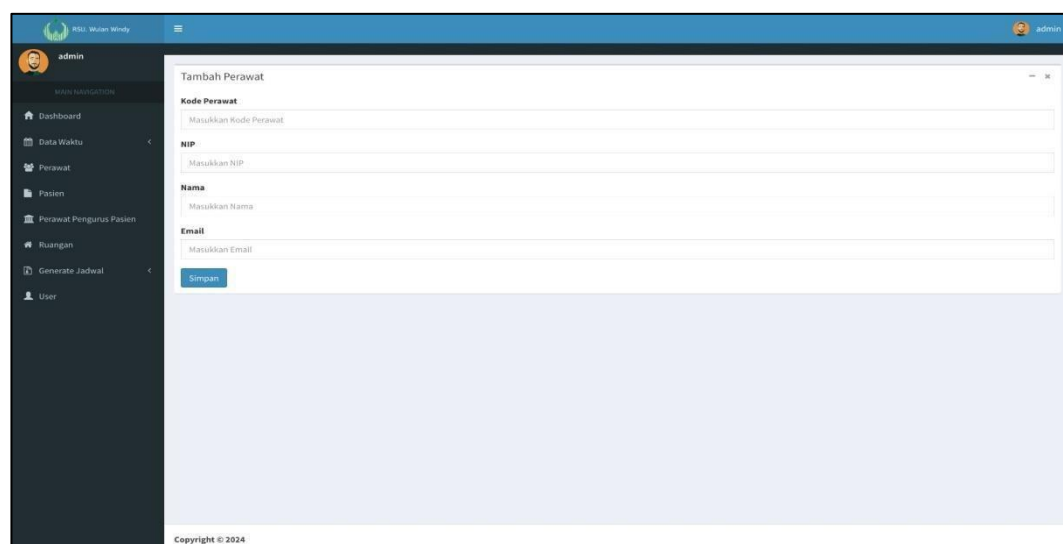
Tampilan pada gambar 4.12 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data perawat. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data perawat. Selain itu terdapat adanya *form* isian pencarian nama



Gambar 4.12 Tampilan Kelola Data Perawat

13. Tampilan Tambah Data Perawat

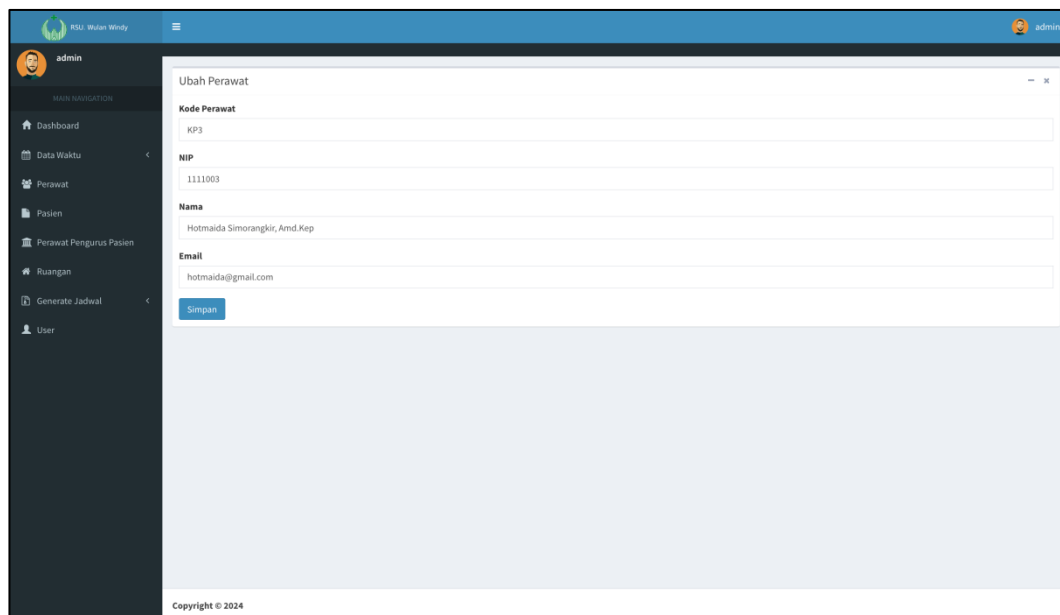
Tampilan pada gambar 4.13 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data perawat. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data perawat ke *database*.



Gambar 4.13 Tampilan Tambah Data Perawat

14. Tampilan Ubah Data Perawat

Tampilan pada gambar 4.14 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data perawat. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data perawat di *database*.

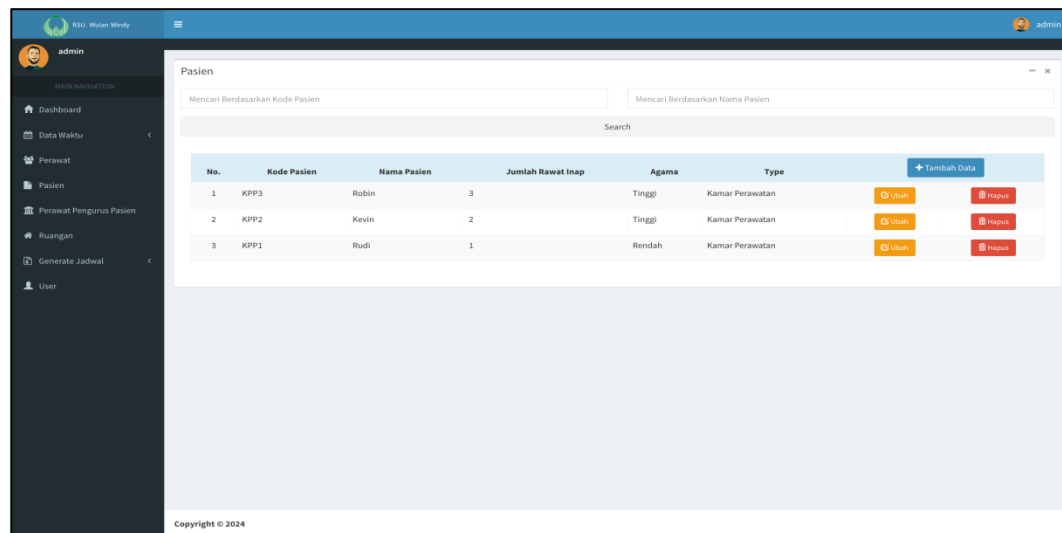


The screenshot shows a web application interface for editing nurse data. The main content area is titled 'Ubah Perawat'. It contains a form with the following fields: 'Kode Perawat' with the value 'KP3', 'NIP' with the value '1111003', 'Nama' with the value 'Hotmaida Simorangkir, Amd.Kep', and 'Email' with the value 'hotmaida@gmail.com'. Below these fields is a blue 'Simpan' button. The interface also features a dark sidebar on the left with navigation links: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top header shows the user 'admin' and a copyright notice 'Copyright © 2024' at the bottom.

Gambar 4.14 Tampilan Ubah Data Perawat

15. Tampilan Kelola Data Pasien

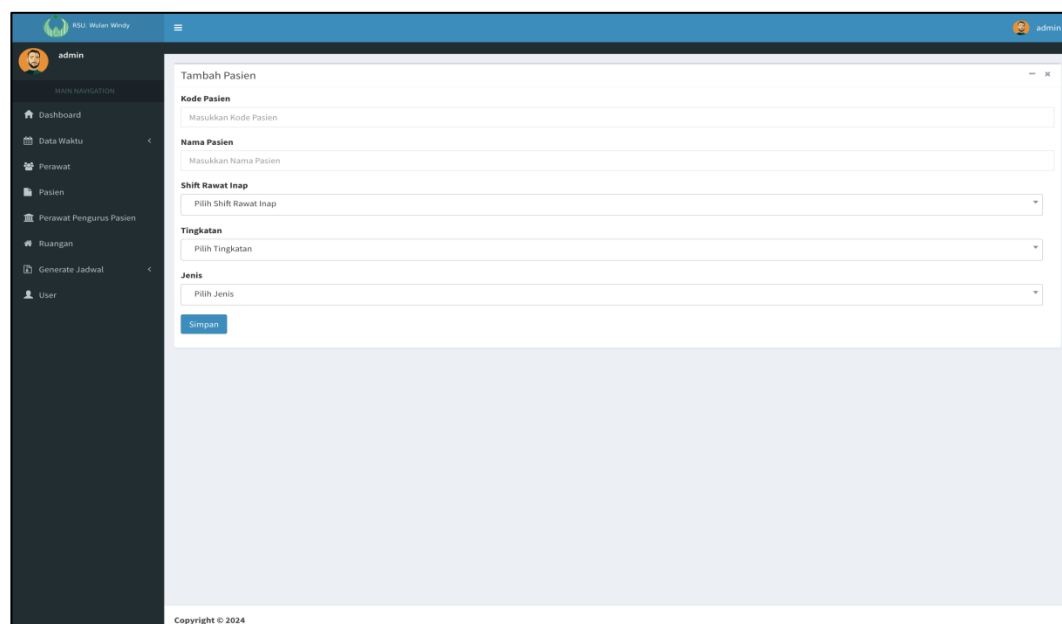
Tampilan pada gambar 4.15 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data pasien. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data pasien. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 4.15 Tampilan Kelola Data Pasien

16. Tampilan Tambah Data Pasien

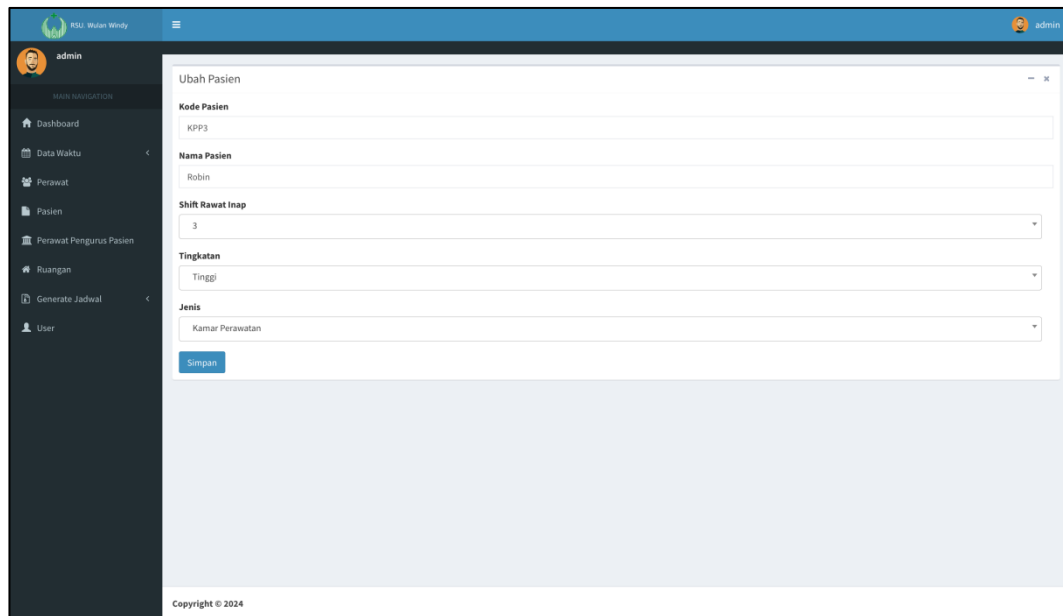
Tampilan pada gambar 4.16 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data pasien. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data pasien ke *database*.



Gambar 4.16 Tampilan Tambah Data Pasien

17. Tampilan Ubah Data Pasien

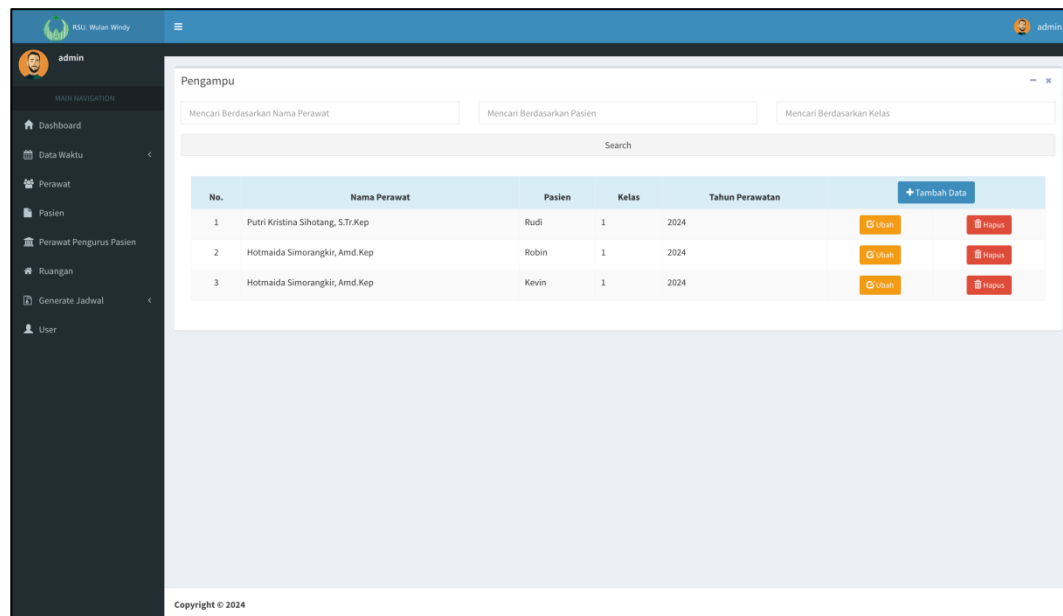
Tampilan pada gambar 4.17 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data pasien. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data pasien di *database*.



Gambar 4.17 Tampilan Ubah Data Pasien

18. Tampilan Kelola Data Perawat Pengurus Pasien

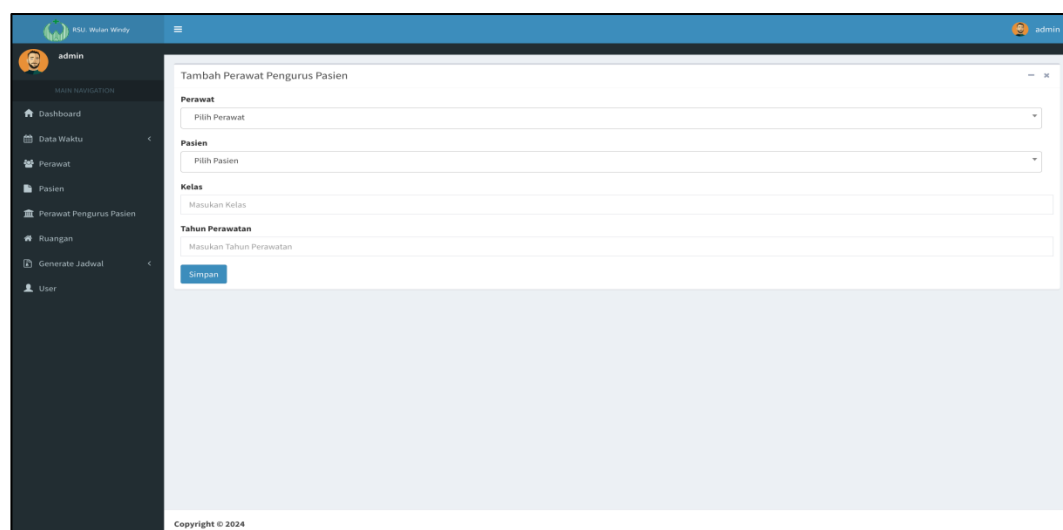
Tampilan pada gambar 4.18 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data perawat pengurus pasien. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data perawat pengurus pasien. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 4.18 Tampilan Kelola Data Perawat Pengurus Pasien

19. Tampilan Tambah Data Perawat Pengurus Pasien

Tampilan pada gambar 4.19 merupakan tampilan yang berisikan *form* untuk menambah data perawat pengurus pasien. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk menambah data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data perawat pengurus pasien ke *database*.



Gambar 4.19 Tampilan Tambah Data Perawat Pengurus Pasien

20. Tampilan Ubah Data Perawat Pengurus Pasien

Tampilan pada gambar 4.20 merupakan tampilan yang berisikan *form* untuk mengubah data perawat pengurus pasien. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk mengubah informasi data perawat pengurus pasien di *database*.

The screenshot shows a web application interface for managing patient nurse data. The main content area is titled 'Ubah Perawat Pengurus Pasien'. It contains a form with the following fields:

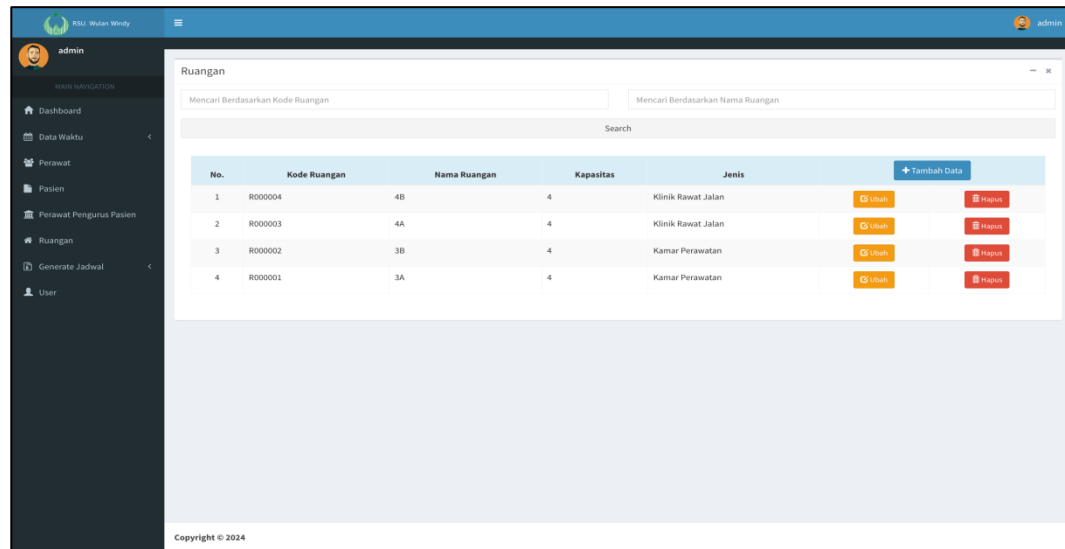
- Perawat:** Putri Kristina Sihotang, S.Tr.Kep
- Pasien:** Rudi
- Kelas:** 1
- Tahun Perawatan:** 2024

At the bottom of the form is a blue 'Simpan' button. The interface includes a dark sidebar with navigation links: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top header shows the user 'admin' and a copyright notice 'Copyright © 2024' at the bottom.

Gambar 4.20 Tampilan Ubah Data Perawat Pengurus Pasien

21. Tampilan Kelola Data Ruangan

Tampilan pada gambar 4.21 merupakan tampilan yang berisikan *list* untuk melakukan pengelolaan data ruangan. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data ruangan. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 4.21 Tampilan Kelola Data Ruangan

22. Tampilan Tambah Data Ruangan

Tampilan pada gambar 4.22 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data ruangan. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data ruangan ke *database*.

Tambah Ruangan

Kode Ruangan
Masukkan Kode Ruang

Nama Ruangan
Masukkan Nama Ruangan

Kapasitas
Masukkan kapasitas

Jenis
Pilih Jenis

Simpan

Gambar 4.22 Tampilan Tambah Data Ruangan

23. Tampilan Ubah Data Ruangan

Tampilan pada gambar 4.23 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data ruangan. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data ruangan di *database*.

The screenshot shows a web application interface for editing room data. The main content area is titled 'Ubah Ruangan'. It contains the following fields:

- Kode Ruangan:** R000004
- Nama Ruangan:** 4B
- Kapasitas:** 4
- Jenis:** Pilih Jenis (dropdown menu)

A blue 'Simpan' button is located below the 'Jenis' field. The left sidebar contains a navigation menu with the following items: Dashboard, Data Waktu, Perawat, Pasien, Perawat Pengurus Pasien, Ruangan, Generate Jadwal, and User. The top bar displays 'RDU: Wulan Windy' and the user 'admin'.

Gambar 4.23 Tampilan Ubah Data Ruangan

24. Tampilan *Form Generate Jadwal*

Tampilan pada gambar 4.24 merupakan tampilan yang berisikan *form* untuk melakukan *generate* jadwal *shift* perawat. Pada tampilan ini terdapat beberapa isian *form* yang harus diisi seperti tahun perawatan, semester, nilai pembangkitan kromosom, maksimal nilai generasi, nilai *crossover*, dan nilai mutasi. Setelah pengaturan dilakukan, maka tekan tombol *generate* untuk melakukan proses *generate* jadwal dengan algoritma LAHC.

Gambar 4.24 Tampilan Form Generate Jadwal

25. Tampilan Hasil *Generate* Optimasi Jadwal

Tampilan pada gambar 4.25 merupakan tampilan yang berisikan informasi hasil penjadwalan *shift* perawat yang di-*generate* berdasarkan algoritma LAHC. Pada tampilan ini juga terdapat beberapa tombol seperti tombol kembali, tombol *export* Excel data ini untuk mengunduh hasil *generate* jadwal dalam bentuk Excel dan pilihan jadwal-jadwal lainnya.

No.	Hari	Jam	Nama Ruangan	Kapasitas Ruangan	Pasien	Perawat	Tingkatan	Shift	Kelas
1	Senin	20:00 - 08:00	3B	4	Robin	Hotmaida Simorangkir, Amd.Kep	Tinggi	3	1
2	Selasa	13:30 - 20:30	3A	4	Kevin	Hotmaida Simorangkir, Amd.Kep	Tinggi	2	1
3	Rabu	20:00 - 08:00	3A	4	Robin	Hotmaida Simorangkir, Amd.Kep	Tinggi	3	1

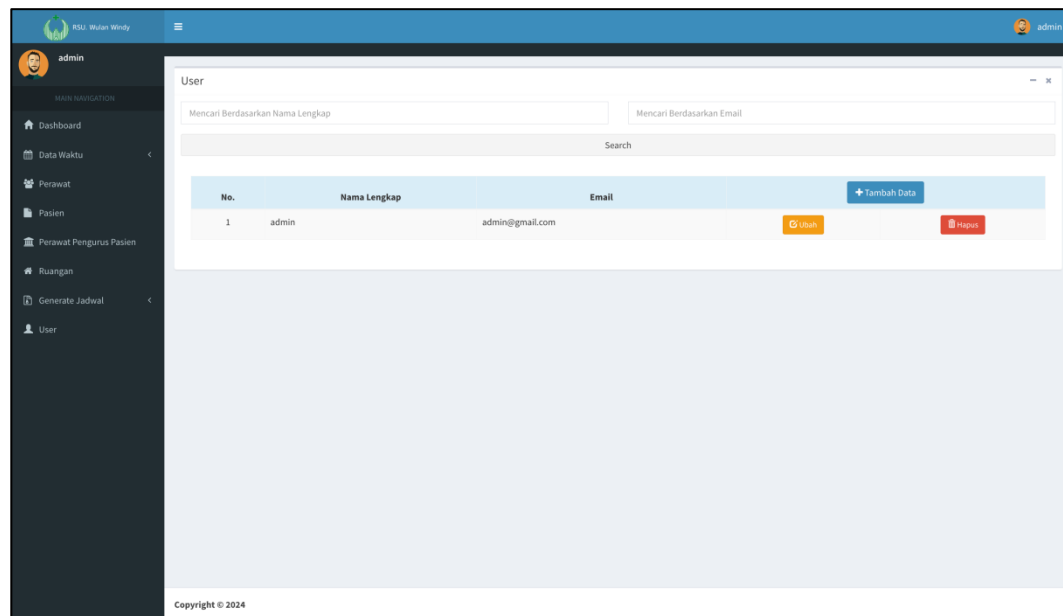
Gambar 4.25 Tampilan Hasil Generate Optimasi Jadwal

No	Hari	Jam	Nama Ruangan	Kapasitas Ruangan	Mata Kuliah	Dosen Pengampu	Semester	SKS	Kelas	Nilai
1	Senin	20:00-08:00	3A		4. Robin	Hotmaida Simorangkir, Amd.Kep	Tinggi	3	1	1
2	Selasa	13:30-20:30	3A		4. Kevin	Hotmaida Simorangkir, Amd.Kep	Tinggi	2	1	1
3	Rabu	20:00-08:00	3A		4. Robin	Hotmaida Simorangkir, Amd.Kep	Tinggi	3	1	1

Gambar 4.26 Tampilan Hasil *Export File Excel* Hasil Generate Optimasi Jadwal

26. Tampilan Kelola Data *User Admin*

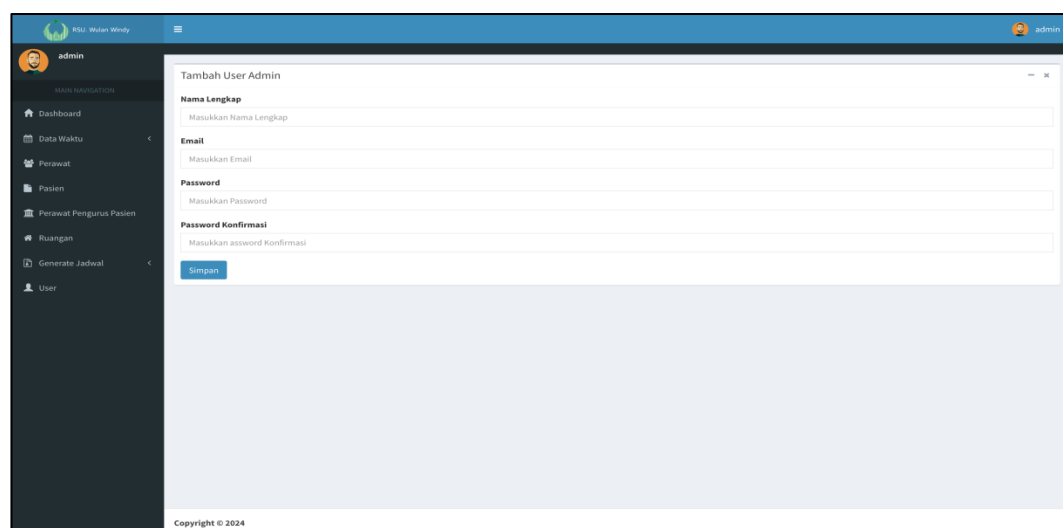
Tampilan pada gambar 4.27 merupakan tampilan yang berisikan *list* untuk mengelola data *user admin*. Pada tampilan ini terdapat beberapa tombol seperti tombol tambah data untuk menampilkan halaman penambahan data, tombol ubah untuk menampilkan halaman perubahan data, dan tombol hapus untuk menghapus data *user admin*. Selain itu terdapat adanya *form* isian pencarian beserta sebuah tombol pencarian untuk mencari data.



Gambar 4.27 Tampilan Kelola Data User Admin

27. Tampilan Tambah Data User Admin

Tampilan pada gambar 4.28 merupakan tampilan yang berisikan *form* untuk melakukan penambahan data *user* admin. Pada tampilan ini disediakan beberapa *form* isian yang harus diisi untuk melakukan penambahan data. Apabila *form* telah terisi, maka cukup menekan tombol simpan untuk menyimpan informasi data *user* admin ke *database*.



Gambar 4.28 Tampilan Tambah Data User Admin

28. Tampilan Ubah Data *User Admin*

Tampilan pada gambar 4.29 merupakan tampilan yang berisikan *form* untuk melakukan perubahan data *user admin*. Pada tampilan ini disediakan beberapa *form* isian yang telah terisi untuk dilakukan perubahan data. Apabila isi *form* telah diubah, maka cukup menekan tombol simpan untuk melakukan perubahan informasi data *user admin* di *database*.

Gambar 4.29 Tampilan Ubah Data *User Admin*

4.2 Pembahasan

Pada sub bab ini dilakukan pembahasan mengenai kelebihan dan kekurangan dari pengembangan sistem optimasi penjadwalan *shift* perawat menggunakan algoritma *Late Acceptance Hill Climbing Hyper-Heuristic* (LAHC) yang telah dibangun. Kelebihan dari sistem yang dibangun yaitu:

1. Sistem yang dibangun mengimplementasikan algoritma LAHC sehingga memungkinkan sistem untuk menghasilkan jadwal *shift* perawat yang

optimal. Hal ini tercapai karena LAHC dapat menerima kandidat solusi yang lebih luas, meningkatkan kemungkinan menemukan solusi yang lebih baik untuk penjadwalan perawat.

2. Sistem yang dibangun mengimplementasikan algoritma LAHC yang menunjukkan efektivitas dan performa yang baik dalam menghasilkan jadwal *shift* perawat. Hal ini berarti sistem dapat membuat jadwal dengan distribusi beban kerja yang lebih adil, mengurangi kemungkinan *overload* pada perawat tertentu dan memastikan semua perawat memiliki waktu istirahat yang memadai.

Kekurangan dari sistem yang dibangun yaitu:

1. Sistem yang dibangun hanya mampu menghasilkan jadwal untuk periode satu minggu. Hal ini bisa menjadi kendala jika rumah sakit membutuhkan penjadwalan untuk periode yang lebih panjang, misalnya satu bulan atau lebih.
2. Sistem ini hanya menggunakan data dari RSU Wulan Windy di Kecamatan Medan Marelan, Kota Medan. Penggunaan data yang terbatas ini bisa mempengaruhi generalisasi hasil penjadwalan ketika diterapkan di rumah sakit lain dengan kondisi yang berbeda.
3. Sistem yang dibangun hanya mencakup perawat yang bertugas di ruang rawat inap tertentu (grup 3A dan 3B). Hal ini berarti sistem belum mencakup seluruh perawat di rumah sakit, seperti perawat di unit gawat darurat atau ruang operasi, yang juga memerlukan penjadwalan *shift*.

4. Sistem ini hanya mendukung penjadwalan dengan tiga *shift* (pagi, sore, malam). Jika rumah sakit memiliki kebutuhan penjadwalan dengan lebih dari tiga *shift* atau dengan fleksibilitas waktu yang lebih tinggi, sistem ini mungkin perlu disesuaikan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berikut ini diuraikan kesimpulan akhir dari penelitian yang telah dilakukan yaitu:

1. Hasil penelitian menunjukkan bahwa algoritma LAHC terbukti efektif dalam menghasilkan jadwal *shift* perawat di RSUD Wulan Windy. Algoritma ini dapat mengakomodasi lebih banyak kandidat solusi, memungkinkan penjadwalan yang lebih adil dan merata.
2. Aplikasi penjadwalan *shift* perawat pada RSUD Wulan Windy yang dibangun berbasis *web* menghasilkan tabel penjadwalan terdiri dari hari, jam, nama ruangan, kapasitas ruangan, pasien, perawat, *shift*, dan kelas. Parameter yang digunakan adalah nilai maksimal generasi, nilai pembangkitan kromosom, nilai *crossover*, dan nilai mutasi.

5.2 Saran

Berikut ini diuraikan saran akhir dari penelitian yang telah dilakukan yaitu:

1. Disarankan untuk mengembangkan sistem agar mampu menghasilkan jadwal untuk periode yang lebih panjang, seperti tiga bulan atau satu tahun. Hal ini akan membantu manajemen rumah sakit dalam perencanaan jangka panjang dan meningkatkan efisiensi operasional.

2. Untuk meningkatkan generalisasi dan aplikasi sistem, disarankan agar sistem diuji dan diimplementasikan dengan data dari berbagai rumah sakit. Hal ini akan membantu memastikan bahwa sistem dapat berfungsi dengan baik di berbagai lingkungan dan kondisi yang berbeda.
3. Disarankan untuk memperluas cakupan data perawat yang diakomodasi oleh sistem, termasuk perawat di unit gawat darurat, ruang operasi, dan unit lain yang memerlukan penjadwalan *shift*. Hal ini akan memastikan bahwa seluruh kebutuhan penjadwalan rumah sakit dapat terpenuhi.
4. Disarankan agar sistem dikembangkan untuk mendukung lebih dari tiga *shift* atau fleksibilitas waktu yang lebih tinggi. Ini akan memberikan manajemen rumah sakit kemampuan untuk menyesuaikan jadwal sesuai dengan kebutuhan spesifik dan kondisi operasional yang dinamis.

DAFTAR PUSTAKA

- [1] I. G. A. Premananda and A. Muklason, "Optimasi Penjadwalan Mata Kuliah Menggunakan Algoritma Late Acceptance Hill Climbing Berbasis Hiper Heuristik," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 774–782, 2021, doi: 10.35957/jatisi.v8i2.778.
- [2] R. R. Utama, T. Nahdliyah, A. N. Nabila, and A. Muklason, "Heuristic Selection Algorithms Comparison for Examination Timetabling Using Hyper-Heuristics Framework," *J. Tek. Elektro dan Komput.*, vol. 12, no. 2, pp. 85–92, 2023.
- [3] A. Muklason *et al.*, "Flexible Automated Course Timetabling System with Lecturer Preferences Using Hyper-heuristic Algorithm," 2022.
- [4] R. R. Utama, "Optimasi Penjadwalan Perawat dengan Algoritma Late Acceptance Hill Climbing Hyper-Heuristic Menggunakan Benchmark Dataset dari Rumah Sakit di Norwegia," Institut Teknologi Sepuluh Nopember., 2020.
- [5] D. A. Pratiwi, *Artikel Ilmiah Pengantar Sistem Informasi*, no. 43219010158. 2020.
- [6] A. Yoraeni *et al.*, *Sistem Informasi Manajemen*, no. November. Jakarta: PT. Scifintech Andrew Wijaya, 2023.
- [7] N. Azis, *Analisis Perancangan Sistem Informasi*. Bandung: Widina Bhakti Persada, 2022.
- [8] P. E. Sudjiman and L. Siregar, "KOMPUTER DALAM PROSES PENGAMBILAN KEPUTUSAN," *J. TeIKa*, vol. 8, pp. 55–67, 2018.
- [9] R. Ernayani *et al.*, *Dasar dan Teori Sistem Informasi Manajemen*. Batam: Yayasan Cendikia Mulia Mandiri, 2022.
- [10] S. Hidayatullah, D. A. Prasetya, D. A. Purnomo, and I. K. Rachmawati, *Hot Fit Model Pengembangan Sistem Informasi*. Ponorogo: Uwais Inspirasi Indonesia, 2022.

- [11] J. Dalle, A. . Karim, and Baharuddin, *Pengantar Teknologi Informasi*. Depok: PT RajaGrafindo Persada, 2020.
- [12] F. Mone and J. E. Simarmata, “Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah,” *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 15, no. 4, pp. 615–628, 2021, doi: 10.30598/barekengvol15iss4pp615-628.
- [13] I. H. Pulu, A. A. Pekuwali, and A. C. Talakua, “Penerapan Algoritme Genetika Penjadwalan Perawat di Rumah Sakit Umum Imanuel Sumba,” *Contar J. Comput. Sci.*, vol. 1, no. 1, pp. 01–05, 2023.
- [14] R. R. Utama and A. Muklason, “Pembentukan Solusi Awal International Timetabling Competition 2021,” *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 4, pp. 1939–1944, 2021.
- [15] R. A. Sukamto and M. Shalahuddin, “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.” Informatika, Bandung, 2018.
- [16] A. O. Sari, A. Abdilah, and Sunarti, *Web Programming*. Yogyakarta: Graha Ilmu, 2019.
- [17] R. A. Putri, *Buku Ajar Basis Data Edisi Kedua*. Bandung: CV. Media Sains Indonesia, 2022.

DAFTAR RIWAYAT HIDUP

Nama : Rama Arviansyah Hutasuhut
Tempat/Tanggal Lahir : Medan/ 12-08-2001
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat : Jl. Jermal Raya Lingk II Sei Mati
Email : ramaarviansyahhutasuhut@gmail.com

RIWAYAT PENDIDIKAN

2007 – 2013 : SD Dr Wahidin Sudirohusodo
2013 – 2016 : SMP Dr Wahidin Sudirohusodo
2016 – 2019 : SMA Dr Wahidin Sudirohusodo
2020 – 2024 : STMIK TIME MEDAN

SURAT KEPUTUSAN
NO. 131/SK/P1/STMIKTIME/XII/2023

Tentang
PENGANGKATAN DOSEN PEMBIMBING SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME

1. **Membaca** : a. Hasil pemeriksaan berkas dan kelengkapan mahasiswa.
b. Presentasi Proposal Judul Skripsi mahasiswa tertanggal 09 Oktober – 18 November 2023.
2. **Mengingat** : a. Kurikulum Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time Program Studi Teknik Informatika.
b. Penyesuaian teknis pelaksanaan Skripsi Program Studi Teknik Informatika.
3. **Menimbang** : Bahwa skripsi mahasiswa merupakan wujud aplikasi kristalisasi sikap kemampuan dan ketrampilan mahasiswa/i dalam upaya menerapkan ilmu pengetahuan yang telah dipelajari dibangku kuliah, serta skripsi merupakan salah satu mata kuliah yang memiliki bobot 6 SKS, maka dipandang perlu untuk menetapkan Dosen Pembimbing I dan Dosen Pembimbing II Skripsi.

4. **Menetapkan** :
Pertama :

Mengangkat Saudara :

Pembimbing I	Dr. Huliman, S.Kom., S.M., M.Kom
Pembimbing II	Devi, S.Kom., S.A.B., M.M

Sebagai Dosen Pembimbing Skripsi atas :

2044133	Rama Arviansyah Hutasuhut	Teknik Informatika
---------	---------------------------	--------------------

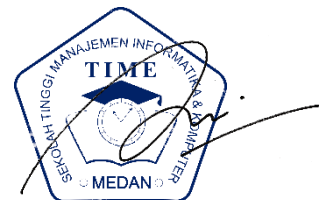
Dengan Judul Skripsi :

Perancangan Aplikasi Penjadwalan Shift Perawat Menggunakan Algoritma Late Acceptance Hill Climbing Hyper - Heuristic (Studi Kasus RSU. Wulan Windy Medan)

Mahasiswa Program Strata Satu (S1) Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time.

- Kedua : Apabila terdapat kekeliruan di kemudian hari tentang penetapan Dosen Pembimbing Skripsi ini akan diperbaiki sebagaimana mestinya
- Ketiga : Surat Keputusan ini berlaku sejak tanggal ditetapkan 07 Desember 2023.

Ditetapkan di
Medan, 07 Desember 2023



Hendri, M.Kom
Pembantu Ketua I STMIK TIME



STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : RAMA ARVIANSYAH HUTASUHUT
NIM : 2044133
JUDUL SKRIPSI : PERANCANGAN APLIKASI PENJADWALAN *SHIFT* PERAWAT MENGGUNAKAN ALGORITMA *LATE ACCEPTANCE HILL CLIMBING HYPER-HEURISTIC* (STUDI KASUS RSU. WULAN WINDY)
NAMA DOSEN : Dr. HULIMAN, M.Kom.
DOSEN PEMBIMBING : ☒ I (SATU) ☐ II (DUA)

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
16 Januari 2024	BAB I: Sesuaikan penyajian dengan pedoman skripsi dan tambahkan detail permasalahan serta metode wawancara untuk pengumpulan data	
25 Januari 2024	BAB I: Perhatikan penggunaan tanda baca dan penulisan kata sesuai KBBI BAB II:	
19 Februari 2024	BAB I: Disetujui BAB II: Tambahkan tinjauan pustaka mengenai website dan database serta algoritma yang digunakan	
4 Maret 2024	BAB II: Disetujui BAB III: Tambahkan kalimat pengarah ke gambar pada sub-bab 3.2 Perancangan	
22 April 2024	BAB III: Disetujui Silakan melanjutkan bimbingan ke Dosen Pembimbing 2	
19 Juli 2024	BAB IV & Aplikasi: Berhubung hasil generate jadwal hanya untuk 1 minggu, maka batasan masalah perlu diperbaiki.	
26 Juli 2024	BAB IV & Aplikasi: Silakan perbaiki nilai input generasi pada aplikasi BAB V: Perbaiki kesimpulan	
27 Juli 2024	BAB IV: Disetujui dengan revisi minor pada aplikasi BAB V: Disetujui	
29 Juli 2024	1. Silakan melengkapi segala dokumen pendukung untuk kelengkapan skripsi disesuaikan dengan pedoman skripsi, termasuk paper (jurnal). 2. Silakan segera mengajukan ujian skripsi jika penyajian skripsi sudah disetujui oleh dosen pembimbing kedua (2). Terima Kasih.	

Ketentuan :

1. Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
2. Formulir Bimbingan wajib diisi dan ditunjukkan ke dosen pembimbing tiap bimbingan
3. Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
4. Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan, Juli 2024
Dosen Pembimbing,

Mahasiswa,



Dr. HULIMAN, M.Kom.

RAMA ARVIANSYAH HUTASUHUT



STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : RAMA ARVIANSYAH HUTASUHUT
NIM : 2044133
JUDUL SKRIPSI : OPTIMASI PENJADWALAN SHIFT PERAWAT MENGGUNAKAN METODE ALGORTIMA LATE ACCEPTANCE HILL CLIMBING HYPER - HEURISTIC (STUDI KASUS RSU. WULAN WINDY MEDAN)
NAMA DOSEN : DEVI, S.KOM., S.AB., M.M
DOSEN PEMBIMBING : ☐ I (SATU) ☒ II (DUA)

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
19 April 2024	Pada rumusan masalah, poin kedua, apakah algoritma pembandingan yang dipakai ?	
20 April 2024	ACC Bab 1, Lanjutkan ke Bab 2	
22 April 2024	Bab 2 - Hapus kata pengulangan sehingga pada kalimat informasi merupakan hasil pengolahan data sehingga sehingga menjadi bentuk ... (hal.12) - Penekanan suatu kata tidak perlu membuat warna yang berbeda (hal.20) - Perhatikan jarak spasi antar baris (bila masih dalam satu bahasan tidak perlu ada jarak antar baris) pada hal. 26, 27, 30 - Kedepannya pengiriman file dengan pdf saja agar tidak ada perbedaan pengetikkan Bab 3 sudah ok	
26 April 2024	ACC Bab 2 dan 3, Silakan lanjutkan proses berikutnya	
26 Juli 2024	Revisi Bab 3, periksa kembali penulisan dan juga penulisan format bahasa asing	
27 Juli 2024	ACC Bab3, Lanjut Bab 4	
29 Juli 2024	ACC Bab 4 dan 5 , silakan kirimkan skripsi lengkap	

Ketentuan :

1. Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
2. Formulir Bimbingan wajib diisi tiap bimbingan
3. Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
4. Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan, **26 April 2024**

Dosen Pembimbing,



Devi,S.kom.,S.A.,M.M

Mahasiswa,

RAMA ARVIANSYAH HUTASUHUT