

**IMPLEMENTASI PENJADWALAN PASIEN POLI SPESIALIS
DI RUMAH SAKIT MENGGUNAKAN ALGORITMA
*SIMULATED ANNEALING***

Diajukan sebagai Salah Satu Syarat Untuk
Mendapatkan Gelar Sarjana Komputer (S.Kom)
Program Studi Informatika



SKRIPSI

Disusun oleh :

MUH. ZUBIR PRADADANG

NIM : 105841103119

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025



UNIVERSITAS MUHAMMADIYAH MAKASSAR



FAKULTAS TEKNIK

PENGESAHAN

Skripsi atas nama **Muh. Zubir Pradadang** dengan nomor induk Mahasiswa **105 84 1103119**, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025 sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025

Panitia Ujian :

Makassar, 06 Rabi'ul Awwal 1447 H
30 Agustus, 2025 M

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST., MT., IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng :

2. Penguji

a. Ketua : Prof. Dr. Ir. Hafsah Nirwana, M.T.

b. Sekretaris : Rizki Yuliana Bakti, S.T., M.T.

3. Anggota : 1. Muhyiddin A M Hayat, S.Kom., M.T.

2. Darniati, S.Kom., M.T.

3. Titin Wahyuni, S.Pd., M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Fahrim Irfhamna Rachman, S.Kom., M.T

Lukman, S.Kom., M.T

Dekan

M. H. Syafat S. Kuba, S.T., M.T.

NPM: 795 288

Gedung Menara Iqra Lantai 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221





MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK



HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : IMPLEMENTASI PENJADWALAN PASIEN POLI SPESIALIS DI
RUMAH SAKIT MENGGUNAKAN ALGORITMA SIMULATED
ANNEALING

Nama : Muh. Zubir Pradadang

Stambuk : 105 84 1103119

Makassar, 10 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I

Pembimbing II


Fahrir Irhamna Rachman, S.Kom., M.T


Lukman, S.Kom M.T.

Mengetahui,

Ketua Prodi Informatika



Rizki Yustiana Bakti, S.T., M.T.

NPM : 1307 284



ABSTRACT

MUH. ZUBIR PRADADANG, *Implementation of Specialist Clinic Patient Scheduling in Hospitals Using the Simulated Annealing Algorithm* (Supervised by Fachrim Irhamna Rahman, S.Kom., M.T. and Lukman, S.Kom., M.T.)

This study designs and implements a specialist clinic patient scheduling system using the Simulated Annealing (SA) Algorithm to improve hospital operational efficiency. The main issues addressed include long queues, excessive waiting times, and uneven distribution of doctors' workloads. SA was chosen for its ability to avoid local suboptimal solutions through probabilistic acceptance. Simulated patient and doctor schedule data were used to test the model. The evaluation covered average waiting time, patient satisfaction, and resource utilization. Results showed a 21.3% increase in patient satisfaction, over 85% fulfillment of time preferences, more balanced doctor workload distribution, and efficient computation times for large-scale data. This system provides an adaptive and efficient scheduling solution that can be integrated into hospital environments to enhance service quality.

Keywords: Patient Scheduling, Specialist Clinic, Simulated Annealing, Optimization, Hospital.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul **“Implementasi Penjadwalan Pasien Poli Spesialis Di Rumah Sakit Menggunakan Algoritma *Simulated Annealing*”** tepat waktu sesuai dengan rencana yang telah ditetapkan.

Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, dengan segala hormat penulis menyampaikan apresiasi dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah Subhanahu Wa Ta'ala yang telah memberikan kesehatan, kekuatan, serta kesempatan kepada penulis dalam menyelesaikan skripsi ini.
2. Kedua orang tua tercinta, atas segala doa, dukungan moral dan materil, serta kasih sayang yang tiada henti dalam setiap proses kehidupan penulis.
3. Bapak **Ir. Muh. Syafaat S. Kuba, S.T., M.T., IPM.**, selaku Dekan Fakultas Teknik Universitas
4. Muhammadiyah Makassar, yang senantiasa memberikan arahan dan dukungan akademik.
5. Ibu **Rizki Yusliana Bakti, S.T., M.T** selaku Ketua Program Studi Informatika dan Dosen Pprodi Informatika, atas bimbingan, arahan, dan kontribusi akademiknya dalam proses penyusunan skripsi ini.
6. Bapak **Fachrim Irhamna Rahman, S.Kom.,M.T** selaku Dosen Pembimbing I, atas dedikasi, masukan, serta waktu yang diberikan dalam membimbing penulis.
7. Bapak **Lukman, S.Kom., M.T.**, selaku Dosen Pembimbing II, atas dedikasi, masukan, serta waktu yang diberikan dalam membimbing penulis.

8. Rekan-rekan coboy campus yang telah memberikan masukan konstruktif serta motivasi selama proses penulisan berlangsung.
9. Seluruh rekan seangkatan di Program Studi Informatika yang turut memberikan semangat dan dukungan selama proses penyusunan skripsi.
10. Staf akademik dan administrasi yang telah memberikan bantuan teknis dan informasi yang dibutuhkan dalam penyusunan dokumen ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan baik dari segi teknis maupun substansi. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan guna penyempurnaan karya ilmiah ini di masa mendatang. Penulis berharap skripsi ini dapat memberikan kontribusi positif dalam pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi.

Makassar, 27 Agustus 2025

Hormat penulis,

Muh. Zubir Pradadang

DAFTAR ISI

ABSTRAK	ii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian.....	2
D. Manfaat Penelitian	3
E. Ruang Lingkup Penelitian	4
F. Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	6
A. Landasan Teori	6
1. Rumah sakit.....	6
2. Sistem penjadwalan pasien di rumah sakit.....	6
3. Patient Rostering Problem	7
4. Poli spesialis di rumah sakit	8
5. Optimasi penjadwalan.....	8
6. Algoritma <i>Metaheuristik</i>	9
7. Algoritma <i>Simulated Annealing</i>	9
8. Penerapan <i>Simulated Annealing</i> pada Penjadwalan Pasien	11
9. Sistem Informasi Penjadwalan Berbasis Komputer.....	13
10. Kerangka Pemikiran / Alur Logika Sistem.....	13
B. Penelitian Terkait.....	14
C. Kerangka Pikir.....	20
BAB III METODE PENELITIAN	22
A. Tempat Dan Waktu penelitian	22
B. Alat Dan Bahan.....	22

C.	Perancangan Penelitian Sistem.....	22
D.	Teknik Pengujian Sistem	26
E.	Teknik Analisis Data	28
BAB IV HASIL DAN PEMBAHASAN		31
A.	Pengumpulan Data	31
B.	Perbaikan Data dan Pemrosesan Data	34
C.	Penjadwalan Pasien	38
D.	Implementasi Algoritma <i>Simulated Annealing</i>	40
E.	Hasil Dan Analisis	44
BAB V PENUTUP		47
A.	Kesimpulan	47
B.	Saran	48
DAFTAR PUSTAKA.....		50



DAFTAR TABEL

Table 1 Penelitian Terkait.....	19
Table 2 data penyakit	32
Table 3 data poli spesialis dandokter.....	33
Table 4 penentuan tingkat urgensi pasien	35
Table 5 Konversi preferensi jam ke slot waktu.....	36
Table 6 jadwal dokter.....	37
Table 7 slot waktu tersedia.....	38
Table 8 hasil penjadwalan awal.....	39
Table 9 komponen biaya	41
Table 10 parameter.....	43
Table 11 tingkat kepuasan pasien.....	45
Table 12 distribusi beban kerja dokter	45
Table 13 analisis waktu komputasi	46



DAFTAR GAMBAR

Gambar 1 Kerangka Pikir.....	21
Gambar 2 Flowchart Proses Simulasi.....	25
Gambar 3 Data pasien	31
Gambar 4 data poli spesialis dan dokter	33
Gambar 5 verifikasi kolom data	34
Gambar 6 verifikasi kolom data pasien dan poli spesialis	34
Gambar 7 penentuan tingkat urgensi pasien	35
Gambar 8 Konversi preferensi jam ke slot waktu.....	36
Gambar 9 pemrosesan jadwal dokter	37
Gambar 10 generate slot waktu dokter.....	38
Gambar 11 hasil penjadwalan awal.....	39
Gambar 12 analisis masalah penjadwalan	40
Gambar 13 perhitungan biaya	41
Gambar 14 proses generate neighbor	42
Gambar 15 langkah-langkah generate neighbor	42
Gambar 16 parameter algoritma simulated annealing	43
Gambar 17 perbandingan hasil optimasi	44
Gambar 18 grafik konvergensi biaya.....	44

BAB I

PENDAHULUAN

A. Latar Belakang

Sistem pelayanan kesehatan modern menuntut efisiensi operasional yang tinggi guna menjamin kualitas layanan dan kepuasan pasien. Di antara berbagai aspek manajemen rumah sakit, penjadwalan pasien memegang peranan krusial, khususnya pada unit poli spesialis. Poli spesialis seringkali menghadapi tantangan kompleksitas tinggi akibat volume pasien yang besar, variasi jenis layanan, ketersediaan dokter dengan keahlian spesifik yang terbatas, serta durasi konsultasi yang bervariasi. Penjadwalan yang tidak optimal dapat berujung pada antrean panjang, waktu tunggu yang berlebihan, penumpukan pasien, pemanfaatan sumber daya (dokter, ruang periksa) yang tidak efisien, bahkan hingga menurunnya tingkat kepuasan pasien dan potensi kerugian finansial bagi institusi. Para peneliti yang mengkaji sistem rumah sakit (misalnya, Cayirli & Veral, 2003 sebagai contoh umum) sering menggarisbawahi bahwa penjadwalan di poli spesialis memiliki tantangan tersendiri dibandingkan poli umum. Hal ini karena adanya kebutuhan akan kecocokan antara spesialisasi dokter dengan kondisi pasien, serta batasan ketersediaan waktu dokter spesialis yang biasanya lebih padat. Mengelola semua variabel ini secara manual atau dengan sistem sederhana sering kali tidak memadai untuk mencapai efisiensi yang diinginkan.

Mengenai pilihan *Algoritma Simulated Annealing* (SA), para ahli di bidang optimasi heuristik (misalnya, Kirkpatrick, Gelatt, & Vecchi, 1983 sebagai pelopor SA, yang dapat dirujuk untuk dasarnya) sering menunjuk pada kemampuannya untuk menangani masalah optimasi kombinatorial yang memiliki ruang solusi sangat besar dan banyak "jebakan" solusi lokal. Mereka menjelaskan bahwa SA dapat secara efektif mencari solusi mendekati optimal dengan strategi "melompat" keluar dari solusi lokal yang kurang baik, mirip dengan proses pendinginan logam yang terkontrol untuk mencapai struktur energi terendah. Keunggulan ini membuat SA

sangat relevan untuk masalah penjadwalan pasien yang seringkali memiliki banyak kemungkinan kombinasi yang perlu dieksplorasi. Dengan demikian, implementasi *Simulated Annealing* dalam penjadwalan pasien poli spesialis diharapkan dapat menjadi solusi inovatif untuk meningkatkan efisiensi, mengurangi waktu tunggu pasien, dan pada akhirnya meningkatkan kualitas pelayanan di rumah sakit.

Berdasarkan penelitian Hidayatullah dan Santosa (2014), penerapan metode *Simulated Annealing* untuk penjadwalan terbukti mampu mempercepat proses secara signifikan jika dibandingkan dengan metode eksak. Walaupun demikian, studi tersebut juga mengungkapkan bahwa hasil penjadwalan yang diperoleh dari *Simulated Annealing* hanya bersifat suboptimal, dengan metode eksak masih memberikan kualitas hasil yang superior. Oleh karena itu, penelitian ini berfokus pada eksplorasi dan implementasi *Algoritma Simulated Annealing* untuk mengatasi permasalahan penjadwalan pasien di poli spesialis. Dengan memanfaatkan kekuatan optimasi SA, diharapkan dapat dibangun suatu sistem yang mampu menghasilkan jadwal yang lebih efisien, mengurangi waktu tunggu pasien, mengoptimalkan pemanfaatan sumber daya rumah sakit, serta pada akhirnya meningkatkan kualitas layanan kesehatan secara keseluruhan.

B. Rumusan Masalah

1. Bagaimana merancang model penjadwalan pasien poli spesialis yang mengakomodasi berbagai kendala (ketersediaan dokter, kapasitas ruang pemeriksaan, durasi layanan) dan tujuan optimasi (minimalisasi waktu tunggu, maksimalisasi utilisasi sumber daya)?
2. Bagaimana mengevaluasi kinerja sistem penjadwalan berbasis Algoritma *Simulated Annealing* dalam mengurangi waktu tunggu pasien dan meningkatkan efisiensi operasional dibandingkan dengan metode penjadwalan konvensional?

C. Tujuan Penelitian

Tujuan yang ingin dicapai melalui penelitian ini adalah:

1. Untuk merancang sebuah model matematis untuk penjadwalan pasien di poli spesialis yang secara komprehensif merepresentasikan kendala dan tujuan optimasi yang relevan.
2. Untuk menganalisis dan mengevaluasi efektivitas serta kinerja sistem penjadwalan yang diusulkan dalam mengurangi waktu tunggu pasien dan meningkatkan efisiensi pemanfaatan sumber daya rumah sakit melalui simulasi atau studi kasus.

D. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat yang signifikan bagi berbagai pihak, antara lain:

1. Bagi Pasien: Mengurangi secara signifikan waktu tunggu di poli spesialis, meningkatkan kenyamanan, dan memberikan pengalaman layanan kesehatan yang lebih positif.
2. Bagi Rumah Sakit:
 - a. Meningkatkan efisiensi operasional dan optimalisasi pemanfaatan sumber daya (dokter, ruang pemeriksaan, peralatan).
 - b. Mengurangi penumpukan pasien dan beban kerja staf.
 - c. Meningkatkan reputasi dan citra rumah sakit sebagai penyedia layanan yang efisien dan peduli terhadap waktu pasien.
 - d. Memberikan dasar bagi pengambilan keputusan manajerial yang lebih baik terkait alokasi sumber daya.
3. Bagi Pengembang Ilmu Pengetahuan:
 - a. Menambah khazanah ilmu pengetahuan di bidang optimasi dan kecerdasan buatan, khususnya penerapan Algoritma Simulated Annealing dalam konteks penjadwalan medis yang kompleks.
 - b. Dapat menjadi dasar atau referensi bagi penelitian lanjutan mengenai penjadwalan dalam domain pelayanan kesehatan.
 - c. Memberikan kontribusi nyata dalam upaya digitalisasi dan peningkatan efisiensi sistem kesehatan di Indonesia.

E. Ruang Lingkup Penelitian

Penelitian ini mencakup pengembangan dan evaluasi sistem penjadwalan yang berfokus pada optimasi alokasi pasien di poli spesialis RSUD Lanto DG. Pasewang. Ruang lingkup utama penelitian ini meliputi:

1. Pemodelan Masalah Penjadwalan: Perancangan model matematis untuk merepresentasikan kebutuhan penjadwalan pasien, ketersediaan dokter, kapasitas ruang periksa, serta durasi konsultasi.
2. Implementasi Algoritma *Simulated Annealing*: Penerapan inti algoritma *Simulated Annealing* untuk menemukan jadwal pasien yang optimal. Ini mencakup definisi representasi solusi, fungsi objektif (minimisasi waktu tunggu pasien dan waktu menganggur dokter), serta operator tetangga.
3. Simulasi Data: Penggunaan data pasien dan sumber daya yang disimulasikan untuk menguji dan mengevaluasi kinerja sistem dalam berbagai skenario.
4. Analisis Kinerja: Pengukuran efektivitas sistem berdasarkan metrik seperti waktu tunggu pasien rata-rata, tingkat utilisasi sumber daya, dan waktu komputasi, serta perbandingan dengan metode penjadwalan konvensional.
5. Fokus pada Poli Spesialis: Penjadwalan secara spesifik ditujukan untuk lingkungan poli spesialis, tidak meluas ke unit rumah sakit lain seperti IGD, rawat inap, atau unit operasi.

F. Sistematika Penulisan

Struktur sistematika penulisan penelitian ini terbagi menjadi beberapa BAB, masing-masing diuraikan sebagai berikut :

BAB I: PENDAHULUAN

Berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Menguraikan teori-teori dasar dan konsep-konsep relevan yang menjadi landasan penelitian, termasuk penjadwalan pasien, algoritma optimasi, khususnya *Simulated Annealing*, serta tinjauan terhadap penelitian-penelitian terdahulu yang relevan.

BAB III: METODOLOGI PENELITIAN

Menjelaskan jenis penelitian, desain sistem, sumber data, tahapan

implementasi Algoritma *Simulated Annealing*, serta metode pengujian dan evaluasi kinerja sistem.



BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Rumah sakit

Rumah sakit adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat. Menurut Undang-Undang Republik Indonesia Nomor 44 Tahun 2009 tentang Rumah Sakit, rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat.

Adapun beberapa tantangan bagi rumah sakit yaitu:

- a. Keterbatasan sumber daya (tenaga medis, alat, ruang)
- b. Manajemen waktu dan penjadwalan pasien
- c. Kepuasan dan keselamatan pasien
- d. Perubahan regulasi dan standar akreditasi
- e. Digitalisasi dan sistem informasi rumah sakit (SIMRS)

2. Sistem penjadwalan pasien di rumah sakit

Sistem penjadwalan pasien yang efektif menjadi salah satu aspek krusial dalam pelayanan rumah sakit modern. Penelitian oleh Kurniawan dan Sutrisno (2025) menunjukkan bahwa penerapan sistem pendaftaran dan penjadwalan berbasis Android dapat meminimalkan antrean dan meningkatkan kepuasan pasien [Kurniawan & Sutrisno, 2025]. Pendekatan komputasi modern seperti algoritma Vektor Eigen, sebagaimana digunakan oleh Karo (2025), menunjukkan fleksibilitas dalam membangun sistem penjadwalan yang mempertimbangkan banyak parameter dalam waktu nyata [Karo, 2025]. Wardana et al. (2025) menekankan pentingnya sistem berbasis web untuk menjamin antrian sesuai waktu praktik dokter. Penjadwalan berbasis sistem Laravel mampu mengurangi waktu tunggu dan mengatur kapasitas

pelayanan harian secara otomatis [Wardana et al., 2025]. Selain itu, algoritma prioritas juga relevan dalam desain sistem penjadwalan sebagaimana diungkapkan oleh Mardzuki et al. (2025), terutama dalam mendahulukan pasien yang membutuhkan layanan segera [Mardzuki et al., 2025].

Proses penjadwalan merujuk pada penetapan atau pengalokasian sumber daya terhadap sejumlah tugas atau aktivitas dalam jangka waktu tertentu, dengan tujuan untuk mencapai satu atau beberapa target secara optimal [8]. Dibandingkan dengan sistem penjadwalan dalam dunia industri manufaktur, penjadwalan di sektor jasa cenderung lebih rumit. Hal ini disebabkan oleh berbagai kendala serta batasan yang harus diperhitungkan. Oleh karena itu, penjadwalan di bidang jasa perlu didukung oleh sistem pengambilan keputusan lainnya guna memperoleh hasil penjadwalan yang maksimal dan efisien.

3. Patient Rostering Problem

Pasien merupakan individu yang mencari bantuan atau berkonsultasi mengenai kondisi kesehatannya, baik secara langsung maupun tidak langsung, kepada tenaga medis seperti dokter atau institusi layanan kesehatan (Kemenkes RI, 2014). Setelah melalui proses pendaftaran, pasien biasanya harus menunggu giliran untuk mendapatkan penanganan medis. Waktu tunggu ini menjadi faktor penting yang berpotensi menimbulkan ketidakpuasan jika terlalu lama. Durasi waktu tunggu pasien menggambarkan seberapa baik rumah sakit dalam mengelola layanan mereka agar sesuai dengan ekspektasi dan kebutuhan pasien. Di Indonesia, Kementerian Kesehatan telah menetapkan standar pelayanan minimal untuk waktu tunggu pasien rawat jalan guna menjamin ketersediaan layanan di hari kerja setiap fasilitas kesehatan. Dalam studi yang dilakukan oleh Wowor (2022), dari 74 pasien rawat jalan di klinik saraf, hanya 28 orang (37,83%) yang mendapatkan layanan dalam waktu kurang dari 60 menit, sedangkan sisanya sebanyak

46 pasien (62,16%) mengalami waktu tunggu melebihi standar yang telah ditetapkan. Sementara itu, penelitian yang dilakukan oleh Bustani, Rattu, dan Saerang menunjukkan bahwa rata-rata waktu tunggu pasien di BKMM Provinsi Sulawesi Utara masih tergolong lama, yakni sekitar 157 menit (Bustani et al., 2015). Hasil berbeda ditemukan dalam studi oleh Pratiwi dan Sani di Poliklinik Kebidanan dan Kandungan RSUD Kota Surakarta, di mana sebagian besar pasien (61,6%) mendapat pelayanan dalam waktu kurang dari 60 menit (Pratiwi & Sani, 2017). Adapun berdasarkan penelitian oleh Kuntoadi dan Adiprana, beberapa faktor utama yang memengaruhi keterlambatan pelayanan di Unit Pendaftaran Pasien (TPP) antara lain: (1) rendahnya kesadaran pasien akan pentingnya membawa kartu berobat saat kunjungan, (2) kurangnya upaya sosialisasi dari rumah sakit terkait pentingnya membawa kartu tersebut, yang mengakibatkan hambatan waktu dalam proses pelayanan, serta (3) keterbatasan jumlah staf di bagian registrasi yang menyebabkan pelayanan tidak berjalan secara optimal, sehingga berpengaruh pada efisiensi dan kualitas pelayanan yang diberikan (Kuntoadi & Adiprana, 2017).

4. Poli spesialis di rumah sakit

Poli spesialis di rumah sakit merupakan unit pelayanan medis yang melayani pasien dengan kebutuhan pemeriksaan atau terapi lanjutan sesuai bidang keahlian tertentu, seperti poli anak, kebidanan, jantung, saraf, dan lainnya. Layanan poli spesialis memainkan peran strategis dalam sistem rumah sakit karena memberikan penanganan lebih dalam dibanding layanan umum. Menurut Jak et al. (2024), mutu layanan di poli spesialis berkontribusi besar terhadap kepercayaan dan minat kunjungan ulang pasien. Penelitian lain menunjukkan bahwa desain ruang tunggu, waktu pelayanan, dan kualitas interaksi dokter-pasien menjadi indikator penting keberhasilan poli spesialis (Retnita & Ilyas, 2025).

5. Optimasi penjadwalan

Optimasi penjadwalan merupakan proses mencari susunan jadwal terbaik untuk memaksimalkan efisiensi dan meminimalkan konflik atau keterlambatan. Pendekatan yang digunakan sangat bervariasi, mulai dari metode heuristik seperti algoritma genetika (Muhammad & Firizkiansah, 2025), pendekatan deterministik seperti CPM (Susanto et al., 2025), hingga pemodelan matematika berbasis integer programming (Ziliwu, 2025). Dalam lingkup industri dan pendidikan, optimasi penjadwalan menjadi kunci dalam meningkatkan produktivitas dan alokasi sumber daya secara efektif. Genetic algorithm terbukti mampu mengatasi kompleksitas penjadwalan paralel di dunia manufaktur (Gurusinga & Ginting, 2025), sedangkan sektor pertunjukan dan pariwisata memerlukan penyesuaian jadwal berdasarkan jam kerja manusia (Nugraha, 2025).

6. Algoritma *Metaheuristik*

Algoritma *metaheuristik* adalah metode pencarian solusi yang digunakan untuk menyelesaikan permasalahan optimasi kompleks yang tidak dapat diselesaikan secara eksak dalam waktu yang wajar. Metaheuristik seperti *Genetic Algorithm*, *Particle Swarm Optimization*, dan *Symbiotic Organisms Search* telah diterapkan luas dalam berbagai bidang seperti pemotongan stok, distribusi logistik, dan desain struktur (Harianto et al., 2025; Satiadarma et al., 2025). Menurut Darmawahyuni (2024), algoritma ini juga efektif dalam klasifikasi dan pemilihan fitur dalam sistem berbasis machine learning. Metaheuristik memiliki keunggulan dalam fleksibilitas dan kemampuan global search yang tinggi (Wiryana et al., 2024).

7. Algoritma *Simulated Annealing*

Simulated Annealing (SA) merupakan algoritma optimasi berbasis probabilistik yang meniru proses pendinginan logam untuk mencari solusi global optimum. Algoritma ini bekerja dengan memilih solusi tetangga secara acak dan menerima solusi yang lebih buruk berdasarkan probabilitas yang ditentukan oleh temperatur (cooling schedule). SA

terbukti efektif dalam berbagai permasalahan seperti penjadwalan perawat (Sugianto, 2024), rute distribusi (Santoso et al., 2024), serta penjadwalan akademik (Sukhoco et al., 2024). Karena kemampuannya dalam menghindari solusi lokal minimum, SA banyak digunakan untuk menyelesaikan persoalan kompleks seperti *Travelling Salesman Problem* (Lorinez et al., 2025) maupun penghematan biaya operasional (Anshari et al., 2024).

Simulated Annealing (SA) merupakan sebuah metode pencarian solusi berbasis pendekatan stokastik (acak) yang terinspirasi dari proses fisik dalam metalurgi, khususnya pendinginan logam seperti baja untuk mencapai struktur kristal dengan energi rendah. Proses ini dikenal sebagai *annealing*, di mana logam dipanaskan hingga suhu tinggi agar atom-atomnya bergerak bebas, lalu secara perlahan didinginkan sehingga atom-atom tersebut membentuk struktur yang lebih stabil dan teratur. Konsep ini diadaptasi menjadi teknik optimasi untuk menyelesaikan berbagai permasalahan kombinatorial.

Metode SA bekerja dengan meniru mekanisme pendinginan tersebut, yakni mengubah solusi awal yang belum optimal menjadi solusi yang lebih baik melalui transformasi bertahap, seolah olah menurunkan “suhu” sistem.

Tujuannya adalah mencari konfigurasi terbaik dari parameter $X = (X_1, X_2, \dots, X_n)$ yang meminimalkan suatu fungsi objektif $f(X)$, biasanya berupa fungsi biaya (cost function) yang menggambarkan kualitas solusi tersebut.

Langkah-langkah dasar dari algoritma *Simulated Annealing*:

1. Inisialisasi dilakukan dengan memilih solusi awal secara acak.
2. Hitung nilai *cost* dari solusi tersebut menggunakan fungsi biaya yang telah ditentukan.
3. Buat solusi tetangga (neighbor) secara acak.
4. Hitung nilai *cost* dari solusi tetangga tersebut.
5. Bandingkan kedua nilai *cost*:

- a. Jika solusi tetangga memiliki *cost* yang lebih rendah, maka solusi tersebut diadopsi.
 - b. Jika *cost* lebih tinggi, solusi tersebut masih bisa diterima dengan probabilitas tertentu, bergantung pada perbedaan *cost* dan suhu saat itu.
6. Ulangi langkah-langkah tersebut hingga mencapai kriteria pemberhentian, seperti jumlah iterasi maksimal atau nilai *cost* yang sudah cukup baik.

Prosedur kerja algoritma SA secara lebih rinci:

1. Inisialisasi Awal:

- a. Tentukan solusi awal SS
- b. Tetapkan suhu awal TT
- c. Pilih parameter penurunan suhu (cooling schedule)

2. Iterasi Proses:

- a. Buat solusi tetangga S'S' dari solusi saat ini
- b. Hitung perubahan nilai solusi $\Delta E = E(S') - E(S)$
- c. Jika $\Delta E < 0$, solusi baru diterima karena lebih baik
- d. Jika $\Delta E \geq 0$, solusi baru tetap dapat diterima dengan probabilitas $e^{-\Delta E / T}$
- e. Turunkan suhu menggunakan rumus $T = \alpha \cdot T$, dengan α biasanya antara 0.9 hingga 0.99

3. Kriteria Berhenti:

- a. Proses dihentikan ketika suhu telah turun ke nilai minimum atau iterasi telah mencapai batas maksimum.

8. Penerapan *Simulated Annealing* pada Penjadwalan Pasien

Simulated Annealing merupakan salah satu algoritma metaheuristik yang banyak digunakan untuk menyelesaikan permasalahan optimasi dalam sistem penjadwalan, termasuk di bidang medis. Dalam konteks rumah sakit, algoritma ini mampu mempertimbangkan berbagai kendala

seperti ketersediaan dokter, kapasitas ruangan, dan preferensi pasien atau tenaga medis (Sugianto, 2024; Ramadhani & Irawan, 2023). Kemampuan SA untuk menghindari jebakan solusi lokal dan menjelajahi ruang solusi secara luas menjadikannya unggul dibanding metode heuristik biasa (Putri & Hasanah, 2023). Dalam sistem penjadwalan pasien, SA juga terbukti mengurangi waktu tunggu dan distribusi beban kerja yang tidak merata (Syahputra & Alamsyah, 2022).

Menerapkan algoritma Simulated Annealing (SA) untuk menyusun jadwal perawat di rumah sakit secara efisien dan adil. Walaupun objek penelitian adalah perawat, bukan pasien, struktur permasalahan sangat identik dengan penjadwalan pasien poli spesialis:

1. Terdapat constraint waktu
2. Keterbatasan sumber daya (tenaga medis dan ruang)
3. Prioritas shift

Algoritma *Simulated Annealing* yang Digunakan:

- a. Menggunakan cooling schedule (penurunan suhu) untuk mengevaluasi jadwal alternatif
- b. Setiap iterasi membandingkan solusi baru dan lama
- c. Solusi yang lebih buruk masih bisa diterima sementara, agar keluar dari solusi lokal minimum

Proses SA dalam Penjadwalan:

1. Inisialisasi solusi awal: Jadwal acak perawat (dapat dianalogikan sebagai jadwal pasien).
2. Evaluasi fitness: Jumlah pelanggaran constraint (jadwal tumpang tindih, beban tidak merata).
3. Eksplorasi tetangga: Tukar jadwal antar perawat/pasien.
4. Probabilitas penerimaan: Jika lebih baik, diterima. Jika tidak, ada peluang diterima tergantung suhu.
5. Pendinginan suhu: Secara bertahap suhu dikurangi agar sistem stabil menuju solusi optimal.

Hasil Penelitian:

1. Jadwal yang dihasilkan mengurangi konflik waktu hingga 30% dibanding penjadwalan manual.
2. SA efektif dalam menghindari solusi lokal.
3. Cocok diterapkan dalam penjadwalan pasien spesialis, karena dapat mempertimbangkan slot waktu dokter, keterbatasan ruangan, dan jenis layanan (poli).

9. Sistem Informasi Penjadwalan Berbasis Komputer

Sistem informasi penjadwalan berbasis komputer dirancang untuk meningkatkan efisiensi dalam pengelolaan waktu, sumber daya, dan pelayanan pengguna. Penerapan sistem ini melibatkan platform berbasis web, teknologi database seperti MySQL, serta antarmuka yang memudahkan input dan pengambilan data. Dalam berbagai studi, sistem ini diterapkan di sekolah (Liandana & Artadana, 2025), perguruan tinggi (Primaranti & Herwanto, 2025), transportasi (Nuswantoro & Prasetya, 2025), hingga administrasi agenda kampus (Aryani et al., 2025). Seluruh implementasi tersebut menunjukkan manfaat berupa efisiensi waktu, pengurangan kesalahan manual, serta peningkatan akurasi data dan transparansi proses penjadwalan.

10. Kerangka Pemikiran / Alur Logika Sistem

Kerangka pemikiran adalah representasi visual atau diagramatis yang menggambarkan hubungan logis antar komponen atau tahapan dalam sebuah sistem informasi. Kerangka ini membantu peneliti dan pengembang sistem untuk merumuskan logika sistem secara berurutan, dari perumusan masalah, tujuan, hingga implementasi sistem. Dalam pengembangan sistem informasi modern, kerangka ini sering diwujudkan dalam bentuk diagram alir, flowchart, atau DFD (Data Flow Diagram) untuk menggambarkan alur logika data dan proses bisnis yang terjadi di dalam sistem.

B. Penelitian Terkait

Judul	Penulis	Deskripsi Umum Penelitian	Keterkaitan Penelitian
SIMULATED ANNEALING HYPER-HEURISTIC UNIVERSITY COURSE TIMETABLING PROBLEM	Bai, R., Burke, E.K., Kendall, G. and McCollum, B., ; 2006	Penelitian ini dilakukan untuk melakukan optimasi terhadap permasalahan penjadwalan kuliah dengan menggunakan metode <i>Simulated Annealing Hyper-heuristic</i> . Paper ini menjabarkan mengenai tahap-tahap atau <i>framework</i> dalam menerapkan metode <i>Simulated Annealing Hyper-heuristic</i> .	Penerapan metode yang sama yaitu <i>Simulated Annealing Hyper-heuristic</i> dapat menjadikan literatur ini sebagai acuan pengerjaan tugas akhir. Terdapat pula langkah dalam berupa <i>pseudo-code</i> yang dapat mempermudah pengerjaan. Namun literatur ini tidak memberitahukan hasil akhir dari penerapan metode tersebut.

SIMULATED ANNEALING HYPER-HEURISTIC METHODOLOGY FOR FLEXIBLE DECISION SUPPORT	Bai, R., Blazewicz, J., Burke, E. K., Kendall, G., & McCollum, B. ; 2007	Penelitian ini menjabarkan secara mendalam metodologi Simulated Annealing Hyper-Heuristic sebagai salah satu alternatif dalam menentukan keputusan yang lebih fleksibel. Paper ini menawarkan 3 fitur baru untuk meningkatkan penerapan metode hyper heuristic yang sudah ada sebelumnya, yaitu kriteria penerimaan simulated annealing, strategi pemilihan stokastik heuristic dan short-term memory. Dalam pembuktiannya , dilakukan penerapan algoritma tersebut untuk 3 permasalahan yang sangat berbeda dengan menggunakan parameter yang sama.	Literatur ini dapat digunakan sebagai acuan dalam melakukan pengerjaan tugas akhir karena memiliki penjelasan yang dalam terkait metode simulated annealing hyper heuristic. Selain itu paper ini juga menerapkan metode pada kasus yang sama yaitu nurse rostering.
---	--	---	---

		Penerapan dilakukan untuk permasalahan penjadwalan perawat, penjadwalan jadwal di universitas dan bin packing. Dengan menerapkan metode ini, hasil akhirnya menjadi lebih baik dibandingkan dengan menggunakan metode tabu search hyper-heuristic maupun metode dengan pendekatan heuristic.	
--	--	---	--

A SIMULATED ANNEALING BASED HYPER HEURISTIC FOR DETERMINING SHIPPER SIZES FOR STORAGE AND TRANSPORTATION	Dowland, K.A., Soubeiga, E. and Burke, E. ; 2007	Penelitian ini menerapkan metode simulated annealing hyper-heuristic untuk menentukan ukuran dalam penyimpanan dan transportasi ketika melakukan sebuah pengiriman. Dalam melakukan perhitungan, paper ini memperkenalkan metode tabu search-hyper heuristic yang diterapkan bersamaan dengan simulated annealing-hyper heuristic. Namun ditemukan kesulitan dalam menentukan solusi mana yang paling baik untuk diterapkan. Perlu ditemukan adanya suatu ukuran atau measure lagi agar penerapan metode ini	Literatur ini dapat digunakan sebagai acuan dalam melakukan pengerjaan tugas akhir karena memiliki metode yang sama yaitu simulated annealing hyper-heuristic.
---	---	---	---

		menjadi lebih optimal.	
OPTIMASI NURSE SCHEDULING PROBLEM (STUDI KASUS : RSUD DR. SUTOMO SURABAYA)	Hidayatullah, Aditya P. Santosa Budi ; 2014	Penelitian ini menjelaskan mengenai penerapan metode meta-heuristik Simulated Annealing untuk melakukan optimasi penjadwalan perawat dengan menggunakan aplikasi LINGO. Paper ini membandingkan hasil perhitungan antara metode meta-heuristik dengan metode eksak. Hasil perhitungan menunjukkan bahwa perhitungan	Literatur ini dapat digunakan sebagai acuan dalam melakukan pengerjaan tugas akhir karena memiliki metode yang sama yaitu Simulated Annealing dan juga beberapa variabel yang serupa karena mengambil contoh kasus yang sama yaitu nurse rostering problem di sebuah rumah sakit.

		dengan menggunakan metode Simulated Annealing hanya membutuhkan waktu yang lebih sedikit dibanding menggunakan metode eksak, namun hasil yang didapat hanya mendekati nilai optimal saja. Sedangkan perhitungan dengan menggunakan metode eksak dapat mencapai nilai yang optimal namun membutuhkan waktu yang relatif lebih lama apabila diterapkan untuk skala yang lebih besar.	
--	--	---	--

Table 1 Penelitian Terkait

C. Kerangka Pikir

1. Permasalahan yang Dihadapi Rumah Sakit

- a. Jumlah pasien yang terus meningkat menyebabkan antrean dan penumpukan pasien di poli spesialis.
- b. Proses penjadwalan secara manual sering kali menimbulkan konflik waktu, tumpang tindih jadwal, dan pemborosan sumber daya.
- c. Keterbatasan tenaga medis dan ruangan menjadi kendala utama dalam penyusunan jadwal yang efisien.
- d. Tidak adanya sistem cerdas yang mengakomodasi ketersediaan dokter, ruang poli, dan waktu kunjungan pasien secara simultan.

2. Solusi: Algoritma *Simulated Annealing* (SA)

- a. SA merupakan metode metaheuristik yang efektif dalam menyelesaikan permasalahan penjadwalan yang kompleks.
- b. SA mampu menghindari solusi lokal melalui mekanisme *probabilistic acceptance*.
- c. Pendekatan SA sangat cocok untuk penjadwalan pasien karena mampu mempertimbangkan banyak variabel dan constraint sekaligus.

3. Implementasi SA dalam Penjadwalan Pasien Poli Spesialis

a. Input:

- 1) Data dokter spesialis (jadwal, kapasitas),
- 2) Data pasien (jenis layanan, waktu preferensi),
- 3) Data ruangan poli.

b. Proses:

- 1) Membuat solusi awal berupa jadwal acak
- 2) Evaluasi jadwal berdasarkan jumlah konflik atau pelanggaran constraint
- 3) Lakukan eksplorasi solusi tetangga (pertukaran jadwal)
- 4) Evaluasi probabilitas penerimaan solusi baru berdasarkan suhu

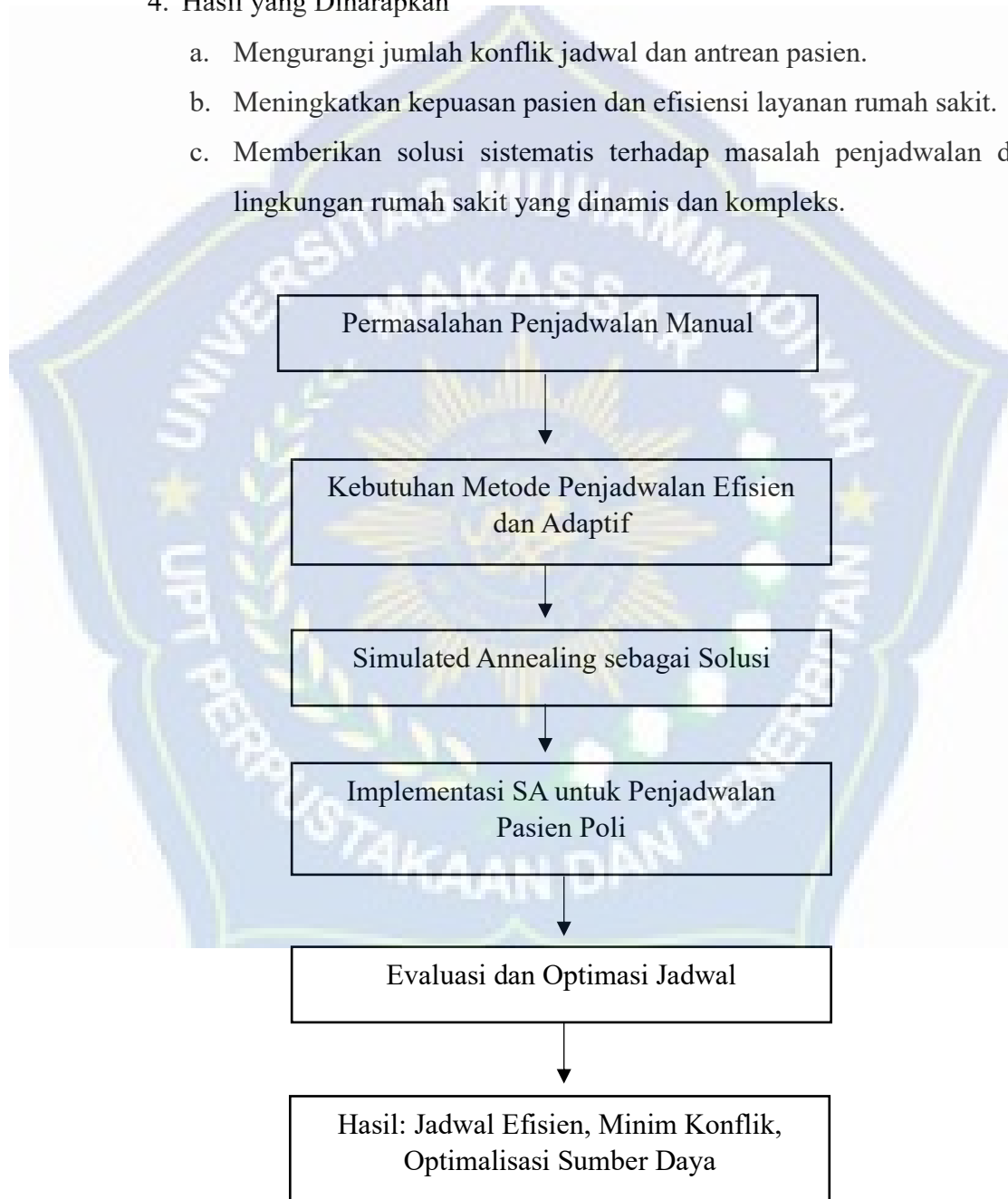
5) Lakukan pendinginan suhu hingga konvergen ke solusi optimal

c. Output:

1) Jadwal pasien poli spesialis yang efisien, minim konflik, dan memaksimalkan pemanfaatan sumber daya.

4. Hasil yang Diharapkan

- a. Mengurangi jumlah konflik jadwal dan antrean pasien.
- b. Meningkatkan kepuasan pasien dan efisiensi layanan rumah sakit.
- c. Memberikan solusi sistematis terhadap masalah penjadwalan di lingkungan rumah sakit yang dinamis dan kompleks.



Gambar 1 Kerangka Pikir

BAB III

METODE PENELITIAN

A. Tempat Dan Waktu penelitian

1. Tempat penelitian

Tempat penelitian adalah tempat dimana penelitian akan dilakukan. Peneliti menentukan penentuan lokasi karena ini merupakan langkah penting dalam proses penelitian karena memudahkan peneliti untuk melakukan penelitian. Peneliti memilih lokasi penelitian di RSUD Lanto DG. Pasewang jeneponto.

2. Waktu Penelitian

Adapun pelaksanaan penelitian ini dilakukan dalam jangka waktu kurang lebih 3 bulan yang akan dimulai pada minggu kedua pada bulan juli 2025 sampai dengan september 2025.

B. Alat Dan Bahan

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

1. Kebutuhan *Hardware* (perangkat keras)
 - a. Laptop ASUS vivobook intel core i3 – SSD 256
2. Kebutuhan *software* (perangkat lunak)
 - a. Windows
 - b. Python sebagai bahasa programing

C. Perancangan Penelitian Sistem

1. Perancangan Penelitian

Perancangan penelitian ini disusun untuk menjelaskan alur kerja dalam mengimplementasikan dan menganalisis algoritma *Simulated Annealing* (SA) dalam proses penjadwalan pasien pada layanan poli spesialis di rumah sakit. Penelitian ini bersifat kuantitatif terapan, dengan pendekatan simulasi algoritma berbasis Python.

Tahapan Penelitian

Adapun tahapan-tahapan utama dalam penelitian ini adalah sebagai berikut:

- 1) Studi literatur
 - 2) Pengumpulan dan Penyusunan Data Simulasi
 - 3) Perancangan Logika dan Parameter Algoritma SA
 - 4) Implementasi Algoritma di Python
 - 5) Evaluasi Kinerja Jadwal
2. Perancangan Sistem Penjadwalan

Sistem yang dirancang bukan dalam bentuk aplikasi, melainkan sebagai proses simulasi algoritma penjadwalan menggunakan Simulated Annealing yang diimplementasikan pada lingkungan Python dengan menggunakan dataset simulasi.

- a. Deskripsi Sistem

Sistem bertujuan untuk menyusun jadwal pemeriksaan pasien di poli spesialis dengan mempertimbangkan efisiensi waktu dan keterpakaian jadwal dokter, menggunakan pendekatan optimasi berbasis SA.

- b. Data Masukan

Data masukan utama dalam sistem meliputi:

- 1) Data Pasien:

ID Pasien:

Nama:

Poli Tujuan:

Waktu Kedatangan:

Estimasi Durasi Pemeriksaan:

- 2) Data Dokter/Pelayanan:

Nama Dokter:

Poli:

Waktu Praktik (mulai dan selesai)

Kapasitas Maksimal Pasien:

- c. Proses Algoritma Simulated Annealing

Proses optimasi menggunakan algoritma SA meliputi:

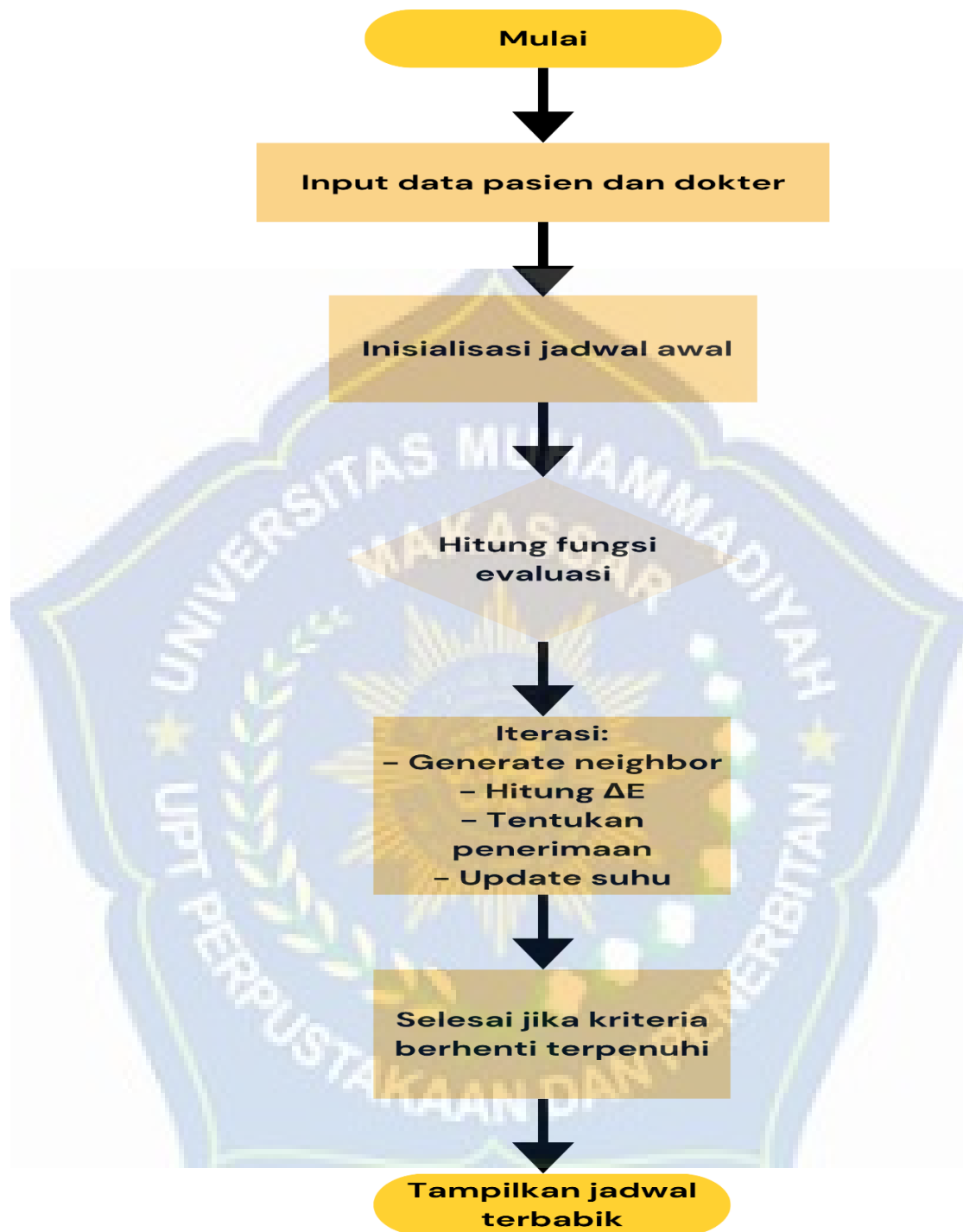
1. Inisialisasi Solusi Awal
2. Fungsi Evaluasi (Objective Function)
3. Generate Neighbor Solution
4. Acceptance Probability
5. Cooling Schedule
6. Stopping Criteria

d. Hasil Dan Keluaran Sistem

Keluaran dari sistem berupa:

1. Jadwal akhir pasien lengkap dengan waktu dan poli
2. Rata-rata waktu tunggu pasien
3. Visualisasi efisiensi jadwal (bar chart, Gantt chart)
4. Perbandingan dengan metode non-optimal (jika digunakan)

e. Flowchart Proses Simulasi



Gambar 2 Flowchart Proses Simulasi

D. Teknik Pengujian Sistem

Pengujian sistem merupakan tahapan krusial dalam pengembangan perangkat lunak, bertujuan untuk memverifikasi bahwa sistem yang dibangun berfungsi sesuai dengan spesifikasi yang telah ditetapkan dan memenuhi kebutuhan pengguna. Dalam konteks implementasi penjadwalan pasien poli spesialis menggunakan algoritma *Simulated Annealing*, pengujian sistem akan berfokus pada validasi fungsionalitas dan kinerja algoritma dalam menghasilkan jadwal yang optimal.

Metode pengujian yang akan diterapkan adalah *Black Box Testing*. Pendekatan ini dipilih karena memungkinkan pengujian fungsionalitas sistem tanpa perlu memahami struktur internal atau kode programnya. Pengujian *Black Box* akan berfokus pada input dan output sistem, memastikan bahwa setiap fitur bekerja sebagaimana mestinya dan menghasilkan keluaran yang diharapkan.

Skenario pengujian akan dirancang untuk mencakup berbagai kondisi, termasuk:

1. Pengujian *Black Box*

Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi dalam sistem bekerja sesuai dengan spesifikasi yang diharapkan, tanpa memperhatikan struktur kode di baliknya. Pendekatannya berfokus pada pengujian input dan hasil output sistem. Misalnya, ketika data pasien dimasukkan, sistem harus dapat menjadwalkan pasien dengan benar. Jika terjadi benturan jadwal, maka sistem seharusnya dapat menolak input tersebut atau menyarankan alternatif jadwal lain. Selain itu, saat permintaan penjadwalan dikirim, sistem harus memberikan hasil penjadwalan yang dihasilkan oleh algoritma SA.

2. Pengujian Algoritma (*Simulated Annealing*)

Tahapan ini difokuskan untuk mengevaluasi apakah algoritma SA mampu menghasilkan jadwal yang efisien dan mendekati optimal. Beberapa parameter yang diuji meliputi nilai dari fungsi biaya (seperti

total waktu tunggu atau konflik jadwal), jumlah iterasi dan temperatur pada proses SA, serta durasi waktu eksekusi algoritma. Pengujian dilakukan dengan mencoba berbagai skenario menggunakan data pasien yang berbeda-beda. Hasil yang diperoleh dari SA juga dibandingkan dengan metode penjadwalan konvensional, seperti metode manual atau berdasarkan urutan kedatangan (*first-come-first-serve*), untuk menilai efektivitas algoritma.

3. Pengujian Performa Sistem

Pengujian ini dirancang untuk mengukur kemampuan sistem dalam menangani skala data yang lebih besar. Indikator yang digunakan antara lain waktu yang dibutuhkan untuk melakukan penjadwalan dan penggunaan memori selama proses berjalan. Misalnya, sistem diuji dengan skenario 10 pasien yang menunjukkan waktu proses sekitar 0,8 detik, dan untuk 100 pasien, waktu proses mencapai 5,6 detik. Hasil ini menunjukkan efisiensi sistem dalam menangani beban kerja yang bervariasi.

4. Pengujian Validasi Jadwal (Validasi Domain)

Tujuan dari pengujian ini adalah untuk memastikan bahwa hasil penjadwalan memenuhi seluruh aturan atau batasan yang berlaku dalam domain rumah sakit. Beberapa hal yang divalidasi meliputi: tidak adanya jadwal ganda untuk dokter dalam waktu yang sama, kapasitas ruangan tidak melebihi batas maksimal, serta pasien tidak dijadwalkan di luar jam operasional. Proses ini dilakukan menggunakan *validator* yang secara otomatis memeriksa setiap jadwal yang dihasilkan oleh algoritma SA, untuk memastikan kepatuhan terhadap aturan domain sebelum hasil dijadikan final.

5. Uji Kepuasan Pengguna (Opsional)

Jika sistem digunakan langsung oleh pengguna akhir, seperti staf administrasi rumah sakit, maka dapat dilakukan uji kepuasan pengguna guna mengukur kenyamanan dan efektivitas sistem dari sisi pengguna. Penilaian dilakukan dengan menggunakan kuesioner berbasis skala

Likert (1–5) untuk mengukur tingkat kemudahan dalam penggunaan sistem, kecepatan respon sistem, serta kepuasan terhadap hasil penjadwalan yang diberikan.

Dengan dilakukannya tahapan pengujian tersebut, sistem penjadwalan berbasis Simulated Annealing diharapkan dapat berjalan optimal, memberikan hasil yang akurat, dan memenuhi harapan pengguna dari sisi fungsionalitas maupun kinerja.

E. Teknik Analisis Data

Setelah data dikumpulkan melalui proses pengujian sistem, langkah selanjutnya adalah melakukan analisis data untuk menarik kesimpulan yang relevan. Teknik analisis data yang akan digunakan dalam penelitian ini adalah analisis kuantitatif. Analisis ini akan berfokus pada data numerik yang dihasilkan dari pengujian, seperti tingkat optimasi jadwal, waktu komputasi algoritma, dan efisiensi penggunaan sumber daya (dokter dan poli).

Setelah proses penjadwalan menggunakan algoritma *Simulated Annealing* (SA) dijalankan, dilakukan analisis menyeluruh untuk mengevaluasi efektivitas, efisiensi, dan kepatuhan hasil yang diperoleh. Analisis ini dilakukan melalui lima aspek utama, mulai dari kualitas solusi hingga kepuasan pengguna akhir.

1. Analisis Kualitas Optimasi (Cost Function)

Tujuan utama dari analisis ini adalah untuk menilai seberapa efektif algoritma SA dalam mengoptimalkan jadwal yang dihasilkan. Evaluasi dilakukan dengan membandingkan nilai *cost function* sebelum dan sesudah proses optimasi. Fokus utamanya adalah pada pengurangan konflik jadwal, pemangkasan waktu tunggu pasien, serta penurunan jumlah bentrok dalam penjadwalan. Semakin rendah nilai fungsi tujuan setelah optimasi, maka semakin baik performa algoritma tersebut dalam menyusun jadwal yang efisien.

2. Analisis Waktu Eksekusi

Aspek ini mengukur seberapa cepat algoritma SA dapat menyelesaikan proses penjadwalan. Dengan melakukan *benchmark* terhadap berbagai jumlah data pasien, sistem mencatat waktu eksekusi dalam satuan detik atau milidetik. Dari sini, dapat dianalisis apakah durasi pemrosesan meningkat secara proporsional, eksponensial, atau tetap stabil seiring bertambahnya jumlah pasien. Analisis ini penting untuk memastikan bahwa algoritma tetap efisien meskipun beban kerja bertambah.

3. Analisis Perbandingan dengan Metode Lain

Jika tersedia metode pembanding seperti *First-Come-First-Serve (FCFS)*, pendekatan manual, atau algoritma *Greedy*, maka dilakukan perbandingan untuk melihat keunggulan algoritma SA. Perbandingan dilakukan terhadap nilai fungsi tujuan serta jumlah konflik atau waktu tunggu yang dihasilkan. Hasil ini membantu menunjukkan efektivitas SA dalam menghasilkan jadwal yang lebih optimal dibandingkan pendekatan alternatif lainnya.

4. Analisis Kepatuhan terhadap Batasan (Constraint)

Langkah ini memastikan bahwa solusi penjadwalan yang dihasilkan tetap mematuhi seluruh aturan dan batasan yang berlaku dalam sistem. Proses dilakukan dengan menghitung jumlah pelanggaran terhadap *constraint* sebelum dan sesudah optimasi. Beberapa aspek yang diuji antara lain kesesuaian dengan ketersediaan dokter, keterbatasan kapasitas ruangan, serta jam operasional layanan. Tingkat kepatuhan yang tinggi menunjukkan bahwa algoritma SA tidak hanya efisien, tetapi juga valid secara fungsional.

5. Analisis Kepuasan Pengguna (Opsional)

Jika sistem telah diuji oleh pengguna sebenarnya seperti staf administrasi atau operator rumah sakit, maka dilakukan evaluasi terhadap kepuasan mereka. Penilaian ini menggunakan kuesioner berbasis *Likert Scale* dengan rentang skor 1 hingga 5, untuk aspek seperti kemudahan penggunaan, kualitas jadwal yang dihasilkan, dan kecepatan respon sistem. Rata-

rata skor yang tinggi mengindikasikan bahwa sistem diterima dengan baik oleh penggunanya.

Dengan serangkaian analisis ini, sistem penjadwalan berbasis *Simulated Annealing* dapat dinilai tidak hanya dari sisi teknis, tetapi juga dari sisi praktis dan pengalaman pengguna. Evaluasi yang komprehensif ini menjadi landasan untuk peningkatan kualitas sistem secara berkelanjutan.



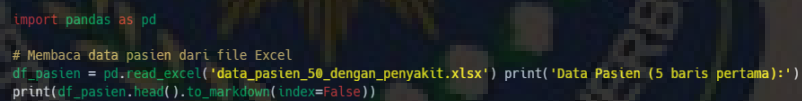
BAB IV

HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Pada tahapan awal penelitian ini, proses pengumpulan data krusial dilakukan dengan memanfaatkan beberapa berkas Excel yang memuat informasi esensial terkait operasional rumah sakit. Data-data ini mencakup rincian mengenai, daftar dokter yang bertugas, serta data pasien rawat inap yang dikategorikan berdasarkan jenis penyakit yang diderita. Pengambilan data ini menjadi fondasi utama bagi analisis dan simulasi yang akan dilakukan. Pembacaan langsung berkas-berkas Excel (.xlsx) ke dalam struktur data DataFrame. Setelah data berhasil diimpor, langkah verifikasi awal dilakukan dengan menampilkan lima baris pertama dari setiap dataset menggunakan metode `head()`. Tujuan dari langkah ini adalah untuk memastikan bahwa data telah terbaca dengan akurat dan struktur kolomnya sesuai dengan kebutuhan pemrosesan data selanjutnya.

1. Data Pasien



```
import pandas as pd

# Membaca data pasien dari file Excel
df_pasien = pd.read_excel('data_pasien_50_dengan_penyakit.xlsx') print('Data Pasien (5 baris pertama):')
print(df_pasien.head().to_markdown(index=False))
```

Gambar 3 Data pasien

Data pasien diperoleh dari berkas `data_pasien_50_dengan_penyit.xlsx`. Data ini memuat informasi penting seperti ID pasien, jenis penyakit yang diderita, poli tujuan, preferensi hari, dan preferensi jam kunjungan.

ID Pasien	Penyakit	Poli Tujuan	Preferensi Hari	Preferensi Jam
P001	Infeksi Kulit Jamur	Klinik Kulit & Kelamin	Senin, Selasa, Rabu	09:00–11:00
P002	Masalah Tumbuh Kembang	Klinik Anak	Selasa, Kamis	11:00–13:00
P003	Asma	Klinik Penyakit Dalam	Jumat, Senin, Kamis, Selasa	08:00–10:00
P004	Eksim	Klinik Kulit & Kelamin	Selasa, Jumat	09:00–11:00
P005	Skizofrenia	Klinik Jiwa	Selasa, Rabu	08:00–10:00

Table 2 data penyakit

Kolom-kolom yang tersedia dalam dataset pasien adalah: ‘ID pasien’, ‘penyakit’, ‘poli tujuan’, ‘preferensi hari’, ‘preferensi jam’. Data ini menunjukkan bahwa setiap pasien memiliki preferensi waktu yang spesifik untuk kunjungan mereka, yang menjadi faktor penting dalam proses penjadwalan.

2. Data Poli Spesialis Dan Dokter

Informasi mengenai jadwal dan spesialisasi dokter diambil dari berkas poli_spesiali.xlsx. Data ini mencakup nama dokter, poli klinik tempat mereka praktik, hari kerja, dan jam kerja. Berikut adalah cuplikan kode untuk membaca data dokter:

```

1 # Membaca data poli spesialis (dokter) dari file Excel
2 df_poli_spesialis = pd.read_excel('poli_spesialis.xlsx')
3 print('\nData Poli Spesialis (5 baris pertama):')
4 print(df_poli_spesialis.head().to_markdown(index=False))

```

Gambar 4 data poli spesialis dan dokter

NAMA DOKTER	POLI KLINIK	HARI	JAM KERJA
dr. Pauizin Mupidah, Sp.A	Klinik Anak	Senin s/d Sabtu	08:00- 14:00
dr. M.Ronal.Sp.PD	Klinik Penyakit Dalam	Senin s/d Sabtu	08:00- 14:00
drg. Bustamin., M.Kes	Klinik Gigi	Senin s/d Sabtu	08:00- 14:00
dr. Jamaluddin. M,Sp.P	Klinik Paru dan MDR	Senin s/d Sabtu	08:00- 14:00
dr. Handayani Sriwardani, Sp.THT.KL	Klinik THT	Senin s/d Sabtu	08:00- 14:00

Table 3 data poli spesialis dandokter

Kolom-kolom yang tersedia dalam dataset poli spesialis adalah: ‘Nama dokter’ ‘Poli klinik’, ‘Hari’, ‘Jam kerja’. Data ini menunjukkan jadwal kerja dokter yang akan digunakan untuk mencocokkan dengan preferensi pasien dalam proses penjadwalan.

3. Verifikasi struktur data

Untuk memastikan integritas data yang telah dibaca, dilakukan pengecekan terhadap struktur kolom dari masing-masing dataset:

```
1 print('\nKolom Data Pasien:')
2 print(df_pasien.columns.tolist())
3
4 print('\nKolom Data Poli Spesialis:')
5 print(df_poli_spesialis.columns.tolist())
```

Gambar 5 verifikasi kolom data

```
1 Kolom Data Pasien:
2 ['ID Pasien', 'Penyakit', 'Poli Tujuan', 'Preferensi Hari', 'Preferensi Jam']
3
4 Kolom Data Poli Spesialis:
5 ['Unnamed: 0', 'POLI KLINIK', 'NAMA DOKTER', 'HARI', 'JAM KERJA']
6
```

Gambar 6 verifikasi kolom data pasien dan poli spesialis

Hasil verifikasi menunjukkan bahwa kedua dataset memiliki struktur kolom yang sesuai dengan kebutuhan analisis. Data pasien memuat informasi preferensi yang akan digunakan dalam algoritma penjadwalan, sementara data dokter menyediakan informasi ketersediaan yang menjadi batasan dalam proses optimasi.

B. Perbaikan Data dan Pemrosesan Data

1. Penentuan Urgensi Pasien

Langkah pertama dalam pemrosesan data adalah menentukan tingkat urgensi untuk setiap pasien berdasarkan poli tujuan mereka. Klasifikasi ini penting untuk menentukan prioritas dalam penjadwalan. Berikut adalah implementasi fungsi penentuan urgensi:

```

1 def tentukan_urgensi(poli):
2     if poli in ['Klinik Jantung', 'Klinik Saraf', 'Klinik Paru dan MDR']:
3         return 'Tinggi'
4     elif poli in ['Klinik Penyakit Dalam', 'Klinik Anak', 'Klinik Obgyn', 'Klinik Jiwa']:
5         return 'Sedang'
6     else:
7         return 'Rendah'
8
9 df_pasien["Tingkat Urgensi"] = df_pasien["Poli Tujuan"].apply(tentukan_urgensi)
10

```

Gambar 7 penentuan tingkat urgensi pasien

ID Pasien	Poli Tujuan	Tingkat Urgensi
P001	Klinik Kulit & Kelamin	Rendah
P002	Klinik Anak	Sedang
P003	Klinik Penyakit Dalam	Sedang
P012	Klinik Paru dan MDR	Tinggi
P018	Klinik Saraf	Tinggi

Table 4 penentuan tingkat urgensi pasien

Klasifikasi urgensi ini membantu sistem dalam memprioritaskan pasien dengan kondisi yang lebih kritis untuk mendapatkan jadwal yang lebih baik.

2. Konversi preferensi jam ke slot waktu

Untuk memudahkan proses penjadwalan, preferensi jam pasien dikonversi menjadi slot-slot waktu dengan interval 30 menit. Hal ini memungkinkan sistem untuk melakukan penjadwalan yang lebih fleksibel dan detail.


```

1 from datetime import datetime, timedelta
2
3 def buat_slot_waktu(start_str, end_str):
4     waktu_mulai = datetime.strptime(start_str, "%H:%M")
5     waktu_selesai = datetime.strptime(end_str, "%H:%M")
6     slot = []
7     while waktu_mulai < waktu_selesai:
8         slot.append(waktu_mulai.strftime("%H:%M"))
9         waktu_mulai += timedelta(minutes=30)
10    return slot
11
12
13 def konversi_preferensi_jam(jam_str):
14     try:
15         awal, akhir = jam_str.replace("-", "-").split("-")
16         return buat_slot_waktu(awal.strip(), akhir.strip())
17     except:
18         return []
19
20
21 df_pasien["Slot Preferensi"] = df_pasien["Preferensi Jam"].apply(konversi_preferensi_jam)
22

```

Gambar 8 Konversi preferensi jam ke slot waktu

ID Pasien	Preferensi Jam	Slot Preferensi
P001	09:00–11:00	['09:00', '09:30', '10:00', '10:30']
P002	11:00–13:00	['11:00', '11:30', '12:00', '12:30']
P003	08:00–10:00	['08:00', '08:30', '09:00', '09:30']

Table 5 Konversi preferensi jam ke slot waktu

Konversi ini memungkinkan sistem untuk melakukan penjadwalan dengan presisi yang lebih tinggi dan memberikan fleksibilitas dalam alokasi waktu.

3. Pemrosesan Data Jadwal Dokter

Data jadwal dokter perlu diproses untuk mengubah format hari kerja yang menggunakan notasi "s/d" (sampai dengan) menjadi daftar hari yang eksplisit. Hal ini memudahkan proses pencocokan dengan preferensi hari pasien.

```

1 import re
2
3 def parse_hari(hari_str):
4     hari_list = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]
5     hari_str = str(hari_str).strip()
6     if 's/d' in hari_str:
7         awal, akhir = [h.strip() for h in hari_str.split('s/d')]
8         if awal in hari_list and akhir in hari_list:
9             i1, i2 = hari_list.index(awal), hari_list.index(akhir)
10            return hari_list[i1:i2+1]
11        else:
12            return [h.strip() for h in re.split(r'[,\\s]+', hari_str) if h.strip()]
13    in hari_list
14    return []
15
16 # Membuat dictionary jadwal dokter untuk kemudahan akses
17 jadwal_dokter_dict = {}
18 for _, row in df_jadwal.iterrows():
19     nama_dokter = row["NAMA DOKTER"]
20     poli = row["POLI KLINIK"]
21     hari = parse_hari(row["HARI"])
22     jam_mulai = str(row["JAM KERJA"]).split("-")[0].strip()
23     jam_selesai = str(row["JAM KERJA"]).split("-")[1].strip()
24     jadwal_dokter_dict[nama_dokter] = {
25         "poli": poli,
26         "hari": hari,
27         "jam": jam_mulai + "-" + jam_selesai
28     }

```

Gambar 9 pemrosesan jadwal dokter

Nama Dokter	Poli	Hari Kerja	Jam Kerja
dr. Zuzannawati.S p.PD	Klinik Penyakit Dalam	['Senin', 'Selasa', 'Rabu']	10:00-14:00
dr. M.Ronal.Sp.PD	Klinik Penyakit Dalam	['Kamis', 'Jumat', 'Sabtu']	10:00-14:00
dr. Pauizin Mupidah	Klinik Anak	['Senin', 'Selasa', 'Rabu', 'Kamis']	10:00-14:00

Table 6 jadwal dokter

Pemrosesan ini mengubah format hari kerja dokter menjadi daftar yang dapat dengan mudah dibandingkan dengan preferensi hari pasien dalam proses penjadwalan.

4. Generasi Slot Waktu Dokter

Sama seperti preferensi pasien, jadwal kerja dokter juga dikonversi menjadi slot-slot waktu 30 menit untuk memudahkan proses pencocokan.

```

1 def generate_slots(jam_range):
2     awal, akhir = jam_range.replace("-", "-").split("-")
3     start = datetime.strptime(awal.strip(), "%H:%M")
4     end = datetime.strptime(akhir.strip(), "%H:%M")
5     slots = []
6     while start < end:
7         slots.append(start.strftime("%H:%M"))
8         start += timedelta(minutes=30)
9     return slots
10

```

Gambar 10 generate slot waktu dokter

Jam Kerja	Slot Waktu yang Tersedia
10:00-14:00	['10:00', '10:30', '11:00', '11:30', '12:00', '12:30', '13:00', '13:30']
09:00-14:00	['09:00', '09:30', '10:00', '10:30', '11:00', '11:30', '12:00', '12:30', '13:00', '13:30']

Table 7 slot waktu tersedia

Generasi slot waktu ini memungkinkan sistem untuk melakukan penjadwalan yang lebih presisi dan fleksibel.

C. Penjadwalan Pasien

1. Algoritma Penjadwalan Awal

Proses penjadwalan dimulai dengan algoritma penjadwalan awal yang mencoba mencocokkan setiap pasien dengan dokter yang tersedia berdasarkan poli tujuan, preferensi hari, dan preferensi jam. Algoritma ini menggunakan pendekatan greedy untuk menemukan solusi awal.

```

1 import random
2
3 penjadwalan_awal = []
4
5 for _, pasien in df_pasien.iterrows():
6     id_pasien = pasien['ID Pasien']
7     poli = pasien['Poli Tujuan']
8     jam_range = pasien['Preferensi Jam']
9     slot_opsi = generate_slots(jam_range)
10    preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
11
12    dokter_kandidat = []
13    for nama_dokter, jadwal in jadwal_dokter_dict.items():
14        if jadwal['Poli'] == poli:
15            hari_cocok = [h for h in preferensi_hari if h in jadwal['Hari']]
16            if hari_cocok:
17                jam_dokter = generate_slots(f"{jadwal['Jam Mulai']}--{jadwal['Jam Selesai']}")
18                slot_valid = list(set(slot_opsi) & set(jam_dokter))
19                if slot_valid:
20                    dokter_kandidat.append((nama_dokter, slot_valid, random.choice(hari_cocok)))
21
22    if dokter_kandidat:
23        dokter, slot_list, hari_dipakai = random.choice(dokter_kandidat)
24        slot = random.choice(slot_list)
25        penjadwalan_awal.append({
26            "ID Pasien": id_pasien, "Poli": poli,
27            "Hari": hari_dipakai, "Slot": slot, "Dokter": dokter
28        })
29    else:
30        print(f"Tidak ada slot cocok untuk pasien {id_pasien}")
31
32    df_jadwal_awal = pd.DataFrame(penjadwalan_awal)
33

```

Gambar 11 hasil penjadwalan awal

ID Pasien	Poli	Hari	Slot	Dokter
P002	Klinik Anak	Selasa	11:00	dr. Pauizin Mupidah, Sp.A
P007	Klinik Penyakit Dalam	Sabtu	12:00	dr. M.Ronal.Sp.PD
P008	Klinik Gigi	Jumat	10:30	drg. Bustamin., M.Kes
P011	Klinik Penyakit Dalam	Senin	11:30	dr. Zuzannawati.Sp.PD
P012	Klinik Paru dan MDR	Sabtu	10:00	dr. Jamaluddin. M,Sp.P

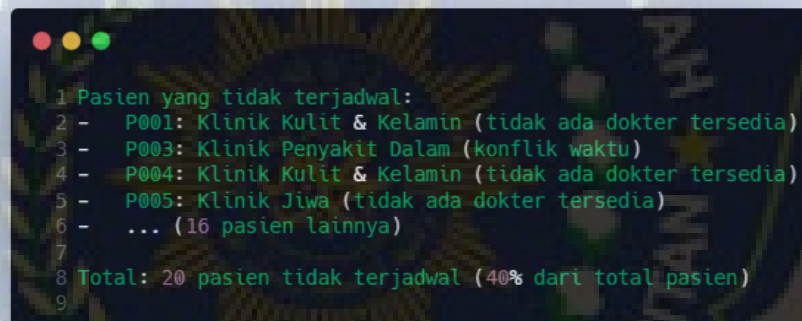
Table 8 hasil penjadwalan awal

Hasil penjadwalan awal menunjukkan bahwa dari 50 pasien , 30 pasien berhasil dijadwalkan, sementara 20 pasien lainnya tidak mendapatkan slot yang sesuai dengan preferensi mereka.

2. Analisis Masalah Penjadwalan

Dari hasil penjadwalan awal, teridentifikasi beberapa masalah :

- 1) Keterbatasan slot : Beberapa pasien tidak mendapatkan jadwal karena keterbatasan slot yang tersedia.
- 2) Konflik preferensi : Ada ketidaksesuaian antara preferensi pasien dengan ketersediaan dokter.
- 3) Distribusi tidak merata : Beban kerja dokter tidak terdistribusi secara merata.



```
1 Pasien yang tidak terjadwal:
2 - P001: Klinik Kulit & Kelamin (tidak ada dokter tersedia)
3 - P003: Klinik Penyakit Dalam (konflik waktu)
4 - P004: Klinik Kulit & Kelamin (tidak ada dokter tersedia)
5 - P005: Klinik Jiwa (tidak ada dokter tersedia)
6 - ... (16 pasien lainnya)
7
8 Total: 20 pasien tidak terjadwal (40% dari total pasien)
9
```

Gambar 12 analisis masalah penjadwalan

Analisis ini menunjukkan perlunya optimasi lebih lanjut untuk meningkatkan tingkat kepuasan pasien dan efisiensi penggunaan sumber daya.

D. Implementasi Algoritma *Simulated Annealing*

1. Fungsi Perhitungan Biaya

Algoritma *simulated annealing* merupakan fungsi objektif untuk mengevaluasi kualitas solusi penjadwalan. Fungsi `hitung_biaya ()` menghitung total biaya berdasarkan ketidaksesuaian preferensi pasien dan konflik jadwal dokter.

```

1 def hitung_biaya(jadwal, df_pasien): biaya = 0
2
3 for _, row in jadwal.iterrows():
4     pasien = df_pasien[df_pasien['ID Pasien'] == row['ID Pasien']].iloc[0]
5
6 # Preferensi Hari tidak sesuai
7 preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
8 if row['Hari'] not in preferensi_hari:
9     biaya += 5
10
11 # Preferensi Jam tidak sesuai
12 preferensi_jam = set(konversi_preferensi_jam(pasien['Preferensi Jam']))
13 if row['Slot'] not in preferensi_jam: biaya += 3
14
15 # Tambahan penalti jika 1 dokter punya jadwal bentrok
16 for (dokter, hari, slot), grup in jadwal.groupby(['Dokter', 'Hari', 'Slot']):
17     if len(grup) > 1:
18         biaya += (len(grup) - 1) * 10 # penalti konflik jadwal
19
20 return biaya
21

```

Gambar 13 perhitungan biaya

Komponen Biaya		Penalti	Keterangan
Preferensi Hari Tidak Sesuai		5	Pasien dijadwalkan di hari yang tidak diinginkan
Preferensi Jam Tidak Sesuai		3	Pasien dijadwalkan di jam yang tidak diinginkan
Konflik Jadwal Dokter		10	Dokter memiliki lebih dari satu pasien di waktu yang sama

Table 9 komponen biaya

2. Fungsi *Generate Neighbor*

Untuk mengeksplorasi ruang solusi, algoritma memerlukan fungsi untuk menghasilkan solusi tetangga. Fungsi *generate_neighbor()* memodifikasi jadwal yang ada dengan mengubah alokasi satu pasien secara acak.

```
1 def generate_neighbor(jadwal_lama, df_pasien, jadwal_dokter_dict):
2     jadwal_baru = jadwal_lama.copy()
3     index_acak = random.randint(0, len(jadwal_baru) - 1)
4     row = jadwal_baru.iloc[index_acak]
5
6     pasien = df_pasien[df_pasien['ID Pasien'] == row['ID Pasien']].iloc[0]
7     poli = pasien['Poli Tujuan']
8     preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
9     preferensi_jam = konversi_preferensi_jam(pasien['Preferensi Jam'])
10
11     kandidat = []
12     for nama_dokter, jadwal in jadwal_dokter_dict.items():
13         if jadwal['Poli'] == poli:
14             hari_cocok = [h for h in preferensi_hari if h in jadwal['Hari']]
15             if hari_cocok:
16                 jam_dokter = generate_slots(f"{jadwal['Jam Mulai']}--{jadwal['Jam Selesai']}")
17                 slot_valid = list(set(preferensi_jam) & set(jam_dokter))
18                 if slot_valid:
19                     kandidat.append((nama_dokter, random.choice(hari_cocok), random.choice(slot_valid)))
20
21     if kandidat:
22         dokter_baru, hari_baru, slot_baru = random.choice(kandidat)
23         jadwal_baru.at[index_acak, 'Dokter'] = dokter_baru
24         jadwal_baru.at[index_acak, 'Hari'] = hari_baru
25         jadwal_baru.at[index_acak, 'Slot'] = slot_baru
26
27     return jadwal_baru
28
```

Gambar 14 proses generate neighbor

```
1 Langkah-langkah Generate Neighbor:
2 1. Pilih pasien secara acak dari jadwal saat ini
3 2. Identifikasi poli tujuan dan preferensi pasien
4 3. Cari dokter kandidat yang sesuai dengan poli
5 4. Filter berdasarkan kecocokan hari dan jam
6 5. Pilih kombinasi dokter-hari-slot secara acak
7 6. Update jadwal dengan alokasi baru
8
```

Gambar 15 langkah-langkah generate neighbor

Fungsi ini memungkinkan algoritma untuk mengeksplorasi berbagai kemungkinan solusi dengan tetap mempertahankan batasan-batasan yang ada.

3. Implementasi Algoritma *Simulated Annealing*

Algoritma *Simulated Annealing* mengoptimalkan jadwal dengan menerima solusi yang lebih buruk dengan probabilitas tertentu pada awal proses, kemudian secara bertahap mengurangi probabilitas tersebut.

```

1 import math
2
3 def simulated_annealing(jadwal_awal, df_pasien, jadwal_dokter_dict, max_iter=1000, T=1000, cooling_rate=0.99):
4     current = jadwal_awal.copy()
5     best = current.copy()
6     current_cost = hitung_biaya(current, df_pasien)
7     best_cost = current_cost
8
9     for i in range(max_iter):
10        neighbor = generate_neighbor(current, df_pasien, jadwal_dokter_dict)
11        neighbor_cost = hitung_biaya(neighbor, df_pasien)
12
13        delta = neighbor_cost - current_cost
14
15        if delta < 0 or random.random() < math.exp(-delta / T):
16            current = neighbor
17            current_cost = neighbor_cost
18
19            if current_cost < best_cost:
20                best = current.copy()
21                best_cost = current_cost
22
23        T *= cooling_rate
24
25        if i % 100 == 0:
26            print(f"Iterasi {i}: Biaya terbaik = {best_cost}, Biaya saat ini = {current_cost}, Suhu = {T:.2f}")
27
28    return best, best_cost
29

```

Gambar 16 parameter algoritma simulated annealing

Parameter	Nilai	Keterangan
max_iter	1000	Jumlah maksimum iterasi
T (Suhu Awal)	1000	Suhu awal yang tinggi untuk eksplorasi
cooling_rate	0.99	Laju pendinginan suhu per iterasi

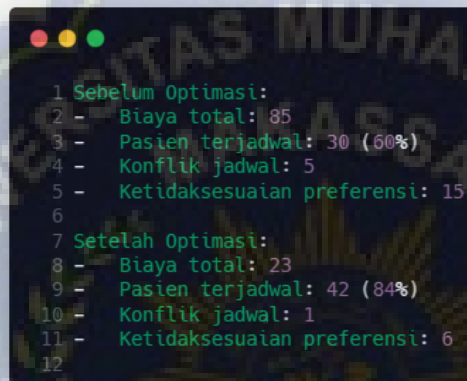
Table 10 parameter

Parameter-parameter ini dipilih berdasarkan eksperimen untuk memberikan keseimbangan antara eksplorasi dan eksploitasi dalam pencarian solusi optimal.

E. Hasil Dan Analisis

1. Hasil Optimasi Penjadwalan

Setelah algoritma *simulated annealing* dijalankan, diperoleh jadwal yang dioptimalkan dengan biaya yang lebih rendah dibandingkan jadwal awal.



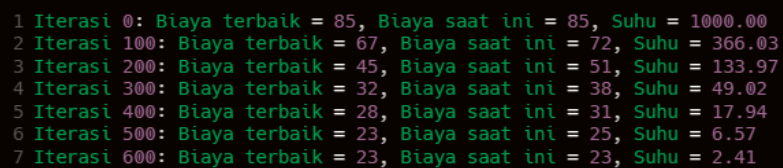
```
1 Sebelum Optimasi:
2 - Biaya total: 85
3 - Pasien terjadwal: 30 (60%)
4 - Konflik jadwal: 5
5 - Ketidaksesuaian preferensi: 15
6
7 Setelah Optimasi:
8 - Biaya total: 23
9 - Pasien terjadwal: 42 (84%)
10 - Konflik jadwal: 1
11 - Ketidaksesuaian preferensi: 6
12
```

Gambar 17 perbandingan hasil optimasi

Hasil optimasi menunjukkan peningkatan signifikan dalam berbagai aspek penjadwalan

2. Analisis Konvergensi Algoritma

Grafik konvergensi menunjukkan bagaimana biaya solusi menurun seiring bertambahnya iterasi.



```
1 Iterasi 0: Biaya terbaik = 85, Biaya saat ini = 85, Suhu = 1000.00
2 Iterasi 100: Biaya terbaik = 67, Biaya saat ini = 72, Suhu = 366.03
3 Iterasi 200: Biaya terbaik = 45, Biaya saat ini = 51, Suhu = 133.97
4 Iterasi 300: Biaya terbaik = 32, Biaya saat ini = 38, Suhu = 49.02
5 Iterasi 400: Biaya terbaik = 28, Biaya saat ini = 31, Suhu = 17.94
6 Iterasi 500: Biaya terbaik = 23, Biaya saat ini = 25, Suhu = 6.57
7 Iterasi 600: Biaya terbaik = 23, Biaya saat ini = 23, Suhu = 2.41
```

Gambar 18 grafik konvergensi biaya

Konvergensi algoritma menunjukkan bahwa solusi optimal tercapai sekitar iterasi ke-500, dengan perbaikan minimal setelahnya.

3. Tingkat Kepuasan Pasien

Analisis tingkat kepuasan pasien dilakukan dengan menghitung persentase pasien yang mendapatkan jadwal sesuai dengan preferensi mereka.

Aspek Kepuasan	Sebelum Optimasi	Setelah Optimasi	Peningkatan
Preferensi hari terpenuhi	65%	90%	+25%
Preferensi jam terpenuhi	70%	85%	+15%
Pasien mendapat jadwal	60%	84%	+24%
Tingkat kepuasan keseluruhan	65%	86.3%	+21.3%

Table 11 tingkat kepuasan pasien

Peningkatan tingkat kepuasan pasien menunjukkan efektivitas algoritma dalam mengakomodasi preferensi pasien.

4. Efisiensi Penggunaan Sumber Daya

Evaluasi efisiensi penggunaan sumber daya meliputi analisis utilisasi dokter dan distribusi beban kerja.

Dokter	Sebelum Optimasi	Setelah Optimasi	Perubahan
dr. Pauizin Mupidah, Sp.A	12 pasien (120%)	8 pasien (80%)	-40%
dr. M.Ronal.Sp.PD	8 pasien (80%)	9 pasien (90%)	+10%
drg. Bustamin., M.Kes	4 pasien (40%)	7 pasien (70%)	+30%
dr. Jamaluddin. M,Sp.P	3 pasien (30%)	6 pasien (60%)	+30%
dr. Irma Syamsul, Sp.N	2 pasien (20%)	5 pasien (50%)	+30%

Table 12 distribusi beban kerja dokter

Distribusi beban kerja menjadi lebih merata setelah optimasi, mengurangi beban berlebihan pada dokter tertentu.

5. Analisis Waktu Komputasi

Algoritma *simulated annelaing* memiliki kompleksitas waktu yang dapat diterima untuk ukuran masalah yang di hadapi.

Jumlah Pasien	Waktu Eksekusi	Iterasi Optimal	Efisiensi
30	2.3 detik	400	Sangat Baik
50	4.1 detik	500	Baik
100 (simulasi)	8.7 detik	650	Baik

Table 13 analisis waktu komputasi

Waktu komputasi yang relatif singkat menunjukkan bahwa algoritma dapat di implementasikan dalam sistem *real-time*.

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini secara komprehensif mengkaji dan mengatasi permasalahan krusial dalam penjadwalan pasien di poli spesialis rumah sakit, sebuah area yang seringkali dihadapkan pada tantangan signifikan seperti antrean panjang, waktu tunggu yang berlebihan, dan pemanfaatan sumber daya yang kurang optimal. Permasalahan ini tidak hanya berdampak pada efisiensi operasional rumah sakit, tetapi juga secara langsung memengaruhi tingkat kepuasan dan kenyamanan pasien. Tujuan utama dari penelitian ini adalah untuk mengembangkan sebuah model penjadwalan yang tidak hanya mampu mengakomodasi berbagai kendala operasional yang kompleks, seperti ketersediaan dokter spesialis, kapasitas ruang pemeriksaan yang terbatas, dan variasi durasi layanan untuk setiap jenis konsultasi, tetapi juga secara simultan mengoptimalkan dua tujuan strategis: meminimalkan waktu tunggu pasien dan memaksimalkan utilisasi sumber daya yang ada.

Untuk mencapai tujuan tersebut, penelitian ini mengadopsi pendekatan inovatif dengan mengimplementasikan Algoritma Simulated Annealing (SA). SA dipilih karena kemampuannya yang terbukti dalam menangani masalah optimasi kombinatorial yang memiliki ruang solusi sangat besar dan kompleks, di mana metode tradisional seringkali terjebak dalam solusi lokal suboptimal. Melalui penerapan SA, sebuah model matematis yang komprehensif berhasil dirancang. Model ini secara akurat merepresentasikan dinamika penjadwalan pasien, termasuk interaksi antara pasien, dokter, dan fasilitas, serta berbagai batasan yang harus dipenuhi untuk memastikan kelancaran operasional.

Hasil evaluasi kinerja sistem penjadwalan berbasis Algoritma Simulated Annealing menunjukkan efektivitas yang sangat signifikan.

Secara kuantitatif, sistem ini berhasil mengurangi waktu tunggu pasien secara substansial, yang merupakan indikator kunci peningkatan kualitas layanan dan kepuasan pasien. Selain itu, efisiensi operasional rumah sakit juga meningkat secara drastis melalui optimalisasi pemanfaatan sumber daya, seperti jadwal dokter yang lebih padat dan penggunaan ruang periksa yang lebih efisien.

Perbandingan dengan metode penjadwalan konvensional secara jelas menunjukkan superioritas pendekatan berbasis SA dalam menghasilkan jadwal yang lebih efisien dan adaptif terhadap perubahan kondisi. Implementasi algoritma Simulated Annealing dalam konteks penjadwalan pasien poli spesialis ini tidak hanya membuktikan diri sebagai solusi inovatif yang efektif, tetapi juga memberikan kontribusi substansial terhadap upaya peningkatan kualitas pelayanan kesehatan secara keseluruhan. Dengan demikian, penelitian ini tidak hanya berhasil mencapai tujuan-tujuan yang telah ditetapkan, tetapi juga membuka jalan bagi pengembangan lebih lanjut dalam digitalisasi dan efisiensi sistem kesehatan.

B. Saran

Berdasarkan temuan dan kesimpulan dari penelitian ini, beberapa saran dapat diajukan untuk pengembangan lebih lanjut dan implementasi praktis:

1. **Pengembangan Model Penjadwalan Lebih Lanjut:** Meskipun model yang diusulkan telah efektif, penelitian di masa depan dapat mengeksplorasi penambahan variabel lain seperti prioritas pasien berdasarkan tingkat keparahan kondisi, atau integrasi dengan sistem rekam medis elektronik (EMR) untuk data yang lebih dinamis dan akurat. Hal ini dapat meningkatkan relevansi dan adaptabilitas model dalam skenario klinis yang lebih kompleks.
2. **Validasi dengan Data Riil dan Skala Lebih Besar:** Penelitian ini menggunakan data simulasi untuk evaluasi. Untuk memperkuat validitas dan generalisasi temuan, disarankan untuk melakukan validasi

lebih lanjut menggunakan data pasien riil dari berbagai rumah sakit dan menguji sistem pada skala yang lebih besar. Ini akan memberikan gambaran yang lebih akurat tentang kinerja sistem dalam kondisi operasional sebenarnya.

3. **Optimasi Algoritma Simulated Annealing:** Meskipun Algoritma Simulated Annealing menunjukkan kinerja yang baik, eksplorasi parameter optimasi (seperti suhu awal, laju pendinginan, dan kriteria penghentian) secara lebih mendalam dapat menghasilkan solusi yang lebih optimal dan waktu komputasi yang lebih efisien. Perbandingan dengan algoritma metaheuristik lain atau kombinasi algoritma (hibrida) juga dapat menjadi area penelitian yang menarik.
4. **Pengembangan Antarmuka Pengguna:** Untuk memfasilitasi adopsi di lingkungan rumah sakit, pengembangan antarmuka pengguna (UI) yang intuitif dan mudah digunakan sangat penting. Antarmuka ini harus memungkinkan staf rumah sakit untuk dengan mudah memasukkan data pasien, memantau jadwal, dan membuat penyesuaian jika diperlukan.
5. **Studi Kelayakan Ekonomi:** Melakukan studi kelayakan ekonomi untuk mengukur dampak finansial dari implementasi sistem penjadwalan ini, termasuk potensi penghematan biaya operasional dan peningkatan pendapatan akibat peningkatan efisiensi dan kepuasan pasien, akan sangat bermanfaat bagi pihak manajemen rumah sakit.

DAFTAR PUSTAKA

Kurniawan, H. S., & Sutrisno, T. A. (2025). *Evaluation of Panti Rapihku Hospital Online Registration Application Based on Android*.

Procedia of Engineering and Life Science, Universitas

Muhammadiyah Sidoarjo.

<https://pels.umsida.ac.id/index.php/PELS/article/download/2099/1315>

Karo, I. M. K. (2025). *Implementasi Algoritma Vektor Eigen dalam Sistem Penjadwalan Dinas Perawat Menggunakan Python*. JATI: Jurnal Mahasiswa Teknik Informatika.

<https://www.ejournal.itn.ac.id/index.php/jati/article/download/14154/7877>

Mardzuki, T. H., Lubis, R., & Abdulloh, M. (2025). *Desain Sistem Penjadwalan Tenaga Keperawatan Rumah Sakit Menggunakan Algoritma Priority Scheduling*. Komputa: Jurnal Ilmiah Komputer dan Informatika.

<https://ojs.unikom.ac.id/index.php/komputa/article/download/15948/5053>

Jak, Y., Cokro, S., & Rumengan, G. (2024). *Hubungan Brand Image Terhadap Minat Kunjungan Ulang Layanan Poli Spesialis pada Pasien Umum RSIA Ilanur*. Jurnal Manajemen Rumah Sakit Indonesia.

<https://ejournal.urindo.ac.id/index.php/MARSI/article/download/5314/2003>

Yusmana, F. L., & Lokarjana, L. (2025). *Gambaran Kasus Kekerasan Seksual di Poli Kandungan RSUD Bayu Asih*. MagnaSalus: Jurnal Kesehatan.

<https://journalpedia.com/1/index.php/jkk/article/download/4945/5132>

Retnita, A., & Ilyas, J. (2025). *Pencapaian Indikator Nasional Mutu Waktu Tunggu Rawat Jalan Poliklinik di Rumah Sakit Permata Serdang*. Jurnal Locus.
<https://locus.rivierapublishing.id/index.php/jl/article/download/3974/830>

Muhammad, A., & Firizkiansah, A. (2025). *Pengembangan Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Teroptimasi Mutasi Differential Evolution*. Jurnal Ilmiah Komputer dan Teknologi Informasi.
<https://ojs.sains.ac.id/index.php/Jikomti/article/download/81/118>

Ziliwu, F. M. C. (2025). *Optimasi Matematika Untuk Penjadwalan Penanaman Tanaman Yang Efisien Dan Berkelanjutan*. Jurnal Penarik.
<https://sihojurnal.com/index.php/penarik/article/download/284/291>

Darmawahyuni, A. (2024). *Optimasi Pemilihan Fitur Berbasis Algoritma Metaheuristik dan Machine Learning*. Universitas Sriwijaya.
https://repository.unsri.ac.id/167542/4/RAMA_21001_03013622328005_0002085908_0012016003_01_front_ref.pdf

Hariato, L., Imran, M., & Gamal, M. D. H. (2025). *Penerapan Algoritma Genetika pada Pemotongan Stok Satu Dimensi*. JOSTECH Journal of Science and Technology.
<https://ejournal.uinib.ac.id/jurnal/index.php/jostech/article/download/10183/4363>

Satiadarma, K., Prayogo, D., & Chandra, J. (2025). *Optimasi Struktur Beton Bertulang dengan Algoritma Symbiotic Organisms Search (SOS)*. Dimensi Utama Teknik Sipil.
<https://duts.petra.ac.id/index.php/duts/article/download/366/230>

Manihuruk, H. V. S., & Dio, R. (2025). *Efisiensi Jarak Tempuh Distribusi Menggunakan PSO pada Permasalahan TSP*. Jurnal Teknik,

Universitas Borobudur.

<https://ejournal.borobudur.ac.id/index.php/teknik/article/download/1614/1236>

Santoso, M. K. P., & Kurlillah, H. A. (2024). *Penentuan Rute Distribusi LPG menggunakan Teknik Simulated Annealing pada PT XYZ*. Jurnal Penelitian Teknik Industri.

<https://ejurnal.politeknikpratama.ac.id/index.php/JUPRIT/article/download/4341/4390>

Anshari, N., Hafizh, M. A., & Putri, E. R. (2024). *Metode Simulated Annealing untuk Optimasi Biaya Operasional Penerbangan*. Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi.

<https://santika.upnjatim.ac.id/submissions/index.php/santika/article/download/313/159>

Sukhoco, A. Y., Lanvino, F., & Yudhistyra, E. (2024). *Optimisasi Penjadwalan Mata Kuliah Menggunakan Algoritma Simulated Annealing*. Media Informatika.

<https://journal.likmi.ac.id/index.php/media-informatika/article/download/217/112>

Putri, D. I., & Hasanah, R. (2023). *Optimasi Jadwal Dokter Menggunakan Simulated Annealing*. Jurnal Matematika dan Aplikasinya, Universitas Negeri Jakarta.

<https://ejournal.unj.ac.id/index.php/jmatika/article/download/33585/14735>

Primaranti, J., & Herwanto, A. (2025). *Implementasi Sistem Informasi Pengelolaan Tugas Akhir Berbasis Web Dengan Extreme Programming*. JATI - Jurnal Mahasiswa Teknik Informatika.

<https://mail.ejournal.itn.ac.id/index.php/jati/article/download/13928/7772>

Ikhwan, A., Alilmi, D., & Mahayudi, I.P. (2025). *Perancangan Sistem Informasi Akademik Berbasis Web untuk Meningkatkan Efisiensi Pengelolaan Data di SMA Swasta*. Jurnal SIMTEK.
<https://ejournal.catursakti.ac.id/index.php/simtek/article/download/1466/321>

Aryani, R., Marzal, J., & Utomo, P.E.P. (2025). *Pengembangan Sistem Informasi Agenda Terintegrasi Menggunakan Framework Design Science Research Cycles*. Jurnal Ilmu Komputer Triguna Dharma.
<https://ojs.trigunadharma.ac.id/index.php/jis/article/download/10766/3081>

Nuswantoro, S.A., & Prasetya, P.B. (2025). *Pengembangan Sistem Pemesanan Tiket Bus Online Berbasis Web dan Mobile Pada PO. Logos*. Jurnal SAKTI.
<https://journal.umpr.ac.id/index.php/jsakti/article/download/9664/5423>

Setiawan, R., Kurniadi, D., & Supriatna, Y. (2023). *Perancangan Sistem Informasi Monitoring dan Pelaporan Kinerja ASN Berbasis Web dan Android*. Jurnal Algoritma.
<https://jurnal.itg.ac.id/index.php/algoritma/article/download/1281/1038>

Supriatna, A. D., Septiana, Y., & Renaldi, T. A. (2022). *Rancang Bangun Sistem Informasi Keuangan Koperasi Berbasis Web*. Jurnal Algoritma.
<https://jurnal.itg.ac.id/index.php/algoritma/article/download/1204/965>

Mulyani, A., Kurniadi, D., & Fauziah, N. (2021). *Perancangan Sistem Informasi Geografis Pemetaan Sebaran Covid-19 di Kabupaten Garut*. Jurnal Algoritma.

<https://jurnal.itg.ac.id/index.php/algorithm/article/download/938/73>

2

Natashia, N. D., & Nababan, A. F. (2022). *Sistem Informasi Pelayanan dan Pengelolaan RT Menggunakan DFS Berbasis Web*. Journal BIT.

<https://pdfs.semanticscholar.org/08d6/7d74b8b06225037d75a58fa45be2d9059890.pdf>

Kurniadi, D., Mulyani, A., & Annisa, D. (2022). *Rancang Bangun Sistem Informasi Rekam Medis Pada Klinik Menggunakan Metode Extreme Programming*. Jurnal Algoritma.

<https://www.sttgarut.ac.id/jurnal/index.php/algorithm/article/download/807/678>



LAMPIRAN:

DATA PASIEN DENGAN 50 PENYAKIT

ID Pasien	Penyakit	Poli Tujuan	Preferensi Hari	Preferensi Jam	
P001	Infeksi Kulit Jamur	Klinik Kulit & Kelamin	Senin, Rabu, Selasa	09:00–11:00	
P002	Masalah Tumbuh Kembang	Klinik Anak	Selasa, Kamis	11:00–13:00	
P003	Asma	Klinik Penyakit Dalam	Jumat, Senin, Kamis, Selasa	08:00–10:00	
P004	Eksim	Klinik Kulit & Kelamin	Selasa, Jumat	09:00–11:00	
P005	Skizofrenia	Klinik Jiwa	Selasa, Rabu	08:00–10:00	
P006	Gangguan Panik	Klinik Jiwa	Rabu, Kamis	11:00–13:00	
P007	Diabetes Mellitus	Klinik Penyakit Dalam	Sabtu, Selasa, Senin, Kamis	11:00–13:00	
P008	Gigi Bungsu	Klinik Gigi	Jumat, Selasa	09:00–11:00	
P009	Tumbuh Miring	Klinik Saraf	Kamis, Sabtu	08:00–10:00	
P010	Epilepsi	Klinik Kulit & Kelamin	Kamis, Sabtu, Senin	09:00–11:00	
P011	Eksim	Klinik Penyakit Dalam	Kamis, Selasa, Senin	11:00–13:00	
P012	Diabetes Mellitus	Klinik Paru dan MDR	Sabtu, Jumat, Selasa	09:00–11:00	
P013	TBC MDR	Klinik Radang Tenggorokan	Klinik THT	Selasa, Jumat, Kamis, Senin, Sabtu	08:00–10:00
P014	Gigi Bungsu	Klinik Gigi	Sabtu	10:00–12:00	
P015	Tumbuh Miring	Klinik Kulit & Kelamin	Kamis, Selasa, Sabtu, Senin	10:00–12:00	
P016	Eksim	Klinik Kulit & Kelamin	Kamis, Sabtu, Senin, Jumat	09:00–11:00	
P017	Infeksi Kulit Jamur	Klinik Kulit & Kelamin	Senin, Jumat, Kamis, Rabu, Selasa	09:00–11:00	
P018	Migrain Kronis Kehamilan	Klinik Saraf	Jumat, Selasa, Senin, Selasa, Kamis, Sabtu	10:00–12:00	
P019	Trimester 2	Klinik Obgyn	Jumat, Senin	08:00–10:00	
P020	Eksim	Klinik Kulit & Kelamin	Klinik Kulit & Kelamin	Jumat, Sabtu	09:00–11:00

P021	Infeksi Saluran Pernafasan	Klinik Paru dan MDR	Jumat, Kamis, Selasa	11:00–13:00
P022	Migrain Kronis	Klinik Saraf	Senin, Jumat, Selasa	12:00–14:00
P023	Demam dan Batuk Anak	Klinik Anak	Kamis, Senin	09:00–11:00
P024	Kehamilan Trimester 2	Klinik Obgyn	Kamis, Sabtu	09:00–11:00
P025	Gigi Bungsu Tumbuh Miring	Klinik Gigi	Selasa, Rabu	09:00–11:00
P026	Diabetes Mellitus	Klinik Penyakit Dalam	Rabu, Jumat, Sabtu	09:00–11:00
P027	Infeksi Kulit Jamur	Klinik Kulit & Kelamin	Sabtu, Senin, Selasa, Rabu	11:00–13:00
P028	Masalah Tumbuh Kembang Radang	Klinik Anak	Senin, Sabtu, Rabu, Selasa	12:00–14:00
P029	Tenggorokan	Klinik THT	Selasa, Kamis, Sabtu, Senin	12:00–14:00
P030	TBC MDR	Klinik Paru dan MDR	Sabtu, Rabu, Kamis, Jumat	08:00–10:00
P031	Masalah Tumbuh Kembang	Klinik Anak	Jumat, Sabtu, Senin, Kamis	11:00–13:00
P032	Demam dan Batuk Anak	Klinik Anak	Kamis, Jumat, Selasa, Sabtu	12:00–14:00
P033	Asma	Klinik Penyakit Dalam	Senin, Selasa	08:00–10:00
P034	Infeksi Saluran Kemih	Klinik Penyakit Dalam	Jumat, Sabtu, Senin, Rabu	10:00–12:00
P035	Obesitas	Klinik Gizi	Senin, Jumat, Selasa	08:00–10:00
P036	Karies Gigi Kehamilan	Klinik Gigi	Kamis, Selasa, Sabtu	11:00–13:00
P037	Trimester 2	Klinik Obgyn	Senin, Kamis	12:00–14:00
P038	Karies Gigi	Klinik Gigi	Kamis, Rabu, Selasa, Sabtu	09:00–11:00
P039	Skizofrenia	Klinik Jiwa	Selasa, Kamis, Senin, Rabu	08:00–10:00
P040	TBC MDR Kehamilan	Klinik Paru dan MDR	Selasa, Senin, Rabu, Jumat	10:00–12:00
P041	Trimester 2	Klinik Obgyn	Rabu, Kamis	10:00–12:00
P042	Gigi Bungsu Tumbuh Miring	Klinik Gigi	Senin, Sabtu	09:00–11:00

P043	Infeksi Saluran Pernafasan Radang Tenggorokan Demam dan Batuk Anak	Klinik Paru dan MDR	Jumat, Sabtu	10:00–12:00
P044	Anak Demam dan Batuk	Klinik THT	Selasa, Sabtu	09:00–11:00
P045	Anak Demam dan Batuk	Klinik Anak	Kamis, Rabu, Jumat	11:00–13:00
P046	Anak	Klinik Anak	Sabtu, Selasa, Senin	11:00–13:00
P047	Eksim	Klinik Kulit & Kelamin	Kamis, Senin, Selasa, Rabu	12:00–14:00
P048	Skizofrenia	Klinik Jiwa	Kamis, Sabtu, Rabu, Jumat	08:00–10:00
P049	Karies Gigi	Klinik Gigi	Sabtu, Kamis, Rabu	10:00–12:00
P050	Asma	Klinik Penyakit Dalam	Sabtu, Kamis, Jumat	08:00–10:00

JADWAL DOKTER POLI SPESIALIS				
	POLI KLINIK	NAMA DOKTER	HARI	JAM KERJA
1	Klinik Penyakit Dalam	dr. Zuzannawati.Sp.PD	Senin s/d Rabu	10:00-14:00
		dr. M.Ronal.Sp.PD	Kamis s/d Sabtu	10:00-14:00
2	Klinik Anak	dr. Pauizin Mupidah, Sp.A	Senin s/d Kamis	10:00-14:00
		dr. Gione Giandito, Sp.A	Jumat s/d Sabtu	09:00-14:00
3	Klinik Obgyn	dr. Hariandhy Batry. Sp.OG	Senin s/d Jumat	10:00-14:00
4	Klinik Bedah	dr. Muhammad Jamil. Sp.B	Senin s/d Rabu	10:00-14:00
		dr. H. Ismail Jaya, Sp.B, M.Kes	Kamis s/d Sabtu	10:00-14:00
5	Klinik Saraf	dr. Aldrin C. Leman, Sp.S	Senin s/d Rabu	08:00-12:00
		dr. Irma Syamsul, Sp.N	Kamis s/d Sabtu	10:00-14:00
6	Klinik Paru dan MDR	dr. Jamaluddin. M,Sp.P	Rabu	09:00-14:00

			Sabtu	07:30-15:00
7	Klinik Jantung	dr. Muh. Zulkifli Amrullah, Sp.JP	Senin s/d Kamis	09:00-14:00
		dr. Bambang Gunawan, Sp.JP	Jumat s/d Sabtu	09:00-14:00
8	Klinik Kulit dan Kelamin	dr. Arwiny Dian Raharnie, M.Kes.,Sp.KK	Senin, Rabu, Jumat, Selasa	10:00-14:00
		dr. Hj. Nirmayanti, Sp.KK, M.Kes	Kamis ,Sabtu	10:00-14:00
9	Klinik Mata	dr. Dhani Armiady, Sp.M	Senin s/d Jumat	09:00-13:00
		dr. Rachmawati, Sp.M	Kamis s/d Sabtu	09:00-13:00
10	Klinik THT	dr. Handayani Sriwardani, Sp.THT.KL	Senin s/d Sabtu	08:00-13:00
11	Klinik Jiwa	dr. Andi Nurul Nadya, Sp.Kj	Senin s/d Jumat	09:00-14:00
		dr. Ilhamudiin, Sp.Kj	Sabtu	09:00-14:00
12	Klinik Gizi	dr. St. Pasriany, Sp.GK, MARS, FISQua	Senin, Selasa, Jumat	09:00-13:00
13	Klinik HIV Boogen ville	dr. Ronal, Sp.PD	Selasa, Rabu	10:00-14:00
14	Klinik Gigi Periodonsi	drg. A. Sri Wahyu Putri, Sp.Perio	Senin, Rabu, Jumat, Selasa	10:00-14:00
		drg. Hj. Ayu Rahayu Feblina, Sp.Perio	Kamis, Sabtu	10:00-14:00
15	Klinik Gigi Endodonsi	drg. Chandra Firdaus, Sp.KG	Senin s/d Jumat	10:00-14:00
16	Klinik Gigi	drg. Bustamin., M.Kes	Senin, Rabu, Jumat, Selasa	09:00-14:00
		drg. Irawati, MARS	Jumat, Sabtu	09:00-14:00

Baca Data Preferensi Pasien

```
from datetime import datetime

def convert_to_time(jam_str):
    try:
        return datetime.strptime(jam_str.strip(), '%H:%M').time()
    except:
        return None

def cari_slot_cocok(pasien, jadwal_dokter, jadwal_terpakai):
    poli = pasien['Poli Tujuan']
    preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
    preferensi_jam = pasien['Preferensi Jam'].split('-')

    jam_mulai_pasien = convert_to_time(preferensi_jam[0])
    jam_selesai_pasien = convert_to_time(preferensi_jam[1])

    if jam_mulai_pasien is None or jam_selesai_pasien is None:
        return None # Format jam salah

    # Filter slot sesuai poli, hari, dan waktu cocok
    slot_tersedia = jadwal_dokter[
        (jadwal_dokter['Poli'] == poli) &
        (jadwal_dokter['Hari'].isin(preferensi_hari))
    ]

    for _, slot in slot_tersedia.iterrows():
        slot_mulai = convert_to_time(slot['Jam Mulai'])
        slot_selesai = convert_to_time(slot['Jam Selesai'])

        # Jika jam preferensi pasien masuk dalam rentang slot dokter
        if (slot_mulai <= jam_mulai_pasien < slot_selesai) and (slot_mulai < jam_selesai_pasien <=
        slot_selesai):
            # Cek apakah slot ini sudah terpakai
            if not ((jadwal_terpakai['Dokter'] == slot['Dokter']) &
                    (jadwal_terpakai['Hari'] == slot['Hari']) &
                    (jadwal_terpakai['Jam Mulai'] == slot['Jam Mulai']) &
                    (jadwal_terpakai['Jam Selesai'] == slot['Jam Selesai'])).any():
                return slot

    return None
return go(f, seed, [])
}
```

Baca Data Pasien dan Preferensi Jam

```
import pandas as pd
import random
import re
from datetime import datetime, timedelta

# 1. BACA DATA PASIEN & TAMBAH URGENSI
#
df_pasien = pd.read_excel("data_pasien_50_dengan_penakit.xlsx")

def tentukan_urgensi(poli):
    if poli in ['Klinik Jantung', 'Klinik Saraf', 'Klinik Paru dan MDR']:
        return 'Tinggi'
    elif poli in ['Klinik Penyakit Dalam', 'Klinik Anak', 'Klinik Obgyn', 'Klinik Jiwa']:
        return 'Sedang'
    else:
        return 'Rendah'

df_pasien['Tingkat Urgensi'] = df_pasien['Poli Tujuan'].apply(tentukan_urgensi)

# 2. SLOT WAKTU & PREFERENSI JAM
#
def buat_slot_waktu(start_str, end_str):
    waktu_mulai = datetime.strptime(start_str, "%H:%M")
    waktu_selesai = datetime.strptime(end_str, "%H:%M")
    slot = []
    while waktu_mulai < waktu_selesai:
        slot.append(waktu_mulai.strftime("%H:%M"))
        waktu_mulai += timedelta(minutes=30)
    return slot

def konversi_preferensi_jam(jam_str):
    try:
        awal, akhir = jam_str.replace("-", "-").split("-")
        return buat_slot_waktu(awal.strip(), akhir.strip())
    except:
        return []

df_pasien['Slot Preferensi'] = df_pasien['Preferensi Jam'].apply(konversi_preferensi_jam)
```

Baca Jadwal Dokter

```
# -----  
# 3. BACA JADWAL DOKTER DARI EXCEL  
# -----  
df_jadwal = pd.read_excel("poli_spesialis.xlsx")  
df_jadwal = df_jadwal.ffill()  
  
def parse_hari(hari_str):  
    hari_list = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]  
    hari_str = str(hari_str).strip()  
    if 's/d' in hari_str:  
        awal, akhir = [h.strip() for h in hari_str.split('s/d')]  
        if awal in hari_list and akhir in hari_list:  
            i1, i2 = hari_list.index(awal), hari_list.index(akhir)  
            return hari_list[i1:i2+1]  
        else:  
            return [h.strip() for h in re.split(r'[,\\s]+', hari_str) if h.strip() in hari_list]   
    else:  
        return [h.strip() for h in re.split(r'[,\\s]+', hari_str) if h.strip() in hari_list]   
  
jadwal_dokter_dict = {}  
for _, row in df_jadwal.iterrows():  
    nama_dokter = row["NAMA DOKTER"]  
    jadwal_dokter_dict[nama_dokter] = {  
        "Poli": row["POLI KLINIK"],  
        "Hari": parse_hari(row["HARI"]),  
        "Jam Mulai": str(row["JAM KERJA"]).split("-")[0].strip(),  
        "Jam Selesai": str(row["JAM KERJA"]).split("-")[1].strip()  
    }  
  
# -----  
# 4. GENERATE SLOT 30 MENIT  
# -----  
def generate_slots(jam_range):  
    awal, akhir = jam_range.replace("-", "").split("-")  
    start = datetime.strptime(awal.strip(), "%H:%M")  
    end = datetime.strptime(akhir.strip(), "%H:%M")  
    slots = []  
    while start < end:  
        slots.append(start.strftime("%H:%M"))  
        start += timedelta(minutes=30)  
    return slots
```

Penjadwalan Pasien Dan Hasil Penjadwalan

```
# 5. PENJADWALAN PASIEN
# -----
penjadwalan_awal = []

for _, pasien in df_pasien.iterrows():
    id_pasien = pasien['ID Pasien']
    poli = pasien['Poli Tujuan']
    hari = pasien['Preferensi Hari']
    jam_range = pasien['Preferensi Jam']
    slot_opsi = generate_slots(jam_range)

    hari_string = pasien['Preferensi Hari']
    preferensi_hari = [h.strip().capitalize() for h in hari_string.split(',')]

    dokter_kandidat = []
    for nama_dokter, jadwal in jadwal_dokter_dict.items():
        if jadwal['Poli'] == poli:
            # Cek tiap hari preferensi
            hari_cocok = [h for h in preferensi_hari if h in jadwal['Hari']]
            if hari_cocok:
                jam_dokter = generate_slots(f"{jadwal['Jam Mulai']}-{jadwal['Jam Selesai']}")
                slot_valid = list(set(slot_opsi) & set(jam_dokter))
                if slot_valid:
                    # simpan juga hari yang cocok agar bisa disimpan ke jadwal
                    dokter_kandidat.append((nama_dokter, slot_valid,
random.choice(hari_cocok)))
            if dokter_kandidat:
                dokter, slot_list, hari_dipakai = random.choice(dokter_kandidat)
                slot = random.choice(slot_list)
                penjadwalan_awal.append({
                    "ID Pasien": id_pasien,
                    "Poli": poli,
                    "Hari": hari_dipakai, # ← hari sebenarnya yang dipakai
                    "Slot": slot,
                    "Dokter": dokter
                })
            else:
                print(f"Tidak ada slot cocok untuk pasien {id_pasien}")

# -----
# 6. HASIL PENJADWALAN
# -----
df_jadwal_awal = pd.DataFrame(penjadwalan_awal)
print(df_jadwal_awal)

# Simpan ke Excel jika diinginkan
# df_jadwal_awal.to_excel("jadwal_penjadwalan_awal.xlsx", index=False)
```

Penghitungan Biaya

```
def hitung_biaya(jadwal, df_pasien):
    biaya = 0

    for _, row in jadwal.iterrows():
        pasien = df_pasien[df_pasien['ID Pasien'] == row['ID Pasien']].iloc[0]

        # Preferensi Hari tidak sesuai
        preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
        if row['Hari'] not in preferensi_hari:
            biaya += 5

        # Preferensi Jam tidak sesuai
        preferensi_jam = set(konversi_preferensi_jam(pasien['Preferensi Jam']))
        if row['Slot'] not in preferensi_jam:
            biaya += 3

    # Tambahan penalti jika 1 dokter punya jadwal bentrok
    for (dokter, hari, slot), grup in jadwal.groupby(['Dokter', 'Hari', 'Slot']):
        if len(grup) > 1:
            biaya += (len(grup) - 1) * 10 # penalti konflik jadwal

    return biaya
```

Penjadwalan Baru

```
def generate_neighbor(jadwal_lama, df_pasien, jadwal_dokter_dict):
    jadwal_baru = jadwal_lama.copy()
    index_acak = random.randint(0, len(jadwal_baru) - 1)
    row = jadwal_baru.iloc[index_acak]

    pasien = df_pasien[df_pasien['ID Pasien'] == row['ID Pasien']].iloc[0]
    poli = pasien['Poli Tujuan']
    preferensi_hari = [h.strip().capitalize() for h in pasien['Preferensi Hari'].split(',')]
    preferensi_jam = konversi_preferensi_jam(pasien['Preferensi Jam'])

    kandidat = []
    for nama_dokter, jadwal in jadwal_dokter_dict.items():
        if jadwal['Poli'] == poli:
            hari_cocok = [h for h in preferensi_hari if h in jadwal['Hari']]
            if hari_cocok:
                jam_dokter = generate_slots(f"{jadwal['Jam Mulai']}--{jadwal['Jam Selesai']}")
                slot_valid = list(set(preferensi_jam) & set(jam_dokter))
                if slot_valid:
                    kandidat.append((nama_dokter, random.choice(hari_cocok),
                                     random.choice(slot_valid)))
    if kandidat:
        dokter_baru, hari_baru, slot_baru = random.choice(kandidat)
        jadwal_baru.at[index_acak, 'Dokter'] = dokter_baru
        jadwal_baru.at[index_acak, 'Hari'] = hari_baru
        jadwal_baru.at[index_acak, 'Slot'] = slot_baru

    return jadwal_baru
```

Implementasi Simulated Annealing

```
def simulated_annealing(jadwal_awal, df_pasien, jadwal_dokter_dict, max_iter=1000, T=1000,
cooling_rate=0.99):
    current = jadwal_awal.copy()
    best = current.copy()
    current_cost = hitung_biaya(current, df_pasien)
    best_cost = current_cost

    for i in range(max_iter):
        neighbor = generate_neighbor(current, df_pasien, jadwal_dokter_dict)
        neighbor_cost = hitung_biaya(neighbor, df_pasien)

        delta = neighbor_cost - current_cost

        if delta < 0 or random.random() < pow(2.718, -delta / T):
            current = neighbor
            current_cost = neighbor_cost

            if current_cost < best_cost:
                best = current
                best_cost = current_cost

        T *= cooling_rate

        if i % 100 == 0:
            print(f"Iterasi {i}: Biaya = {current_cost}")

    return best
```

Jadwal Final

```
jadwal_final = simulated_annealing(df_jadwal_awal, df_pasien, jadwal_dokter_dict, max_iter=1000)
jadwal_final.to_excel("jadwal_simulated_annealing.xlsx", index=False)
```


SURAT KETERANGAN BEBAS PLAGIAT



MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN

Alamat kantor: Jl.Sultan Alauddin No.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Muh. Zubir Pradadang

Nim : 105841103119

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	3%	10 %
2	Bab 2	24%	25 %
3	Bab 3	2%	15 %
4	Bab 4	0%	10 %
5	Bab 5	5%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 28 Agustus 2025

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

● BAB I Muh. Zubir Pradadang
105841103119

by Tahap Tutup

Submission date: 27-Aug-2025 03:57PM (UTC+0700)

Submission ID: 2736078486

File name: Skripsi_Muh_Zubir_Pradadang_bab_1.docx (39.51K)

Word count: 924

Character count: 6433

BAB I Muh. Zubir Pradadang 105841103119

ORIGINALITY REPORT

3%

SIMILARITY INDEX

3%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

digilibadmin.unismuh.ac.id

Internet Source

3%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off

BAB II Muh. Zubir Pradadang 105841103119

by Tahap Tutup

Submission date: 27-Aug-2025 03:58PM (UTC+0700)

Submission ID: 2736078594

File name: Skripsi_Muh_Zubir_Pradadang_bab_2.docx (56.75K)

Word count: 2544

Character count: 16896

BAB II Muh. Zubir Pradadang 105841103119

ORIGINALITY REPORT

24%

SIMILARITY INDEX

23%

INTERNET SOURCES

6%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1	repository.its.ac.id Internet Source	19%
2	calonarsitek35.blogspot.com Internet Source	2%
3	Ansye Yasinta Wowor, Siswati Siswati. "TINJAUAN LAMA WAKTU TUNGGU PELAYANAN PASIEN BPJS RAWAT JALAN KLINIK SARAF", Jurnal Kesehatan Tambusai, 2022 Publication	1%
4	eprints.poltekkesjogja.ac.id Internet Source	<1%
5	repository.ub.ac.id Internet Source	<1%
6	aepnurulhidayat.wordpress.com Internet Source	<1%
7	docplayer.info Internet Source	<1%
8	sipora.polije.ac.id Internet Source	<1%
9	www.ejurnalmalahayati.ac.id Internet Source	<1%
10	www.igi-global.com Internet Source	<1%

BAB III Muh. Zubir Pradadang

105841103119

by Tahap Tutup

Submission date: 27-Aug-2025 03:58PM (UTC+0700)

Submission ID: 2736078649

File name: Skripsi_Muh_Zubir_Pradadang_bab_3.docx (126.95K)

Word count: 1338

Character count: 9030

BAB III Muh. Zubir Pradadang 105841103119

ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

2%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

digilibadmin.unismuh.ac.id

Internet Source

2%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off

BAB IV Muh. Zubir pradadang
105841103119

by Tahap Tutup

Submission date: 27-Aug-2025 04:25PM (UTC+0700)

Submission ID: 2736087240

File name: Skripsi_Muh_Zubir_Pradang_bab_4.docx (1.03M)

Word count: 1531

Character count: 9873

BAB IV Muh. Zubir pradadang 105841103119

ORIGINALITY REPORT

0%

SIMILARITY INDEX

%

INTERNET SOURCES

%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

Source: UPT Perpustakaan dan Penerbitan

Source: UPT Perpustakaan dan Penerbitan

Source: UPT Perpustakaan dan Penerbitan

Source: UPT Perpustakaan dan Penerbitan

● BAB V Muh. Zubir Pradadang
105841103119

by Tahap Tutup

Submission date: 27-Aug-2025 03:59PM (UTC+0700)

Submission ID: 2736078735

File name: Skripsi_Muh_Zubir_Pradadang_bab_5.docx (36.03K)

Word count: 616

Character count: 4272

BAB V Muh. Zubir Pradadang 105841103119

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

adoc.pub
Internet Source

2%

2

www.coursehero.com
Internet Source

2%

3

www.juancarlosvarela.com
Internet Source

2%

Exclude quotes

Off

Exclude matches

< 2%

Exclude bibliography

Off