

**RANCANG BANGUN SISTEM INFORMASI KLINIK  
KEPERAWATAN LUKA**

**Skripsi**

**Disusun untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Komputer**



**Oleh:  
Muhammad Insan Khamil  
3145161580**

**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS NEGERI JAKARTA**

**2023**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>iii</b>
<b>DAFTAR GAMBAR</b>	<b>iv</b>
<b>DAFTAR TABEL</b>	<b>v</b>
<b>I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang Masalah . . . . .	1
1.2 Rumusan Masalah . . . . .	5
1.3 Pembatasan Masalah . . . . .	5
1.4 Tujuan Penelitian . . . . .	6
1.5 Manfaat Penelitian . . . . .	6
<b>II KAJIAN PUSTAKA</b>	<b>7</b>
2.1 Sistem Informasi . . . . .	7
2.2 Unified Modeling Language (UML) . . . . .	8
2.2.1 <i>Use Case Diagram</i> . . . . .	8
2.2.2 <i>Class Diagram</i> . . . . .	8
2.2.3 <i>Activity Diagram</i> . . . . .	9
2.2.4 <i>Sequence Diagram</i> . . . . .	9
2.3 Pengantar <i>Flask</i> . . . . .	9
2.4 MongoDB . . . . .	12
2.5 Scrum . . . . .	16
<b>III METODOLOGI PENELITIAN</b>	<b>22</b>
3.1 Pengumpulan Data . . . . .	22

3.2	Analisa Kebutuhan . . . . .	23
3.3	Perancangan Sistem Menggunakan <i>Scrum</i> . . . . .	23
3.3.1	<i>Product Backlog</i> . . . . .	24
3.3.2	<i>Sprint Backlog</i> . . . . .	25
3.3.3	<i>Sprint</i> . . . . .	25
3.3.4	<i>Deploy</i> . . . . .	25
<b>DAFTAR PUSTAKA</b>		<b>28</b>

## DAFTAR GAMBAR

Gambar 2.1	Contoh registrasi Blueprint pada dokumen “auth.py”.	
	Sumber:	
	<a href="https://flask.palletsprojects.com/en/2.1.x/tutorial/views/">https://flask.palletsprojects.com/en/2.1.x/tutorial/views/</a>	10
Gambar 2.2	Dokumen <code>__init__.py</code> .	Sumber:
	<a href="https://flask.palletsprojects.com/en/2.1.x/tutorial/views/">https://flask.palletsprojects.com/en/2.1.x/tutorial/views/</a>	11
Gambar 2.3	Contoh pembuatan <i>routing</i> dengan <i>Blueprint</i> .	Sumber:
	<a href="https://flask.palletsprojects.com/en/2.1.x/tutorial/views/">https://flask.palletsprojects.com/en/2.1.x/tutorial/views/</a>	11
Gambar 2.4	Pemodelan RDBMS.	Sumber:
	<a href="https://www.mongodb.com/nosql-explained/">https://www.mongodb.com/nosql-explained/</a>	12
Gambar 2.5	Pemodelan NoSQL.	Sumber:
	<a href="https://www.mongodb.com/nosql-explained/">https://www.mongodb.com/nosql-explained/</a>	13
Gambar 2.6	Menambahkan Ekstensi PyMongo pada Flask.	Sumber:
	Dokumentasi PyMongo,	
	<a href="https://flask-pymongo.readthedocs.io/en/latest/">https://flask-pymongo.readthedocs.io/en/latest/</a>	14
Gambar 3.1	Tahapan penelitian	22

## DAFTAR TABEL

Tabel 2.1	Fungsi-fungsi Metode HTTP . . . . .	10
Tabel 3.1	<i>Product Backlog</i> . . . . .	24

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Luka kronis adalah masalah kritis dalam kesehatan. Di Amerika Serikat, sekitar 6,5 juta orang menderita luka kronis dan biaya perawatan luka kronis menghabiskan sekitar \$20 miliar per tahun. Bahkan di negara maju, sekitar 1-2% dari seluruh populasi terkena luka kronis selama hidup mereka. (Biswas et al., 2018). Luka kronis berdampak terhadap finansial dan penurunan kualitas hidup pasien. Kerusakan fisik, sosial, dan emosional seperti penurunan mobilitas, rasa sakit, ketidaknyamanan, membatasi kinerja aktivitas sehari-hari. Isolasi sosial, frustrasi, dan reaksi psikologis lainnya yang menimbulkan dampak pada kehidupan pasien. (Vogt et al., 2020). Di Indonesia sendiri pengidap luka kronis berjumlah sekitar 24% dari 8,6% total populasi terhadap kasus diabetes. (Safitri et al., 2022).

Pada penelitian sebelumnya yang dilakukan oleh Salsa yang berjudul "Rancang Bangun Aplikasi dan *Web Service* Pengkajian Luka Kronis Khusus Modul Pengolahan Citra Berbasis Android". Salsa melakukan wawancara dengan Ratna Aryani, M.Kep., Dosen Politeknik Negeri Jakarta I, diperoleh bahwa saat melakukan penggantian balutan luka dan pengecekan awal kondisi luka dilakukan pengkajian luka. Berikut langkah-langkah pengkajian luka diawali dengan balutan luka dibuka, lalu luka dicuci, dan diakhiri dengan proses pengkajian luka. Instrumen yang dipilih saat melakukan pengkajian luka ialah Bates-Jensen *Wound Assesment Tools* (BWAT). Pada BWAT ada 13 kategori penilaian yakni beberapa di antaranya tepi luka, ukuran luka, epitalisasi dan jumlah eksudat (cairan tubuh yang keluar dari jaringan selama peradangan). Saat ini data pengkajian luka masih dilakukan secara

tradisional dicatat dalam arsip atau catatan kertas, maka dari itu salsa mengusulkan untuk mendigitalisasi pencatatan data luka yang sudah dikaji. (Rahmadati, 2023).

Penelitian lain yang terkait juga dilakukan oleh Ardiansyah, menjelaskan bahwa Rumah Sakit Umum Kambang Jambi masih memakai cara tradisional dalam pelayanannya seperti mendapatkan nomor antrian berobat, informasi mengenai jadwal dokter dan jumlah seluruh pasien. Hal ini menyebabkan pemanfaatan informasi menjadi kurang maksimal, berjalan kurang efektif dan lama dalam prosesnya. Dengan adanya permasalahan yang terjadi ardiansyah dan kawannya menyimpulkan bahwa dibutuhkannya Sistem Informasi Manajemen Rumah Sakit Berbasis *Website* untuk menyelesaikan permasalahan yang ada. Sehingga meminimalkan kekurangan dan ketidak efektifan dalam pelayanan. (Ardiansyah and Effiyaldi, 2021).

Dalam jurnal berjudul "Sistem Informasi Rekam Medis Pada Rumah Sakit Umum Daerah (RSUD) Pacitan Berbasis *Web Base*" oleh Gunawan Susanto. Pencatatan riwayat dan data rekam medis kesehatan milik pasien merupakan hal yang krusial dalam dunia medis karena data tersebut digunakan untuk pemeriksaan pasien selanjutnya. Sistem pencatatan yang dipakai memiliki kelemahan. Hal ini dikarenakan data rekam medis pasien hanya disimpan secara lokal di tempat pasien diperiksa dan dirawat serta pertukaran data langsung antara divisi medis tidak diperbolehkan. Maka dari itu dilakukan pengembangan sistem informasi rekam medis yang memiliki tujuan untuk menyelesaikan kelemahan yang dimiliki oleh sistem pencatatan rekam medis pasien yang sebelumnya, yaitu alternatif teknologi yang dapat diterapkan di masa yang akan datang untuk pencatatan dan penyampaian data rekam medis. (Susanto and Sukadi, 2021).

Pada penelitian yang dilaksanakan oleh Inah Carminah yang berjudul "Aplikasi Monitoring Perawatan Luka Diabetes Melitus Berbasis *Website*". Proses

pelayanan yang masih menggunakan *paper base system* memiliki risiko kerusakan atau kehilangan data rekam medis pasien. Selain itu membuat perawat kewalahan ketika mencari data rekam medis pasien secara satu-persatu ketika dibutuhkan ketika pasien datang untuk berobat kembali. Berangkat dari permasalahan di atas memotivasi instansi untuk membuat aplikasi dengan tujuan untuk meningkatkan kualitas pelayanan pasien saat berobat. (Carminah et al., 2021).

Didalam buku berjudul “Rancang Bangun Aplikasi *Mobile* Android Sebagai Alat Deteksi Warna Dasar Luka Dalam Membantu Proses Pengkajian Luka Kronis Dengan Nekrosis”, Teknik pengkajian luka berdasarkan warna luka yang umum digunakan salah satunya The RYB (*Red-Yellow-Black*) *wound classification system*. Metode ini digunakan dengan mengandalkan subyektifitas dari perawat luka. Hasil penelitian pada buku ini menunjukkan bahwa perawat mampu mengetahui perbedaan warna luka secara otomatis yang membantu proses pengkajian luka kronis dengan nekrosis. (Aryani et al., 2018). Ia juga meneliti dan menemukan bahwa perban basah membantu mempercepat proses penyembuhan luka. Perawat harus mempertimbangkan untuk menggunakan balutan basah daripada perawatan standar untuk meningkatkan penyembuhan. Namun, perawat harus melindungi luka dari kelembapan yang berlebihan karena dapat merusak kulit di sekitar luka atau di dalam luka. (Aryani, 2016).

Pada payung penelitian *medical imaging* yang sama dengan peneliti juga sudah pernah dilakukan penelitian mengenai Pengaruh Penggunaan *Color Model* LAB dalam Kalibrasi Warna Luka Menggunakan Metode Segmentasi *K-Means* dan *Mean Shift* oleh rekan sesama peneliti. (Khairunnisa, 2021). Dan Muhamad rizki juga melakukan penelitian deteksi tepi luka menggunakan metode *Active Contour* yang ditambah interpolasi. (Rizki, 2022). Kedua penelitian tersebut merupakan penelitian berdasarkan dua kategori pengkajian luka yaitu warna luka dan tepi luka,



algoritma yang dikembangkan pada penelitian tersebut direncanakan akan terintegrasi dalam satu ekosistem aplikasi, yakni sistem informasi keperawatan luka. Dimana pada penelitian Salsa Rahmadati melakukan perancangan aplikasi pengkajian luka kronis berbasis Android sesuai modul *image processing*. (Rahmadati, 2023).

Berdasarkan hal di atas, penulis tertarik untuk melakukan penelitian yang bertujuan untuk membuat sistem informasi keperawatan luka dengan dasar pengembangan menggunakan data paparan presentasi bersama ibu Irma Puspita Arisanti selaku pemilik klinik *moist care* dan sesuai dengan proposal PKM-PI dengan judul "Pengembangan Pelayanan Sistem Informasi Klinik Serta Fitur Keperawatan Luka Pada Aplikasi Untuk Mendukung Integrasi Data Kesehatan Dan Ketahanan Nasional Bidang Kesehatan" yang dibuat oleh Hafiz dan tim. Melanjutkan penelitian sebelumnya yang dilakukan oleh Salsa, dimana pengkajian luka masih dilakukan dengan cara manual atau arsip kertas sehingga Salsa membuat aplikasi untuk mengarsipkan data secara digital dan peneliti mengembangkan *web* aplikasi yang berkaitan dengan aplikasi sebelumnya untuk dapat diakses datanya oleh klinik dengan maksud seluruh staff klinik yang berkepentingan dapat dengan mudah mengaksesnya.

Sistem Informasi tersebut diharapkan dapat menambah opsi pendaftaran berobat secara *online* selain daripada pendaftaran secara *offline*, membantu pengelolaan antrian, membantu integrasi data pasien dan perawat secara digital, manajemen inventaris, beserta verifikasi dan validasi biaya tagihan sehingga dapat mempermudah pelayanan.

## 1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang yang diutarakan di atas, maka perumusan masalah pada penelitian ini adalah bagaimana membuat rancang bangun sistem informasi keperawatan luka?

## 1.3 Pembatasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Sistem informasi keperawatan luka dibuat dengan dasar instrumen pengkajian Bates-Jensen *Wound Assessment Tool* (BWAT).
2. *User* aplikasi sistem informasi keperawatan luka adalah perawat dan admin klinik.
3. Sistem informasi keperawatan luka dibuat berbasis *Website*
4. Sistem informasi keperawatan luka dibuat berdasarkan paparan presentasi bersama pemilik klinik *Moist Care* yaitu ibu Irma Puspita Arisanti.
5. Model pengembangan yang digunakan untuk mengembangkan sistem informasi keperawatan luka adalah scrum.
6. Fitur-fitur yang diimplementasi pada sistem informasi keperawatan luka, diantaranya adalah pembuatan akun pasien, *dashboard* klinik, pemeriksaan kesehatan dan sebagian proses pengobatan luka (*view* dan *web service* inventaris dan layanan).

#### **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk membuat rancang bangun sistem informasi keperawatan luka di klinik *Moist Care*.

#### **1.5 Manfaat Penelitian**

##### **1. Bagi penulis**

Penelitian yang dilakukan merupakan media penerapan dari berbagai ilmu pengetahuan, khususnya dalam perancangan sistem informasi keperawatan luka pada klinik *Moist Care*.

##### **2. Bagi Program Studi Ilmu Komputer**

Penelitian ini dapat menjadi pintu gerbang untuk penelitian selanjutnya di masa depan.

##### **3. Bagi Universitas Negeri Jakarta**

Menjadi evaluasi akademik program studi Ilmu Komputer dalam penulisan skripsi sehingga dapat meningkatkan kualitas pendidikan program studi Ilmu Komputer di Universitas Negeri Jakarta.

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Sistem Informasi**

Sistem informasi tersusun dari dua kata yaitu sistem dan informasi. Menurut Jerry FitzGerald, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. (FitzGerald et al., 1981)

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi yaitu *input*, proses dan *output*. Menurut Robert A. Leitch, Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (A. Leitch and Davis, 2001)

## 2.2 Unified Modeling Language (UML)

Subbab ini ditulis berdasarkan (Suendri, 2019). *Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar dalam memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). Siti Fatima mengatakan UML memberikan standar penulisan dalam sebuah sistem *blueprint* yang meliputi konsep bisnis proses, penulisan kelas-kelas yang spesifik dalam bahasa program, skema database, dan komponen-komponen yang dibutuhkan dalam sistem *software* Fatima (2013). Diagram *Unified Modelling Language* (UML) antara lain sebagai berikut :

### 2.2.1 Use Case Diagram

*Use case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. Model *use case* dapat diartikan sebagai diagram *use case*, tetapi diagram ini tidak sama dengan model sebab model memiliki cakupan yang lebih luas dari diagram. *Use case* harus sanggup dalam menggambarkan susunan atau urutan aktor yang menghasilkan nilai terukur. (Widodo and Prabowo, 2011)

### 2.2.2 Class Diagram

Kelas merupakan suatu set objek dengan atribut dan perilaku yang sama. Kelas dapat disebut juga sebagai kelas objek. (Whitten et al., 2004)

Kelas memiliki tiga area pokok yaitu :

1. Nama, kelas haruslah mempunyai sebuah nama.
2. Atribut, merupakan kelengkapan yang melekat pada kelas. Suatu kelas memiliki nilai yang hanya bisa diproses sebatas pada atribut yang dimiliki.

3. Operasi, merupakan proses yang dilakukan oleh sebuah kelas kepada kelas itu sendiri ataupun kelas lainnya.

### 2.2.3 *Activity Diagram*

Diagram *activity* menunjukkan aktivitas sistem yang berbentuk kumpulan dari aksi-aksi, bagaimana masing-masing aksi tersebut dinilai, keputusan yang mungkin terjadi hingga berakhirnya aksi. Selain itu, *activity diagram* dapat menggambarkan lebih dari satu proses aksi dalam waktu yang bersamaan. “*Activity diagram* adalah aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas”. (Haviluddin, 2011)

### 2.2.4 *Sequence Diagram*

“Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*.” (Haviluddin, 2011)

## 2.3 Pengantar *Flask*

*Flask* merupakan *web framework* yang memiliki dasar bahasa pemrograman *python*. *Web framework* adalah koleksi dari modul-modul dan *packages* yang membuat pengembang dapat membuat aplikasi *web* atau *web service* tanpa harus memikirkan detail-detail dasar seperti protokol, socket, atau manajemen proses.

*Core* yang dimiliki *flask* tergolong sederhana dan bersifat ringan, selain itu *flask* juga bersifat *simplicity* dan *flexibility* sehingga pengembangan dapat menyesuaikan dengan kebutuhan oleh penambahan ekstensi yang ada. Ekstensi yang dimiliki *flask* salah satunya adalah *blueprint*. *Blueprint* memiliki fungsi untuk

mempermudah dalam pembuatan pengaturan minimal RESTful APIs. RESTful APIs merupakan layanan atau metode yang berfungsi untuk mentransmisikan data dengan menggunakan protokol HTTP.

*Routing* pada *flask* diartikan sebagai bantuan ekstensi *blueprint* yang mempermudah akses kepada beberapa metode *Hypertext Transfer Protocol*(HTTP) hanya dengan mendefinisikan metode yang digunakan pada *routing* yang akan digunakan. Metode permintaan HTTP yang bisa gunakan antara lain sebagai berikut:

**Tabel 2.1:** Fungsi-fungsi Metode HTTP

Metode HTTP	Fungsi
GET	Menerima informasi dari server yang diberikan menggunakan URI yang spesifik. Permintaan menggunakan metode GET hanya menerima data tanpa adanya efek perubahan pada data.
POST	Mengirimkan data ke server seperti unggahan file, informasi pelanggan dan lain-lain menggunakan form HTML.
HEAD	Sama seperti metode GET, namun hanya memberikan data status dan seksi header saja.
PUT	Mengganti semua representasi dari target <i>resource</i> dengan konten yang diunggah.
DELETE	Menghapus semua representasi dari target yang didefinisikan pada URI.

```
import functools

from flask import (
    Blueprint, flash, g, redirect, render_template, request, session, url_for
)
from werkzeug.security import check_password_hash, generate_password_hash

from flaskr.db import get_db

bp = Blueprint('auth', __name__, url_prefix='/auth')
```

**Gambar 2.1:** Contoh registrasi Blueprint pada dokumen “auth.py”.

Sumber: <https://flask.palletsprojects.com/en/2.1.x/tutorial/views/>

**Gambar 2.1** berisi kode pada halaman sebelumnya berfungsi untuk menterjemahkan penggunaan ekstensi *blueprint* pada suatu dokumen bernama *'auth'*. Agar *routing* dapat berjalan maka harus di registrasikan pada dokumen *init.py* yang merupakan tempat *flask* akan berjalan.

```
def create_app():
    app = ...
    # existing code omitted

    from . import auth
    app.register_blueprint(auth.bp)

    return app
```

**Gambar 2.2:** Dokumen `__init__.py`.

Sumber: <https://flask.palletsprojects.com/en/2.1.x/tutorial/views/>

**Gambar 2.3** merupakan contoh dari pembuatan *routing log in* dengan URL *routing* *"/login"* yang mendefinisikan metode GET dan POST. Saat proses *log in* sukses maka akan diarahkan ke URL *"auth/login.html"*.

```
@bp.route('/login', methods=('GET', 'POST'))
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        db = get_db()
        error = None
        user = db.execute(
            'SELECT * FROM user WHERE username = ?', (username,)
        ).fetchone()

        if user is None:
            error = 'Incorrect username.'
        elif not check_password_hash(user['password'], password):
            error = 'Incorrect password.'

        if error is None:
            session.clear()
            session['user_id'] = user['id']
            return redirect(url_for('index'))

        flash(error)

    return render_template('auth/login.html')
```

**Gambar 2.3:** Contoh pembuatan *routing* dengan *Blueprint*.

Sumber: <https://flask.palletsprojects.com/en/2.1.x/tutorial/views/>



## 2.4 MongoDB

MongoDB adalah basis data yang menggunakan konsep *Not Only SQL* (NoSQL) yang menyimpan data berorientasikan dokumen. NoSQL tidak memiliki sistem tabular dan mempunyai perbedaan penyimpanan dari tabel relasional. *Database* dengan konsep NoSQL memberikan pengembang fleksibilitas untuk menyimpan struktur data dalam jumlah besar.

Kunci perbedaan NoSQL dan *Relational Database Management System* (RDBMS) ialah bagaimana sebuah data dimodelkan pada *database*. RDBMS menggunakan pemodelan yang masih menggunakan tabel berstruktur dengan setiap kolom baris bersifat tetap antara satu dengan lainnya, sedangkan pemodelan data pada NoSQL, khususnya pada MongoDB, menggunakan dokumen dimana setiap barisnya mempunyai kolom yang dapat berbeda dengan baris yang lain. berikut merupakan contoh perbedaan basis data menggunakan konsep RDBMS dan NoSQL dapat dilihat pada **Gambar 2.4** (Pemodelan RDBMS) dan **Gambar 2.5** (Pemodelan NoSQL).

Users				
ID	first_name	last_name	cell	city
1	Leslie	Yepp	8125552344	Pawnee

Hobbies		
ID	user_id	hobby
10	1	scrapbooking
11	1	eating waffles
12	1	working

**Gambar 2.4:** Pemodelan RDBMS.

Sumber: <https://www.mongodb.com/nosql-explained/>

```
{
  "_id": 1,
  "first_name": "Leslie",
  "last_name": "Yepp",
  "cell": "8125552344",
  "city": "Pawnee",
  "hobbies": ["scrapbooking", "eating waffles", "working"]
}
```

**Gambar 2.5:** Pemodelan NoSQL.

Sumber: <https://www.mongodb.com/nosql-explained/>

Pada **Gambar 2.4** pada halaman sebelumnya, digunakan untuk menyimpan data *user* dan data *hobbies* dibutuhkan dua tabel terpisah dimana hal ini tidak dibutuhkan pada pemodelan NoSQL (**Gambar 2.5**) yang dapat menggabungkan dua data *user* dan *hobbies* pada satu dokumen serta baris yang sama. Dengan NoSQL ketika ingin memanggil dua data tersebut secara bersamaan hanya membutuhkan satu dokumen saja tanpa menggunakan *joins*, yang menghasilkan *queries* jauh lebih cepat dibandingkan dengan RDBMS.

#### 1. Integrasi MongoDB dan *Flask*

*Database* MongoDB dapat diintegrasikan dengan *framework flask* dengan menggunakan ekstensi yang tersedia, PyMongo adalah salah satunya. PyMongo memiliki perintah yang sama dengan perintah CLI MongoDB diantaranya membuat data, mengakses data, dan memodifikasi data. Untuk mengintegrasikan Flask dan MongoDB diperlukan terlebih dahulu untuk menginisialisasikan proyek *flask* dan mengimpor ekstensi Flask-PyMongo.

```

from flask import Flask
from flask_pymongo import PyMongo

app = Flask(__name__)
app.config["MONGO_URI"] = "mongodb://localhost:27017/myDatabase"
mongo = PyMongo(app)

```

**Gambar 2.6:** Menambahkan Ekstensi PyMongo pada Flask.

Sumber: Dokumentasi PyMongo, <https://flask-pymongo.readthedocs.io/en/latest/>

Inisialisasi MongoDB pada proyek *flask* dilakukan dengan menggunakan konstruktor PyMongo yang menerima objek app Flask dan URI string dari *database* MongoDB. Setelah *flask* dan MongoDB terintegrasi, fungsi-fungsi yang dapat kita lakukan adalah sebagai berikut:

(a) Membuat Dokumen

Metode PyMongo yang digunakan untuk menambahkan data ke dalam *database* adalah `db.colection.insert_one()` jika terdapat hanya satu data dan `db.colection.insert_many()` jika terdapat lebih dari satu data. Untuk menambahkan dokumen ke dalam koleksi MongoDB, diperlukan untuk mendefinisikan *dictionary* yang terdiri atas *fields* dan *values*.

```

@bp.route("/add_many")
def add_many():
    db.colection.insert_many([
        {'_id': 1, 'judul': "todo title one ", 'desc': "desc body one "},
        {'_id': 2, ' judul ': "todo title two", ' desc': " desc body two"},
        {'_id': 3, ' judul ': "todo title three", 'desc': " desc body three"},
        {'_id': 4, ' judul ': "todo title four", 'desc': " desc body four"},
        {'_id': 5, ' judul ': "todo title five", 'desc': " desc body five"},
        {'_id': 1, ' judul ': "todo title six", 'desc': " desc body six"},
    ])
    return flask.jsonify('message':True)

```

Ketika mendefinisikan lebih dari satu data yang sama *BulkWriteError*

akan muncul, yang berarti hanya ada satu data yang terekam dan data lainnya yang sama akan hilang. Untuk mencegah hal tersebut, parameter *ordered* pada fungsi *insert\_many()* harus didefinisikan sebagai *false* kemudian menangkap eksepsi *BulkWriteError*.

#### (b) Membaca Dokumen

Flask-PyMongo memiliki beberapa metode dalam menerima data dari *database*. Penerimaan semua dokumen dari koleksi menggunakan metode *find()* untuk menerima semua data di *database* dan *find\_one()* untuk menerima satu data sesuai dengan ID yang diberikan. Metode *find()* dapat menerima parameter yang digunakan sebagai *filter*. Parameter *filter* yang digunakan menjelaskan diksi yang mendefinisikan properti yang akan dicari.

#### (c) Memperbaharui dan Mengganti Dokumen

Metode yang digunakan dalam memperbaharui data pada *database* adalah *update\_one()* atau *replace\_one()*. Metode *replace\_one()* mempunyai beberapa argumen sebagai berikut:

- i. *Filter*: berupa *query* yang mendefinisikan data pada ID yang akan diganti,
- ii. *Replacement*: berupa data yang akan menggantikan data yang dihapus.
- iii. *Upsert*: adalah opsi *boolean* yang jika dijadikan sebagai *true* dapat membuat dokumen baru jika tidak terdapat target dokumen yang dimaksud.

#### (d) Menghapus Dokumen

PyMongo menyediakan dua metode untuk menghapus satu atau lebih

koleksi *database* yaitu, *delete\_one()* untuk menghapus satu koleksi dan *delete\_many()* untuk menghapus beberapa koleksi.

```
@bp.route("/delete_todo/<int:ID>", methods=['DELETE'])
def delete_todo(ID):
    todo = db.todos.delete_one({'_id': 1})
    return todo.raw_results
```

Contoh kode di atas ketika menjalankan *request* seperti `http://localhost:5000/delete_todo/5` PyMongo akan mencari entri berdasarkan ID yang diberikan dan menghapusnya.

#### (e) Menyimpan dan Menerima *Files*

MongoDB mengizinkan pengembang untuk menyimpan data biner ke dalam *database* menggunakan spesifikasi GridFS. Ekstensi Flask-PyMongo menyediakan metode *save\_file()* untuk menyimpan *file* ke GridFS dan metode *send\_file()* untuk menerima *file* dari GridFS

```
@bp.route("/uploads/<filename>", methods=["POST"])
def save_upload(filename):
    mongo.save_file(filename, request.files["file"])
    return redirect(url_for("get_upload", filename=filename))
```

Kode di atas, dibuat form untuk menangani unggahan file dan mengembalikan nama file yang telah terunggah.

## 2.5 Scrum

*Scrum* merupakan salah satu struktur kerja yang digunakan untuk mengembangkan produk. *Scrum* diumumkan pertama kali oleh Ken Schwaber pada tahun 1995 pada konferensi Austin, namun fondasi daripada metode *scrum* sudah

ada sejak tahun 1980 (Ozierańska et al., 2016). *Scrum* dibuat berdasarkan empirisme yang dicapai dengan beberapa kualitas. Hasil survei dari literatur, kualitas yang membangun empirisme *scrum* adalah kejelasan dari setiap proses, inspeksi untuk mendeteksi masalah dan adaptasi terhadap perubahan

Setiap produk dihantarkan dengan cara yang fleksibel dan iteratif dalam kerangka kerja *scrum* dimana setiap akhir *sprint* terdapat produk nyata yang dapat dihantarkan. *Requirement* yang dibutuhkan dalam suatu proyek berupa *product backlog* yang diperbaharui secara berkala.

*Scrum* mempunyai tiga elemen, di antaranya:

#### 1. *Roles*

*Role* dalam *scrum* terbagi menjadi empat *role* utama, yaitu:

##### (a) *Tim Scrum*

Tim *scrum* merupakan kelompok kecil yang terdiri dari satu *scrum master*, satu *product owner*, dan pengembang. Pada tim *scrum* tidak terdapat tim kecil ataupun hierarki. Seluruh anggota tim memiliki kemampuan penting untuk memberikan nilai ke dalam setiap *sprint* dan fokus dengan satu tujuan pada satu waktu, *product goal*.

Tim *scrum* bertanggung jawab dalam setiap aktivitas produk seperti kolaborasi dengan *stakeholder*, *maintenance*, verifikasi, *research*, *operation*, *experimentation* dan pengembangan. Tim *scrum* menghantarkan produk secara *iterative* menggunakan *sprint*, oleh karena itu tim *scrum* juga bertanggung jawab untuk menciptakan nilai pada setiap *sprint*-nya.

(b) *Scrum Master*

*Scrum master* memiliki tanggung jawab dalam merealisasikan *scrum* yang terdefinisi pada panduan *scrum*. Setiap anggota tim dibantu *scrum master* untuk mengerti bagaimana teori dan praktik kerangka kerja pada metode *scrum*. Selain itu, menjaga efektivitas dari tim *scrum* juga menjadi tanggung jawab *scrum master*.

(c) *Product Owner*

*Product owner* memiliki tanggung jawab untuk meningkatkan nilai komersial produk yang dihasilkan oleh *development team* dan mengelola *product backlog* agar lebih maksimal. Hanya *product owner* yang memiliki tanggung jawab untuk mengelola *product backlog*. Adapun pengelolaan *product backlog*:

- i. Penyampaian isi *product backlog*.
- ii. Memastikan *development team* memahami *product backlog*.
- iii. Memastikan isi daripada *product backlog* transparan dan jelas bagi seluruh anggota tim.
- iv. Mengurutkan item pada *product backlog* untuk mencapai tujuan secara optimal.

(d) *Development Team*

*Development team* atau tim pengembang adalah profesional yang mengeksekusi isi yang tercantum di dalam *product backlog*. Tim Pengembang berkomitmen untuk membuat semua aspek *increment* yang dapat berfungsi pada setiap *sprint*. Namun, tim Pengembang juga selalu bertanggung jawab untuk:

- i. Membuat rancangan *sprint* atau dikenal dengan *sprint backlog*.

- ii. Membuat definisi penyelesaian sebuah *task*.
- iii. Mengadaptasikan semua *plan* setiap hari sampai *sprint goal*.
- iv. Mengurutkan *item* pada *product backlog* untuk mencapai tujuan secara optimal.

## 2. Artifacts

Artefak *scrum* dirancang untuk memaksimalkan transparansi informasi utama dan kesempatan untuk menginspeksi dan mengadaptasi.

### (a) Product Backlog

*Product backlog* atau umumnya disebut dengan *user stories* merupakan kumpulan fitur-fitur yang terdapat pada suatu produk . *User stories* dapat ditambahkan, dimodifikasi, atau dihilangkan dari *product backlog* selama proyek berjalan.

### (b) Sprint Backlog

*Sprint backlog* adalah beberapa *user stories* yang diambil dari *product backlog* untuk dijalankan pada satu *sprint*. *Sprint backlog* mencakup seluruh kegiatan kerja yang diperlukan untuk mencapai *sprint goal*.

Pada satu *sprint* terdapat *increment* yang merupakan manifestasi dari *user stories* yang diselesaikan dan total *increment* dari seluruh *sprint* sebelumnya.

## 3. Events

*Event* merupakan wadah dari semua *event* yang terdapat pada *scrum*. *Event* dibuat sebagai perwujudan salah satu dari tiga pilar *scrum*. Seluruh *event* berjalan secara bersamaan untuk mengurangi kompleksitas.



(a) *Sprint*

*Sprint* adalah komponen utama kerangka kerja *scrum*, dimana sebuah ide menjadi sebuah nilai. Lama durasi *sprint* bersifat tetap yaitu satu hingga empat minggu untuk menjaga konsistensi.

*Sprint* berfokus untuk menghantarkan beberapa *user stories* pada *product backlog*. Setiap satu *sprint* memiliki beberapa kegiatan diantaranya *sprint planning*, *sprint review* dan *sprint retrospective*.

(b) *Sprint Planning*

*Sprint backlog* adalah beberapa *user stories* yang diambil dari *product backlog* untuk dijalankan pada satu *sprint*. *sprint backlog* mencakup semua kegiatan kerja yang dibutuhkan untuk mencapai *sprint goal*.

Sebelum memulai *sprint*, perencanaan apa yang akan dilaksanakan pada saat *sprint* dilakukan pada saat *sprint planning* oleh seluruh anggota tim *scrum*. Waktu untuk melaksanakan *sprint planning* terbatas dengan lama durasi hingga delapan jam. Fungsi dari *sprint planning* adalah memutuskan apa yang dapat dimasukkan ke dalam *increment* dari *sprint* dan bagaimana penyelesaian yang dibutuhkan untuk menghantarkan *increment*.

(c) *Daily Scrum*

*Daily scrum* adalah kegiatan 15 menit bagi para pengembang untuk memeriksa perkembangan menuju *sprint goal* dan menyesuaikan pekerjaan yang akan dikerjakan selama 24 jam ke depan. Pada kegiatan *daily scrum*, hal apa saja yang akan dikerjakan hari ini, apa yang telah dikerjakan kemarin dan hambatan yang telah dialami dalam mencapai *sprint goal* akan didiskusikan oleh tim pengembang.

(d) *Sprint Review*

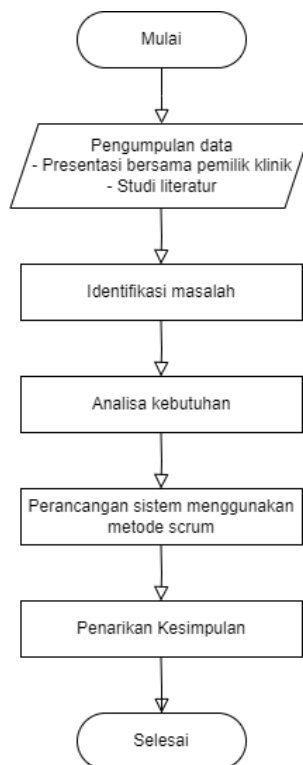
Tahap ini dilaksanakan pada akhir *sprint*, tujuannya untuk mengawasi apa yang telah diselesaikan di *sprint*. Menurut hasil tinjauan serta perubahan *product backlog*, tim *scrum* menentukan kembali pekerjaan selanjutnya yang dapat mengoptimalkan produk.

*Sprint review* bersifat informal dan diselenggarakan dengan lama durasi empat jam untuk sprint dalam waktu satu bulan. Semakin singkat durasi *sprint*, maka semakin singkat juga durasi *sprint review*.

## BAB III

### METODOLOGI PENELITIAN

Melalui penelitian yang dilakukan oleh penulis, akan menghasilkan produk tertentu. Penelitian yang dilakukan oleh penulis juga termasuk dalam jenis penelitian dan pengembangan. Berikut adalah tahapan-tahapan penelitian yang penulis lakukan dalam perancangan sebuah aplikasi:



**Gambar 3.1:** Tahapan penelitian

#### 3.1 Pengumpulan Data

Peneliti mengambil data dari presentasi bersama dengan pemilik klinik *moist care* dan klien dari penelitian ini yaitu ibu Irma Puspita Arisanti. Untuk dokumentasi foto pada saat presentasi dapat dilihat pada Lampiran A. Peneliti juga melakukan

studi literatur dengan membaca jurnal-jurnal yang berkaitan dengan topik penelitian serupa.

### 3.2 Analisa Kebutuhan

Berikut merupakan perangkat keras dan perangkat lunak yang penulis butuhkan dalam merancang sistem informasi keperawatan luka:

Perangkat keras berupa:

1. Laptop dengan spesifikasi Processor Intel Core i5 generasi ke-3 dan RAM 12 GB.

Perangkat lunak berupa:

1. Windows 10 *Operating System*.
2. Figma sebagai alat untuk mendesain tampilan UI/UX.
3. *Visual Studio Code* untuk pembuatan sistem informasi keperawatan luka.
4. *Python* sebagai bahasa pemrograman yang peneliti gunakan.
5. *Flask* sebagai web framework yang akan digunakan.
6. MongoDB sebagai basis data

### 3.3 Perancangan Sistem Menggunakan *Scrum*

*Website* aplikasi yang dibuat dalam penelitian ini dikembangkan dengan menggunakan metode *scrum*. Penjelasan rinci tentang metode *scrum* akan disajikan pada sub bab di bawah ini.

### 3.3.1 *Product Backlog*

Tahap *Product backlog* ini berfungsi untuk menterjemahkan seluruh fitur yang akan diimplementasikan pada aplikasi. Rincian *Product Backlog* yang akan diimplementasikan pada *website* aplikasi dapat dilihat pada tabel di bawah ini.

**Tabel 3.1:** *Product Backlog*

No	User Story	Priority	Sprint No.
1	Pembuatan akun pasien	High	1 & 2
2	Dashboard klinik	High	1 & 2
3	Pemeriksaan kesehatan	High	1 & 3
4	Proses pengobatan luka	High	1 & 3
5	Pendaftaran pasien berobat	High	1 & 3
6	Pengelolaan antrian	High	1 & 4
7	Administrasi keuangan	High	1 & 4

*Product backlog* yang dibuat memiliki 4 kolom yang di antaranya adalah sebagai berikut:

1. *User Story*

Kolom *user story* berisi fitur-fitur yang akan dibuat pada aplikasi.

2. *Priority*

Kolom *priority* berisi tingkat prioritas dari *user story*, dimana prioritas *high* merupakan fitur yang mempunyai peran penting pada penelitian ini.

3. *Sprint No.*

Kolom *sprint no.* berisi informasi tentang urutan pengerjaan fitur *sprint* tersebut akan dibuat.

### 3.3.2 *Sprint Backlog*

Sebelum *sprint* dimulai dilakukan *sprint backlog*, *sprint backlog* berisikan daftar pekerjaan yang keputusannya diambil dari *product backlog*. Dengan adanya *sprint backlog* semua anggota tim bisa melihat perkembangan dari setiap pekerjaan. Pada penelitian peneliti menggunakan tiga status perkembangan yaitu harus dikerjakan, sedang dikerjakan, selesai, next *sprint* dan tidak selesais.

### 3.3.3 *Sprint*

Setelah perencanaan *sprint backlog* sudah dibuat, maka pengerjaan *sprint* sudah bisa dimulai dan mengikuti jadwal pengerjaan yang telah disepakati bersama tim. Interval *sprint* yang digunakan adalah dua minggu.

### 3.3.4 *Deploy*

Setelah semua pekerjaan *sprint* yang telah direncanakan pada *sprint backlog* selesai maka aplikasi akan di *deploy*.

## DAFTAR PUSTAKA

- A. Leitch, R. and Davis, K. R. (2001). *Sistem Informasi*. PT. Prenhallindo.
- Ardiansyah and Effiyaldi (2021). Analisis dan Perancangan Sistem Informasi Manajemen Rumah Sakit Berbasis Website Pada Rumah Sakit Umum Kambang Kota Jambi. *Jurnal Ilmiah Sistem Informasi, Teknologi Informasi dan Sistem Komputer*, 6(1):188–197.
- Aryani, R. (2016). Accelerating Wound Healing Process By Using Moist Dressing.
- Aryani, R., Yusro, M., Suryana, M. E., and Firmansyah, I. (2018). Rancang Bangun Aplikasi Mobile Android Sebagai Alat Deteksi Warna Dasar Luka Dalam Membantu Proses Pengkajian Luka Kronis Dengan Nekrosis.
- Biswas, T., Fauzi, M. F. A., Abas, F. S., and Nair, H. K. (2018). Superpixel Classification with Color and Texture Features for Automated Wound Area Segmentation. *2018 IEEE 16th Student Conference on Research and Development, SCORED 2018*, pages 1–6.
- Carminah, I., Suheryadi, A., and Puspaningrum, A. (2021). Aplikasi Monitoring Perawatan Luka Aplikasi Monitoring Perawatan Luka Diabetes Mellitus Berbasis Website. *Seminar Nasional Teknologi Terapan (SEMITERA)*, pages 130–138.
- Fatima, S. (2013). Perancangan Sistem Informasi Penjualan Mebel Online pada UD. Melindo Jaya. *Kisaran: AMIK Royal Kisaran*.
- FitzGerald, J., Stallings, W. D., and Fitzgerald, A. F. (1981). *Fundamentals of systems analysis*. John Wiley & Sons, Inc.
- Haviluddin, H. (2011). Memahami Penggunaan UML (Unified Modelling Language).
- 6.

- Khairunnisa, A. (2021). Pengaruh Penggunaan Color Model Lab Dalam Kalibrasi Warna Luka Menggunakan Metode Segmentasi K-Means dan Mean Shift.
- Ozierańska, A., Skomra, A., Kuchta, D., and Rola, P. (2016). The critical factors of Scrum implementation in IT project– the case study. *Journal of Economics and Management*, 25(3):79–96.
- Purnama, S. (2013). Metode Penelitian Dan Pengembangan (Pengenalan Untuk Mengembangkan Produk Pembelajaran Bahasa Arab). *LITERASI (Jurnal Ilmu Pendidikan)*, 4(1):19.
- Rahmadati, S. (2023). Rancang Bangun Aplikasi dan Web Service Pengkajian Luka Kronis Khususnya Modul Pengolahan Citra Berbasis Android.
- Rizki, M. (2022). Deteksi Keliling Luka Kronis Menggunakan Active Contour ( Snake ) dan Active Contour yang Ditambahkan Interpolasi.
- Safitri, N. A. N., Purwanti, L. E., and Andayani, S. (2022). Hubungan Perilaku Perawatan Kaki Dengan Kualitas Hidup Pasien Diabetes Melitus Di RSU Muhammadiyah Dan Klinik Rulia Medika Ponorogo. *Health Sciences Journal*, 6(1):67–74.
- Suendri (2019). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer dan Informatika*, 3(1):1–9.
- Susanto, G. and Sukadi (2021). Sistem Informasi Rekam Medis Pada Rumah Sakit Umum Daerah (RSUD) Pacitan Berbasis Web Base. *Journal Speed Sentra Penelitian Engineering dan Edukasi*, 3(4):18–24.



Vogt, N. T., Koller, F. J., Santos, P. N. D., Lenhani, B. E., Guimarães, P. R. B., and Kalinke, L. P. (2020). Quality of life assessment in chronic wound patients using the Wound-QoL and FLQA-Wk instruments. *Investigacion y Educacion en Enfermeria*, 38(3):1–12.

Whitten, J. L., Bentley, L. D., and Dittman, K. C. (2004). *Metode desain dan analisis sistem*. Andi Offset, Yogyakarta.

Widodo, P. P. and Prabowo, H. (2011). *Menggunakan UML*. Informatika Bandung.