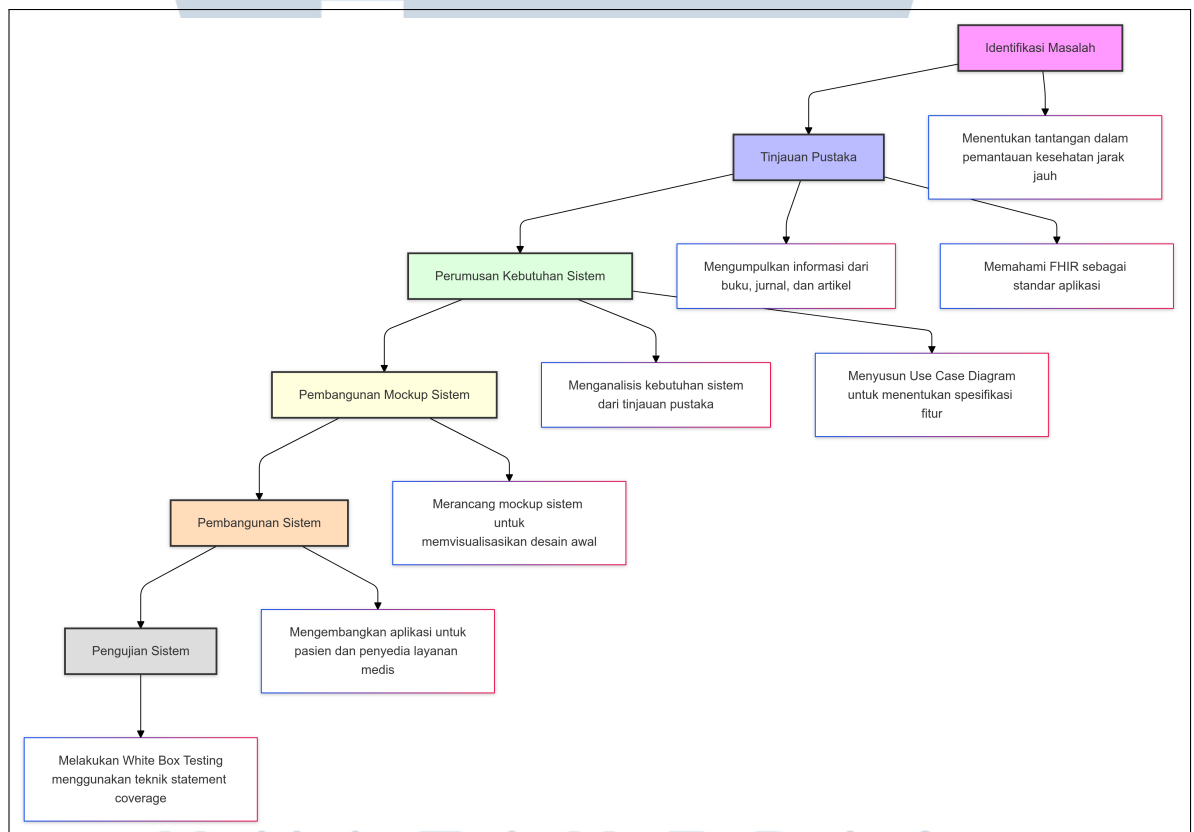


BAB 3

METODE PENELITIAN

Dalam penelitian ini, pengembangan aplikasi *Oncology App* dilakukan dengan pendekatan *Software Development Life Cycle* (SDLC). SDLC adalah kerangka kerja sistematis yang mencakup beberapa tahap untuk memastikan pengembangan perangkat lunak dilakukan secara terstruktur dan menghasilkan aplikasi yang sesuai dengan kebutuhan. Penelitian ini menggunakan model SDLC, yang terdiri dari enam tahapan utama: perencanaan, analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Berikut adalah penjelasan setiap tahapan yang dilakukan dalam penelitian ini:



Gambar 3.1. Alur Tahapan Penelitian

3.1 Perencanaan (Planning)

Tahapan perencanaan bertujuan untuk menentukan tujuan pengembangan aplikasi, ruang lingkup proyek, serta jadwal pelaksanaan kegiatan secara terperinci.

Jadwal pelaksanaan ditentukan berdasarkan skala prioritas dari setiap aktivitas yang telah diidentifikasi selama diskusi tim dan evaluasi kebutuhan proyek. Metode brainstorming digunakan untuk mengidentifikasi masalah utama yang menjadi dasar pengembangan aplikasi, seperti kurangnya alat untuk pemantauan kesehatan pasien kanker secara *real-time*. Selain itu, proses ini melibatkan penyusunan *timeline* yang realistis dengan mempertimbangkan sumber daya yang tersedia, kapasitas tim, serta batasan waktu proyek, guna memastikan keberhasilan implementasi. Dalam tahap ini, dilakukan:

- Identifikasi Masalah: Permasalahan yang dihadapi adalah kurangnya aplikasi yang dapat memantau gejala, kepatuhan pengobatan, dan perubahan kondisi pasien kanker secara *real-time*.
- Penentuan Tujuan: Aplikasi *Oncology App* bertujuan untuk menyediakan fitur pelaporan gejala, pemantauan tanda vital, dan manajemen kondisi pasien sesuai dengan standar FHIR.
- Pengumpulan Tim: Tim pengembang terdiri dari *frontend* dan *backend developer*, serta peneliti yang fokus pada penerapan standar FHIR.

Output dari tahap ini adalah dokumen rencana proyek yang mencakup deskripsi masalah, tujuan, kebutuhan sumber daya, dan jadwal.

3.2 Analisis Kebutuhan (Requirement Analysis)

Pada tahap ini, dilakukan analisis kebutuhan aplikasi dengan pendekatan yang melibatkan berbagai metode, seperti wawancara mendalam dengan dokter untuk memahami kebutuhan klinis pasien dan survei terhadap pasien untuk mengetahui preferensi serta kesulitan dalam pemantauan kesehatan. Berdasarkan hasil analisis, kebutuhan aplikasi didefinisikan sesuai dengan standar FHIR, meliputi fitur-fitur seperti pencatatan tanda vital melalui *Form Observation* dan dokumentasi kondisi medis melalui *Form Condition*. Proses ini juga mencakup identifikasi kebutuhan keamanan data pasien, termasuk *hashing* untuk melindungi informasi sensitif, serta evaluasi kebutuhan kompatibilitas perangkat agar aplikasi dapat diakses dengan baik di berbagai *platform*. Proses yang dilakukan meliputi:

- Kebutuhan Fungsional:
 - *Form Observation* untuk mencatat tanda vital pasien.

- *Form Condition* untuk mencatat kondisi medis pasien.
- Integrasi data dengan server HAPI FHIR.
- Kebutuhan Non-Fungsional:
 - Keamanan data pasien dengan *hashing*.
 - Kompatibilitas dengan berbagai perangkat (responsif).
 - Skalabilitas untuk mendukung banyak pengguna.

Output dari tahap ini adalah dokumen spesifikasi kebutuhan perangkat lunak (Software Requirement Specification/SRS).

3.3 Desain (Design)

Tahap desain bertujuan untuk merancang arsitektur sistem, antarmuka pengguna, dan struktur *database* yang digunakan dalam aplikasi. Proses desain diawali dengan identifikasi kebutuhan teknis dan alur kerja yang dirancang melalui diagram arsitektur sistem. Visualisasi ini menggambarkan bagaimana setiap komponen aplikasi, seperti *form Observation* dan *Condition*, berinteraksi dengan *server* HAPI FHIR dan modul *hashing* data. Antarmuka pengguna dirancang menggunakan *wireframe* untuk memastikan kemudahan navigasi dan pengalaman pengguna yang intuitif. Pada sisi *backend*, *server* HAPI FHIR digunakan sebagai penyedia layanan penyimpanan *resource* FHIR, yang dilengkapi dengan algoritma *hashing* untuk melindungi data sensitif. Struktur database dirancang dengan format JSON yang sesuai dengan standar FHIR untuk memastikan interoperabilitas data. Berikut adalah diagram arsitektur sistem untuk memberikan gambaran menyeluruh tentang hubungan antara komponen aplikasi: Proses desain meliputi:

- Desain Frontend:
 - Menggunakan React.js untuk membangun antarmuka form yang interaktif.
 - *Wireframe* dirancang untuk memastikan pengalaman pengguna yang optimal.
- Desain Backend:
 - Menggunakan *server* HAPI FHIR untuk menyimpan dan mengelola *resource* FHIR seperti *Observation* dan *Condition*.

- Algoritma *hashing* diterapkan pada data sensitif.
- Desain Database:
 - *Resource* FHIR seperti *Patient*, *Observation*, dan *Condition* disimpan dalam format JSON pada *server* FHIR dan Firebase.

Output dari tahap ini adalah dokumen desain sistem (System Design Document) dan prototipe awal antarmuka.

3.4 Implementasi (Implementation)

Pada tahap implementasi, pengembangan dilakukan berdasarkan desain yang telah dibuat, dengan menghadapi beberapa tantangan teknis yang berhasil diatasi melalui pendekatan sistematis. Salah satu tantangan utama adalah integrasi *form Observation* dan *Condition* dengan *server* HAPI FHIR untuk memastikan data dapat disimpan dalam format JSON yang sesuai dengan standar FHIR. Di sisi *frontend*, tantangan mencakup pengelolaan validasi data secara dinamis menggunakan React.js untuk mencegah kesalahan input. Untuk *backend*, penerapan algoritma *hashing* pada data sensitif memerlukan optimisasi agar tidak memengaruhi kinerja sistem secara signifikan. Solusi yang diterapkan mencakup pemanfaatan *library hashing modern* dan pengujian ekstensif untuk memastikan kompatibilitas penuh dengan standar FHIR. Implementasi ini didukung oleh kolaborasi tim lintas fungsi dan iterasi pengembangan yang bertujuan untuk mengoptimalkan fungsionalitas dan keamanan aplikasi secara keseluruhan. Proses implementasi meliputi:

- Frontend:
 - Mengembangkan *form Observation* dan *Condition* menggunakan React.js.
 - Menggunakan validasi dinamis untuk memastikan data yang dimasukkan sesuai.
- Backend:
 - Mengintegrasikan *form* dengan RESTful API server HAPI FHIR.
 - Menambahkan fitur *hashing* data sensitif sebelum dikirimkan ke server.

Kode yang dihasilkan diuji secara lokal untuk memastikan bahwa aplikasi berjalan sesuai dengan spesifikasi.

3.5 Pengujian (Testing)

Tahap pengujian bertujuan untuk memastikan aplikasi berjalan dengan baik dan bebas dari bug. Pengujian dilakukan melalui berbagai metode, termasuk unit testing untuk memverifikasi fungsi individual seperti validasi form dan pembuatan resource FHIR, serta integration testing yang menguji alur data antara frontend dan backend. Diagram alur pengujian membantu menggambarkan proses pengujian yang sistematis, mulai dari identifikasi skenario uji hingga dokumentasi hasilnya: Proses pengujian meliputi:

- Unit Testing:
 - Menguji fungsi-fungsi individu seperti validasi form dan pembuatan resource FHIR.
- Integration Testing:
 - Menguji integrasi antara frontend dan backend untuk memastikan data dikirim dan diterima dengan benar.

Output dari tahap ini adalah laporan pengujian yang mencakup hasil dan perbaikan yang dilakukan.

3.6 Pemeliharaan (Maintenance)

Tahap pemeliharaan dilakukan setelah aplikasi diimplementasikan untuk memastikan aplikasi tetap berfungsi dengan baik. Aktivitas yang dilakukan meliputi:

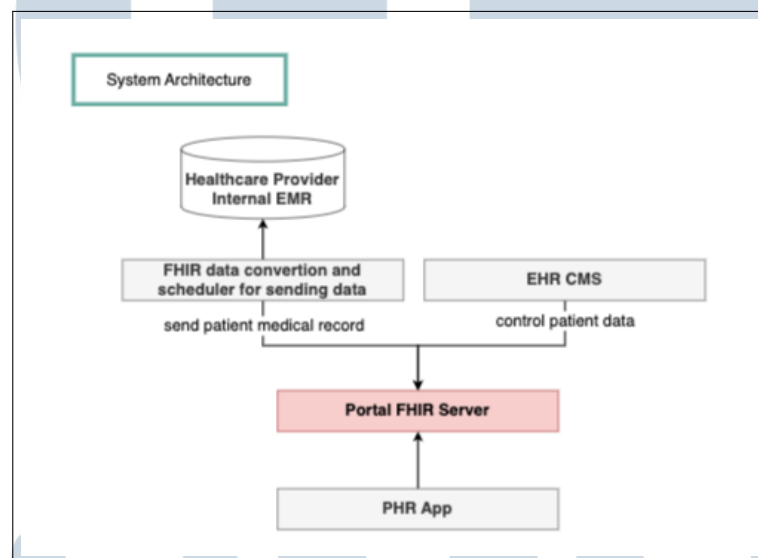
- Perbaikan Bug:
 - Mengatasi masalah yang ditemukan setelah aplikasi digunakan.
- Pembaruan Fitur:
 - Menambahkan fitur baru berdasarkan masukan dari pengguna.
- Monitoring Sistem:

- Memastikan server dan aplikasi berjalan dengan performa optimal.

Output dari tahap ini adalah dokumentasi pembaruan dan laporan pemeliharaan aplikasi.

3.7 Arsitektur Aplikasi

Arsitektur aplikasi dibutuhkan untuk membangun *environment* sistem agar bisa dikembangkan secara terus menerus. Gambar 3.2. Arsitektur Aplikasi Pasien Kanker adalah Arsitektur yang akan dibangun kedepannya pada penelitian ini.

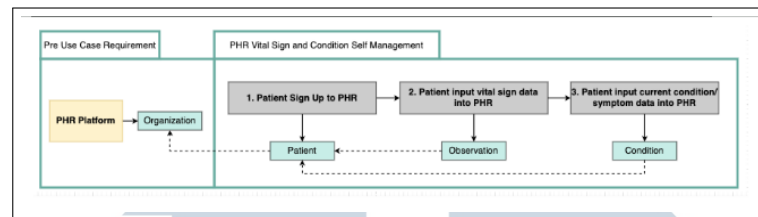


Gambar 3.2. Arsitektur Aplikasi

Pada Gambar 3.2. Arsitektur Aplikasi Pasien Kanker, PHR app atau aplikasi pasien kanker akan mendapatkan input seperti observasi, *vital sign* dan kondisi dari pasien kanker. Kemudian data yang ada akan dimonitoril pada portal sistem untuk memudahkan pihak rumah sakit atau fasilitas kesehatan memonitoring kondisi pasien kanker yang melakukan rawat jalan. Portal ini akan terintegrasi dengan CMS dan *Scheduler* untuk *converting data* sesuai format FHIR yang selanjutnya akan disimpan pada setiap database yang ada di masing masing fasilitas kesehatan.

3.8 Diagram User Scenario

Diagram User Scenario digunakan untuk memberikan gambaran terhadap perancangan fitur-fitur yang akan dibangun. Berikut adalah Gambar 3.3 *User Scenario* pada Aplikasi Pasien Kanker yang akan dibuat dalam penelitian ini:



Gambar 3.3. Diagram User Scenario

Sumber:

Pada Gambar 3.3 *User scenario* aplikasi Pasien Kanker ada tiga skenario utama yang akan dilakukan pada aplikasi. pertama, pasien kanker perlu melakukan registrasi pada aplikasi. Data registrasi yang akan diberikan seperti nama pasien, tanggal lahir, rumah sakit terdaftar pasien dan lain lainnya. Kemudian jika pasien sudah berhasil registrasi, setiap hari pasien akan melakukan observasi dan cek kondisi secara mandiri atau dengan bantuan dari tenaga medis seperti dokter dan perawat. Pada skenario kedua, *Patient* akan melakukan observasi dengan mengisi data *vital sign* seperti berat badan, suhu badan dan tinggi badan. yang terakhir dari skenario yang dilakukan adalah mengisi berupa kuesioner hasil observasi apakah ada pening, muntah, rambu rontok dan observasi lanjut jika baru melakukan terapi.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA