BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian yang dilakukan oleh Inda Rusdia Sofiani1, Rafli Kharisma2, Lailis Syafa'ah3. Judul penelitian ini Sistem *Monitoring Heart Rate* dan Oksigen Dalam Darah Berbasis Lora pada tahun 2021. Hasil dari penelitian ini menyatakan bahwa tujuan penelitian ini adalah menggunakan sensor untuk membangun dan mengembangkan perangkat yang mudah digunakan untuk mengukur atau memantau kadar oksigen darah dan detak jantung. Laptop dapat menampilkan hasil teknik menggunakan sensor NodeMCU MAX30100. Sebagai hasil dari penelitian ini, WiFi ESP8266 dapat diaktifkan di NodeMCU, memungkinkan perangkat terhubung ke jaringan dan memantau kadar oksigen dan detak jantung. Arduino IDE dan program aplikasi perangkat lunak kontrol berbasis Lora merupakan program yang digunakan pada NodeMCU[8].

Penelitian yang dilakukan oleh Della Rahmawarni dan Harmadi berjudul "Sistem Monitoring Saturasi Oksigen dan Denyut Nadi dalam Darah Menggunakan Sensor MAX30100 via Telegram Berbasis IoT" pada tahun 2021. Hasil dari penelitian ini menunjukkan bahwa sensor MAX30100 berbasis IoT yang terhubung ke Telegram digunakan dalam sistem ini untuk melacak saturasi oksigen dan denyut nadi. Untuk mengukur intensitas cahaya

dan menerjemahkannya menjadi sinyal listrik, sensor ini memanfaatkan fotodetektor dan LED. Telegram dan LCD menerima data dari sensor yang telah diproses oleh Wemos D1. Selain itu, sistem ini dilengkapi dengan bel sebagai sinyal peringatan jika denyut nadi atau saturasi oksigen berada di luar kisaran 60 hingga 100 denyut per menit.[9]

Penelitian yang dilakukan oleh Sunardi & Ghaffar Mudzakkir Daud berjudul "Monitoring Detak Jantung dan Menampilkan Suhu Tubuh Mengg unakan MLX90614 Berbasis Android" pada tahun 2023. Penelitian ini membahas sistem pemantauan kesehatan yang dirancang dengan menggunakan sensor MAX30102 sebagai alat pengukur detak jantung dan sensor MLX90614 sebagai alat pengukur suhu tubuh. Tujuan dari penelitian ini adalah membantu mengatasi masalah pendataan di posyandu untuk mempersingkat waktu pengambilan data pasien, karena masih banyak yang menggunakan alat manual untuk mengukur detak jantung dan suhu tubuh. Hasil penelitian ini menunjukkan keberhasilan perancangan sistem pemantauan suhu tubu dan detak jantung menggunakan sensor MLX90614 dan MAX30102, yang dikendalikan olehmodul WiFi ESP32 untuk terhubung ke internet dan aplikasi berbasis Android.[10]

Penelitian yang dilakukan oleh Zain Bahaul Anwar1, Arif Widodo2, Nur Kholis3, Nurhayati4 dengan judul Sistem Monitoring Pasien Isolasi Mandiri Covid-19 Berbasis *Internet Of Things* pada tahun 2021.Hasil dari penelitian ini Untuk mencegah beban rumah sakit semakin besar, Tujuan dari penelitian ini adalah untuk menciptakan sistem yang dapat

mengawasi individu yang melakukan isolasi mandiri. Mikrokontroler ESP32 dapat digunakan sebagai perangkat *Internet of Things* untuk memantau dan mengirimkan SpO2,detak jantung, dan suhu tubuh ke server *ThingSpeak*. Ketidakakuratan SpO2 sistem yang rendah, yang berada di bawaikriteria akurasi SpO2 ≤4%, berarti bahwa pasien yang melakukan isolasi mandiri dapat memanfaatkannya dengan baik, menurut temuan penelitian [11].

Penelitian yang dilakukan oleh M. Taufiq Tamam1, Itmi Hidayat Kurniawan2, Anis Kusumawat3,. dengan judul Sistem pemantau Detak Jantung dan Saturasi Oksigen dalam darah (SpO2) Berbasis IoT pada tahun 2023. Hasil dari Penelitian ini membahas tentang membuat aplikasi monitoring detak jantung dan saturasi oksigen dalam darah (SpO2) berbasis IoT yang dapat diakses secara daring dalam waktu nyata kapanpun dan di manapun. Sensor dipasang di ujung jari pasien. Proses pengamatan berupa perubahan sinyal analog yang diolah dengan aritmatik fast fourier transform sebagai pemrosesan sinyal digital. Proses akhir pengolahan menampilkan besaran nilai detak jantung dan saturasi oksigen dalam darah (*SpO2*) yang terukur[12].

2.2 Landasan Teori

2.2.1 Laravel

Menurut ID CloudHost, Laravel adalah framework PHP yang dirilis dibawah lisensi MIT dan dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang bertujuan untuk menurunkan biaya pengembangan awal dan pemeliharaan, serta meningkatkan pengalaman bekerja dengan aplikasi dengan sintaks yang ekspresif, jelas, dan menghemat waktu [11]. MVC adalah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. Pendekatan MVC membagi aplikasi berdasarkan komponennya, seperti pengontrol, manipulasi data, dan antarmuka pengguna..

1. Model mewakili struktur data; biasanya, model berisi fungsi fungsi yang membantu orang mengelola basis data, seperti memasukkan data ke dalam basis data, pembaruan data, dan sebagainya.
2. Tampilan adalah bagian yang mengatur tampilan untuk pengguna. Bisa dianggap sebagai halaman web..
3. Controller merupakan bagian yang menjembatani model dan view.

Gambar 2.1 Laravel

2.2.2 Database Management System (DBMS)

Sistem manajemen basis data seperti MySQL atau PostgreSQL digunakan untuk menyimpan, mengelola, dan mengamankan data pengguna aplikasi. DBMS memastikan keakuratan, konsistensi, dan keamanan data selama proses operasional aplikasi.

Gambar 2.2 DBMS

2.2.3 CSS

Cascading Style Sheets (CSS) adalah bahasa desain yang digunakan untuk Mengatur tampilan dan tata letak halaman web. CSS bekerja Bersama HTML untuk memberikan gaya pada elemen-elemen web, seperti warna, ukuran, tata letak, animasi, dan responsivitas, sehingga membuat halaman web lebih menarik dan mudah digunakan.

Gambar 2.3 CSS

2.2.4 HTML

HyperText Markup Language (HTML) adalah bahasa markup standar yang digunakan untuk membuat dan mengatur struktur konten halaman web. Ini memberi tahu browser bagaimana menampilkan teks, gambar, video, dan elemen lainnya. Fokus HTML lebih pada struktur dan konten daripada tampilan visual. CSS membantu mengatur desain dan tata letak visual halaman web..

Gambar 2.4 HTML

2.2.5 Unified Modelling Languange

UML dirancang untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan sistem perangkat lunak. Itu menggunakan diagram dan teks-teks pendukung untuk memodelkan dan berkomunikasi mengenai sistem.

1. Use Case Diagram

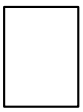
   Komponen pembentuk use case diagram adalah:

   1) Aktor (actor), menggambarkan pihak-pihak yang berperan dalam sistem.
   2) Use Case, aktifitas yang disiapkan oleh sistem.
   3) Hubungan (link), aktor mana saja yang terlibat dalam use case ini, UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori object-oriented dan sebagian lagi memfokuskan pada detail rancangan

dan konstruksi. Semuanya dimaksudkan sebagai sarana komunikasi antar team programmer maupun dengan pengguna.

Tabel 2.1 Use Case Diagram

No	Simbol	Nama	Keterangan
1.		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2.		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
3.		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4.		*Include*	Menspesifikasikan bahwa use case sumber secara eksplisit.
5.		*Extend*	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

7.		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlai dan elemen-elemennya (sinergi).
10.		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Activity Diagram

Diagram ini memodelkan model bisnis dan software. Diagram aktifitas menunjukkan aksi-aksi sistem. Diagram aktifitas menunjukkan pemanggilan fungsi tertentu, seperti panggilan, dalam pemodelan software. Di sisi lain, dalam pemodelan bisnis, diagram ini menunjukkan aktifitas yang disebabkan oleh kejadian di luar, seperti pemesanan, atau kejadian internal.

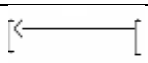
Tabel 2.2 Activity Diagram

No.	Simbol	Nama	Keterangan
1.		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4.		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5.		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa Aliran

3. Equence diagram

Serangkaian diagram dirancang untuk membangun komunikasi antara objek daripada memanipulasi data saat berkomunikasi.

Tabel 2.3 Equence diagram

No.	Simbol	Nama	Keterangan
1.		*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
2.		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang Terjadi
3.		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi

4. Class Diagram

Diagram kelas adalah dasar dari proses pemodelan objek. Forward engineering mengubah model menjadi kode program, sedangkan reverse engineering merubah kode program menjadi model.

Diagram kelas memiliki fitur, seperti atribut dan operasi. Perilaku suatu kelas digambarkan oleh attributnya (atribut) dan operasinya (operasi). Perluasan kelas, seperti stereotypes, nilai yang dilabelkan, dan batasan, juga merupakan ciri-ciri kelas.

Tabel 2.4 Class Diagram

No.	Simbol	Nama	Keterangan
1.		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2.		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		*Dependecy*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang tidak mandiri.
7.		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.