

BAB III

METODOLOGI PENELITIAN

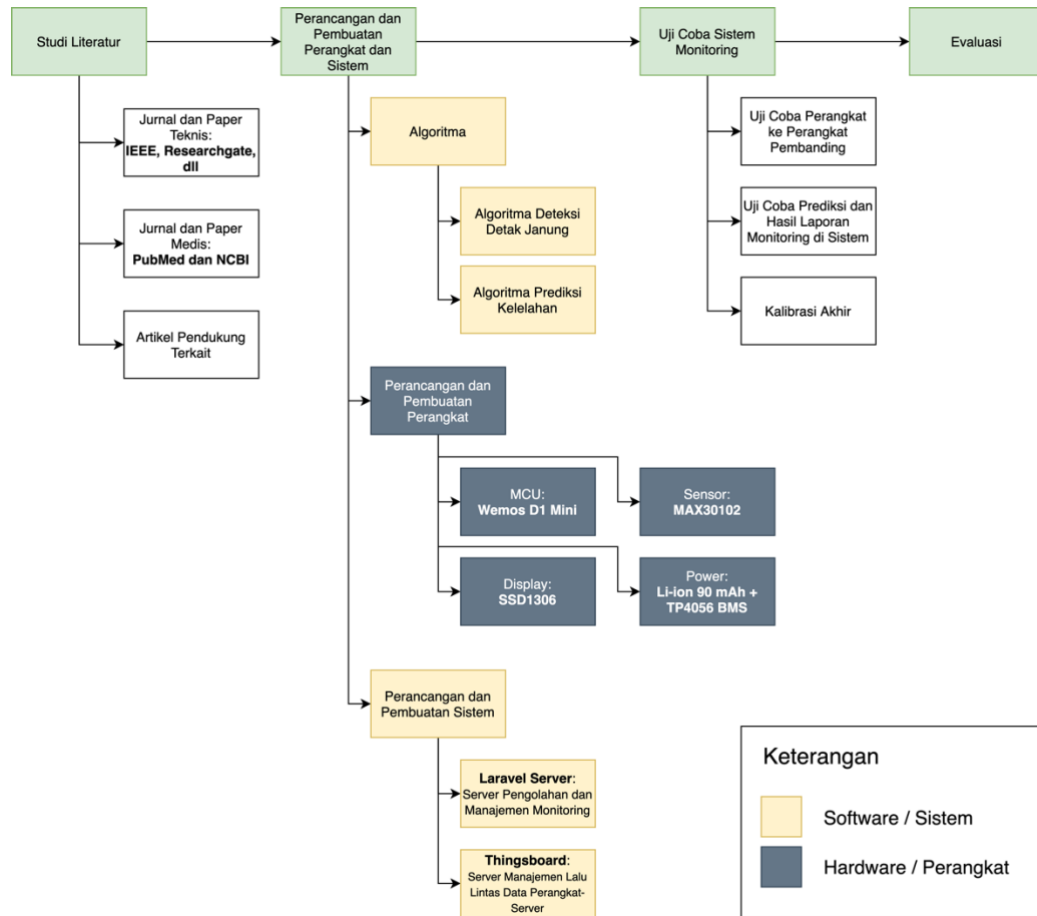
3.1 Metode Penelitian

Pada penelitian ini, digunakan metode penelitian *Research and Development*. Metode penelitian ini akan dilakukan dengan cara mempelajari penelitian sejenis dan terkait yang telah ada sebelumnya, dan membuat dan mengembangkan produk berdasarkan referensi desain dan fungsi dari produk yang telah ada sebelumnya, dan dikembangkan berdasarkan fokus dan target penelitian, yaitu mengembangkan sistem *monitoring* yang didesain khusus untuk memonitor tingkat kelelahan.

Secara umum, penelitian ini akan dilakukan dalam beberapa tahapan. Tahapan-tahapan tersebut adalah studi literatur, merancang dan membuat sistem *monitoring*, uji coba sistem *monitoring*, dan evaluasi.

3.2 Tahapan Penelitian

Tahapan pada penelitian ini akan dilakukan dalam 4 tahap, yaitu studi literatur, perancangan dan pembuatan perangkat dan sistem, serta evaluasi perangkat dan sistem. Ringkasan tahapan penelitian ini dapat dilihat pada diagram tahapan penelitian pada gambar 3.1.



Gambar 3. 1. Diagram Tahapan Penelitian

3.2.1 Studi Literatur

Tahap pertama dari penelitian ini adalah melakukan studi literatur yang meliputi studi mengenai teori pendukung, penelitian sejenis, perangkat terkait, serta pencarian data terkait situasi dan kendala di lapangan untuk menambah masukan terhadap pengembangan sistem serta kebutuhan spesifikasi teknis. Studi literatur dari sisi materi teknis akan direferensikan dari paper dan jurnal yang terbit di IEEE, Researchgate, dan beberapa sumber jurnal lainnya. Lalu studi literatur dari sisi materi medis akan direferensikan dari paper dan jurnal yang ada di PubMed NCBI. Serta, akan digunakan juga artikel-artikel pendukung terkait yang ada di beberapa website terkait yang memiliki reputasi di bidangnya.

3.2.2 Perancangan dan Pembuatan Sistem *Monitoring*

Dalam penelitian ini, pada proses pendeteksian *fatigue* akan dilakukan dengan cara memonitor parameter detak jantung menggunakan sensor berbasis

modul MAX30102. Terdapat metode lain yang dapat digunakan dalam pendeteksian kelelahan ini, contohnya menggunakan teknologi visi komputer yang banyak diterapkan di mobil. Namun untuk penerapannya pada area yang luas akan sangat sulit karena *monitoring* hanya dapat dilakukan di tempat tertentu yang terliput oleh jangkauan kamera.

Perangkat yang akan dirancang dan dibuat dalam program ini utamanya akan terdiri atas 3 modul utama, yaitu sensor MAX30102, mikrokontroler Wemos D1 Mini, dan modul OLED SSD1306. Mikrokontroler Wemos D1 Mini akan membaca tingkat detak jantung dan nilai *Heart Rate Variability* / HRV. Kemudian hasil pembacaannya akan dikirimkan ke *server* untuk diolah, serta akan ditampilkan melalui modul layar OLED bagi pengguna. Ketiga modul tersebut akan dirancang dan dirakit dalam sebuah PCB secara bertumpuk untuk memperoleh desain yang ringkas.

Pada perancangan *software* yang dilakukan, dirancang *software* untuk perangkat *monitoring* dan untuk *server monitoring*. Pada *server monitoring*, *software* untuk *real-time monitoring* dan pengolahan hasil *monitoring* akan dibangun dengan berbasiskan *platform* IoT Thingsboard dan PHP *framework* Laravel. *Software* pada komponen *server* utamanya akan berlaku sebagai pencatat dan pengolah hasil *monitoring*, manajemen perangkat dan pengguna, serta untuk memberi prediksi dari hasil *monitoring* yang didapat.

Lalu pada *software* di perangkat *monitoring*, penulisan dan perancangan *software* akan dilakukan menggunakan Arduino IDE. Di mana program Arduino IDE ini telah kompatibel dengan mikrokontroler yang digunakan dalam program penelitian ini.

3.2.3 Uji Coba Sistem *Monitoring*

Pada tahap ini, akan dilakukan uji coba terhadap keseluruhan sistem. Pada tahap uji coba, akan dilakukan uji coba perangkat *monitoring* terhadap perangkat pembanding, serta *server monitoring* juga akan diuji untuk memonitor dan memprediksi tingkat kelelahan.

Pada perangkat *monitoring*, kinerja sensor akan diuji coba secara uji statis dan non-statis dan dibandingkan ke perangkat pembanding yaitu perangkat Mi

Band 3. Serta pada server *monitoring*, pengujian akan meliputi pengujian performa dari algoritma prediksi yang dipakai serta hasil dari laporan dan prediksi yang dihasilkan.

3.2.4 Evaluasi

Pada tahap ini, akan dilakukan evaluasi terhadap kinerja sistem secara keseluruhan. Proses evaluasi ini akan dilakukan dengan cara membandingkan kinerja perangkat yang dikembangkan terhadap perangkat referensi yang dijadikan acuan, serta kinerja dari *server monitoring* yang dikembangkan. Lalu akan dievaluasi berdasarkan poin-poin kelebihan dan kekurangan dari keseluruhan sistem untuk dijadikan sebagai kesimpulan penelitian ini serta saran pada penelitian sejenis kedepannya.

3.3 Sistem *Monitoring* dan Deteksi Kelelahan

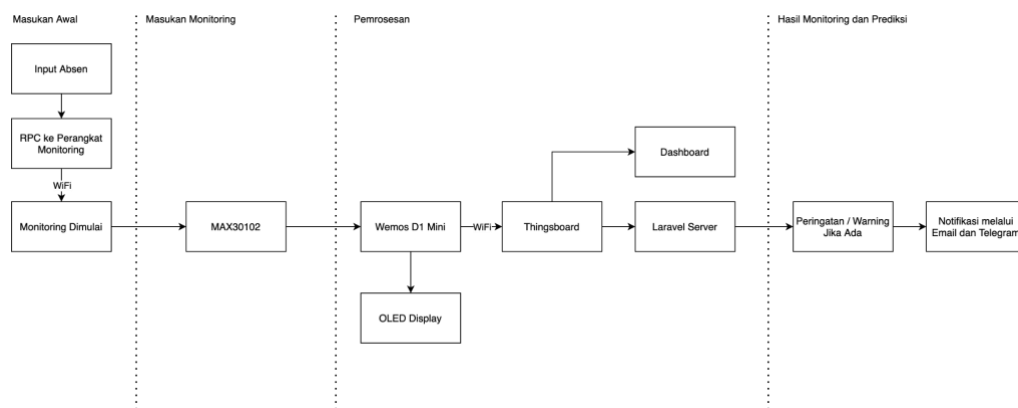
3.3.1 Perangkat dan Sistem Penunjang

Perangkat utama yang digunakan pada penelitian ini merupakan perangkat *monitoring* yang dirancang dari mikrokontroler Wemos D1 Mini, modul MAX30102. Sistem pendukung utama yang digunakan dalam penelitian ini merupakan *server monitoring* yang dirancang menggunakan PHP *framework* Laravel yang dikombinasikan dengan *platform* IoT Thingsboard untuk membantu manajemen dan lalu lintas data dari perangkat *monitoring* ke sistem *monitoring* utama.

Selain itu, perangkat dan sistem pendukung yang digunakan pada penelitian ini di antaranya adalah *Operating System* Ubuntu sebagai basis dari *server*, dan Arduino IDE sebagai *compiler* untuk perangkat *monitoring* yang dirancang.

3.3.2 Prinsip kerja

Prinsip kerja perangkat dan sistem yang dirancang dapat digambarkan seperti pada diagram di gambar 3.2.



Gambar 3. 2. Diagram Prinsip Kerja Sistem *Monitoring*

Pada penelitian ini, prinsip kerja dari sistem keseluruhan terbagi atas 2 tahap masukan, yaitu masukan awal ke *server monitoring*, dan masukan hasil *monitoring* dari perangkat. Serta tahap pemrosesan dan hasil.

Pada tahap masukan, masukan awal merupakan proses masukan ke *server monitoring* yang akan menerima inputan berupa perintah untuk mengaktifkan perangkat *monitoring* yang akan dipakai melalui absensi. Setelah perangkat *monitoring* aktif, maka perangkat *monitoring* akan mengirimkan masukan hasil *monitoring* ke sistem yang berisi parameter detak jantung, *R-R Interval* / jarak interval antar detak jantung yang pembacaannya dilakukan melalui modul MAX30102. Hasil pembacaan pada perangkat *monitoring* akan dikirimkan ke *server monitoring* dengan interval pengiriman setiap 2 detik.

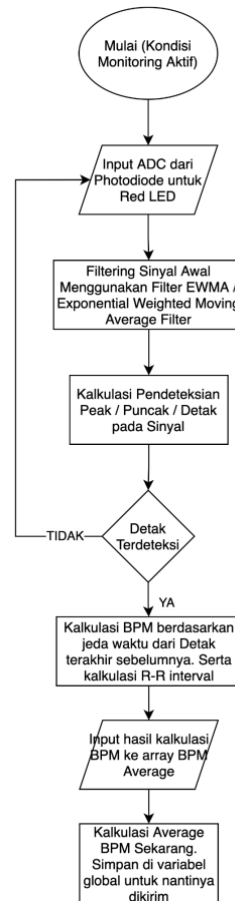
Pada tahap pemrosesan, tahap pertama akan diawali dengan mikrokontroler Wemos D1 Mini memperoleh inputan dari modul MAX30102, lalu mengirimkan hasil *monitoring*nya ke *platform* IoT Thingsboard untuk dapat ditampilkan ke *real-time dashboard*, dan meneruskan data hasil *monitoring* nya ke *server* Laravel. Pada *server* Laravel, data hasil *monitoring* akan dicatat dan diolah untuk keperluan *monitoring* dan prediksi tingkat kelelahannya.

Lalu pada hasil yang didapat, akan diperoleh hasil *monitoring* berupa laporan dan prediksi, serta juga peringatan / *warning* untuk hasil prediksi tingkat kelelahan yang tinggi, ataupun hasil nilai HRV yang rendah pada *monitoring*. Selanjutnya, hasil *monitoring* dan prediksi ini akan dikirimkan ke *user* melalui Telegram dan Email.

3.3.3 Algoritma

Pada perancangan sistem *monitoring* pada penelitian ini, terdapat 2 algoritma utama untuk mendukung jalannya. Algoritma tersebut adalah algoritma pembacaan detak jantung dan algoritma prediksi tingkat lelah.

3.3.3.1 Algoritma Pembacaan Detak Jantung



Gambar 3. 3. Diagram Algoritma Pembacaan Detak Jantung

Diagram pada gambar 3.3 merupakan gambaran dari algoritma yang digunakan untuk pembacaan detak jantung melalui modul MAX30102. Sensor akan secara terus menerus membaca inputan berupa nilai sinyal hasil konversi ADC dari *Photodiode*, yang kemudian sinyal tersebut akan di proses menggunakan filter *Exponential Weighted Moving Average Filter* untuk menghilangkan derau dan menstabilkan nilai pembacaan. Setelah itu, sinyal akan diproses untuk dilakukan pendeteksian detak melalui fungsi dari *library* yang disertakan dari sensor.

Ketika detak terdeteksi, maka akan dilakukan perhitungan detak jantung dalam satuan BPM / *Beat per Minute* dengan formula berikut.

Hersyanda Putra Adi, 2021

RANCANG BANGUN SISTEM MONITORING TINGKAT KELELAHAN BERDASARKAN DETAK JANTUNG UNTUK PENCEGAHAN BURNOUT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

$$BPM(now) = \frac{60}{(millis() - lastBeat)/1000}$$

millis() : waktu detak sekarang berdasarkan internal timer mikrokontroler ; *lastBeat* : waktu detak terakhir.

Selanjutnya, data BPM tersebut akan diinputkan ke *array* $BPM_{average}$ untuk kemudian dicari rata-rata BPM dari data yang ada di *array* $BPM_{average}$ agar menghasilkan pembacaan yang lebih konsisten. Selain itu, selang waktu antar detak yang digunakan pada formula perhitungan $BPM(now)$ akan menjadi nilai *R-R interval* untuk perhitungan HRV.

3.3.3.2 Algoritma Prediksi tingkat Lelah

Pada sistem *monitoring* ini, prediksi tingkat kelelahan (*fatigue*) akan dilakukan menggunakan 2 parameter utama, yaitu menggunakan tingkat lelah (*tiredness*) dan penilaian nilai HRV. Serta, proses prediksi ini akan dilakukan setelah proses *monitoring* / proses sesi kerjanya selesai.

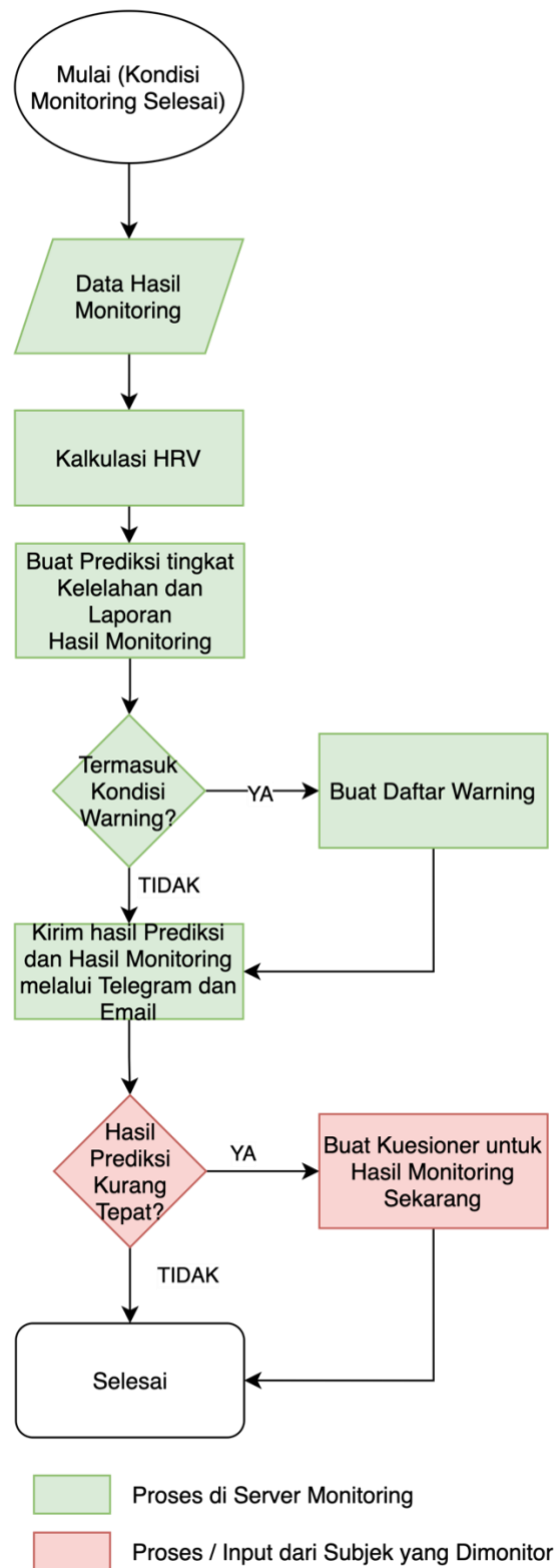
Prediksi nilai lelah akan dilakukan menggunakan metode klasifikasi *k-Nearest Neighbor* / *k-NN* dengan *dataset* yang digunakan berasal dari data hasil *monitoring* dan data kuesioner dari *monitoring* terkait sebelumnya. Pada sistem ini, setiap *user* akan memiliki *dataset* nya sendiri yang berasal dari hasil *monitoring* dan hasil kuesioner masing-masing oleh karena tingkat kelelahan setiap orang yang sifatnya subjektif.

Pada algoritma prediksi ini, algoritma *k-Nearest Neighbor* dijalankan dengan menggunakan perhitungan *distance* secara teknik *Manhattan Distance*. Hal ini dikarenakan berdasarkan referensi, penggunaan *Manhattan Distance* lebih diutamakan daripada *Euclidean Distance* pada perhitungan pada dimensi yang tinggi (Aggarwal dkk., 2001), oleh karena adanya *Curse of Dimensionality* (Aggarwal dkk., 2001; Domingos, 2012). Di mana pada pengujian ini, *dataset* yang digunakan memiliki 8 parameter / dimensi untuk sampel, dengan 1 label untuk kategori tingkat lelah dari hasil kuesioner. Selain itu, penggunaan angka “k” atau penggunaan referensi banyak data terdekat akan juga divariasikan berdasarkan jumlah *dataset* yang ada dengan harapan agar dapat mendapatkan prediksi yang

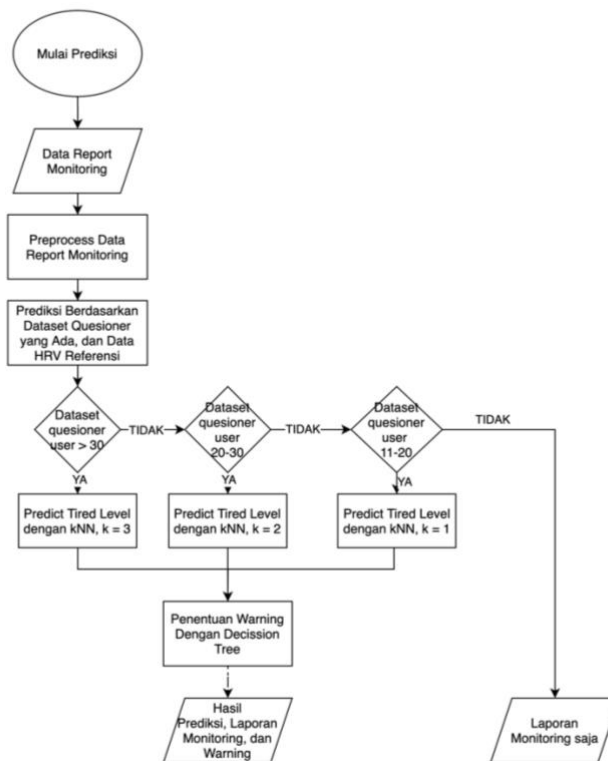
akurasinya lebih tinggi dan representatif ketika jumlah *dataset* nya masih belum begitu banyak.

Serta, penilaian nilai HRV akan dilakukan berdasarkan nilai referensi HRV dari EliteHRV. EliteHRV sendiri merupakan sebuah perusahaan dan sebuah *software* yang fokus untuk *monitoring* kesehatan, tingkat stres, tingkat *recovery*, dan keseimbangan sistem saraf dengan fokus utama metode pendeteksiannya menggunakan penilaian nilai *Heart Rate Variability* / HRV. Angka referensi pada penelitian ini digunakan dari EliteHRV dikarenakan nilai referensi dari EliteHRV merupakan nilai referensi yang mencakup cakupan untuk usia yang luas, serta datanya diperoleh dari survey dari EliteHRV dengan jumlah data lebih dari 40000 (Moore, 2021).

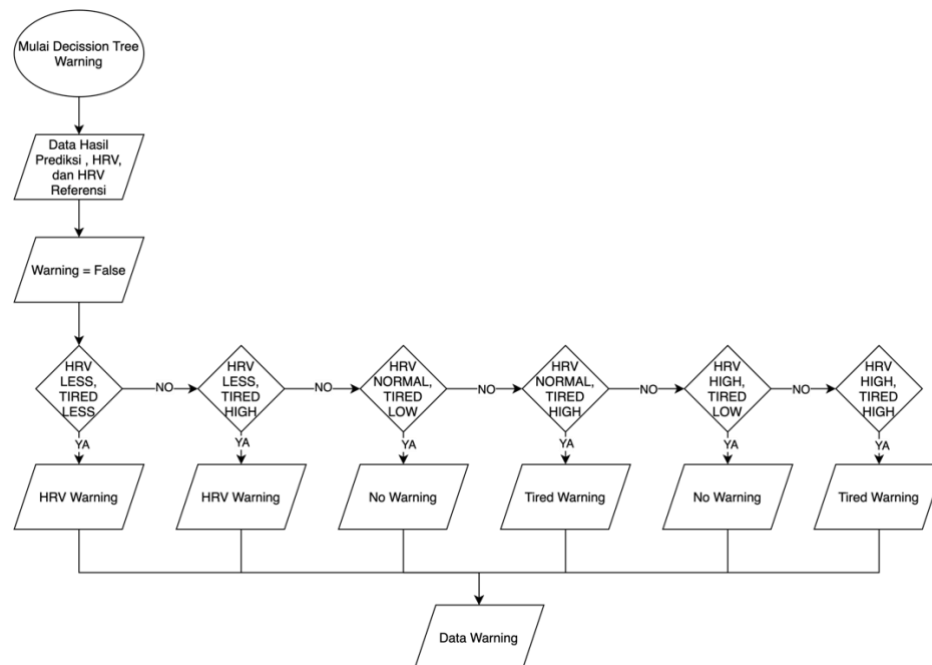
Untuk gambaran proses untuk pelaksanaan prediksinya, akan digambarkan melalui diagram pada gambar 3.4, gambar 3.5, dan gambar 3.6.



Gambar 3. 4. Diagram Proses Prediksi



Gambar 3. 5. Diagram Proses Prediksi Penentuan tingkat Lelah



Gambar 3. 6. Diagram Penentuan Hasil Prediksi Akhir

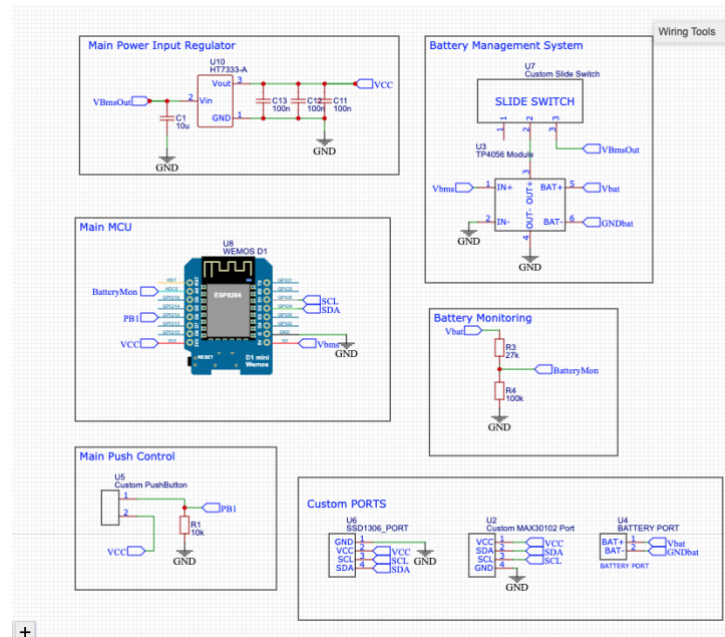
Proses prediksi tingkat lelah (*tiredness*) menggunakan *k-Nearest Neighbor* akan dilakukan dengan nilai parameter *k* yang berbeda-beda, tergantung dari total *dataset* yang telah tersedia dari *user* tersebut. Hal ini dilakukan agar akurasi prediksi dapat ditingkatkan seiring dengan bertambahnya *dataset* dari *user*. Selanjutnya, angka skor dari prediksi lelah (*tiredness*) sebelumnya akan diproses kembali dengan data perhitungan HRV melalui proses penentuan seperti pada diagram di gambar 3.6.

Setelah prediksi dilakukan, hasil *monitoring* dan prediksi akan dikirimkan kepada *user*, beserta dengan kuesioner jika *dataset* dari *user* masih dalam jumlah yang sedikit. Selain itu, kuesioner juga akan dapat diisi oleh *user* ketika hasil prediksi yang diberikan oleh sistem dinilai salah / tidak representatif. Sehingga diharapkan pada prediksi selanjutnya, hasil prediksinya dapat lebih akurat.

3.3.4 Perancangan Perangkat *Monitoring*

Pada perangkat *monitoring* yang dirancang pada penelitian ini, perangkat dirancang dengan pendekatan *modular*. Sensor-sensor dan komponen pendukung disatukan dalam suatu PCB sehingga menghasilkan sebuah *shield* (papan sirkuit tambahan) untuk perangkat mikrokontroler utama yang digunakan yaitu Wemos D1 Mini.

Proses perancangan PCB ini dilakukan menggunakan *platform* EasyEDA dengan hasil skematik lengkapnya seperti pada gambar 3.7. Serta detail komponen yang dipakai pada perangkat *monitoring* ini dijabarkan pada tabel 3.1.

Gambar 3. 7. Skematik Rangkaian Perangkat *Monitoring*

| No | Komponen | Keterangan |
|----|------------------------------|---|
| 1 | Wemos D1 Mini | Mikrokontroler |
| 2 | TP4056 | <i>Battery Management System</i> |
| 3 | HT7333-A | LDO <i>Regulator</i> untuk <i>Main Input Supply</i> |
| 4 | SSD1306 | <i>Display OLED</i> |
| 5 | MAX30102 | Modul Detak Jantung |
| 6 | 401030 Li Ion Battery 90 mAh | |

Tabel 3. 1. Daftar Komponen Perangkat *Monitoring*

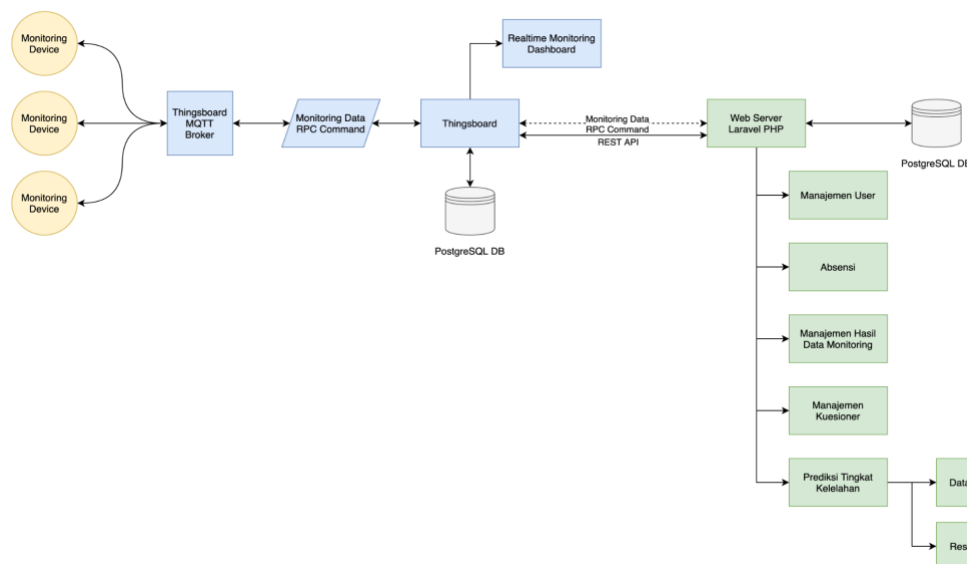
Pada perangkat *monitoring* ini, sumber daya akan menggunakan sebuah baterai Li-Ion dengan kapasitas 90 mAh, dengan proses pengisian daya nya akan di kontrol menggunakan BMS TP4056. Pada komunikasi ke sensor dan ke *display* OLED, seluruhnya dilakukan menggunakan protokol komunikasi I2C sehingga

penggunaan pin GPIO pada MCU Wemos D1 Mini yang dipakai dapat lebih hemat oleh karena minimnya jumlah pin GPIO yang tersedia.

3.3.5 Perancangan Server *Monitoring*

Pada server *monitoring* yang dirancang pada penelitian ini, server dirancang dan dibuat oleh 2 komponen pendukung utama, yaitu PHP *Framework* Laravel dan IoT *platform* Thingsboard. Pada hal ini, untuk penyimpanan data *monitoring* ke *database*, pengolahan data, dan prediksi akan dilakukan di *web server* utama yang dibangun menggunakan PHP *Framework* Laravel. Serta Thingsboard akan berlaku sebagai *microservice* pendukung *web server* utama yang fungsinya adalah untuk membantu dalam hal manajemen perangkat *monitoring*, dan proses pengambilan data dari perangkat *monitoring* untuk kemudian diteruskan ke *web server* utama. Komunikasi antara *web server* utama dengan Thingsboard dilakukan dengan REST API menggunakan protokol HTTP.

Proses perancangan sistem yang dilakukan akan menghasilkan sistem *monitoring* dengan arsitektur seperti yang digambarkan pada diagram di gambar 3.8.



Gambar 3. 8. Diagram Arsitektur Sistem *Monitoring*