

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penulis mengumpulkan beberapa penelitian yang terdahulu terkait sistem monitoring yang dapat membantu penulis dalam penelitian:

2.1.1 Smart Helmet As Driving Safety [2]

Penelitian ini bertujuan untuk mengembangkan helm pintar berbasis *Arduino* sebagai perangkat keselamatan dan keamanan berkendara bagi pengendara sepeda motor. Helm ini dilengkapi dengan beberapa sensor untuk mendeteksi kondisi pengendara dan keamanan helm itu sendiri. Salah satu fitur utama adalah deteksi kantuk menggunakan sensor detak jantung *XD58C* dan sensor kemiringan kepala *MPU6050*. Jika detak jantung pengendara berada dalam rentang tertentu dan kepala terdeteksi miring lebih dari 30 derajat, helm akan memperingatkan pengendara untuk beristirahat melalui speaker yang tertanam.

Selain itu, helm ini juga dilengkapi dengan fitur keamanan untuk mencegah pencurian. Dengan memanfaatkan klip tali helm yang terhubung ke alarm, sistem akan mendeteksi saat helm dilepas tanpa izin dan membunyikan alarm untuk memperingatkan pemilik. Helm ini menggunakan *NRF24L01* untuk mengirim sinyal antara helm dan motor. Ketika helm terpasang dengan benar, relay akan terhubung dan memungkinkan motor untuk dihidupkan. Namun, jika helm tidak terdeteksi, sistem mencegah motor dari dihidupkan, menambah lapisan keamanan bagi pengendara.

Dari hasil pengujian, sistem helm pintar ini berhasil mendeteksi kondisi kantuk, memastikan penggunaan helm yang benar, dan memberikan peringatan saat helm dicuri. Pada uji koneksi helm dan

motor, sinyal tetap stabil hingga jarak 70 meter tergantung kondisi lingkungan. Sistem keamanan menunjukkan tingkat keberhasilan 100% dalam mendeteksi pelepasan klip helm dan membunyikan alarm. Selain itu, uji kantuk menunjukkan bahwa alarm peringatan berbunyi saat *BPM* pengendara berada di bawah 60 dan kemiringan kepala lebih dari 30°. Dengan menggunakan *Arduino Nano* dan *Arduino Pro Mini* sebagai prosesor utama, helm ini efektif meningkatkan keselamatan pengendara dengan peringatan dini terkait kantuk dan pencegahan pencurian.

Beberapa poin penting yang dapat diambil oleh penulis adalah sebagai berikut:

1. Sensor kemiringan *MPU6050* dan pemanfaatan klip tali helm terbukti dapat menjadi *trigger* yang efektif pada sebuah sistem monitoring. Namun, sensor kemiringan *MPU6050* kurang cocok diimplementasikan pada sistem monitoring perangkat *IoT* karena perangkat *IoT* dipasang pada sebuah tiang besi yang tidak memungkinkan untuk mengalami kemiringan. Penulis hanya mengimplementasikan cara kerja sistem *Smart Helmet as Driving Safety* dalam memanfaatkan klip tali helm dengan cara mengecek apakah klip helm terlepas sebagai *trigger*, lalu memodifikasi cara kerja tersebut dengan cara mengecek apakah perangkat *IoT* terlepas atau tercabut dari tiang besinya.
2. Sistem *alerting* pada *Smart Helmet as Driving Safety*, yaitu alarm, kurang cocok diimplementasikan pada sistem monitoring perangkat *IoT* karena tujuan sistem monitoring perangkat *IoT* adalah memberitahu ke admin bahwa perangkat *IoT* telah terlepas dari tiang besi.
3. Pengujian sistem monitoring dilakukan dengan cara mengukur tingkat keberhasilan sistem tersebut dengan melakukan beberapa kali tes.

2.1.2 LoRa Technology Applied in a Cocoa Environment [3]

Penelitian ini bertujuan untuk menguji kinerja teknologi LoRa dalam lingkungan perkebunan kakao dengan menganalisis parameter komunikasi seperti *Received Signal Strength Indicator (RSSI)*, *Signal-to-Noise Ratio (SNR)*, *airtime*, dan *data rate* berdasarkan berbagai faktor penyebaran (*Spreading Factor/SF*). Sistem yang digunakan terdiri dari node sensor TTGO LoRa32, *gateway* berbasis Raspberry Pi 4 dengan modul RAK2287, serta server TTN untuk pemantauan data. Pengujian dilakukan dengan menerapkan topologi *star* dimana data dari sensor kelembaban tanah dikirim ke *gateway* dan dianalisis berdasarkan jarak serta nilai SF yang berbeda.

Hasil penelitian menunjukkan bahwa nilai RSSI dan SNR menurun seiring dengan peningkatan jarak antara node dan *gateway*. Dengan SF7 hingga SF10, *airtime* meningkat dari 0,046 detik menjadi 0,329 detik, yang berarti bahwa semakin tinggi SF, semakin lama waktu yang dibutuhkan untuk mengirimkan data, tetapi cakupan jarak meningkat. Dalam pengujian lapangan, jarak maksimum yang dapat dicapai adalah 1,1 km menggunakan SF10, tetapi dengan konsekuensi penurunan *data rate* menjadi 0,9 Kb/s. Untuk jarak yang lebih pendek seperti 225–675 meter, SF7 dan SF8 memberikan kinerja optimal dengan waktu transmisi lebih singkat dan *data rate* lebih tinggi.

Secara keseluruhan, penelitian ini menegaskan bahwa konfigurasi SF sangat mempengaruhi efisiensi komunikasi dalam jaringan LoRa. Meskipun SF yang lebih tinggi memungkinkan jangkauan lebih jauh, hal ini berdampak pada penurunan *data rate* dan peningkatan *airtime*. Oleh karena itu, pemilihan SF harus disesuaikan dengan kondisi lingkungan dan kebutuhan aplikasi agar mendapatkan keseimbangan optimal antara jangkauan, efisiensi energi, dan kecepatan transmisi data.

Dari penelitian diatas, ada beberapa poin yang dapat dimanfaatkan pada penelitian ini yaitu:

1. Pada penelitian diatas, dilakukan pengujian pengukuran pengujian LoRa dimana didapatkan bahwa pada jarak 225, 450 dan 675 m, LoRa dapat mengirimkan data dengan waktu pengiriman yang cukup cepat yaitu dibawah 1 detik. Hasil pengujian tersebut dapat digunakan untuk dijadikan perbandingan pada penelitian ini.

2.1.3 Implementasi Firebase Cloud Messaging Pada Aplikasi Broadcast Informasi Perkuliahan Untuk Mempercepat Pendistribusian Informasi Di STMIK KHARISMA Makassar [4]

Penelitian ini bertujuan untuk mengimplementasikan Firebase Cloud Messaging (FCM) pada aplikasi broadcast informasi perkuliahan guna mempercepat pendistribusian informasi di STMIK Kharisma Makassar. Aplikasi ini dikembangkan menggunakan Android Studio, dengan Cloud Firestore sebagai basis data dan Firebase Cloud Messaging untuk mengelola pemberitahuan secara real-time. Dalam pengujian fungsionalitas, digunakan metode black box testing, sedangkan kecepatan pendistribusian informasi diukur melalui uji performa. Selain itu, kualitas aplikasi dinilai melalui kuesioner kepada pengguna untuk mengetahui tingkat kepuasan mereka terhadap sistem yang dikembangkan.

Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan layak digunakan dan mampu mempercepat pendistribusian informasi. Dari aspek fungsionalitas, pengujian dengan metode black box membuktikan bahwa aplikasi berjalan dengan baik tanpa kesalahan. Dari sisi kecepatan, aplikasi ini mampu mendistribusikan informasi sekitar 2,367 detik. Aplikasi ini lebih cepat sekitar 2,006 detik dibandingkan metode sebelumnya. Selain itu, hasil tanggapan pengguna melalui kuesioner menunjukkan tingkat kepuasan sebesar 92%, yang masuk dalam kategori "Sangat Setuju". Dengan demikian, aplikasi ini terbukti

efektif dalam meningkatkan efisiensi penyebaran informasi perkuliahan di STMIK Kharisma Makassar.

Dari penelitian diatas, ada beberapa poin yang dapat dimanfaatkan pada penelitian ini:

1. Hasil dari pengujian yang dilakukan dari penelitian di atas dapat dijadikan acuan dalam pengujian waktu yang diperlukan server untuk mengirimkan notifikasi melalui Firebase Cloud Messaging.

2.1.4 Implementasi Modul GPS Ublox 6M Dalam Rancang Bangun Sistem Keamanan Motor Berbasis *Internet Of Things* [5]

Penelitian ini dilakukan dengan tujuan untuk mengembangkan sistem keamanan kendaraan bermotor berbasis *IoT*. Sistem yang diimplementasikan menggunakan modul GPS *U-Blox 6M* yang digunakan untuk mendapatkan data posisi kendaraan secara real-time. Tujuan dari penelitian ini adalah untuk menciptakan keamanan yang memungkinkan pengguna untuk melacak kendaraan mereka ketika dalam terjadi kasus pencurian.

Sistem keamanan bermotor yang dirancang berbentuk perangkat keras (*hardware*) dan perangkat lunak (*software*). Pada bagian *hardware*, sistem menggunakan modul GPS untuk mendapatkan titik koordinat lokasi kendaraan. Data tersebut kemudian akan dikirimkan melalui *NodeMCU ESP8266* yang terhubung dengan jaringan internet. Sistem juga menggunakan relay 4 channel yang digunakan untuk mematikan mesin kendaraan jika diperlukan. Input yang diperlukan sistem didapatkan dari *Telegram* sebagai pengontrol dan data yang diterima ditampilkan secara real-time pada aplikasi *Blynk*.

Pada bagian *software*, *Arduino IDE* digunakan untuk memprogram *NodeMCU* dengan library GPS. Library ini memungkinkan *NodeMCU*

membaca data koordinat dari modul GPS melalui pin IO yang telah diinisialisasi pada modul GPS. Proses pengendalian mencakup penerimaan perintah dari *Telegram*, membaca data lokasi GPS, dan mengeksekusi fungsi-fungsi seperti pengendalian relay untuk keamanan kendaraan.

Pada penelitian ini, ada beberapa hal yang dilakukan pengujian yaitu waktu yang diperlukan GPS untuk memperoleh data titik koordinat dan akurasi data koordinat yang dihasilkan. Pengujian waktu yang diperlukan GPS untuk memperoleh data titik koordinat dilakukan sebanyak 5 kali dan pengujian akurasi dilakukan sebanyak 4 kali. Berdasarkan hasil pengujian, waktu yang diperlukan GPS untuk memperoleh data adalah 7 - 12 detik dan akurasi yang didapatkan sekitar 1 - 1,3 meter.

Dari penelitian diatas, ada beberapa poin yang dapat diimplementasikan pada penelitian ini:

1. Pada penelitian di atas, pengujian yang dilakukan dengan mengukur waktu yang diperlukan GPS untuk mendapatkan data koordinat lokasi kendaraan bermotor. Pengujian ini berguna untuk melihat kinerja dari GPS yang akan berpengaruh ke sistem keamanan. Cara pengujian ini dapat diterapkan pada sistem monitoring *IoT* karena penelitian ini juga menggunakan modul GPS untuk mengetahui lokasi dari perangkat *IoT* tersebut.
2. Hasil dari pengujian yang dilakukan dari penelitian di atas dapat dijadikan acuan dalam pengujian waktu yang diperlukan GPS untuk mendapatkan data koordinat karena modul GPS yang digunakan merupakan modul yang diproduksi oleh *Ublox*.

2.2 Tinjauan Teori

2.2.1 Internet of Things

Internet of Things (IoT) adalah jaringan perangkat yang saling terhubung dan berkomunikasi serta bertukar data satu sama lain melalui internet. [6] Teknologi ini diterapkan dalam berbagai bidang seperti rumah pintar, perawatan kesehatan, transportasi, pertanian, dan otomatisasi industri. IoT memungkinkan perangkat untuk mengumpulkan dan berbagi data, yang mengarah pada peningkatan efisiensi, kenyamanan, dan proses pengambilan keputusan.

Konsep *Internet of Things* (IoT) telah berkembang pesat sejak 1990-an ketika perangkat pertama yang terhubung ke internet, seperti pemanggang roti John Romkey dan mesin penjual otomatis di Carnegie Mellon, muncul. Kemajuan signifikan seperti sistem satelit andal DARPA pada 1993 membuka jalan bagi akses internet yang lebih luas dan GPS. Kevin Ashton menciptakan istilah “IoT” pada 1999, menggambarkan objek berperangkat sensor yang berkomunikasi melalui internet.

Pada 2000-an, teknologi nirkabel dan solusi *machine-to-machine* (M2M) meningkat, dengan inovasi seperti lemari es terhubung internet dari LG dan pengembangan teknologi NFC oleh Sony dan Philips. Pada pertengahan 2000-an, sensor hemat daya dan komputasi awan mendorong IoT lebih jauh, dibuktikan dengan penggunaan RFID oleh Walmart untuk pelacakan inventaris. 2010-an ditandai oleh munculnya *Industrial IoT* (IIoT) dengan adopsi oleh perusahaan besar seperti Cisco dan IBM serta inisiatif seperti “*Smart City Switzerland*.” Munculnya *edge computing* membantu mengatasi keterbatasan komputasi awan. Pada 2020-an, kombinasi AI dan ML dengan IoT membuka kemampuan baru dalam analisis data, otomatisasi, dan pengambilan keputusan prediktif, mendorong inovasi di bidang kesehatan, kota pintar, dan kembar digital. [3]

Sistem Internet of Things biasanya memiliki 3 tahapan arsitektur yaitu: [7]

1. *Physical perception layer* terdiri dari perangkat-perangkat fisik seperti sensor, aktuator, dan terminal mobile. Fungsi utamanya adalah menjembatani dunia fisik dengan dunia virtual melalui pengumpulan data lingkungan, mendigitalisasikannya, dan mengirimkannya ke lapisan di atasnya. Lapisan ini mengabstraksi objek-objek fisik, menghasilkan data yang berguna bagi aplikasi-aplikasi yang memerlukannya.
2. *Network layer* mencakup semua komponen jaringan, termasuk berbagai teknologi dan protokol komunikasi. Tugas utamanya adalah mengangkut data yang diperoleh dari lapisan persepsi fisik melalui teknologi seperti jaringan seluler, jaringan sensor nirkabel, dan jaringan ad hoc. Lapisan ini memastikan bahwa data dienkode, ditransmisikan, digabungkan, dan diproses pada level dasar sebelum diteruskan ke lapisan aplikasi.
3. *Application layer* memuat logika bisnis dari berbagai sektor IoT yang menggunakan informasi bernilai tambah yang dihasilkan dari perangkat fisik. Aplikasi-aplikasi ini dapat diterapkan di salah satu lapisan (*cloud, fog, atau edge*), bergantung pada kompleksitas pemrosesan dan kebutuhan kualitas layanan. Lapisan ini bertanggung jawab untuk menyediakan layanan akhir kepada pengguna berdasarkan data yang telah diproses.

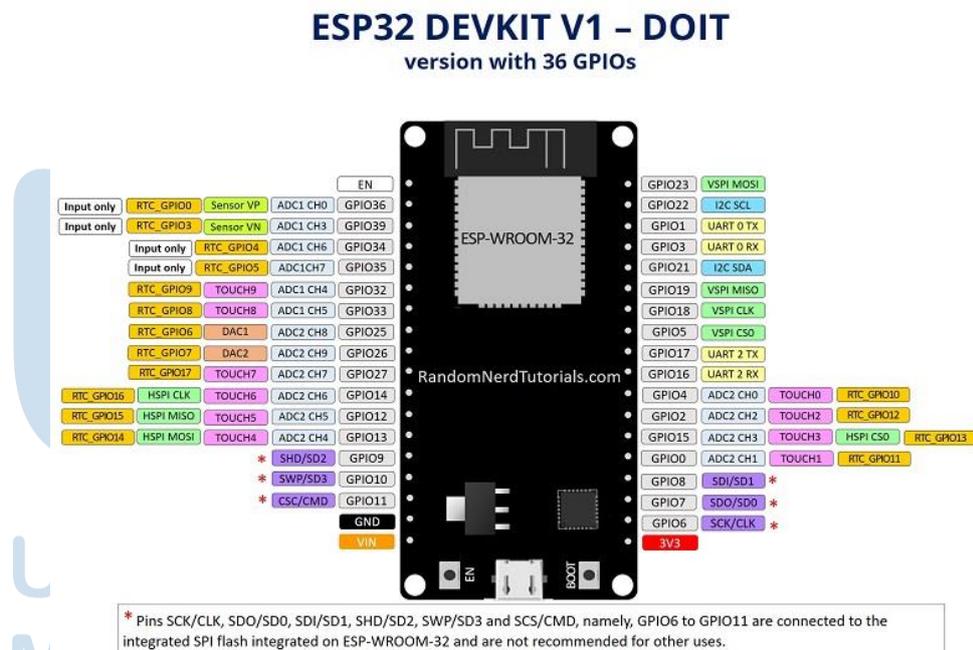
UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.2.2 ESP32

ESP32 adalah mikrokontroler yang dikembangkan oleh sebuah perusahaan China bernama Espressif Systems yang *low-cost* serta telah terintegrasi dengan bluetooth dan wifi. ESP32 sangat populer karena memiliki beragam fitur serta keunggulan lainnya seperti mikrokontroler dengan *dual-core processor* dan kapasitas memori yang besar.

Selain fitur dan spesifikasinya, ESP32 juga mendukung beberapa bahasa pemrograman sehingga penggunaanya lebih *flexible* dalam menggunakan ESP32. Bahasa pemrograman utama yang digunakan adalah C/C++, terutama melalui Espressif IoT Development Framework (ESP-IDF) dan Arduino IDE. Selain itu, **Python** dalam bentuk MicroPython juga didukung, memungkinkan pengembangan yang lebih cepat dan sederhana, terutama untuk prototipe. **Lua** adalah bahasa lain yang kompatibel dengan ESP32 melalui platform seperti NodeMCU.

Berikut merupakan Pinout dari ESP32 terlihat pada gambar 2.1.



Gambar 2.1 ESP32 Pinout [8]

Dari gambar diatas, dapat dilihat bahwa ESP32 memiliki 38 Pin dimana setiap pin memiliki periferalnya masing-masing. Berikut beberapa periferel yang terdapat pada ESP32:

1. 18 ADC (*Analog To Digital*) Converter
2. 3 SPI (*Serial Peripheral Interface*)
3. 3 UART (*Universal Asynchronous Receiver-Transmitter*)
4. 2 I2C (*Inter-Integrated Cicuit*) Interfaces
5. 16 PWM (*Pulse Width Modulation*) Output Channels
6. 2 DAC (*Digital To Analog*) Converter
7. 2 I2S (*Inter-IC Sound*) Interfaces
8. 10 *Capacitive Sensing GPIOs*

ESP32 sendiri memiliki spesifikasi yang terlihat pada tabel 2.1. [9]

Tabel 2.1 Spesifikasi ESP32

<i>Wifi Protocols</i>	802.11 b/g/n (802.11n up to 150 Mbps)
<i>Bluetooth Protocols</i>	Bluetooth v4.2 BR/EDR and Bluetooth LE specification
<i>Module interfaces</i>	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®), compatible with ISO11898-1 (CAN Specification 2.0)
<i>Operating voltage/Power supply</i>	3.0 V ~ 3.6 V
<i>Operating current</i>	Average: 80 mA
<i>Operating temperature range</i>	-40 °C ~ +85 °C

<i>Total Pin</i>	38
ROM	448 KB
RAM	<ul style="list-style-type: none"> ● 520 KB on-chip SRAM (<i>Static Random Access Memory</i>) ● 8 KB RTC SRAM (<i>RTC FAST Memory</i>) ● 8 KB RTC SRAM (<i>RTC SLOW Memory</i>)

2.2.3 Material Konduktif [10]

Material konduktif adalah jenis material yang memiliki kemampuan tinggi untuk menghantarkan listrik. Hal ini disebabkan oleh keberadaan elektron bebas di dalam strukturnya. Elektron-elektron ini tidak terikat kuat pada atom tertentu, sehingga mereka bisa bergerak bebas di dalam material, membentuk arus listrik ketika ada beda potensial. Struktur logam pada umumnya bersifat kristalin, di mana atom-atom tersusun dalam pola yang teratur dan tidak acak.

Dalam material konduktif, elektron dari pita valensi dapat dengan mudah melewati tingkat Fermi (tingkat energi tertinggi yang ditempati elektron pada suhu nol mutlak) dan masuk ke pita konduksi. Pita konduksi inilah yang memungkinkan elektron bebas bergerak dan menyebabkan aliran listrik. Karena elektron-elektron ini tidak bergantung pada atom pusat, konduktivitas listrik pada material ini sama di semua arah (isotropik) baik secara in-plane (sejajar dengan permukaan) ataupun through-plane (menembus lapisan).

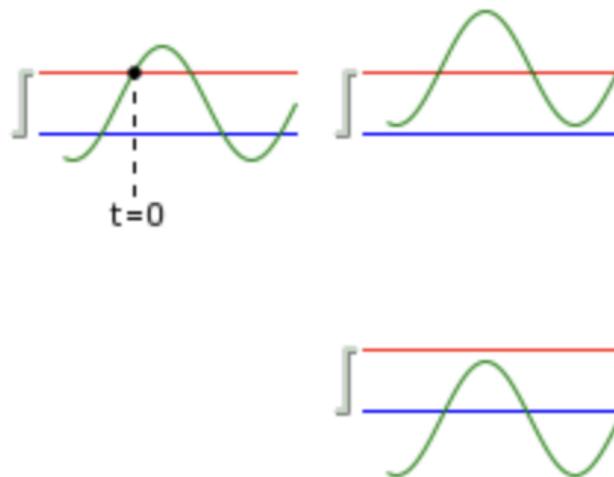
Salah satu contoh material konduktif adalah logam. Logam memiliki konduktivitas listrik yang sangat tinggi, dengan nilai konduktivitas listrik lebih dari 1000 hingga 15.000 S/cm (Siemens per centimeter). Ini karena

keberadaan elektron-elektron bebas yang melimpah, memungkinkan arus listrik mengalir dengan resistansi yang sangat rendah.

2.2.4 *Signal Edge Detection*

Signal Edge Detection adalah teknik yang banyak digunakan dalam perangkat lunak embedded, pengembangan berbasis model, dan elektronik [11]. *Signal Edge* disebut sebagai perubahan sinyal dari keadaan *HIGH* ke keadaan *LOW* atau sebaliknya. Berdasarkan jenis transisi, terdapat tiga jenis deteksi tepi yang berbeda:

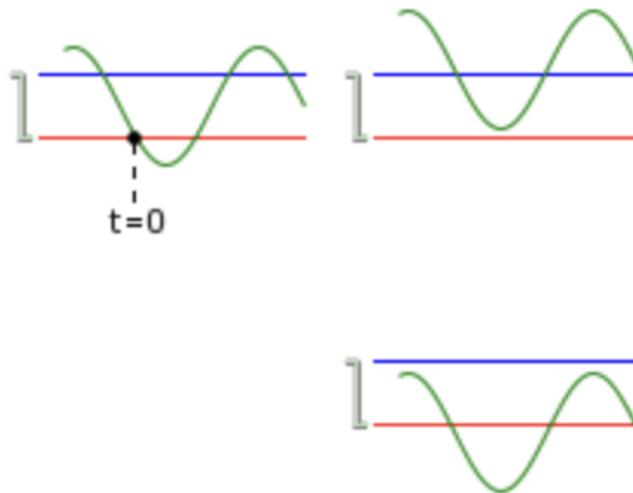
1. *Rising Edge*: ketika sinyal input bertransisi dari keadaan rendah ke keadaan tinggi seperti yang terlihat pada gambar 2.2.



Gambar 2.2 *Rising Edge* [12]

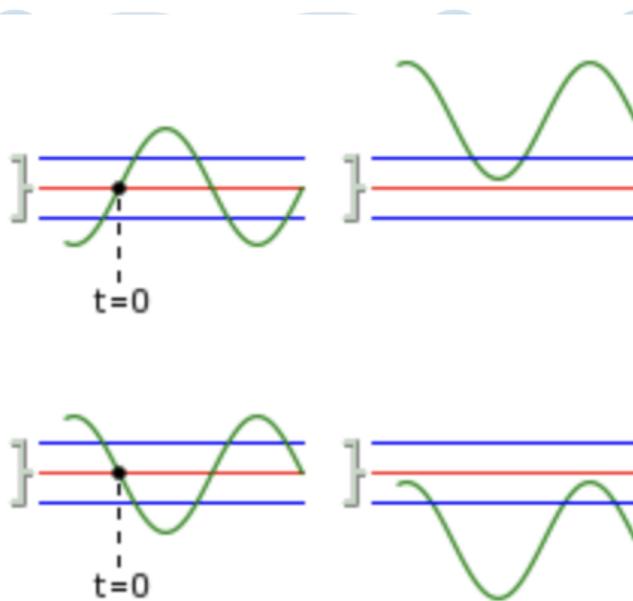
UNIVERSITAS
MULTIMEDIA
NUSANTARA

2. *Falling Edge*: ketika sinyal input bertransisi dari keadaan tinggi ke keadaan rendah seperti yang terlihat pada gambar 2.3.



Gambar 2.3 *Falling Edge* [12]

3. *Either Edge*: ketika sinyal input berubah keadaan, dari tinggi ke rendah atau dari rendah ke tinggi seperti yang terlihat pada gambar 2.4.



Gambar 2.4 *Either Edge* [12]

Signal edge detection sering digunakan sebagai pemicu (trigger) dalam sistem elektronik. Dalam aplikasi embedded, *edge detection* digunakan untuk mengawasi sinyal input dari sensor, tombol atau perangkat lainnya. Dengan *signal edge detection*, sebuah sistem dapat menjalankan logika yang sesuai berdasarkan interaksi pengguna atau kondisi lingkungan.

2.2.5 GPS Module

GPS (*Global Positioning System*) adalah sebuah sistem navigasi yang memungkinkan orang atau perangkat untuk menentukan lokasi mereka di bumi secara presisi dimana saja. Menurut Hofmann-Wellenhof, GPS terbagi menjadi 3 segmen yaitu: [13]

1. Segmen luar angkasa (*Space Segment*)

Segmen antariksa GPS terdiri dari konstelasi 24 satelit yang berada pada ketinggian sekitar 26.560 km dari pusat bumi. Mengingat radius bumi sekitar 6.378 km, maka konstelasi satelit tersebut berada pada ketinggian sekitar 20.192 km di atas permukaan bumi. Satelit-satelit ini ditempatkan di ketinggian yang tinggi untuk memastikan stabilitas dan presisi, karena pada ketinggian tersebut gerakan satelit tidak terpengaruh oleh hambatan atmosfer. Konstelasi ini terbagi menjadi enam orbit dengan sudut inklinasi 55 derajat terhadap ekuator bumi, dan setiap orbit dipisahkan dengan sudut 60 derajat untuk memastikan bahwa setiap titik di permukaan bumi dapat dipantau oleh satelit.

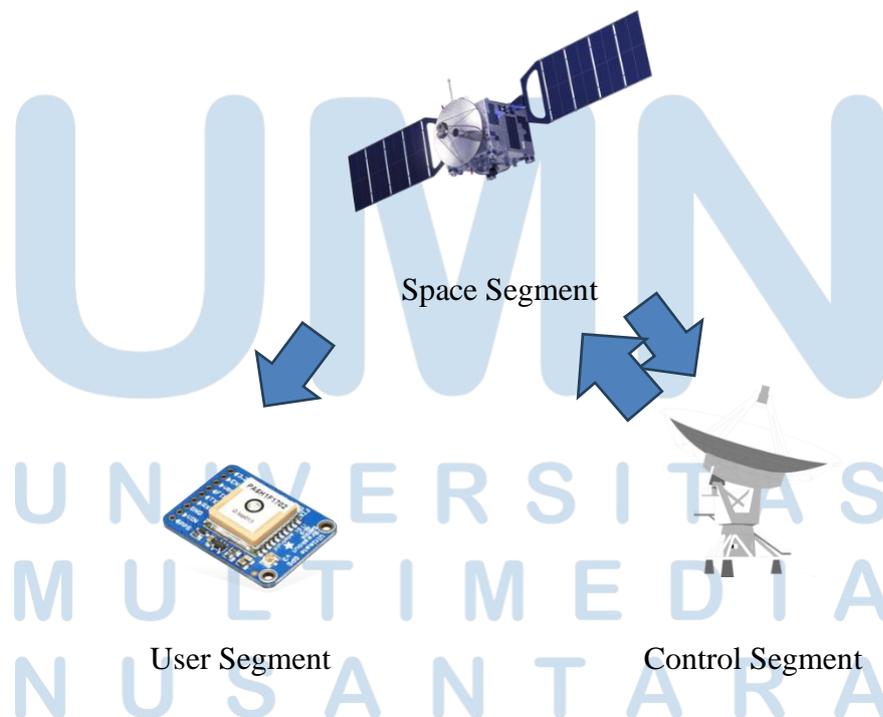
2. Segmen kontrol (*Control Segment*)

Segmen kontrol, yang sering disebut sebagai segmen darat, terletak di permukaan bumi. Pada awalnya, GPS hanya memiliki lima stasiun kontrol yang terletak di Hawaii, Colorado, Pulau Ascension di Samudra Atlantik Selatan, Diego Garcia di Samudra Hindia, dan Kwajalein di Samudra Pasifik Utara. Namun, dalam perkembangannya, jumlah stasiun kontrol ini bertambah menjadi enam belas untuk memastikan setiap satelit dapat dipantau oleh

setidaknya tiga stasiun kontrol. Fungsi utama stasiun-stasiun ini adalah untuk memantau apakah satelit-satelit tetap berada dalam orbitnya dan untuk menyinkronkan waktu pada satelit.

3. Segmen pengguna (*User Segment*)

Segmen pengguna mengacu pada penerima GPS yang terdiri dari antena, rantai frekuensi radio (RF), dan konverter analog-ke-digital (ADC). Antena digunakan untuk menerima sinyal yang dikirim oleh satelit, kemudian rantai RF akan memperkuat sinyal ke frekuensi yang sesuai, dan sinyal yang diinginkan akan didigitalkan oleh konverter ADC. Untuk menentukan posisi pengguna, segmen pengguna dilengkapi dengan perangkat lunak yang digunakan untuk memproses sinyal output yang telah didigitalkan. Segmen pengguna juga memiliki jam lokal, namun berbeda dengan satelit yang menggunakan jam atom, penerima GPS menggunakan jam kristal yang lebih murah dan disesuaikan dengan waktu satelit. Masalah waktu yang akurat ini dapat diatasi dengan menggunakan trik triangulasi dari empat satelit.



Gambar 2.5 Cara kerja GPS

GPS bekerja dengan menggunakan tiga segmen utama: segmen antariksa, segmen kontrol, dan segmen pengguna. Satelit di segmen antariksa secara terus-menerus mengirimkan sinyal radio ke bumi, yang diterima oleh stasiun kontrol di segmen kontrol dan perangkat penerima di segmen pengguna. Ketika penerima GPS diaktifkan, ia akan menerima informasi waktu pengiriman sinyal dan lokasi satelit dari sinyal tersebut. Dengan menggunakan informasi ini, penerima GPS dapat menentukan posisi pengguna.

Untuk menentukan lokasi secara akurat, GPS memanfaatkan teknik matematika yang disebut trilaterasi. Setiap satelit memberikan informasi tentang jarak antara satelit dan penerima, yang digambarkan sebagai sebuah bola dengan satelit sebagai pusatnya. Ketika penerima GPS menerima sinyal dari tiga satelit, tiga bola terbentuk dan berpotongan di dua titik. Satu dari dua titik ini adalah lokasi yang benar dari penerima di permukaan bumi. Dengan demikian, penerima GPS dapat menentukan posisi pengguna dalam tiga dimensi (x, y, z).

2.2.6 Backend (Express JS)

Backend adalah *software* yang berjalan di server dan bertanggung jawab atas segala *logic* yang diperlukan pada sebuah sistem atau aplikasi, melakukan pengelolaan data, dan komunikasi dengan frontend. Backend menerima permintaan dari pengguna melalui frontend, kemudian memproses permintaan tersebut, dan mengirimkan respons kembali ke pengguna. Ini mencakup pengelolaan basis data, otentikasi pengguna, dan pemrosesan data secara real-time.

Backend berperan penting dalam mengatur interaksi antara pengguna dan aplikasi. Ini termasuk penyimpanan data pengguna, validasi data, pengaturan sesi pengguna, dan pengiriman data yang diminta oleh pengguna. Backend juga mengelola API (*Application*

Programming Interface) yang memungkinkan komunikasi antara aplikasi dan sistem eksternal atau aplikasi lain.

2.2.6.1 ExpressJS

Salah satu framework backend yang paling populer adalah ExpressJS. Framework ExpressJS merupakan framework *web application* yang dibangun di atas Node.js (sebuah platform *compiler* Javascript yang dibangun di atas Javascript V8 milik Google Chrome). Express JS menawarkan beberapa fitur mencakup dukungan untuk berbagai metode HTTP, integrasi yang mudah dengan basis data, dan kemampuan untuk membuat API RESTful. Selain itu, kerangka kerja ini memungkinkan penggunaan fungsi *middleware* untuk menangani permintaan dan respons. ExpressJS banyak digunakan karena menyederhanakan proses pembangunan aplikasi web dan API di Node.js, sehingga mempercepat dan mempermudah pengembangan aplikasi server-side yang andal.

2.2.7 Frontend (React JS)

Frontend adalah bagian dari aplikasi web yang berinteraksi langsung dengan pengguna. *Frontend* mencakup semua yang dilihat dan berinteraksi oleh pengguna. *Frontend* dikembangkan menggunakan teknologi web seperti HTML, CSS, dan JavaScript, yang semuanya bekerja bersama untuk menciptakan antarmuka pengguna yang responsif dan fungsional.

2.2.5.1 React JS

Frontend bertujuan untuk menyediakan pengalaman pengguna yang menarik dan efisien. Ini termasuk merancang tata letak yang mudah dinavigasi, memastikan responsivitas desain untuk berbagai perangkat, serta mengelola interaksi pengguna dengan sistem backend melalui API (*Application Programming Interface*). *Frontend* juga berperan untuk

memastikan bahwa informasi disampaikan kepada pengguna dengan jelas dan efektif. Secara keseluruhan, frontend adalah bagian penting dari aplikasi yang mempengaruhi bagaimana pengguna berinteraksi dengan sebuah sistem atau aplikasi.

React.js adalah *framework* JavaScript open-source yang digunakan untuk membangun antarmuka pengguna (UI) untuk aplikasi web. React.js terkenal karena berbasis komponen yang memungkinkan pengembang untuk membuat komponen UI yang dapat digunakan kembali. Penggunaan *virtual DOM* di React meningkatkan efisiensi dengan meminimalkan pembaruan langsung ke DOM, sehingga aplikasi web dapat berjalan lebih cepat dan responsif.

Salah satu keunggulan React.js adalah kemampuannya untuk digunakan dalam pengembangan *single-page applications* (SPA). SPA adalah aplikasi web yang memuat satu halaman HTML statis dan hanya memperbarui bagian dari halaman tersebut sesuai interaksi pengguna, tanpa harus melakukan *reload* seluruh halaman.

Selain itu, React.js sangat fleksibel dan dapat diintegrasikan dengan berbagai pustaka atau alat lain, salah satunya adalah untuk membangun *Progressive Web Apps* (PWA), yang menawarkan pengalaman aplikasi web yang menyerupai aplikasi native. PWA dirancang agar dapat diakses melalui browser web, tetapi juga menawarkan fitur-fitur canggih seperti kemampuan untuk bekerja secara offline, pengiriman *push notifications*, serta pengalaman yang responsif di berbagai perangkat. Salah satu komponen utama dari PWA adalah penggunaan *service worker*, yaitu skrip JavaScript yang berjalan di latar belakang dan memungkinkan aplikasi untuk menyimpan data secara lokal dan berfungsi tanpa internet.