

SISTEM PENGIRIMAN DATA DAN TAMPILAN SIMULASI DINAMIKA PERUBAHAN TEKANAN UDARA MATRAS DEKUBITUS DAN KOORDINAT LOKASI ALAT MEMANFAATKAN WEB SERVER

Christophorus K. Wisma Nugraha¹, Hartono Pranjoto², Lanny Agustine³

^{1,2,3}Jurusan Teknik Elektro, Fakultas Teknik

Universitas Katolik Widya Mandala Surabaya

e-mail: ¹christophorusk29@gmail.com, ²pranjoto@ukwms.ac.id*,

³lanny.agustine@ukwms.ac.id

ABSTRAK

Pemantauan kesehatan pasien jarak jauh untuk menjelaskan kondisi kesehatan pasien telah mendorong manusia untuk membuat sistem pemantauan kesehatan berbasis IoT (*Internet Of Things*). Pemantauan dirasa penting untuk pasien yang berbaring di kasur dan tidak bergerak dalam jangka waktu yang lama, karena kondisi tersebut dapat menimbulkan ulkus dekubitus. Untuk mencegah masalah tersebut digunakan matras pereduksi dekubitus yang mengubah titik tumpu pada kulit secara berkala. Ada kondisi saat matras mengalami malafungsi, dan matras tidak lagi mengubah titik tumpu pada kulit secara berkala. Untuk itu perlu adanya pemantauan tekanan kantung udara matras sehingga kondisi tersebut bisa diatasi. Raspberry Pi 3 menjalankan simulasi perubahan siklus tekanan kantung udara pada matras secara periodik. Data simulasi tersebut akan dikirimkan ke *server* menggunakan modul GPS/GPRS SIM868 dan akan disimpan dalam *database*. Lokasi alat dapat diketahui dengan menggunakan modul GPS yang terintegrasi dalam modul GPS/GPRS SIM868. Dalam halaman web ditampilkan simulasi kembang kempis kantung udara matras yang diindikasikan dengan warna. Modul GPS/GPRS SIM868 dapat berkomunikasi dengan Raspberry Pi dan dapat mengirimkan data simulasi perubahan siklus tekanan udara pada matras. Pengujian tampilan halaman web dengan siklus simulasi kembang kempis kantung udara dengan indikasi perubahan warna pada matras sudah sesuai dengan data simulasi tekanan kantung udara yang dikirimkan dari Raspberry Pi ke *database*.

Kata kunci: Dekubitus, Matras, Modul GPS/GPRS, Web Server

ABSTRACT

Remote patient health monitoring to explain how the patient's health condition has encouraged humans to create an IoT (Internet of Things) based health monitoring system. Monitoring is considered important for patients who lie in bed and do not move for long periods of time, because these conditions can cause ulkus decubitus. To prevent this problem, pressure-reducing mattresses are used to periodically change the fulcrum of the skin. But there are conditions when the mattress is malfunctioning, so the mattress no longer changes the fulcrum of the skin on a regular basis. For this reason, it is necessary to monitor the pressure of the air sacs of the mattress so that the condition can be overcome. Raspberry Pi 3 simulates changes in the cycle of air bag pressure on the mattress periodically. The simulation data will be sent to the server using the SIM868 GPS/GPRS module and will be stored in the database. The location of the device can be determined using the GPS module integrated in the SIM868 GPS/GPRS module. The web page displays a simulation of the inflation of the mattress air bag which is indicated by color. The SIM868 GPS/GPRS module can communicate with the Raspberry Pi and can transmit simulated data on changes in the air pressure cycle on the

mattress. Testing the appearance of a web page with a simulation cycle of the air bag deflating with an indication of the color change on the mat is in accordance with the air bag pressure simulation data sent from the Raspberry Pi to the database.

Keywords: Decubitus, Mattress, GPS/GPRS Module, Web Server

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi senantiasa mengalami peningkatan pesat pada pemanfaatannya. Salah satu yang menjadi sorotan adalah penggunaan teknologi informasi dalam bidang kesehatan.

Pemantauan data kesehatan saat ini dilakukan secara langsung di rumah sakit oleh tim medis. Salah satu kondisi pasien yang butuh pemantauan adalah pasien dekubitus. Untuk mencegah masalah tersebut saat ini digunakan matras pereduksi dekubitus yang mengubah titik tumpu pada kulit secara berkala. Namun ada kondisi saat matras mengalami malafungsi dan tidak diketahui oleh paramedis, sehingga matras tidak lagi mengubah titik tumpu pada kulit secara berkala dan dapat menyebabkan timbulnya ulkus dekubitus. Untuk itu perlu adanya pemantauan tekanan kantung udara matras sehingga kondisi tersebut bisa diatasi dan pasien terhindar dari resiko ulkus dekubitus.

Seiring dengan kemajuan teknologi yang serba mudah dan praktis dengan memanfaatkan internet dapat dibuat sistem simulasi untuk memonitor perubahan tekanan kantung udara matras pereduksi dekubitus secara *real-time* dan jarak jauh. Parameter tekanan kantung udara dan waktu menjadi hal penting untuk membantu paramedis dalam memonitor pasien dekubitus. Karena jika terjadi malafungsi, maka paramedis dapat langsung bertindak dengan

memindahkan posisi pasien secara manual.

TINJAUAN PUSTAKA

A. Raspberry Pi 3 Model B[4]

Raspberry Pi adalah salah satu platform mini PC yang berfungsi sebagai komputer untuk memproses data yang diterima.



Gambar 1. Bentuk Fisik Raspberry Pi 3 Model B

Tabel 1. Spesifikasi Board Raspberry Pi 3 Model B

Chip Mikrokontroler	Broadcom BCM2837 64bit CPU
Kecepatan Prosesor	1.2 GHz
Tegangan Input (USB Type-B)	5 Volt (2.5A)
Port USB	4x USB 2.0 ports
Penyimpanan	Menggunakan Micro SD
SDRAM	1GB LPDDR2
Dimensi	85 mm x 56 mm
Temperatur Operasi	0-50 °C suhu sekitar

Raspberry Pi akan menggunakan koneksi serial UART untuk dapat berkomunikasi dengan modul GSM / GPRS SIM868. Koneksi serial menggunakan *baudrate* 115200 bps dengan pin 8 (TDX) sebagai pengirim dan pin 10 sebagai penerima (RXD). Bentuk fisik dari Raspberry Pi 3 Model B dapat dilihat pada Gambar 1. Spesifikasi Raspberry Pi 3 Model B dapat dilihat pada Tabel 1.

B. GSM/GPRS/GNSS HAT SIM868[6]

SIM868 merupakan modul GPS/GPRS *quad-band* yang menggabungkan teknologi GNSS untuk navigasi satelit. Modul ini memiliki 3 buah fitur yaitu untuk *bluetooth*, GPS dan juga GPRS. *Bluetooth* digunakan untuk transfer data dengan perangkat lain. GPS digunakan untuk mengetahui lokasi alat. GPRS digunakan untuk menghubungkan mikrokontroler dengan jaringan internet, sehingga memungkinkan untuk mengirim data melalui jaringan internet. Koneksi serial UART digunakan untuk dapat berkomunikasi dengan Raspberry Pi. Modul GSM/GPRS SIM868 termasuk dalam GPRS kelas B, yang berarti modul dapat menggunakan *service* GPRS dan GSM secara bersamaan dengan interferensi terhadap GPRS oleh GSM. GPRS akan terhenti saat menerima telepon / SMS dan akan



Gambar 2. Bentuk Fisik GSM/GPRS/GNSS HAT SIM868

terhubung lagi saat *service* GSM telah selesai. Gambar fisik modul dapat dilihat pada Gambar 2 dan tabel spesifikasi pada Tabel 2.

Tabel 2. Spesifikasi GSM/GPRS/GNSS HAT SIM868

Kecepatan <i>downlink</i> dan <i>uplink</i>	Maksimum 85,6 kbps
Tegangan Operasi	5V
Dimensi	30,2 mm x 65 mm
Temperatur Operasi	-40 °C - 50 °C suhu sekitar

C. Protokol Komunikasi[7]

Protokol adalah sistem peraturan yang memungkinkan terjadinya hubungan, komunikasi dan perpindahan data antara dua perangkat atau lebih. Protokol dapat diterapkan pada perangkat keras dan perangkat lunak. Protokol di standarisasi oleh *Internasional Standard Organization* (ISO).

Open System Interconnection (OSI) merupakan standar komunikasi yang diterapkan untuk jaringan komputer. Standar ini digunakan untuk menentukan aturan sehingga seluruh alat komunikasi bisa saling terkoneksi melalui jaringan internet. Dalam model OSI terdapat 7 *layer*. Ketujuh lapisan OSI *layer* tersebut adalah:

- Application Layer* bertanggung jawab atas antarmuka *user* dengan jaringan.
- Presentation Layer* bertanggung jawab dalam mengkodekan dan konversi data dari *application layer*, serta memastikan semua data yang berasal dari *application layer* dapat dibaca pada sistem lainnya.
- Session Layer* bertugas untuk membangun dan memelihara hubungan antara dua koneksi.

- d. *Transport Layer* memastikan komunikasi dari ujung ke ujung terjadi.
- e. *Network Layer* menghubungkan segmen-segmen pada jaringan agar dapat saling berhubungan.
- f. *Datalink Layer* menyediakan link untuk data dan menghubungkan titik ke titik secara langsung.
- g. *Physical Layer* bertugas untuk mengirimkan dan menerima data mentah pada media fisik.

Salah satu jenis protokol komunikasi adalah HTTP (*Hyper Text Transfer Protocol*). HTTP dipergunakan untuk mengirim dokumen dalam *World Wide Web*. HTTP adalah sebuah protokol untuk meminta/menjawab antara *client* dan server. Sebuah *client* HTTP seperti web browser, biasanya memulai permintaan kepada *host* dengan membuat hubungan TCP/IP. Sebuah server HTTP tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET /HTTP/1.1". Begitu kode permintaan tersebut diterima, server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan eror atau pesan lainnya. HTTP mendefinisikan beberapa metode (*Request Method*) yang mengindikasikan aksi yang diinginkan untuk dijalankan pada *resource* yang teridentifikasi, yaitu:

- a. GET untuk meminta sebuah respons dari *resource* yang spesifik. Sejauh ini metode GET merupakan metode yang paling sering digunakan.
- b. HEAD untuk meminta respons identik ke suatu yang menanggapi permintaan GET, tetapi tanpa *response body*. Hal ini berguna untuk menerima meta-information

- yang ditulis pada *header* tanpa harus mengirim seluruh *content*.
- c. POST untuk mengirim data (dari *form* HTML) ke *resource* yang teridentifikasi. Data tersebut dimasukkan ke dalam *body* dari *request*.
- d. DELETE untuk menghapus *resource* yang spesifik (jarang diimplementasikan).
- e. TRACE untuk mengirimkan kembali *request* yang diterima, sehingga *client* dapat melihat apa yang sedang ditambahkan atau diubah pada *request* oleh server.

D. Hypertext Markup Language (HTML)[1]

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan format *hypertext* sederhana yang ditulis ke dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan ke dalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML. HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C).

E. JavaScript[3]

JavaScript adalah suatu bahasa skrip yang didasarkan pada konsep pemrograman berbasis prototipe. Bahasa ini digunakan untuk menyediakan akses skrip untuk objek yang dinamakan di aplikasi lain. *JavaScript* diadopsi dari

bahasa C/C++ yang dikembangkan menjadi bahasa pemrograman web *client-side*. *JavaScript* didesain untuk dapat bekerja sama dengan HTML untuk membuat web *page* yang interaktif, beranimasi dan memiliki visual *effect* yang dinamis. Kode dari *JavaScript* harus diapit oleh tag `<script language="Javascript">` dan diakhiri dengan tag `</script>`.

F. Global Positioning System (GPS)[8]

GPS (*Global Positioning System*) adalah sistem satelit navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi dan kecepatan tiga-dimensi serta informasi mengenai waktu, secara kontinu di seluruh dunia tanpa bergantung waktu dan cuaca, bagi banyak orang secara simultan. GPS dapat memberikan informasi posisi dengan ketelitian bervariasi dari beberapa milimeter (orde nol) sampai dengan puluhan meter. NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System*) adalah nama asli dari Sistem GPS, yang mempunyai tiga segmen yaitu: satelit (*Space Segment*), pengendali (*Control Segment*), dan penerima/pengguna (*User*



Gambar 3. Bidang Orbit Satelit

Segment).

Cara kerja GPS adalah Teknologi GPS memerlukan 24 satelit buatan (mengorbit pada ketinggian 20.200 km), yang disebut juga *space segment* agar semua titik di permukaan bumi dapat terpantau. Gambar 3 menggambarkan 6 bidang orbit satelit yang masing-masing bidang berjarak 60° (6 bidang agar memenuhi 360°), dan tiap bidang orbit terdapat 4 satelit. Dengan susunan seperti ini, diharapkan semua titik di permukaan bumi dapat dipantau oleh 5-10 satelit dalam waktu bersamaan sehingga dapat menyediakan data dan informasi yang sangat akurat.

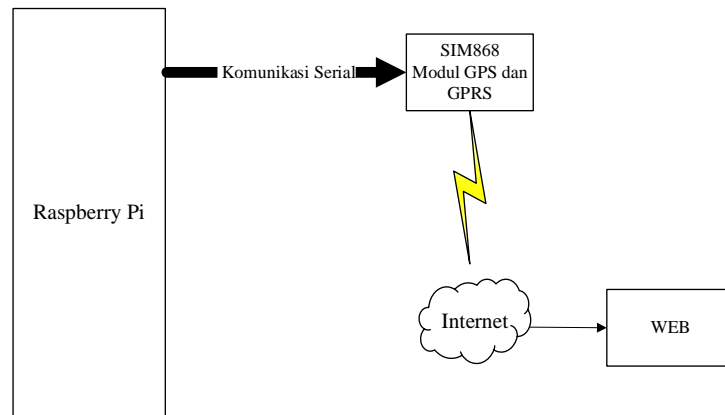
G. MySQL[2]

MySQL merupakan salah satu jenis *database* server, MySQL menggunakan bahasa SQL (*Structured Query Language*) sebagai dasar untuk mengakses *database*. MySQL bersifat *open source* dan bisa digunakan pada berbagai platform.

Pada MySQL digunakan istilah tabel, baris, dan kolom karena MySQL termasuk jenis RDBMS (*Relational Database Management System*). Dalam MySQL terdapat sebuah *database* yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah kolom dan setiap kolom mengandung satu atau beberapa baris.

METODE PENELITIAN

Pada bagian ini akan dibahas mengenai perancangan dan pembuatan sistem. Secara garis besar perancangan Sistem Pengiriman Data dan Tampilan Simulasi Dinamika Perubahan Tekanan Udara Matras Dekubitus dan Titik Lokasi Alat Memanfaatkan Web Server terbagi menjadi perancangan web dan perancangan perangkat lunak. Diagram blok alat dapat dilihat pada Gambar 4.



Gambar 4. Diagram Blok Alat



Gambar 5. Diagram Proses Alat

Raspberry Pi berfungsi sebagai *chip* utama untuk mengatur keseluruhan kerja sistem. *Power Supply* akan terhubung langsung ke Raspberry Pi. Modul GPS/GPRS SIM868 terhubung dengan Raspberry Pi dengan komunikasi serial. Pada Raspberry Pi akan dibuat data simulasi tekanan udara pada matras dekubitus. Data simulasi dibuat berdasarkan siklus perubahan tekanan dari sensor tekanan udara pada matras yang mengacu pada penelitian “Monitor Tekanan dan Kebocoran Udara pada Matras Pereduksi Dekubitus Berbasis Raspberry Pi”[8]. Keluaran data simulasi tersebut kemudian dikirimkan ke modul GPS/GPRS SIM868 dengan komunikasi serial UART. Pengiriman data simulasi ke *database* dilakukan oleh modul GPS/GPRS SIM868 dengan jaringan internet GPRS. Data simulasi yang telah dikirimkan akan disimpan pada *database*. Modul GPS/GPRS SIM868 juga berfungsi untuk mendapatkan data posisi koordinat alat.

Data yang berada di *database* ditampilkan pada *website* dan akan diperbarui 1 detik sekali. Akan dibuat matras virtual lengkap dengan simulasi kembang kempis kantung udara dengan variasi warna. Warna merah untuk kondisi pembuangan angin, warna kuning untuk pengisian matras, dan warna hijau untuk kondisi saat kantung udara matras penuh. Gambaran proses alat dapat terlihat dalam diagram proses alat pada Gambar 5.

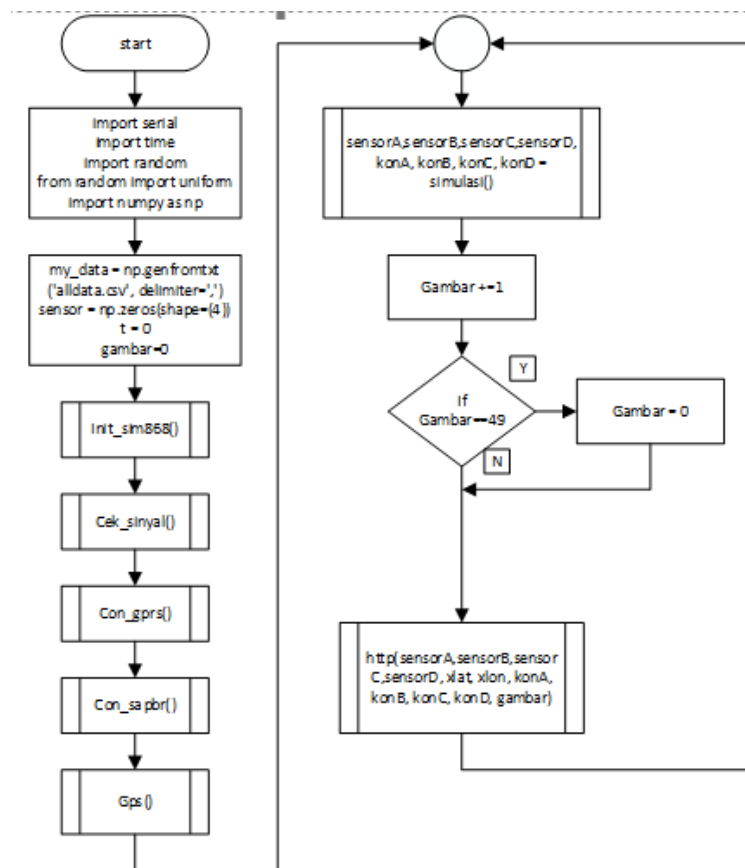
A. Perancangan Perangkat Lunak

Raspberry Pi mengambil data GPS dari GSM/GPRS/GNSS HAT SIM868 serta membuat data simulasi perubahan tekanan 4 sensor tekanan udara kemudian mengirimkan data simulasi dan data GPS tersebut ke *database* menggunakan komunikasi GPRS secara berkala selama 10 detik sekali. Pada Raspberry Pi dilakukan inisialisasi sistem diantaranya-Nya pengaturan *baudrate* GSM/GPRS/GNSS

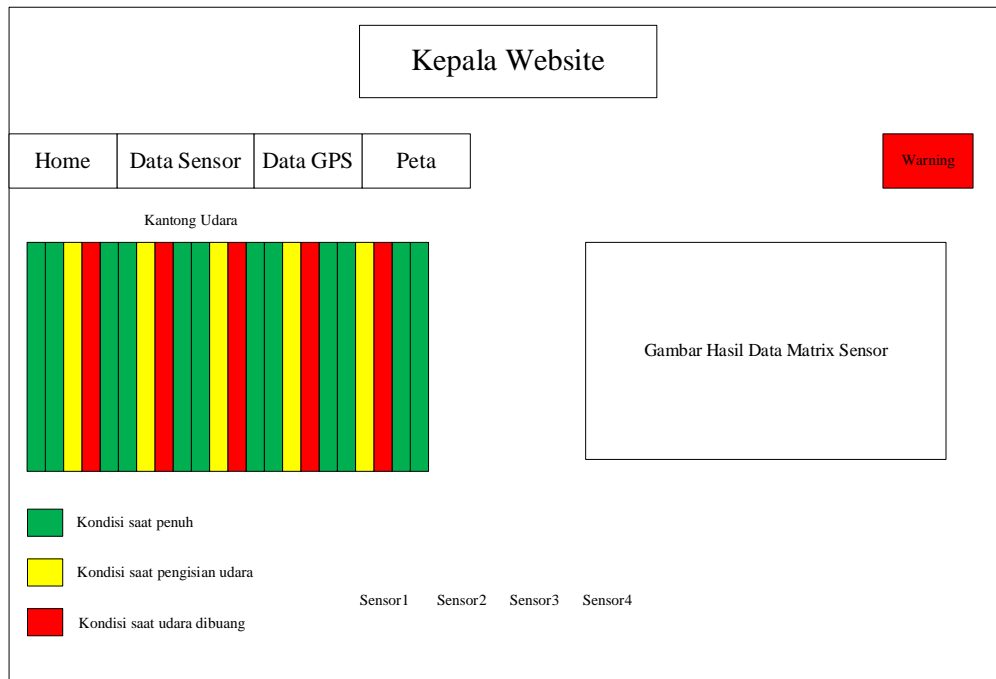
HAT SIM868 dan pengaturan APN yang dibutuhkan untuk komunikasi dengan menggunakan GPRS. Komunikasi serial menggunakan *baudrate* 115200 agar waktu yang diperlukan untuk pertukaran data lebih sedikit, sehingga dapat mengurangi *delay* waktu antara tampilan gambar pada *output* dan perubahan pada *user*. Pada modul GPRS pengaturan *baudrate* bersifat *autobaud* maka modul tersebut akan menyesuaikan pengaturan *baudrate* pada Raspberry Pi. APN diatur menyesuaikan kartu provider yang digunakan, pada artikel ini menggunakan provider Telkomsel sehingga pengaturan APN diatur “internet”. Untuk mensimulasikan kondisi perubahan tekanan pada kantung udara matras dibuat simulator. Simulator tersebut akan menghasilkan rentang nilai keluaran sensor dari 0 - 1 (dalam satuan

psi) yang berubah-ubah setiap siklusnya. Pemilihan rentang nilai ini berdasarkan survei dari penelitian “Monitor Tekanan dan Kebocoran Udara pada Matras Pereduksi Dekubitus Berbasis Raspberry Pi”[5]. Nilai 1 psi diasumsikan sebagai nilai maksimal pada tekanan kantung udara, sehingga akan menampilkan warna hijau (kondisi penuh). Untuk setiap perubahan nilai sensor yang lebih besar (dari 0 ke 0.333) akan menampilkan warna kuning karena kondisi kantung udara terisi, dan jika perubahan nilai menjadi lebih kecil (1 ke 0), maka akan menampilkan warna merah karena kondisi kantung udara mengeluarkan udara. Algoritma keseluruhan kerja Raspberry Pi dapat dilihat pada Gambar 6.

B. Perancangan Halaman Web
Rancangan halaman web tersebut



Gambar 6. Flowchart Kerja Alat



Gambar 7. Rancangan Tampilan Halaman Utama

dapat dilihat pada Gambar 7. Dalam Gambar 7 terlihat tampilan halaman utama web, terdapat beberapa sub menu pada halaman ini yaitu menu *home*, menu Data Sensor, menu Data GPS, dan menu Peta. Menu *Home* merupakan halaman utama yang di dalamnya berisikan 20 kantong udara virtual dengan warna yang nantinya akan berubah-ubah sesuai dengan kondisinya. Pada sisi bawah akan ter tampil nilai sensor1 sampai sensor4 yang diambil dari *database*. Menu Data Sensor akan menampilkan tabel data simulasi sensor dari *database* lengkap dengan waktu diterimanya data tersebut. Data yang ditampilkan pada tabel adalah 10 data terakhir yang masuk ke dalam *database*.

C. Rancangan Struktur Data pada MySQL

Perancangan struktur data pada MySQL bertujuan agar data dapat tersimpan sesuai dengan kebutuhan. Terdapat *database* dengan nama

“*ckwf8695 Datasensor*”, dan di dalamnya terdapat tabel dengan nama “*sensor*”. Tabel “*sensor*” menampung data simulasi dari 4 sensor udara dan data GPS (*latitude* dan *longitude*).

PENGUJIAN SISTEM

A. Pengujian Komunikasi Serial Modul Dengan Raspberry Pi

Pengujian komunikasi modul GPS/GPRS SIM868 dengan Raspberry Pi bertujuan untuk menguji komunikasi data UART TTL antara modul GPS dengan Raspberry Pi. Dibutuhkan beberapa *command* spesifik untuk dapat melakukan pertukaran data antara mikroprosesor dan internet. AT *Command* dituliskan pada program Minicom.

Hasil dari pengujian komunikasi GPS dapat dilihat pada Gambar 8. Gambar tersebut merupakan tampilan program minicom ketika Raspberry Pi menerima data dari modul GPS/GPRS


```

AT
OK
AT+CGNSPWR=1
OK
AT+CGNSPWR?
+CGNSPWR: 1

OK
AT+CGNSIPR=115200
OK
AT+CGNSINF
+CGNSINF: 1,1,20201208032808.000,-7.259749,112.772873,-6.433,0.00,259.9,1,,0.6,,

OK
AT+CGNSPWR=0
OK
_
    
```

Gambar 8. Pengujian Komunikasi GPS

```

Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyS0, 00:29:20

Press CTRL-A Z for help on special keys

AT
OK
AT+CSQ
+CSQ: 25,0

OK
AT+COPS?
+COPS: 0,0,"TELKOMSEL"

OK
AT+CGATT=1
OK
AT+SAPBR=3,1,"Contype","GPRS"
OK
AT+SAPBR=3,1,"APN","internet"
OK
AT+SAPBR=1,1
OK
AT+SAPBR=2,1
+SAPBR: 1,1,"10.41.237.246"

OK
AT+HTTPIPINIT
OK
AT+HTTTPARA="CID",1
OK
AT+HTTTPARA="URL","http://ckwn-web.xyz/Inputv1.php?P1=1&P2=2&P3=3&P4=4&xlat=7.4&xlong=112"
OK
AT+HTTTPACTION=1
OK

+HTTTPACTION: 1,200,25
AT+HTTTPREAD
+HTTTPREAD: 25
Koneksi Database Berhasil
OK
_
    
```

Gambar 9. Pengujian Komunikasi GPRS

SIM868. Dari gambar tersebut terlihat bahwa Raspberry Pi dapat menerima data GPS dari modul GPS/GPRS SIM868 dengan *baudrate* 115200 bps.

Hasil pengujian komunikasi modul GPS dan Raspberry Pi tampak modul dapat memberikan respon dari

AT *command* yang diberikan sehingga data GPS (*latitude* dan *longitude*) berhasil didapatkan. Pada pengujian komunikasi modul GPRS dan Raspberry Pi terlihat bahwa modul dapat memberikan respons atas AT *command* yang diberikan sehingga modul dapat terkoneksi dengan server (ditandai

dengan respons server “Koneksi Database Berhasil”).

B. Pengujian Data GPS

Pengujian dilakukan dengan cara mengambil data GPS melalui modul GPS/GPRS SIM868, lalu membandingkan keluaran data tersebut dengan data GPS dari *handphone* dengan menggunakan aplikasi “GPS Coordinates”.

Tabel 3. Pengujian Data GPS di Lab Instrumentasi

No.	Data GPS Lab Instrumentasi			
	Modul GPS SIM868		Aplikasi GPS Coordinates	
	latitude	longitude	latitude	longitude
1	7,2598 22	112,77281	-7,25982	112,77283
2	7,2598 22	112,77281	-7,25982	112,77283
3	7,2598 21	112,77281	-7,25971	112,77272
4	7,2598 39	112,77273	-7,25982	112,77286
5	7,2598 39	112,77273	-7,25982	112,77286

Hasil pada Tabel 3 menunjukkan perolehan data koordinat GPS dari modul GPS/GPRS SIM868 dan aplikasi Coordinates tampak menunjukkan hasil yang menyerupai. Dan saat dimasukkan ke dalam layanan Google Maps akan menunjuk titik area yang sama.

C. Pengujian Pengiriman Data Simulasi ke Database

Pengujian pengiriman simulasi data ke *database* bertujuan untuk menguji penulisan simulasi data sensor pada *database* yang ada di dalam server. Data yang dikirimkan merupakan data simulasi 4 sensor tekanan udara, titik koordinat GPS (*latitude* dan *longitude*).

```

+AT+CGNSINF
data= +CGNSINF: 1,1,20201211122948.000,-7.328940,112.681543,24.698,0.00,195.4,2
latitude = -7.328940
longitude = 112.681543
+AT+HTTPINIT
OK
OK
AT+HTTPPARA="URL","http://ckwm-web.xyz/Inputv2.php?xlat=-7.328940&xlong=112.681
OK
OK
+HTTPACTION: 1,200,25
+HTTPREAD: 25
OK
P1 = 1.0 P2 = 1.0 P3 = 1.0 P4 =0.0
+AT+HTTPINIT
OK
OK
AT+HTTPPARA="URL","http://ckwm-web.xyz/Inputv1.php?P1=1.0&P2=1.0&P3=1.0&P4=0.0
OK
OK
+HTTPACTION: 1,200,25
+HTTPREAD: 25
OK
P1 = 0.666 P2 = 1.0 P3 = 1.0 P4 =0.333
+AT+HTTPINIT
OK
    
```

Gambar 10. Pengiriman Data dengan GPRS

ID	waktu	P1	P2	P3	P4
992	2020-12-11 19:31:37	1	0	1	1
991	2020-12-11 19:31:26	0.666	0.333	1	1
990	2020-12-11 19:31:15	0.333	0.666	1	1
989	2020-12-11 19:31:04	0	1	1	1
988	2020-12-11 19:30:53	0	1	1	1
987	2020-12-11 19:30:42	0	1	1	1
986	2020-12-11 19:30:30	0	1	1	1
985	2020-12-11 19:30:19	0.333	1	1	0.666
984	2020-12-11 19:30:08	0.666	1	1	0.333
983	2020-12-11 19:29:57	1	1	1	0

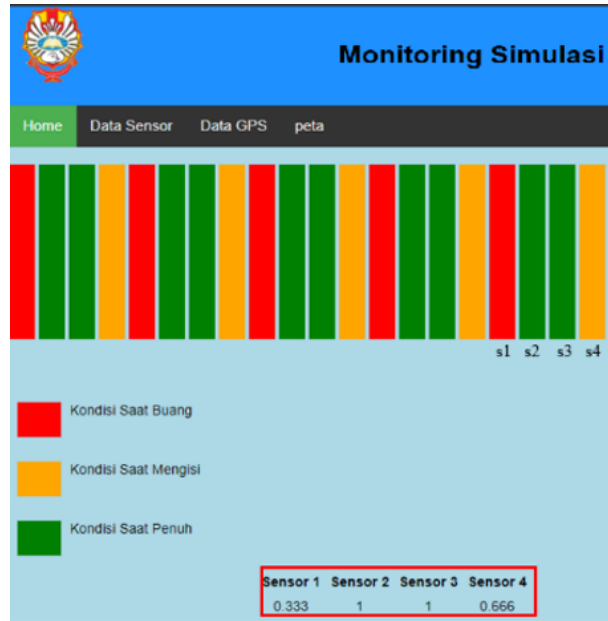
ID	waktu	xlat	xlong
23	2020-12-11 19:29:51	-7.32894	112.682
22	2020-12-11 19:29:15	-7.32894	112.682
21	2020-12-11 19:18:16	-7.32894	112.682
20	2020-12-11 19:10:19	-7.32903	112.682

Gambar 11. Data Pada Database

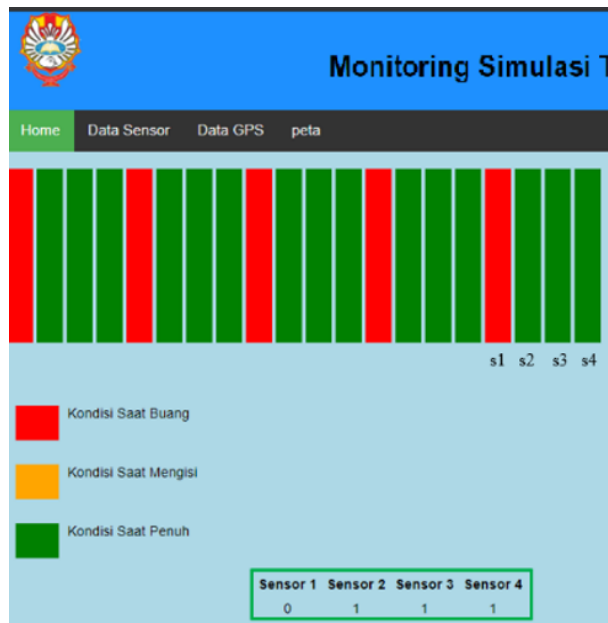
Terlihat pada Gambar 10 dan Gambar 11 yang diberi kotak hijau merupakan *output* dari simulasi sensor tekanan udara, sedangkan kotak berwarna merah merupakan data GPS (*latitude* dan *longitude*). Kedua data tersebut nilainya sama seperti yang ada dalam *database* dan dapat dikatakan pengiriman ke *database* berhasil.

D. Pengujian Tampilan Web

Tampilan pada halaman web berupa beberapa kantong udara virtual



Gambar 12. Tampilan Utama Halaman Web I



Gambar 13. Tampilan Utama Halaman Web II

yang warnanya berubah-ubah sesuai kondisi dari keluaran nilai simulasi 4 sensor tekanan udara dan gambar hasil rekaman *pressure matrix sensor* seperti pada Gambar 12. Halaman web ini dapat dibuka dengan alamat *ckwn-web.xyz*. Dalam halaman web tersebut juga terdapat menu untuk menampilkan tabel simulasi data *output* sensor tekanan

udara dan data GPS (*latitude* dan *longitude*) yang tersimpan di *database*. Menu peta untuk menampilkan peta dan penanda dimana lokasi tempat alat berada Gambar 17.

Pada Gambar 12, Gambar 13 dan Gambar 14 kotak merah dan hijau, data yang dikirimkan oleh Raspberry Pi berhasil ditampilkan pada halaman web.

Sedangkan pada Gambar 12 dan Gambar 13 terjadi perubahan nilai data Sensor1 dan Sensor4. Sensor1 yang awalnya bernilai 0.333 psi berubah menjadi 0 psi dan Sensor4 yang awalnya 0.666 psi berubah menjadi 1 Psi. Perubahan ini akan menyebabkan perubahan warna pada kantung udara, kantung kelompok s4 dari berwarna kuning menjadi hijau yang artinya kondisi kantung telah penuh (dalam simulasi sensor tekanan udara diasumsikan nilai 1 psi adalah nilai maksimal). Pada kelompok s1 tidak terjadi perubahan warna karena masih menjalani proses buang dari nilai 0.333 psi ke 0 psi. Maka dapat disimpulkan bahwa sistem mampu mengakses keluaran simulasi data sensor tekanan udara dan menggunakan data tersebut untuk menghasilkan tampilan kantong udara virtual.

```
P1 = 0.333 P2 = 1.0 P3 = 1.0 P4 = 0.666
AT+HTTPINIT
OK
OK
AT+HTTPPARA="URL","http://ckwm-web.xyz/Inputv1.php?P1=0.333&P2=1.0&P3=1.0&P4=0.666
OK
OK
+HTTPACTION: 1,200,25
+HTTPREAD: 25
OK
P1 = 0.0 P2 = 1.0 P3 = 1.0 P4 = 1.0
AT+HTTPINIT
OK
OK
AT+HTTPPARA="URL","http://ckwm-web.xyz/Inputv1.php?P1=0.0&P2=1.0&P3=1.0&P4=1.0
OK
OK
+HTTPACTION: 1,200,25
+HTTPREAD: 25
OK
```

Gambar 14. Tampilan Pada Jendela Terminal

Pada kolom data sensor akan ditampilkan rekaman data simulasi yang dikirimkan dari Raspberry Pi dan tersimpan dalam *database*. Terlihat pada Gambar 15, data simulasi yang dikirimkan dari Raspberry Pi berhasil ditampilkan pada halaman Data Sensor. Data simulasi perubahan tekanan kantong udara dalam satuan psi.

No	Date	Sensor1	Sensor2	Sensor3	Sensor4
997	2020-12-12 09:41:41	0	1	1	1
996	2020-12-12 09:38:25	0.333	1	1	0.666
995	2020-12-12 09:36:23	0.666	1	1	0.333
994	2020-12-12 09:34:49	1	1	1	0
993	2020-12-12 09:33:58	1	1	1	0
992	2020-12-11 19:31:37	1	0	1	1
991	2020-12-11 19:31:26	0.666	0.333	1	1
990	2020-12-11 19:31:15	0.333	0.666	1	1

Gambar 15. Tampilan Kolom Data Simulasi Sensor

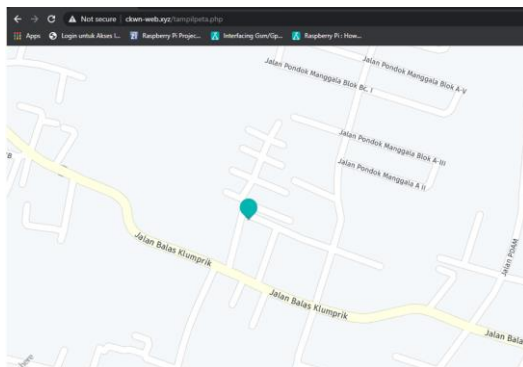
Untuk kolom Data GPS akan ditampilkan tabel data koordinat *latitude* dan *longitude*. Pada Gambar 16 tampak data GPS yang dikirimkan oleh Raspberry Pi dapat ditampilkan pada halaman web kolom Data GPS.

No	Date	Latitude	Longitude
25	2020-12-12 10:47:29	-7.32894	112.681
24	2020-12-12 09:54:09	-7.32894	112.682
23	2020-12-11 19:29:51	-7.32894	112.682
22	2020-12-11 19:29:15	-7.32894	112.682
21	2020-12-11 19:18:16	-7.32894	112.682
20	2020-12-11 19:10:19	-7.32903	112.682

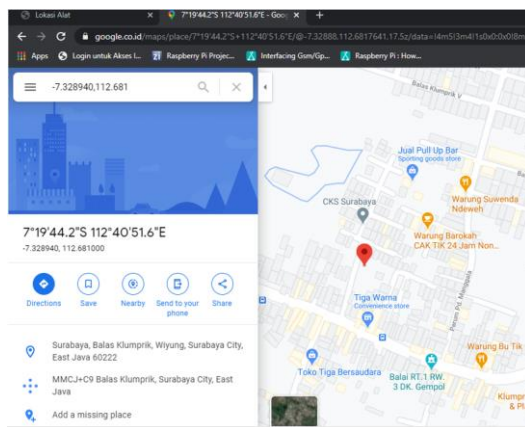
Gambar 16. Tampilan Kolom Data GPS

Untuk tampilan peta pada halaman web *marker* yang ditunjuk mengacu pada data *latitude* dan *longitude* yang terakhir tersimpan dalam *database*. Data GPS seperti Gambar 16 dimasukkan pada layanan Google Maps

untuk membandingkan titik lokasi yang ditunjuk. Pada Gambar 17 dan Gambar 18 tampak posisi *marker* telah berada di area yang sama, maka dapat disimpulkan bahwa sistem telah mampu mendapatkan data GPS berupa titik koordinat (*latitude* dan *longitude*) dari modul GPS/GPRS SIM868 dan menghasilkan tampilan peta beserta *marker* yang menunjukkan lokasi data GPS tersebut.



Gambar 17. Tampilan Peta



Gambar 18. Tampilan Google Maps

KESIMPULAN

Dari hasil perancangan, dan pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Raspberry Pi dapat berkomunikasi dengan GSM/GPRS/GNSS HAT SIM868 menggunakan komunikasi serial sehingga data

GPS berhasil didapatkan dan *device* dapat terhubung dengan server.

2. Hasil pengujian keluaran data GPS dari GSM/GPRS/GNSS HAT SIM868 menghasilkan data yang menyerupai dengan keluaran data GPS dari *handphone* menggunakan aplikasi GPS Coordinates.
3. Data simulasi dari Raspberry Pi berhasil dikirimkan ke *database* dengan layanan GPRS menggunakan GSM/GPRS/GNSS HAT SIM868 dengan provider Telkomsel.
4. Data simulasi yang tersimpan dalam *database* telah dapat diakses dan ditampilkan pada halaman web dalam bentuk tabel waktu diterimanya *output* data simulasi terhadap data tekanan udara masing-masing sensor tekanan gas, data GPS, dan kantong udara virtual.

DAFTAR PUSTAKA

- [1] Harison., Syarif, A., “Sistem Informasi Geografis Sarana Pada Kabupaten Pasaman Barat”, Jurnal TEKNOIF, Vol. 4, No.2 Oktober 2016
- [2] Lavarino, Dio., Yustanti, Wiyli., “Rancang Bangun E – Voting Berbasis Website di Universitas Negeri Surabaya”, Jurnal Manajemen Informatika, Vol. 6, No. 1 tahun 2016
- [3] Nixon, Robin. May 2018. “Learning PHP, MySQL, and Javascript, 5th Edition” United States of America:O’Reilly Media, Inc.
- [4] *Raspberry Pie 4 Model B* (Oktober 2019),

- <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [5] Sarwono, W.C., "Monitor Tekanan dan Kebocoran Udara pada Matras Pereduksi Dekubitus Berbasis Raspberry Pi".
- [6] SIMCOM. SIMCOM.EE smart machines, smart decision (November 2019).
<https://simcom.ee/modules/gsm-gprs-gnss/sim868/>
- [7] Sukaridhoto, Sritrusta. 2014. "Buku Jaringan Komputer I" (Desember 2019),
<http://dhoto.lecturer.pens.ac.id/publications/book/2014/Dhoto-JaringanKomputer1.pdf>
- [8] Susilo, Y.S., Pranjoto, H., "Sistem Pelacakan dan Pengamanan Kendaraan Berbasis GPS dengan Menggunakan Komunikasi GPRS", *Jurnal Ilmiah Widya Teknik*, Vol. 13, No. 1 tahun 2014