

Smart Trash Untuk Membantu Petugas Kebersihan Menggunakan Arduino

Riesky Akbar M. S.
Program Studi Informatika
Fakultas Teknologi Industri
Universitas Kristen Petra
Jl. Siwalankerto 121-131
(031)2983455, Surabaya 60236
smkp11rizky@gmail.com

Djoni Haryadi Setiabudi
Program Studi Informatika
Fakultas Teknologi Industri
Universitas Kristen Petra
Jl. Siwalankerto 121-131
(031)2983455, Surabaya 60236
djonih@petra.ac.id

Handry Khoswanto
Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Kristen Petra
Jl. Siwalankerto 121-131
(031)2983455, Surabaya 60236
handry@petra.ac.id

ABSTRAK

Seringkali dijumpai tempat-tempat sampah di tempat umum sudah penuh bahkan melebihi kapasitas atau *overflow* yang menyebabkan kondisi yang tidak higienis serta membuat tempat tersebut tidak enak dipandang. Tempat sampah yang sudah penuh dibiarkan menumpuk dan menunggu sampai diambil kembali oleh petugas kebersihan. Untuk menghindari situasi tersebut, dibuat smart trash untuk membantu petugas kebersihan dalam memantau muatan tempat sampah. Navghane [5] telah membuat smart trash yang dapat memantau muatan tempat sampah melalui web browser. Maka pada penelitian ini akan dibuat tempat sampah dengan push notification melalui aplikasi pada smartphone sehingga lebih memudahkan petugas mengambil tempat sampah secara tepat waktu.

Tempat sampah dipasang sistem Arduino WeMos D1 R1 dan sensor proximity E18-D80NK yang akan menunjukkan status sampah pada level saat ini. Data yang didapat dari sensor akan dikirim dengan konektivitas Wi-Fi yang terdapat pada mikrokontroler menuju database. Komunikasi tersebut akan memicu pertukaran data secara real-time antara aplikasi pada smartphone pengguna, dengan perangkat smart trash.

Hasil yang didapatkan dari penelitian ini adalah perangkat dapat mendeteksi dan menampilkan status ketinggian sampah dalam tempat sampah secara real-time dalam bentuk persentase dan gambar 2D pada aplikasi smartphone, serta push notification yang akan muncul pada saat ketinggian sampah sudah penuh atau mencapai batas atas. Sensor proximity E18-D80NK rentan membaca sampah kering yang bersifat transparan dan tipis, namun akurat membaca sampah kering yang bersifat solid dan tidak tembus pandang.

Kata Kunci: *Smart Trash, Arduino, Aplikasi Smartphone, Pemantauan Muatan Tempat Sampah*

ABSTRACT

Many times, the garbage bins at public places are overloaded even exceeding capacity or overflow which creates unhygienic conditions for people as well as ugliness to that place and leaving bad smell. The full garbage bins are left to pile up and wait until they are taken back by the janitor. To avoid that kind of situation, a smart trash is made for assisting the janitor to monitoring the level content of the garbage bins. Navghane [5] has created a smart trash that can monitoring level of garbage by the web browser. This thesis will create a smart trash system with push

notification by a smartphone application that will be easier for janitors pick up the garbage bin just in time.

The garbage bin is installed with Arduino WeMos D1 R1 system and proximity sensor E18-D80NK which will show the status of the garbage at the current level. Data obtained from the sensor will be sent with a microcontroller that connected with Wi-Fi to the database. That will be data exchange in real-time between an application on the user's smartphone with the smart trash device.

The result of this thesis is the device can detect and show the garbage level in real-time with percentage and 2D images on the smartphone application, and provide a push notification that appears when the garbage level has full or reached the upper limit. Proximity sensor E18-D80NK is prone to detect dry trash, which is transparent and thin, but can detect dry trash which are solid and not transparent..

Keywords: *Smart Trash, Arduino, Smartphone Application, Monitoring Level of The Garbage Bins*

1. PENDAHULUAN

Seringkali dijumpai tempat-tempat sampah di tempat umum sudah penuh bahkan melebihi kapasitas atau *overflow* yang menyebabkan kondisi yang tidak higienis serta membuat tempat tersebut menjadi tidak enak dipandang. Penyebab utamanya adalah pengambilan sampah yang tidak teratur. Selama ini, pengangkutan sampah dilakkan 2-3 hari per minggu per TPS. Tempat sampah yang sudah penuh harus menunggu sampai diambil kembali oleh petugas kebersihan sehingga dibiarkan menumpuk. Ketersediaan Wi-Fi di tempat umum dapat dimanfaatkan dengan memasang perangkat alat pada tempat sampah untuk membantu petugas kebersihan memantau muatan tempat sampah.

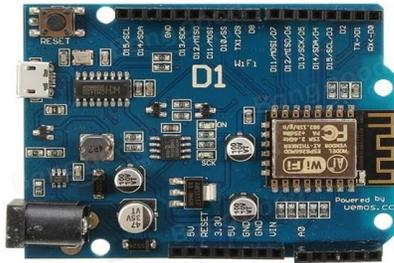
Untuk mengatasi *problem* di atas, maka penelitian ini akan dibuat *Smart Trash* Untuk Membantu Petugas Kebersihan Menggunakan Arduino dengan cara mengirim notifikasi pada aplikasi *Android* serta mengubah sensor IR (TSOP 1738) dengan menggunakan sensor proximity E18-D80NK pada *smart trash*.

2. TINJAUAN PUSTAKA

2.1 Arduino

Menurut Syahwil *Arduino* adalah kit elektronik atau papan rangkaian elektronik *opensource* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel [9]. *Arduino* yang digunakan pada

penelitian menggunakan modul Wi-Fi bernama WeMos D1 R1 ESP-8266.



Gambar 1. WeMos D1 R1 ESP-8266

Gambar 1 adalah gambar penampakan fisik dari WeMos D1 R1. Menurut Widiyaman WeMos D1 R1 ESP-8266 merupakan modul Wi-Fi yang berfungsi sebagai perangkat tambahan seperti membuat koneksi TCP/IP [10]. Kemampuan mikrokontroler WeMos D1 R1 bisa dioperasikan menggunakan kabel dan harganya lebih terjangkau jika dibanding dengan mikrokontroler Arduino yang lain.

		D15	SCL	GPIO6
		D14	SDA	GPIO4
		GND		
		D13	SCK	GPIO14
		D12	MISO	GPIO12
		D11	MOSI	GPIO13 Rx0*
		D10	SS	GPIO15 Tx0*
5V		D9	TX1	GPIO2 10K Pull up Built in led
RST		D8		GPIO0 10K Pull up
3V3				
5V				
GND				
VIN				
		D7	MOSI	GPIO13
		D6	MISO	GPIO12
		D5	SCK	GPIO14
		D4	SDA	GPIO4
		D3	SLC	GPIO6
		D2		GPIO16
		D1	Tx	GPIO1
		D0	Rx	GPIO3

Gambar 2. Datasheet WeMos D1 R1

Gambar 2 adalah *datasheet* dari WeMos D1 R1. GPIO digunakan dalam menentukan pin dari papan mikrokontroler yang diinisialisasikan di Arduino IDE. Contoh, pada papan mikrokontroler WeMos D1 R1, GPIO dari pin “D9” adalah GPIO 2, artinya saat menginisialisasikan pin pada aplikasi Arduino IDE, harus menggunakan pin “D2” bukan pin “D9”, karena pin “D9” pada WeMos D1 R1 merupakan pin “D2” pada aplikasi Arduino IDE. GPIO dapat dicek dengan cara melihat langsung di papan mikrokontroler atau dengan membaca pinout dengan mencari *datasheet*-nya (Alfalah, n.d.). [1]

2.2 Proximity E18-D80NK



Gambar 3. Sensor Proximity E18-D80NK

Gambar 3 penampakan fisik sensor *proximity* tipe E18-D80NK. Menurut Noval [6], sensor *proximity* E18-D80NK mampu mendeteksi suatu objek dengan jangkauan tertentu, apabila sensor mendeteksi keberadaan suatu objek maka output rangkaian sensor akan berlogika “1” atau “HIGH”. Begitu juga sebaliknya, ketika

sensor tidak mendeteksi suatu objek maka output rangkaian sensor akan berlogika “0” atau “LOW”. Sensor ini dapat beroperasi dengan tegangan sebesar 5 volt. Arus maksimal (I_{max}) 25 mA. Frekuensi 490 Hz. Mendeteksi jarak 3 cm–80 cm. Secara fisik, *proximity* E18-D80NK memiliki panjang 45 mm (1.77”), diameter 17 mm (0.67”), dan panjang kabel 115 cm (45”).

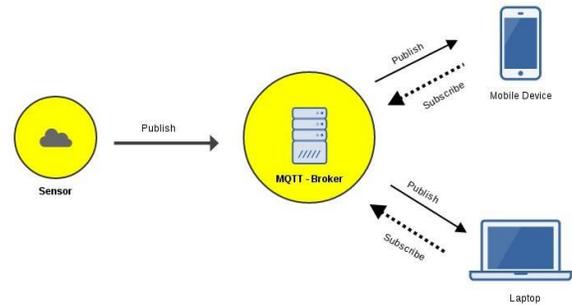
2.3 Library

Library digunakan pada perangkat *smart trash* ini adalah ESP-8266 Wi-Fi dan *pubsubclient.h*. *Library* Wi-Fi untuk ESP8266 telah dikembangkan berdasarkan ESP-8266 SDK, menggunakan konvensi penamaan dan filosofi fungsi keseluruhan dari *library* Wi-Fi Arduino [3]. *Library* *pubsubclient* menyediakan klien untuk melakukan *publish* berlangganan pesan sederhana dengan server yang mendukung MQTT [7]

2.4 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi *Android* berdasarkan IntelliJ IDEA (www.developer.android.com, 2017, para. 1) [2]. *Android Studio* menawarkan banyak fitur yang dapat meningkatkan produktivitas dalam membangun aplikasi *Android*.

2.5 MQTT



Gambar 4. Protokol MQTT [4]

Gambar 4 menurut Saputra et al [8], protokol MQTT (Message Queuing Telemetry Transport) adalah protokol pesan ringan berbasis *publish* atau *subscribe* yang digunakan di atas protokol TCP/IP. Protokol ini mempunyai ukuran paket data low overhead kecil (minimum 2 byte) sehingga dalam penggunaannya hanya konsumsi daya kecil. MQTT bersifat terbuka, simple, dan didesain mudah untuk diimplementasikan, yang mampu menangani ribuan client jarak jauh dengan hanya menggunakan satu server.

2.6 Tinjauan Studi

Navghane [5], membuat perangkat berupa tong sampah yang dapat mendeteksi level pada tong sampah berdasarkan 3 level yaitu 0%, 50%, dan 90% (*full*). Penelitian tersebut menggunakan sensor inframerah TSOP 1738. Data kemudian dikirimkan ke mikrokontroler ARM LPC2148 lalu diteruskan ke halaman HTML pada *web browser* yang dapat diakses melalui *personal computer* atau pada *mobile device*.

3. ANALISIS DAN DESAIN

3.1 Analisis

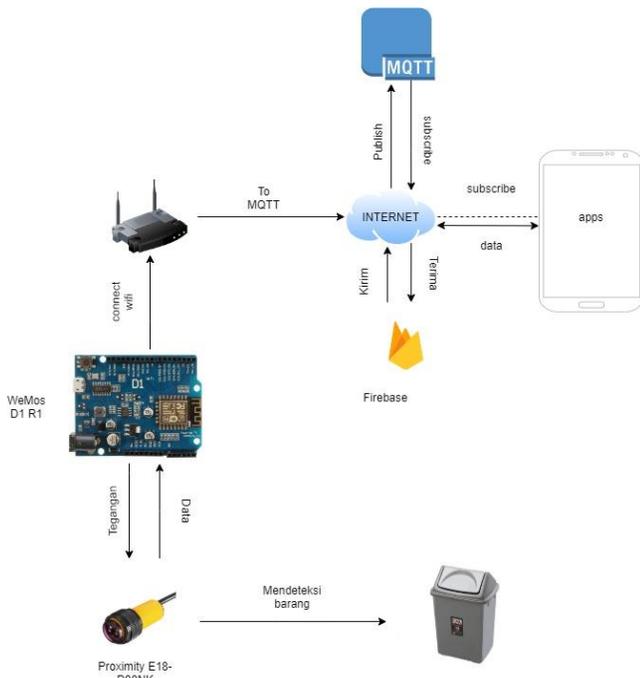
Navghane [5] telah melakukan penelitian untuk mengatasi permasalahan penumpukan sampah pada tempat sampah dengan memonitoring tempat sampah lewat *website*. Namun kekurangan pada penelitian tersebut yakni tidak ada notifikasi untuk

memberitahu petugas kebersihan tepat waktunya. Maka pada penelitian ini akan dibuat *smart trash* dengan sensor *proximity* E18-D80NK dan memonitoring tempat sampah melalui aplikasi yang akan memberi notifikasi apabila tempat sampah telah penuh, sehingga petugas kebersihan dapat segera melakukan pengangkutan sampah. Data yang didapat dari sensor akan dikirim dengan konektivitas Wi-Fi yang terdapat pada mikrokontroler WeMos D1 R11 menuju *database*. Komunikasi tersebut akan menimbulkan pertkaran data secara *real-time* antara aplikasi pada *smartphone* pengguna dengan perangkat *smart trash*.

Sensor *proximity* E18-D80NK bisa mendeteksi benda atau barang yang menghalangi sensor dalam jarak jauh atau dekat dalam artian sensitivitas dari sensor ini lebih baik daripada sensor TSOP 1738 yang telah digunakan sebelumnya. Berbeda dengan sensor TSOP 1738 yang memiliki sensitivitas yang jaraknya cukup panjang dan jaraknya tidak dapat diatur seperti sensor *Proximity* E18-D80NK.

3.2 Desain

Terdapat 2 jenis desain dalam penelitian ini, yakni desain perangkat keras dan desain perangkat lunak dari aplikasi *monitoring*.



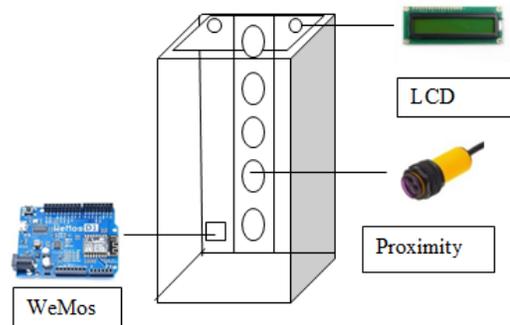
Gambar 5. Desain Arsitektur Sistem

Gambar 5 adalah tentang desain dari arsitektur system. Sensor *proximity* E18-D80NK mendeteksi barang. WeMos memberi tegangan sensor *proximity* E18-D80NK. Sensor *proximity* E18-D80NK bekerja dan mengirimkan data ke WeMos untuk nantinya dikirim ke MQTT melalui koneksi internet. Sensor *proximity* E18-D80NK diprogram untuk memproses benda dalam waktu 10 detik. Benda yang sudah jatuh dan berada didepan sensor atau menghalangi sensor, maka sensor akan membaca benda tersebut. Jika benda tersebut hanya melewati sensor maka program akan mengecek ulang. WeMos akan mengolah dan akan meneruskan sinyal tersebut ke MQTT *Broker*, jika kondisi tertentu terpenuhi maka informasi tersebut dikirimkan melalui koneksi Wi-Fi.

Setelah terkoneksi internet WeMos *publish* data ke MQTT *broker*, MQTT *broker* akan meneruskan sinyal tersebut ke aplikasi pada *smartphone* yang terkoneksi pada internet. *Smartphone* akan *subscribe* data dari MQTT *broker*. Koneksi antara WeMos D1 R1 dan aplikasi, terhubung dalam satu jaringan Wi-Fi yang sama. Menggunakan *database* *Firebase* karena data akan disimpan sebagai json, kemudian *sync* secara *real-time* ke setiap aplikasi *smartphone*. *Firebase database* ini digunakan untuk menyimpan data tempat sampah, yang nantinya bisa dimanfaatkan untuk menyimpan data user dan history tempat sampah.

3.2.1 Desain Perangkat Keras

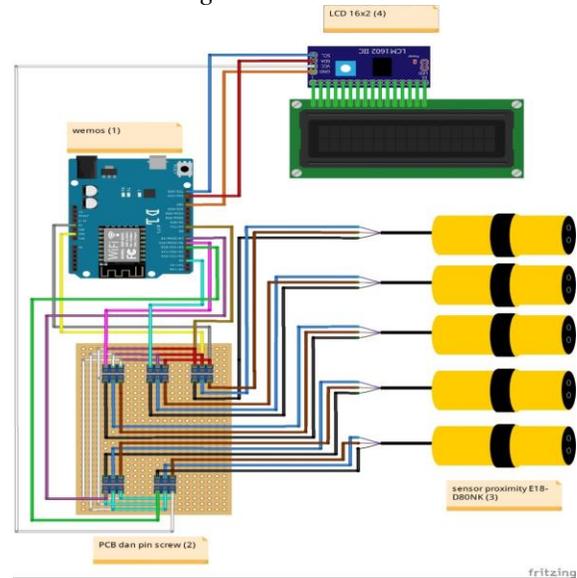
3.2.1.1 Desain Perangkat Smart Trash



Gambar 6. Desain Perangkat

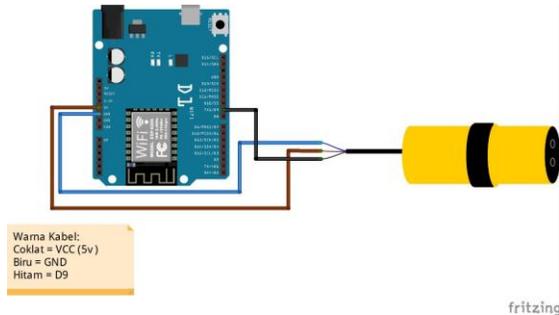
Gambar 6 merupakan penampilan ilustrasi desain perangkat *hardware* bila sudah terpasang semua. Pada sisi belakang tempat sampah terpasang sensor *proximity* E18-D80NK dan disisi atas tempat sampah terpasang LCD. Sensor *proximity* dan *display* disambungkan ke WeMos diberi tegangan. Sensor *proximity* mengecek sampah hingga mencapai ketinggian tempat sampah. Tempat sampah akan memberikan tampilan ketinggian jika sensor membaca benda.

3.2.1.2 Desain Rangkaian Listrik



Gambar 7. Desain Rangkaian Listrik Keseluruhan

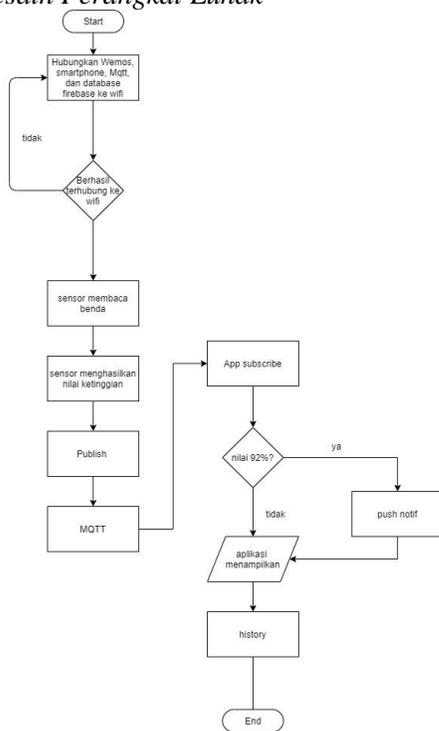
Gambar 7 menampilkan desain rangkaian listrik dari perangkat keras (*hardware*) yang adalah media untuk mendapatkan data dari tempat sampah. WeMos D1 R1 ESP-8266 (1) sebagai mikrokontroler utama yang memiliki modul Wi-Fi, ESP-8266, PCB dan pin screw (2) sebagai penyambung kabel 5V, GND, I/O Digital, serta sensor proximity E18-D80NK (3) sebagai sensor yang dipasang pada tempat sampah untuk mendeteksi sampah, dan LCD ukuran 16x2 (4) untuk menampilkan persentase level ketinggian sampah.



Gambar 8. Desain Rangkaian Listrik

Gambar 8 menampilkan rancangan pemasangan pin pada perangkat. Sensor *proximity* E18-D80NK memiliki beberapa kabel yang tersambung dengan mikrokontroler WeMos D1 R1, antara lain kabel berwarna coklat *Vcc* terhubung dengan pin 5V pada WeMos, kabel berwarna biru GND terhubung dengan pin GND pada WeMos. Kabel dan GND bertindak sebagai *power supply* bagi sensor *proximity* E18-D80NK. Kemudian kabel berwarna hitam *signal* terhubung dengan pin D9 pada WeMos D1 R1. Pin D9 digunakan sebagai jalur *input-output* (I/O) berupa data digital.

3.2.2 Desain Perangkat Lunak



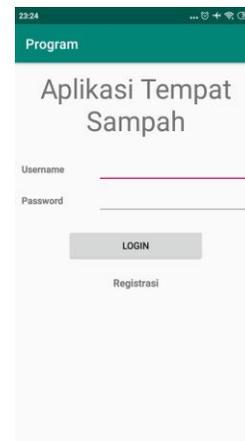
Gambar 9. Flowchart Cara Kerja Sistem

Gambar 9 adalah gambaran garis besar cara kerja sistem pada perangkat yang ditampilkan dalam bentuk *flowchart*. Pertama, dilakukan proses inialisasi, seperti inialisasi pin pada mikrokontroler WeMos, inialisasi variabel program WeMos, inialisasi *library* yang digunakan program, inialisasi seluruh konektivitas yang diperlukan yaitu konektivitas perangkat dengan MQTT dan *database*. Kemudian jika semua sudah terkoneksi, maka proses pengambilan nilai ketinggian dari sensor *proximity* E18-D80NK pada tempat sampah. Nilai ketinggian yang didapat oleh sensor diteruskan WeMos D1 R1 melalui ESP-8266 yang telah terhubung jaringan Wi-Fi untuk menuju MQTT. *Publish* nilai kirim MQTT, *app subscribe* untuk menampilkan nilai, jika nilai ketinggian 92% akan ditampilkan di aplikasi dan *push notif* akan dikirim, jika tidak akan ditampilkan dan dicatat di *history*.

3.3 Desain UI

Desain UI terdiri dari beberapa tampilan, yakni tampilan login, halaman *Register* Akun, dan halaman *Home* pada aplikasi.

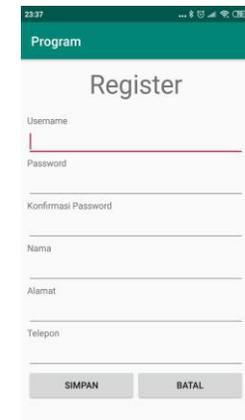
3.3.1 Login



Gambar 10. Tampilan Layout Login

Gambar 10 berisi tampilan *form login* aplikasi. *User* akan diminta untuk *login* ke akun *user* masing-masing jika sudah membuat akun sebelumnya. Jika belum, maka disediakan tombol *register* agar *user* dapat segera membuat akunnya. Setelah *user* berhasil melakukan *login*, maka *user* akan masuk ke *main menu* personal dari *user*, di mana *user* akan memonitoring tempat sampah.

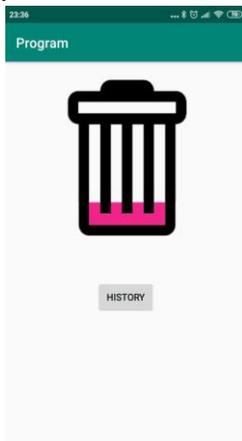
3.3.2 Register



Gambar 11. Tampilan Layout Register

Gambar 11 adalah tampilan ketika *user* baru mendaftar agar mendapat akun. *User* akan diminta untuk mengisi *username*, *password*, konfirmasi *password*, nama, alamat, dan telepon yang akan digunakan dalam aplikasi ini. Setelah menekan tombol simpan, maka *user* akan diarahkan kembali ke menu *login*. Akun baru yang telah didaftarkan akan tersimpan ke dalam *database* untuk digunakan dalam *menu login*.

3.3.3 Main Menu



Gambar 12. Tampilan Main Menu

Gambar 12 adalah tampilan ketika *user* sudah melakukan *login* dan berhasil masuk ke dalam aplikasi. Terdapat tombol pada menu, yaitu *History*. Kemudian tombol “*History*” untuk membuka halaman “*history*” untuk melihat riwayat sampah level ketinggian sampah mulai kosong hingga terisi sampah.

3.4 Desain Database

Berikut adalah tabel-tabel gambaran desain *database* dari aplikasi monitoring tempat sampah yang terdiri dari dua *entity* yaitu *user*, *history*.

Tabel 1. Desain Entity User

User		
Nama Field	Tipe Data	Keterangan
User	String	Tempat menyimpan email
password	String	Tempat menyimpan password
Telepon	String	Tempat menyimpan telephone
Nama	String	Tempat menyimpan nama user
user_uid	String	Tempat menyimpan user id

Entity user pada tabel 1 digunakan untuk menyimpan data *user* untuk *login* dan *sign up* pada aplikasi. Dalam *entity user* terdapat lima buah atribut yaitu “*user*” sebagai penampung *email* untuk *login*. Kemudian “*password*” sebagai penampung *password* untuk *login*, “*telephone*” sebagai penampung nomer telepon user, “*nama*” sebagai nama user, “*user_uid*” yang menampung id yang diberikan secara unik oleh *Firebase*.

Table 2. Desain Entity History

History		
Nama Field	Tipe Data	Keterangan
level	String	Tempat menyimpan id pengisian
tanggal	String	Tempat menyimpan durasi

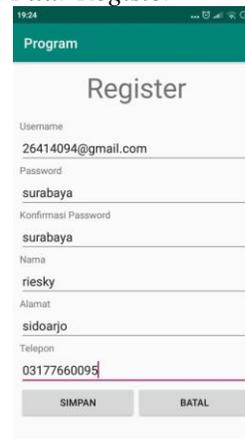
		pengisian
History_id	string	Tempat menyimpan history id

Entity history pada tabel 2 terdiri dari 3 buah atribut yaitu “*level*” yang berisi ketinggian dari tempat sampah, “*tanggal*” tanggal dan jam dari pencatatan ketinggian tempat sampah, dan “*id*” yang berisi unik id setiap *history*.

4. PENGUJIAN SISTEM

4.1 Pengujian Aplikasi

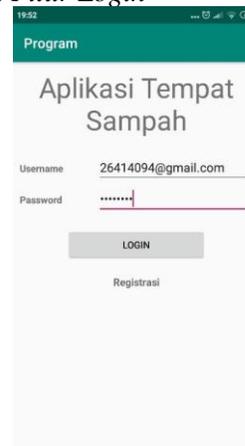
4.1.1 Pengujian Fitur Register



Gambar 13. Input pada Halaman Register

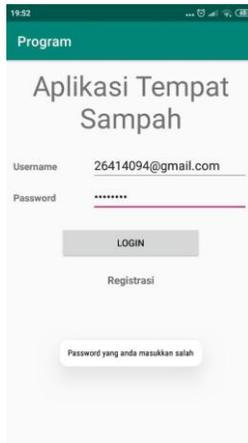
Pada gambar 13 dilakukan uji coba dengan memasukkan *email* “26414094@gmail.com” dan *password* “surabaya”, dan setelah itu tombol “SIMPAN” ditekan untuk menyimpan akun baru di *database*. Jika sudah memiliki akun user bisa menekan tombol “BATAL”.

4.1.2 Pengujian Fitur Login



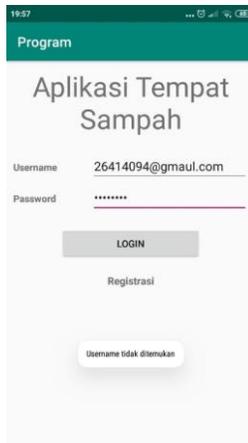
Gambar 14. Halaman Login

Gambar 14, halaman *login* diisi *email* dan *password* yang sudah diregistrasi sebelumnya. Setelah berhasil memasukan *login* dan *password user* dapat masuk ke main menu untuk memonitoring level sampah dengan tampilan 2D dan terdapat menu *history* untuk mengetahui *history* sampah.



Gambar 15. Login Gagal Salah Password

Pada gambar 15 menampilkan akun yang gagal login karena salah input password akun.



Gambar 16. Login Gagal Salah Username

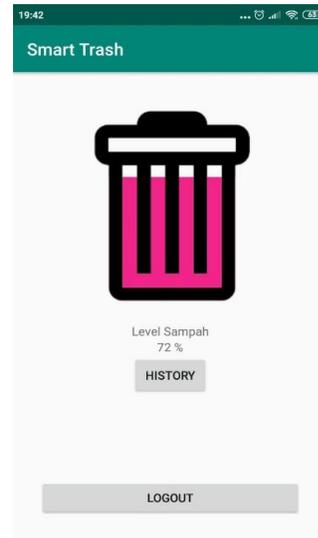
sementara gambar 16 menampilkan akun yang gagal login karena salah input username atau karena akun belum terdaftar.

4.1.3 Pengujian Fitur Ketinggian Sampah Secara Real-Time

Pada fitur ini, user dapat melihat ketinggian sampah di tempat sampah dalam bentuk persentase dan sebuah gambar tempat sampah yang terisi sampah sesuai dengan persentase tersebut.



Gambar 17. Tampilan Sebelum Perubahan



Gambar 18. Tampilan Setelah Perubahan

Gambar 17 adalah tampilan halaman “main menu” sebelum adanya perubahan (pada ketinggian 52%), sedangkan gambar 18 merupakan tampilan sesudah adanya perubahan data (pada ketinggian 72%).

4.1.4 Pengujian Fitur History



Gambar 19. Tampilan Halaman History

Gambar 19 menunjukkan adanya penambahan data history yang didapat dari WeMos setelah ketinggian sampah bertambah. Data yang ditambahkan dalam bentuk objek yang berisi “level” dan “tanggal”.

4.1.5 Pengujian Push Notification

Fitur push notification muncul ketika Firebase mendeteksi ada perubahan data pada database sesuai dengan kriteria yang ditentukan.

Gambar 20 menampilkan push notification yang berhasil keluar dengan keterangan, “Peringatan, Tempat Sampah Penuh”. Notifikasi akan keluar jika tempat sudah mencapai persentase 92%.



Gambar 20. Push Notification

4.2 Pengujian Perangkat Keras

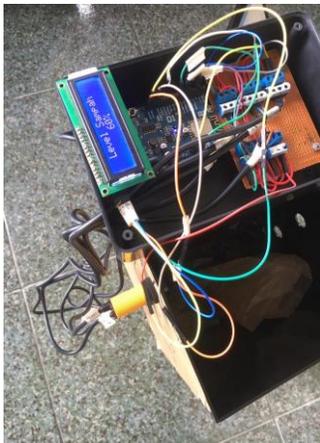


Gambar 21. Foto Keseluruhan Perangkat Keras

Gambar 21 perangkat yang digunakan adalah tempat sampah *acrylic*, sensor *proximity* E18-D80NK, *Arduino* WeMos D1 R1, dan LCD 16x2.

4.2.1 Pengujian Sensor Proximity E18-D80NK

Dari pengujian yang dilakukan, telah mendapatkan hasil bahwa sensor *proximity* E18-D80NK berhasil mengukur level ketinggian sampah yang ada di dalam tempat sampah. Sensor dipasang tegak lurus menghadap bagian dalam tempat sampah. Keadaan perangkat tampak atas dapat dilihat pada gambar 22.



Gambar 22. Foto Perangkat Tampak Atas

4.2 Pengujian Keakuratan Perangkat

4.2.1 Pengujian Keakuratan Tinggi Sampah

Pengujian keakuratan tinggi sampah dilakukan dengan membandingkan tinggi sampah yang terdeteksi dan ditampilkan pada LCD dan aplikasi dengan tinggi sampah secara manual.

Tabel 3. Hasil Simulasi Tinggi Sampah

Data	Tinggi Sampah di Aplikasi (%)	Tinggi Sampah Diukur Manual (max=50cm)	
		Dalam cm	Dalam %
1	12	6	12
2	32	16	32
3	52	26	52
4	72	36	72
5	92	46	92

Jika terdapat objek atau sampah pada level 1 atau batas bawah, tinggi sampah dihitung secara manual adalah 6 cm dengan persentase sebesar 12%. Besaran persentase tersebut sesuai dengan persentase yang ditampilkan pada LCD dan aplikasi.

Pada keterangan ukuran yang tercantum pada tabel 3, batas atas sebesar 46 cm dengan tingkat persentase 92% telah cukup untuk membaca ketinggian sampah pada keadaan penuh, karena apabila sensor dipasang pada ketinggian sama dengan tinggi tempat sampah (yaitu 50 cm) maka tempat sampah menjadi terlalu penuh. Ketika sampah menumpuk dan mencapai sensor paling atas pada ketinggian 46 cm, maka LCD dan aplikasi akan menampilkan persentase 92% dan pemberitahuan bahwa sampah sudah penuh.

4.2.2 Pengujian Keakuratan Jarak Sensor

Pengujian keakuratan jarak sensor dilakukan dengan meletakkan objek pada rentang jarak tertentu yang diukur secara manual dari sensor. Pengujian dilakukan dengan 8 macam sampah kering dengan jarak 0 cm sampai batas lebar tempat sampah, yakni 19,5 cm. Hasil pengujian ditunjukkan sebagai berikut:

Tabel 4. Pengujian Akurasi Jarak Sensor

Objek	Jarak			
	0-5 cm	6-10 cm	11-15 cm	16-19.5 cm
Benda Putih:				
- Plastik Bening	x	x	v	V
- Mika	v	v	x	X
- Botol Mineral Plastik	v	v	v	V
- Kaca	v	v	v	V
Benda Hitam:				
- Plastik Hitam	V	x	x	X
- Kemasan Makanan Ringan	V	v	v	V
- Kertas	V	v	v	V
- Kain Perca	V	v	v	V

Tabel 4 menunjukkan pengujian keakuratan jarak sensor dalam mendeteksi sampah kering. Pada pengujian benda putih, botol mineral plastik dan kaca berhasil dibaca oleh sensor dalam rentang jarak 0-19,5 cm dibandingkan dengan 2 macam sampah lainnya. Hal ini disebabkan karena botol mineral dan plastik

meskipun berwarna transparan, keduanya bertekstur lebih padat dan solid, sehingga sensor lebih akurat membaca adanya benda. Sementara plastik bening tidak dapat dibaca pada jarak 0-10 cm karena adanya kemungkinan bahwa cahaya sensor menembus medium, sementara pada jarak 11-19,5 cm cahaya sensor terpantul oleh plastik bening, sehingga sensor menganggap ada benda. Sebaliknya, untuk sampah mika dapat terbaca pada radius 0-10 cm karena tekstur mika yang lebih tebal dari plastik bening sehingga sensor tidak tembus. Namun, mika tidak dapat terbaca pada jarak 11-19,5 cm karena sensor tidak sampai dan sifat mika yang tidak bisa memantulkan cahaya sensor seperti plastik bening.

Pada pengujian benda hitam, hanya plastik hitam yang hanya terbaca maksimal pada jarak 5 cm. Hal tersebut disebabkan karena plastik hitam, meskipun warnanya hitam, namun ketika disinari cahaya masih bisa tembus karena mediumnya yang tipis.

Dari uraian di atas dapat disimpulkan bahwa pengujian akurasi sensor dalam membaca objek pada jarak tertentu bergantung pada spesifikasi objek itu sendiri. Cara kerja sensor membaca objek adalah melalui cahaya, maka untuk objek yang tipis, rentan, dan tembus cahaya akan sulit terbaca. Namun, tidak semua objek transparan tidak dapat dibaca oleh sensor. Objek transparan yang bersifat lebih tebal, padat, dan solid seperti botol plastik dan kaca, berhasil dibaca sensor seperti yang ditunjukkan pada tabel 4.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian sistem, dapat disimpulkan beberapa hal berikut:

1. Perangkat sistem monitoring tinggi sampah dapat melakukan pengiriman data melalui jaringan Wi-Fi, ke *Firestore real-time database*.
2. Aplikasi dapat memonitor sampah secara *real-time* dan menampilkan hasilnya berupa persentase dan gambar.
3. Aplikasi dapat memberikan notifikasi berupa *Push Notification* yang muncul pada saat ketinggian sampah sudah mencapai batas atas.
4. Perangkat berhasil melakukan monitoring ketinggian sampah dengan presentase sebesar 80%, yang mana objek yang bersifat transparan dan tipis tidak dapat dibaca oleh sensor, sementara objek yang bersifat tidak transparan, solid, tebal, dan padat dapat dibaca sensor secara akurat.
5. Petugas harus setting ulang untuk pergantian ssid dan *password* untuk menyambung ke Wi-Fi.

5.2 Saran

Berikut ini merupakan beberapa hal yang dapat dijadikan saran untuk pengembangan aplikasi:

1. Aplikasi perlu dicoba pada OS *Android* terbaru, bilamana terjadi *error*.
2. Pengembangan aplikasi dengan mengubah *script* pemrograman yang dapat meningkatkan ketepatan hasil *monitoring*.
3. Penggunaan gambar 3D untuk menunjukkan ketinggian sampah bisa menjadi lebih menarik daripada gambar 2D.

4. Penambahan sensor yang digunakan untuk mengukur ketinggian tempat sampah.
5. Penggunaan aplikasi sebaiknya dilakukan dengan koneksi internet yang stabil.
6. Menambahkan sensor servo untuk buka tutup otomatis pada tempat sampah.
7. Untuk menghindari kehilangan alat pada tempat sampah diberikan sensor GPS agar lebih aman dan petugas juga bisa mengecek tempat sampah mana yang sudah penuh.
8. Peneliti selanjutnya perlu menambahkan sensor untuk sampah basah.
9. Peneliti selanjutnya dapat memberi CCTV atau *webcam* agar tempat sampah aman dan menghindari kehilangan tempat sampah.

6. DAFTAR PUSTAKA

- [1] Alfalah, Vidy. n.d. Cara Menggunakan WeMos D1 R1/WeMos D1 Mini/Node MCU. Retrieved July 1, 2020, from: <https://www.instructables.com/id/Cara-Menggunakan-WeMos-D1-R1-WeMos-D1-Mini-NodeMCU/>
- [2] Android Studio. 2017. Meet Android Studio. Retrieved January 13, 2020, from: <https://developer.android.com/studio/intro>
- [3] ESP8266 Arduino Core. 2017. WiFi (ESP8266WiFi library). Retrieved December 10, 2019, from: <https://arduino-esp8266.readthedocs.io/en/latest/libraries.html>
- [4] Medium. 2015. Mengenal MQTT – Pemrograman – Medium. Retrieved January 13, 2020, from <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>
- [5] Navghane, S. S., Killedar, M. S., & Rohokale, V. M. 2016. IoT Based Smart Garbage And Waste Collection Bin. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 5(5), 1576-1578.
- [6] Noval, Faisal Ahmad. 2018. Rancang Bangun Sistem Kendali Kelengasan Tanah, Suhu Lingkungan, dan Perangkat Hama untuk Budidaya Tanaman Cabai Merah Keriting (*Capsicum annum L.*) Berbasis Mikrokontroler. Skripsi. Universitas Lampung
- [7] Randomnerdtutorials.com. 2015. ESP32 MQTT-Publish and Subscribe with Arduino IDE. Retrieved Mei 20, 2020, from: <https://randomnerdtutorials.com/esp32-mqtt-publish-subscribe-arduino-ide/>
- [8] Saputra, G. Y. dkk. 2017. Penerapan Protokol Mqtt Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya). *Jurnal Infomatika Mulawarman* Vol.12, No.2 September 2017.
- [9] Syahwil, M. 2013. Panduan Mudah Simulasi dan Praktek Mikrokontroler Arduino. *Yogyakarta: Andi*.
- [10] Widiyaman, Tresna. 2016. Pengertian Modul Wi-fi ESP8266. Retrieved December 10, 2019, from: <http://www.warriornux.com/pengertian-modul-Wi-fi-esp8266/>