

# Sistem Monitoring Penggunaan Air di Apartemen/Mal MultiTenant dengan NodeMCU dan Raspberry Pi

Albert Ong  
Program Studi Informatika,  
Universitas Kristen Petra  
Jl.Siwalankerto 121-131  
Surabaya 60236, Indonesia  
Telp. (031) - 2983455  
Albertong27@Gmail.com

Andreas Handojo  
Program Studi Informatika,  
Universitas Kristen Petra  
Jl.Siwalankerto 121-131  
Surabaya 60236, Indonesia  
Telp. (031) - 2983455  
handojo@petra.ac.id

Resmana Lim  
Program Studi Teknik Elektro,  
Universitas Kristen Petra  
Jl.Siwalankerto 121-131  
Surabaya 60236, Indonesia  
Telp. (031) - 2983455  
resmana@petra.ac.id

## ABSTRAK

Selanjutnya dengan hasil survei yang dilakukan ditemukan bahwa sulitnya melakukan *monitoring* penggunaan air dan melakukan manajemen penggunaan air. Penggunaan air menjadi tidak teratur dan hasil survei yang dilakukan ditemukan bahwa responden setuju dengan adanya sistem *monitoring* dan manajemen penggunaan air berbasis *web*. Pada penelitian ini akan dilakukan uji coba untuk membantu menyelesaikan masalah tersebut dengan cara memodernisasi meter air atau mengimplementasikan teknologi *Internet of Things* pada setiap meter air. Kemudian untuk sistem pembayaran menggunakan sistem *token*. Implementasi IoT pada meter air adalah pada segi pengukuran penggunaan air dan pembatasan penggunaan air. Hasil pembuatan sistem yang dilakukan didapat sistem yang dibuat mampu melakukan *monitoring* penggunaan air, membatasi penggunaan air, dan melakukan penambahan *limit* penggunaan air menggunakan *token*.

**Kata Kunci:** Meter Air, *Multitenant*, *Multibuilding*, Raspberry Pi, ESP8266

## ABSTRACT

As survey results conducted, it was found that it was difficult to monitor water consumption and to manage water consumption. Water consumption become very unregulated and the survey conducted found that respondents agreed with the existence of a web-based water consumption monitoring and management system. In this study, a trial will be conducted to help solve this problem by modernizing water meters or implementing Internet of Things technology on each water meter. Then for payment system will be using a token system. The implementation of IoT in water meters is in terms of measuring water consumption and limiting the water consumption. The results of the system found that the system was capable of monitoring water consumption, limiting water consumption, and increase water consumption limit using token.

**Keywords:** Water Meter, *Multitenant*, *Multibuilding*, Raspberry Pi, ESP8266

## 1. PENDAHULUAN

Berdasarkan survei yang dilakukan masalah yang dihadapi oleh banyak pihak adalah sulitnya untuk melakukan monitoring penggunaan air, hal ini banyak terjadi pada *tenant-tenant* di mall dan apartemen. Kesulitan yang ditemui oleh para responden antara lain, lokasi meter air yang jauh dari lokasi responden, lokasi meter air terletak di area yang sulit untuk dijangkau oleh para responden, dan adanya meter air yang rusak.

Dari survei yang dilakukan juga ditemukan banyak responden yang masih menggunakan sistem pembayaran pascabayar (bayar setelah pakai). Dari survei ini juga didapatkan bahwa keseluruhan responden mendukung adanya sistem prabayar. Keseluruhan responden juga mendukung adanya sistem *monitoring* dan manajemen air menggunakan *web*.

Dari sistem yang saat ini masih dipakai di Indonesia ditemukan *monitoring* atau pencatatan penggunaan air masih dilakukan secara *manual* atau melihat meter air satu per satu, hal ini sangat rentan terjadinya kesalahan pencatatan yang disebabkan oleh *human error* dan hasil pencatatan penggunaan air tidak dapat dilihat secara *real time*.

Skripsi ini dibuat dengan tujuan untuk memudahkan pengguna dalam melakukan *monitoring* penggunaan air dan manajemen penggunaan air dengan fitur-fitur sebagai berikut, menggunakan *web interface* sebagai *media* untuk melakukan *monitoring* dan mengatur penggunaan air, menggunakan sistem prabayar untuk menggantikan metode pembayaran pascabayar di banyak *tenant* (*multi-tenant*), dapat digunakan di banyak gedung (*multi-building*) pada satu *server*, tiap *tenant* dapat memiliki biaya penggunaan air yang berbeda-beda biaya dapat ditentukan oleh pemilik bangunan, *administrator* dapat menentukan fitur apa saja yang dapat diakses oleh *user* tertentu (*customizable role*).

## 2. TINJAUAN PUSTAKA

### 2.1 Tinjauan Studi

Menurut Masruchi, Vecky, dan Repi, dimana dalam penelitian tersebut menggunakan *flowmeter*, dan aplikasi sistem SMS (*Short Message Service*) untuk memberi laporan hasil kepada *user*. Dalam uji cobanya, *Flowmeter* yang digunakan sebenarnya terdiri dari tutup katup plastik, *rotor* air, dan sensor *Hall Effect*. Pengukuran debit air dapat dilakukan ketika *flowmeter* dilalui arus air didalamnya, sehingga membuat *rotor* air berputar dan putaran *rotor* air ini dideteksi oleh sensor *Hall Effect* dengan menghitung jumlah putaran *rotor*. Semakin banyak putaran, maka semakin besar pula jumlah debit air. Kemudian, hasil pengukuran ditampilkan ke *user* melalui sistem SMS pada telepon genggam. Hasilnya, penggunaan *flowmeter* yang memanfaatkan sensor *Hall Effect* menghasilkan nilai akurasi (*flowrate*) sebesar 88,9%. Artinya, penggunaan sensor *Hall Effect* menghasilkan akurasi data yang tinggi. [1] Sedangkan pemanfaatan sistem SMS sebagai pusat informasi akan lebih baik jika diganti dengan sebuah *Web Interface* yang dapat menampilkan dan menkonfigurasi setiap titik/*end-point*. Penggunaan sensor *hall effect* dapat mengurangi akurasi pembacaan sensor apabila sensor menerima intervensi magnet.

Pada penelitian Paskah Abraham, beliau menggunakan WeMos D1 R2 sebagai *device* yang terhubung ke sensor TCRT5000, *device* ini terhubung langsung ke *server* dan menggunakan *media* wifi/WLAN sebagai koneksi jaringannya, beberapa fitur masih membutuhkan *user* untuk melakukan perubahan *script*(code). [2]. *Direct connection* dari *device* ke *server* akan lebih baik jika melalui *gateway* dan *Single user application* akan lebih baik jika dikembangkan menjadi *multi user/tenant* dan *multi building*.

Pada penelitian D.Anandhavalli dkk, menggunakan NodeMCU dan *flow* sensor, mereka juga menggunakan *serial interface* sebagai *output* dari hasil pembacaan sensor [3]. *Serial interface* dapat diganti dengan mengirimkan *request data* ke API *server* dan menggunakan *web interface* untuk menampilkan hasil pembacaan sensor.

## 2.2 TCRT5000

TCRT5000 merupakan sensor reflektif yang menggunakan pemancar *infra red* dan *photo-transistor*. Signal yang diterima adalah refleksi dari pemancar *infra red*. *Output* dari sensor ini dapat berupa signal *digital* dan *analog*. Pada skripsi ini sensor ini berfungsi untuk mendeteksi putaran jarum pada meter air. [4]

## 2.3 ESP8266

ESP8266 merupakan SoC (*System on Chip*) yang diproduksi oleh Espressif Systems. ESP8266 dapat digunakan sebagai *external* wifi *module*, menggunakan *standard AT Command* yang terhubung menggunakan signal UART. GPIO pada ESP8266 memungkinkan SoC ini untuk menggunakan *interface* seperti *Analog and Digital IO*, plus PWM, SPI, I2C. [5]

## 2.4 18650 Shield

18650 *Shield* adalah sebuah *battery module* yang berfungsi sebagai *usb power supply*, 18650 *Shield* memiliki fitur: LED *indicator*, *battery overcharge protection* dan *deep discharge*. [6]

## 2.5 Solenoid Valve

*Solenoid Valve* adalah katup yang digerakan oleh energi listrik melalui *solenoid/solenoida*. Pada bagian inti *solenoid* terdapat kumparan sebagai penggerak yang berfungsi untuk membuka/menutup katup. Prinsip kerja dari *solenoid* adalah ketika kumparan tersebut dialiri tegangan listrik maka pin katup akan tertarik karena adanya gaya magnet sehingga katup tertutup/terbuka. [7]

## 2.6 Itron Cyble Multimag Water Meter

Itron Cyble Multimag *Water Meter* adalah sebuah meter air yang digunakan untuk mengukur besaran debit air, perangkat ini memiliki fitur *Anti-Fraud 3G System* yang terdiri dari cincin merah yang ditempatkan di bawah penutup meter sehingga unit tidak dapat dibuka tanpa merusak cincin merah tersebut dan perangkat ini juga memiliki fitur *Magnetic Shield* sehingga alat ini aman dari *magnet tampering*. [8]

## 2.7 Flask

Flask adalah WSGI *framework* aplikasi *web* ringan, Flask di desain untuk mempercepat dan mempermudah proses memulai.

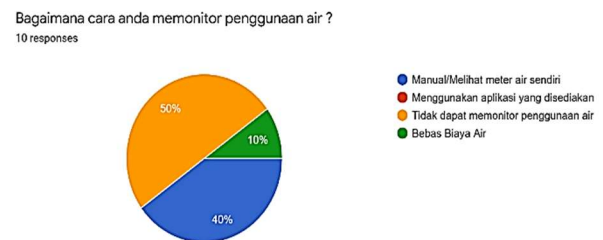
Flask memiliki kemampuan untuk *scale up* ke aplikasi yang lebih kompleks. [9]

## 3. ANALISA DAN DESAIN SISTEM

### 3.1. Analisa Sistem

#### 3.1.1 Sistem Yang Sering Dipakai Saat Ini

Sistem *monitoring* penggunaan air yang sering dipakai *tenant-tenant mall*, apartemen, dan *food court* di Indonesia saat ini adalah *monitoring* secara *manual*. Dari hasil *survey* yang dilakukan sebesar 50% dari responden tidak dapat me *monitor* penggunaan air dan sebesar 40% dari responden masih melakukan *monitoring* secara *manual*. Pada gambar 1 dapat dilihat hasil *survey*.



Gambar 1. Hasil Survey

Untuk *survey* kesulitan yang mereka alami dalam melakukan *monitoring* di dapat 10% dari responden meter air sudah rusak, 10% dari *responden* memberikan informasi jika lokasi meter air jauh dan 70% dari *responden* memberikan informasi jika lokasi meter air berada di area yang sulit mereka jangkau. Pada gambar 2 dapat dilihat hasil *survey* kesulitan.



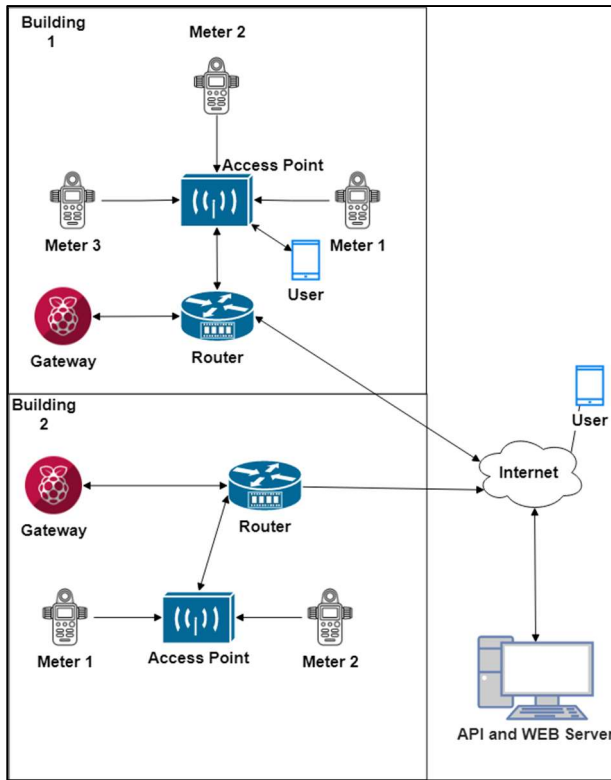
Gambar 2. Grafik Hasil Survey Kesulitan yang Dialami

#### 3.1.2 Sistem Yang Dibutuhkan

Berdasarkan sistem yang sering dipakai saat ini, untuk melakukan *monitoring* air dibutuhkan suatu sistem yang dapat memudahkan pemilik *tenant* dan pemilik bangunan dalam melakukan *monitoring* penggunaan air dan me *manage* penggunaan air. Untuk sistem pembayaran, dibutuhkan sistem *token* yang berguna untuk menambah batas penggunaan air.

## 3.2 Desain Sistem

### 3.2.1 Arsitektur Sistem



Gambar 3. Arsitektur Sistem

Gambar 3 adalah desain arsitektur dari meter air. Pada arsitektur sistem ini terdapat beberapa komponen antara lain:

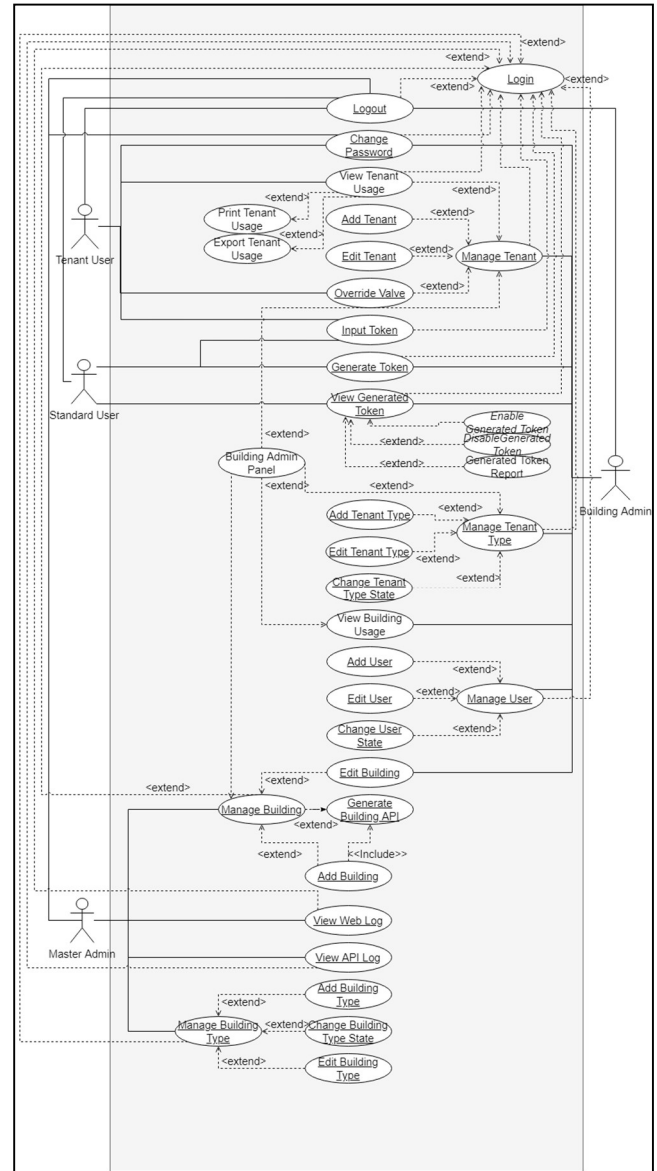
- User
- Meter air. Masing-masing Meter air terdapat:
  - Wemos D1 Mini
  - Wemos Relay Module
  - Battery Module
  - Power Supply
  - Itron Water Meter
  - Solenoid Valve
- Access Point
- Router
- Raspberry Pi yang berfungsi sebagai gateway
- API dan Web Server

Komunikasi dari meter air ke Raspberry Pi gateway menggunakan HTTP API dan dari Raspberry Pi gateway ke API server menggunakan Flask API.

### 3.2.2 Usecase Diagram

Bagian ini akan menjelaskan mengenai actor yang ada dalam sistem, relasi antar actor, serta hal apa saja yang dapat dilakukan oleh actor.

Pada Gambar 4 dapat dilihat Usecase Diagram.



Gambar 4. Usecase Diagram

## 4. IMPLEMENTASI SISTEM

### 4.1 Implementasi Sistem Deteksi Putaran Meter Air Pada ESP8266

Implementasi sistem ini akan mendeteksi putaran meter air, deteksi putaran meter air dilakukan menggunakan sensor TCRT5000 yang dihubungkan pada ESP8266. Sensor TCRT5000 dipasang mengarah ke jarum yang terdapat bagian metal reflective, setiap putaran jarum tersebut akan menandakan penggunaan air sudah mencapai  $0.01M^3$ , jika sensor tersebut mendeteksi adanya putaran pada meter air, sensor akan mengirimkan sinyal digital ke ESP8266 kemudian diteruskan ke gateway untuk dikirimkan ke API server. Pada sistem deteksi

putaran meter air terdapat sistem *buffer* yang berguna menyimpan jumlah putaram meter air saat ESP8266 terputus dari jaringan *Wifi*.

## 4.2 Implementasi Sistem Routine Task Pada ESP8266

*Routine task* merupakan *task* yang dijalankan secara berkala, *routine task* terdiri dari pengecekan *status valve* dari *gateway* dan pengiriman *status valve* saat ini ke *gateway*. *Routine task* di jalankan setiap 30 detik atau setiap 300 *loop cycle*.

Pengecekan *status valve* berfungsi untuk mematikan aliran air dengan cara memutuskan aliran listrik ke *solenoid valve* apabila penggunaan air sudah mencapai batas *limit* atau *user* melakukan *override valve*. Pengiriman *status valve* saat ini berfungsi untuk memberikan informasi terbaru pada *status valve* saat ini.

## 4.3 Implementasi Sistem Generate Token Pada Web Server

Sistem *generate token* berfungsi menghasilkan 10-digit string yang dihasilkan dari hasil *hashing md5* dari *random value*, sistem *generate token* akan melakukan *generate token* hingga tidak ditemukan adanya *token* yang sama.

## 4.4 Implementasi Sistem Input Token Pada Web Server

Sistem *input token* berfungsi untuk menambahkan *limit* penggunaan pada salah satu *tenant*, Pertama sistem melakukan pengecekan *token* dalam *database*, jika *token* dengan id *tenant* dan id *building* yang sesuai maka dilanjut pada proses pengambilan *limit* penggunaan saat ini untuk *tenant* yang akan di tambahkan *limit*, kemudian data *limit* di *update* dengan *limit* ditambah dengan *value* dari *token* tersebut.

## 4.5 Implementasi Sistem Meter Rotate Pada API Server

Sistem *meter rotate* berfungsi menambahkan *usage tenant* saat ini. *Method* yang digunakan untuk sistem ini adalah *POST*, *data* yang dibutuhkan oleh sistem *meter rotate* antara lain kode API, ID *building*, dan ID *tenant*.

## 4.6 Implementasi Sistem Get Relay Status Pada API Server

Sistem *get relay status* berfungsi untuk mendapatkan *state relay* saat ini, sistem ini menggunakan *method GET*, *data* yang dibutuhkan antara lain kode API, ID *building*, dan ID *tenant*. Sistem *get relay status* memberikan *output*: “0” jika *usage* sudah mencapai *limit* atau *override* status dalam posisi *off* dan *output* “1” jika *usage* belum mencapai *limit* atau *override status* dalam posisi *on*.

## 4.7 Implementasi Sistem Update Relay State Pada API Server

Sistem *update relay state* berfungsi untuk men-*set status relay* saat ini, sistem ini menggunakan *method POST*, *data* yang

dibutuhkan antara lain kode API, ID *building*, ID *tenant*, dan *state* saat ini. *State* “0” menandakan *relay* sedang tidak aktif dan *state* “1” menandakan *relay* sedang aktif.

## 5. PENGUJIAN SISTEM

Pada Gambar 5 dapat dilihat *form add building*, *form add building* berfungsi untuk menambahkan *building* baru, setelah *user* membahkan *building* baru *user* akan mendapatkan kode API untuk *building* baru Gambar 6 merupakan contoh *output* dari kode API.

Gambar 5. Halaman Add Building

Gambar 6. Kode API

Pada Gambar 7 dapat dilihat tampilan dari *building management*.

ID	Name	Address	Type
17	Pakuwon Apartment	Surabaya, Indonesia	Apartment
16	Pakuwon Mall	Surabaya, Indonesia	Mall

Gambar 7. Tampilan Building Management

Kemudian *tenant* baru akan ditambahkan pada salah satu *building*, setelah *user* menambahkan *tenant* baru *user* akan mendapatkan ID untuk *tenant* baru, ID tersebut berfungsi untuk di masukkan ke dalam *variable* di ESP8266. Kemudian pada Gambar 8 dapat dilihat tampilan dari *tenant management* salah satu *building*.

ID	Name	Usages	Limits	Valve status	Valve override
16-63590	Tenant A2	1.65	416.00	On	Auto
16-5E270	Tenant A1	1.48	295.00	On	Auto

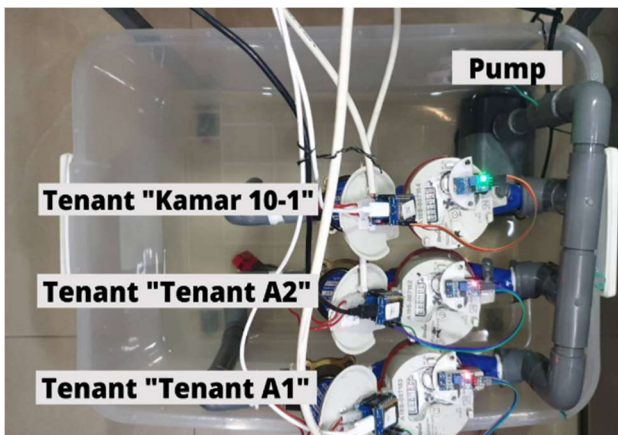
Gambar 8. Tampilan *Tenant Management*

Kemudian ditambahkan *user* baru sehingga terdapat tiga *user* baru seperti yang terdapat pada Gambar 9, pada Gambar 9 dapat dilihat tampilan dari *user management*.

ID	Username	Updated On
30	tenantA2	
29	tenantA1	
18	pm	

Gambar 9. Tampilan *User Management*

Pada Gambar 10 dapat dilihat foto meter air yang terpasang di *simulator*.



Gambar 10. Foto Meter Air Yang Terpasang di *Simulator*

### 5.1 Pengujian *Generate Token*

Pada Gambar 11 dapat dilihat tampilan dari halaman pertama *token generate*, halaman pertama merupakan halaman di mana *user* memilih *building* dan *tenant* yang akan di *token generate* nya. Pada Gambar 12 dapat dilihat tampilan dari halaman kedua *generate token*, halaman kedua *token generate* berfungsi untuk

menentukan jumlah pengisian dan yang terakhir pada Gambar 13 dapat dilihat tampilan hasil *token generate*.

Building  
Pakuwon Mall

Tenant  
Select Tenant

Select Tenant

Tenant A1

Tenant A2

Gambar 11. Tampilan *Token Generate Pertama*

Building  
Pakuwon Mall

Tenant  
Tenant A1

Price Per M3  
12000

Pay  
120000

Volume  
10

Generate Cancel

Gambar 12. Tampilan *Token Generate Kedua*

Building  
Pakuwon Mall

Tenant  
Tenant A1

Price Per M3  
12,000

Pay  
120000

Volume  
10

Token  
FFBF7F4DED

Ok

Gambar 13. Tampilan *Output Dari Generate Token*

## 5.2 Pengujian Token Input

Pengujian *token input* dilakukan dengan memasukkan *token* yang telah di *generate* sebelumnya, pada Gambar 14 dapat dilihat tampilan dari halaman *token input confirmation* yang berisi informasi *token*.

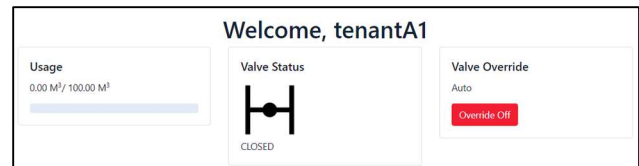
Gambar 14. Halaman *Token Input Confirmation*

## 5.3 Pengujian Sistem Valve Control

Sistem *valve control* berfungsi untuk mengatur *solenoid valve* pada meter air, *solenoid valve* berfungsi untuk menghentikan aliran air dan membuka kembali aliran air. Sistem *valve control* dapat di *override* melalui halaman *tenant management*, *override valve* berfungsi untuk “memaksa” *valve* yang seharusnya terbuka menjadi tertutup atau *valve* yang seharusnya tertutup menjadi terbuka, sistem *override* pada *valve control* memiliki tiga opsi yaitu *off*, *auto*, dan *on*. Opsi *auto* akan memungkinkan *valve* untuk terbuka hingga penggunaan sudah mencapai *limit*. Pada Gambar 15 dapat dilihat tampilan dari *tenant management*, pada bagian kanan tabel *tenant management* merupakan bagian *override option*. Kemudian pada Gambar 16 dapat dilihat opsi *valve override* pada halaman *user tenant* pada bagian kanan.

Valve status	Valve override	Updated On	Override Option
On	Auto		<input type="button" value="OFF"/> <input type="button" value="AUTO"/> <input type="button" value="ON"/>
On	Auto		<input type="button" value="OFF"/> <input type="button" value="AUTO"/> <input type="button" value="ON"/>
			Previous <input type="button" value="1"/> Next

Gambar 15. Tampilan *Tenant Management*



Gambar 16. Halaman Utama *User Tenant*

## 6. KESIMPULAN

Dari keseluruhan sistem yang dibuat, terdapat beberapa kesimpulan yang didapat, antara lain:

- Teknologi yang dibuat dapat memberikan informasi penggunaan air secara *real time* pada *user*.
- Teknologi yang dibuat memungkinkan *user* untuk membuka dan menutup aliran air secara *remote*.
- Teknologi yang dibuat memungkinkan aliran air untuk dihentikan secara otomatis saat *usage* mencapai *limit*.

## 7. DAFTAR PUSTAKA

- [1] Masruchi, Vecky, & Repi. 2016. Perancangan Sistem Pengukuran dan Monitoring Pemakaian Air Rumah PDAM Berbasis SMS (Short Message Service). *Skripsi. Jakarta: Universitas Nasional.*
- [2] Abraham, A. P., Setiabudi, D. H., & Lim, R. 2019. Sistem Monitoring Konsumsi Air Multi-Tenant Menggunakan Arduino dan Mobile Apps. *Skripsi. Surabaya: Universitas Kristen Petra.*
- [3] Anandhavalli, D., Sangeetha, K., Dharshini, V. P., & Fathima, B. L. 2018. Smart Meter for Water Utilization using IoT. *Skripsi. Tamil Nadu: Velammal College of Engineering & Technology, 1003-1004.*
- [4] Subramanian. 2018, September 14. IR Sensor-TCRT5000 Pin and Working Details. Retrieved from AndroidErode: <https://www.androiderode.com/ir-sensor-tcrt5000-pin-and-working-details/>
- [5] Basile, F. 2015. ESP8266 introduction. Retrieved from fabacademy: <http://fabacademy.org/archives/2015/doc/networking-esp8266.html>
- [6] Hareendran, T. (t.thn.). Portable Power- 18650 Battery Shield for Raspberry Pi & Arduino. Diambil kembali dari Electro Schematics: <https://www.electroschematics.com/battery-shield/>
- [7] Dermanto, T. 2013, September 29. Pengertian dan Prinsip Kerja Solenoid Valve. Diambil kembali dari Desain Sistem Kontrol: <http://trikueni-desain-sistem.blogspot.com/2013/08/Solenoid-Valve.html>
- [8] Itron. 2011. Multimag Cyble. Diambil kembali dari Itron: <https://www.itron.com/-/media/feature/products/documents/brochure/multimag-cyble-dn1520-mid--brochure-english.pdf>
- [9] Flask. (t.thn.). Diambil kembali dari The Pallets Projects: <https://palletsprojects.com/p/flask/>