

# Aplikasi *Mobile* Untuk Pemantauan Simulasi Ketinggian Permukaan Air Berbasis *Internet of Things* (IoT)

Andreas Setiawan Hidayat, Resmana Lim

Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya - 60236, Indonesia

Email: m23413019@john.petra.ac.id ; resmana@petra.ac.id

**Abstrak**--Pada *paper* ini, dibuat sistem pemantauan ketinggian permukaan air berbasis *Internet of Things*. Sistem terdiri dari aplikasi *mobile*, Raspberry Pi, sensor ultrasonik, Pi Camera, dan *Cloud*. Diharapkan sistem dapat membantu untuk antisipasi banjir.

Aplikasi *mobile* berfungsi sebagai antarmuka pengguna dengan sistem. Raspberry Pi dan sensor ultrasonik berfungsi untuk mengambil dan mengelola data ketinggian permukaan air. Pi Camera berfungsi untuk melihat keadaan perairan secara langsung. *Cloud* berfungsi untuk menyimpan data ketinggian permukaan air.

Dari pengujian, didapatkan bahwa aplikasi *mobile* dapat melakukan komunikasi dengan *Cloud* untuk mengambil dan mengirim data. Raspberry Pi dapat melakukan komunikasi dengan *Cloud* untuk mengirim data. Pi Camera dapat diakses oleh aplikasi. Pembacaan dengan sensor ultrasonik menggunakan 3 filter: tanpa filter, filter median, moving average filter. Pengujian menunjukkan hasil pembacaan terbaik dihasilkan filter median. Rata-rata kesalahan pembacaan ketinggian permukaan air menggunakan sensor ultrasonik dengan filter median dibandingkan dengan pembacaan menggunakan meter ukur tanpa gangguan sebesar 6,2%, dan dengan gangguan sebesar 6,3%.

**Kata kunci :**

Pemantauan Ketinggian Air, Aplikasi *Mobile*, *Cloud* Ubidots

## I. PENDAHULUAN

Mayoritas individu di kota besar saat ini sangat bergantung pada penggunaan *smartphone* dalam menjalankan aktivitas sehari-hari. Penggunaan teknologi *smartphone* tentunya beriringan dengan pemakaian data dan internet dari penggunaannya. [1]

Perkembangan teknologi informasi dan komunikasi yang begitu cepat telah banyak memberikan kemudahan bagi manusia terutama untuk melakukan komunikasi dan mendapatkan informasi tanpa dibatasi oleh waktu. Salah satu pemanfaatan teknologi ini adalah dalam bidang perairan, khususnya informasi mengenai pasang surut air laut. Pasang surut air laut ini merupakan salah satu fenomena alam yang setiap harinya

terjadi. Informasi mengenai pasang surut air laut ini sangat berguna bagi kegiatan manusia terutama yang berhubungan dengan perairan.

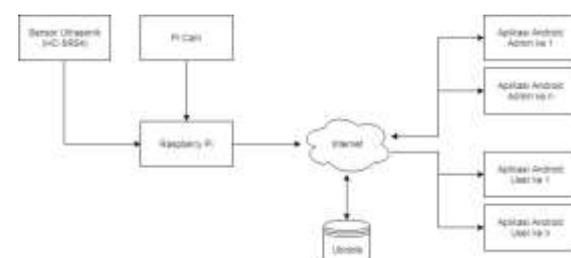
Banjir merupakan salah satu bencana alam yang kerap terjadi di berbagai daerah di Indonesia. Banjir rob merupakan banjir yang berasal dari pasangannya air laut hingga air laut yang pasang tersebut menggenangi daratan. Banjir rob ini juga dikenal sebagai banjir genangan. Pada perumahan, terutama yang dekat dengan permukaan laut, banyak rumah yang tergenang oleh rob sehingga dapat mengakibatkan kerusakan bangunan.

Untuk mencegah kerugian yang diakibatkan oleh banjir rob, perlu adanya sistem untuk memantau ketinggian permukaan air. Dengan mengetahui ketinggian permukaan air, maka tindakan lanjutan yang diperlukan dapat diambil sesuai dengan ketinggian permukaan air.

Pada sistem yang dikembangkan di *paper* ini, pembacaan ketinggian air akan dilakukan secara *real-time* untuk mendapatkan data yang lebih akurat. Sebelum pengiriman data, data akan dilewatkan filter untuk mendapatkan hasil yang *smooth*. Selain itu, sistem akan dilengkapi dengan fitur *streaming* Pi Camera untuk dapat melihat keadaan lapangan secara langsung.

## II. PERANCANGAN SISTEM

Berikut ini adalah skema sistem secara keseluruhan dari *paper* ini:



Gambar 1. Skema Sistem

Pada Gambar 1, Raspberry Pi berfungsi untuk mengambil data ketinggian air dan mengirimkan

data ketinggian air menuju *cloud* dimana pada *paper* ini digunakan *cloud* Ubidots. Selain itu, Raspberry Pi berfungsi untuk menjadi *streaming server* Pi Camera.

Selanjutnya, data yang ada di *cloud* Ubidots akan dikirim ke aplikasi Android. Setiap kali ada data baru yang dikirim Raspberry Pi ke Ubidots, Ubidots akan mengirimkan data baru tersebut ke aplikasi Android.

Pi Camera berfungsi untuk *streaming* kondisi perairan secara langsung dan dapat diakses oleh aplikasi Android melalui menu yang tersedia di aplikasi.

Koneksi antara Raspberry Pi dengan Ubidots menggunakan protokol MQTT. Koneksi antara aplikasi Android dengan Ubidots menggunakan protokol MQTT dan HTTP. Protokol *streaming* yang dipakai aplikasi Android adalah HTTP

Aplikasi Android memiliki 2 jenis pengguna, yakni User dan Admin. Tabel 1 menunjukkan perbedaan layanan dari pengguna User dengan Admin.

Tabel 1. Layanan Dari User dan Admin

Layanan	User	Admin
Melihat data ketinggian permukaan air	✓	✓
Melihat secara langsung keadaan lapangan melalui IP Camera	✓	✓
Mengubah nilai notifikasi ketinggian air	✗	✓

A. Desain Raspberry Pi

Berikut ini adalah tabel dari daftar komponen yang dibutuhkan untuk sistem ini:

Tabel 2. Daftar Komponen Yang Diperlukan

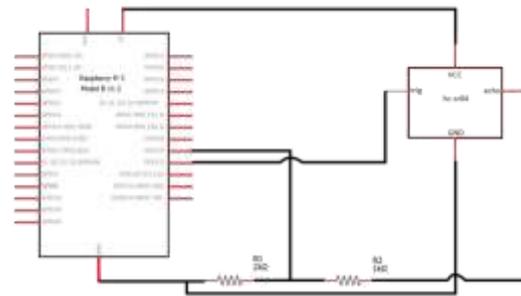
No	Komponen	Jumlah
1	Raspberry Pi 3 Model B	1
2	HC-SR04 Sensor Ultrasonik	1
3	Resistor 1KΩ	1
4	Resistor 2KΩ	1
5	Pi Camera	1

Berikut ini adalah tabel konfigurasi pin Raspberry Pi yang dipakai pada sistem ini:

Tabel 3. Konfigurasi Pin Raspberry Pi

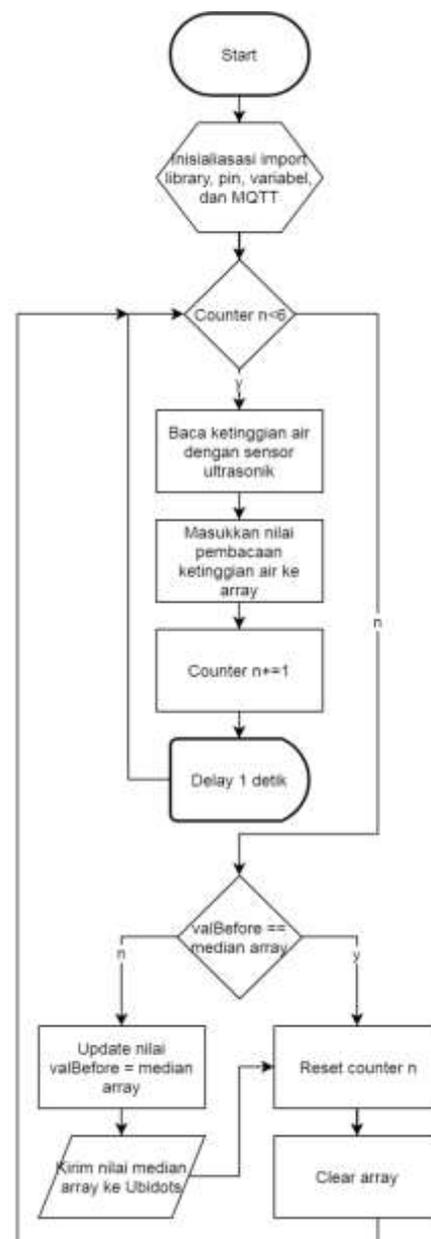
Pin	Fungsi
5V	Sumber tegangan HC-SR04
GND	Ground sumber
GPIO 23	Pin <i>Trigger</i> HC-SR04
GPIO 24	Pin <i>Echo</i> HC-SR04
CSI	Pi Camera

Setelah komponen dan konfigurasi pin telah ditentukan, maka rangkaian yang dipakai menjadi seperti Gambar 2.



Gambar 2. Rangkaian Raspberry Pi

Gambar 3 adalah *flowchart* dari program Raspberry Pi:



Gambar 3. Flowchart Program Raspberry Pi  
 Desain program pada Raspberry Pi menggunakan beberapa *library* yaitu RPI.GPIO, time, paho.mqtt.client, dan numpy. RPI.GPIO digunakan untuk mendapat akses pin GPIO

Raspberry Pi, time digunakan untuk menghitung waktu, paho.mqtt.client digunakan untuk koneksi MQTT, dan numpy digunakan untuk menghitung nilai median.

Pada awal program akan dilakukan inisialisasi *import library*, pin, variabel, dan MQTT. Setelah inisialisasi, program akan langsung mulai mengambil nilai ketinggian air dan menyimpannya di *array* sebanyak 6 kali dengan interval 1 detik. Untuk menghitung sudah berapa kali pembacaan dilakukan, digunakan *counter* n.

Setelah pengambilan data selesai, program akan membandingkan nilai median dari array nilai ketinggian air dengan *valBefore*. Apabila nilai *valBefore* tidak sama dengan nilai median, maka nilai median akan dimasukkan ke *valBefore* dan dikirim ke Ubidots. Apabila nilai *valBefore* sama dengan nilai median, program akan langsung ke langkah selanjutnya. Selanjutnya, program akan *me-reset counter* n dan mengosongkan *array*. Setelah *reset* dilakukan, program akan mengulang pembacaan ketinggian air lagi.

**B. Desain Aplikasi Android**

Tabel 4 menunjukkan desain layanan dari aplikasi Android.

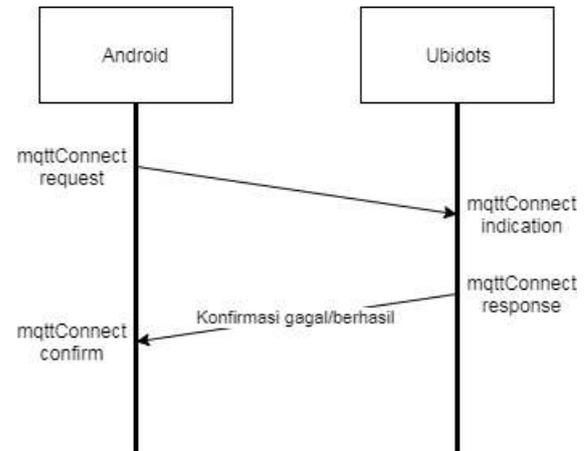
Tabel 4. Desain Layanan Aplikasi Android

Servis Primitif	Parameter	Deskripsi
mqttConnect request	clientID, username, password	Layanan ini adalah prosedur koneksi dengan Ubidots
mqttConnect indication	clientID, username, password	
mqttConnect response	returnCode	
mqttConnect confirm	returnCode	
mqttSubscribe request	topic, QoS	Layanan ini adalah prosedur subscribe Android ke Ubidots untuk mendapatkan nilai terbaru dari topic setiap kali ada update
mqttSubscribe indication	topic, QoS	
mqttSubscribe response	returnCode	
mqttSubscribe confirm	returnCode	
mqttCallback request	topic, message	
mqttCallback indication	topic, message	
mqttDisconnect request	-	Layanan ini adalah prosedur disconnect Android dengan Ubidots
mqttDisconnect indication	-	
mqttDisconnect response	returnCode	
mqttDisconnect confirm	returnCode	
asynhttpGet	topic, token	Layanan ini

request		adalah prosedur pengambilan nilai notifikasi yang ada di Ubidots
asynhttpGet indication	topic, token	
asynhttpGet response	statusCode, responseString	
asynhttpGet confirm	statusCode, responseString	Layanan ini adalah prosedur update nilai notifikasi yang ada di Ubidots
asynhttpPost request	topic, key, value	
asynhttpPost indication	topic, key, value	
asynhttpPost response	statusCode	
asynhttpPost confirm	statusCode	

Pada aplikasi Android, terdapat 5 servis elemen yaitu:

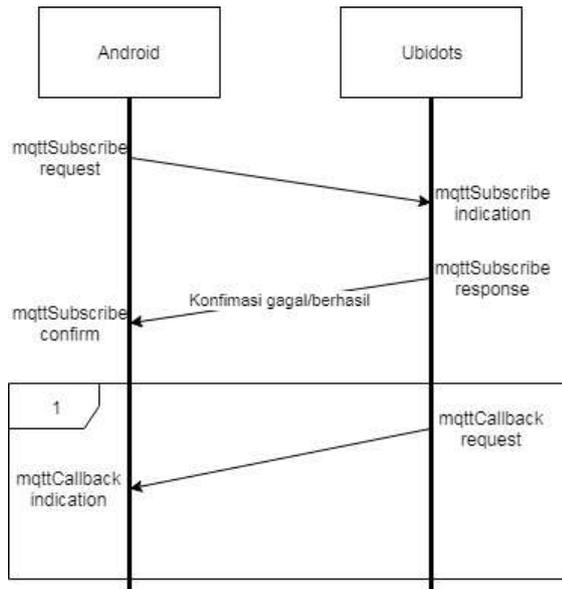
1. Koneksi dengan Ubidots
2. *Subscribe topic*
3. *Disconnect* dengan Ubidots
4. Mengambil nilai notifikasi
5. *Update* nilai notifikasi



Gambar 4. Desain Protokol Koneksi Aplikasi Android dengan Ubidots

Pada servis elemen pertama yaitu koneksi dengan Ubidots, aplikasi akan mengajukan permintaan untuk membuka koneksi antara aplikasi dengan Ubidots. Aplikasi akan mengirim 3 parameter, yakni clientID, username, dan password. Username diisi dengan token dari akun Ubidots. Fungsi token di *username* supaya Ubidots mengenali akun Ubidots mana yang ingin diakses oleh aplikasi.

Aplikasi akan menerima konfirmasi dari Ubidots mengenai berhasil atau tidaknya koneksi dibangun. Proses koneksi gagal apabila token yang dikirim tidak valid. Apabila koneksi berhasil, Ubidots akan mengirim *connection acknowledgement* dan akan membuka koneksi antara aplikasi dengan Ubidots sampai ditutup.

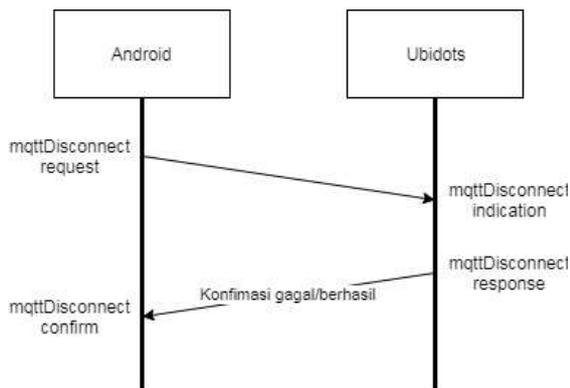


Gambar 5. Desain Protokol *Subscribe* Aplikasi Android ke Ubidots

Pada servis elemen kedua yaitu *subscribe topic*, aplikasi mengajukan permintaan ke Ubidots untuk *subscribe* ke *topic* yang diperlukan. Aplikasi akan mengirim 2 parameter yakni *topic*, dan *QoS*. *Topic* diisi dengan variabel di Ubidots yang diinginkan.

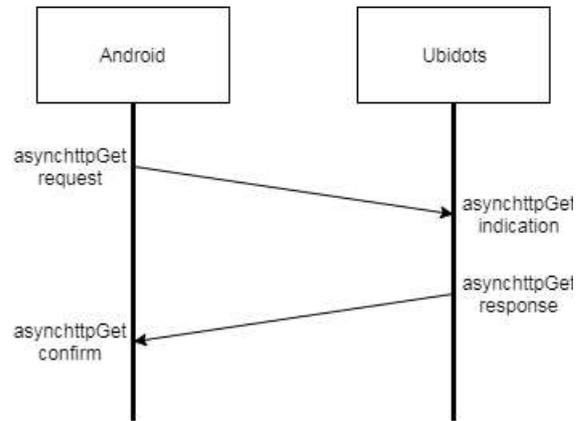
Aplikasi akan menerima konfirmasi dari Ubidots mengenai berhasil tidaknya aplikasi melakukan *subscribe*. Apabila berhasil, Ubidots akan mengirim *connection acknowledgement*.

Setelah berhasil melakukan *subscribe*, tiap kali ada data baru pada variabel yang *unsubscribe*, maka Ubidots akan mengirimkan data baru tersebut ke aplikasi.



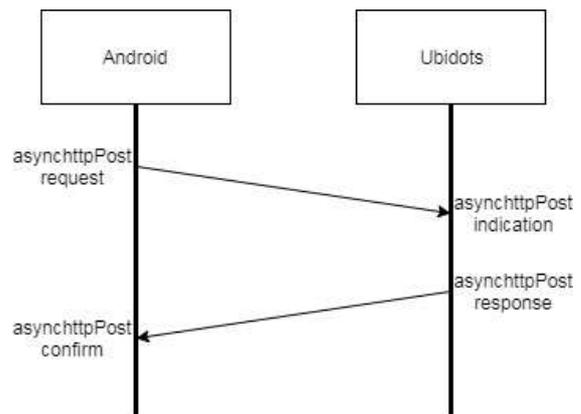
Gambar 6. Desain Protokol *Disconnect* Aplikasi Android dengan Ubidots

Pada servis elemen ketiga yaitu *disconnect* dengan Ubidots, aplikasi mengajukan permintaan untuk memutus sambungan koneksi dengan Ubidots tanpa parameter. Setelah itu, Ubidots akan membalas dengan konfirmasi gagal atau berhasilnya pemutusan sambungan koneksi.



Gambar 7. Desain Protokol Pengambilan Nilai Notifikasi dari Ubidots

Pada servis elemen keempat yaitu mengambil nilai notifikasi, aplikasi mengajukan permintaan untuk mengambil nilai notifikasi dari Ubidots dengan 2 parameter yakni *token*, dan *topic*. Apabila permintaan berhasil, Ubidots akan mengirimkan nilai notifikasi terbaru ke aplikasi dalam *responseString*.

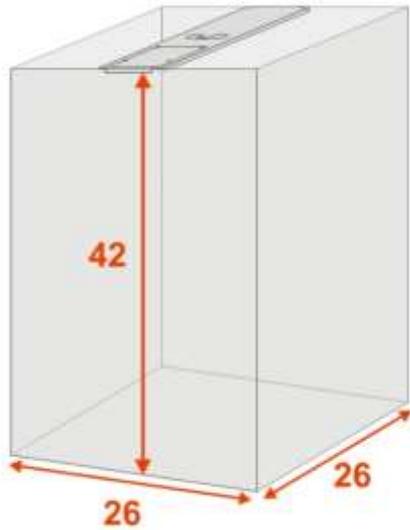


Gambar 8. Desain Protokol Pengiriman Nilai Notifikasi Baru ke Ubidots

Pada servis elemen kelima yaitu *update* nilai notifikasi, aplikasi akan mengirimkan nilai notifikasi baru dengan 3 parameter, yakni *topic*, *key*, dan *value*. *Topic* adalah URL variabel yang dituju, dan *value* adalah nilai notifikasi baru. Setelah itu, Ubidots akan memberi konfirmasi gagal atau berhasilnya permintaan untuk *update* nilai notifikasi.

### C. Desain Plant Simulasi

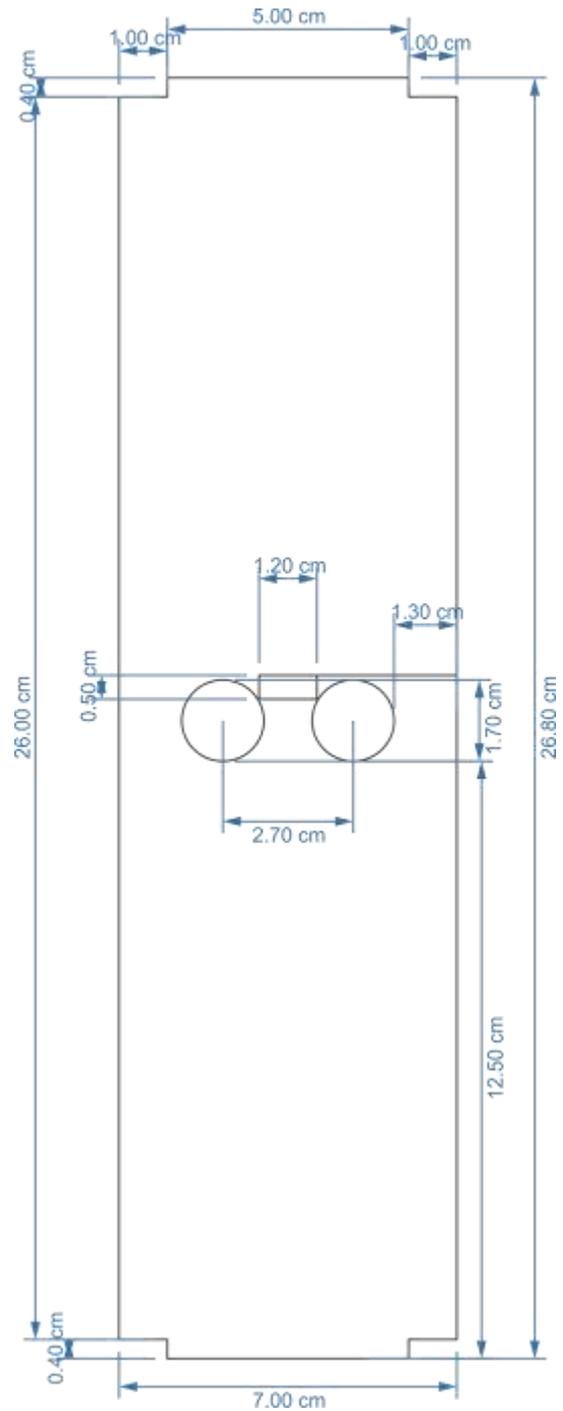
*Plant* simulasi menggunakan bahan akrilik dengan ketebalan 4mm. Pada bagian dalam *plant*, diberi *sealant* agar air tidak bocor keluar. Pada bagian dalam *plant*, dipasang pompa air untuk mengeluarkan air dari *plant*. Pada salah satu sisi luar *plant*, dipasang meter ukur untuk mempermudah pengukuran.



Gambar 9. Desain *Plant* Simulasi



Gambar 10. *Plant* Simulasi



Gambar 11. Desain *Platform* Sensor

### III. PENGUJIAN SISTEM

#### A. *Pembacaan Ketinggian Air dan Pengaruh Filter Terhadap Pembacaan*

Pembacaan dilakukan menggunakan 2 cara, meter ukur yang dipasang di badan *plant* dan sensor ultrasonik. Pembacaan sensor sendiri dibagi menjadi 3, yakni:

1. Pembacaan tanpa filter
2. Pembacaan dengan filter median
3. Pembacaan dengan filter moving average

Pembacaan dengan meter ukur dijadikan patokan pada percobaan ini. Perhitungan kesalahan

pembacaan menggunakan sensor dibanding meter ukur ditampilkan dalam bentuk persentase.

Pembacaan dilakukan sebanyak 10 kali. Titik ukur dari pengujian ini adalah keberhasilan pembacaan sensor dan kemampuan filter untuk menghasilkan pembacaan yang akurat dan tahan terhadap gangguan.

Pengujian pertama dilakukan tanpa adanya gangguan terhadap sensor ultrasonik.

Tabel 5. Pengujian Pembacaan Ketinggian Air Tanpa Gangguan

Pengujian Pembacaan Ketinggian Air Tanpa Gangguan				
Pengambilan Data	Tanpa filter (cm)	Median (cm)	Moving Average (cm)	Meter Ukur (cm)
1	8	9	8	9
2	11	11	9	13
3	17	15	10	17
4	20	19	12	21
5	23	22	14	24
6	27	26	17	28
7	31	29	22	31
8	34	33	25	34
9	35	35	28	35
10	36	36	31	36

Tabel 6. Kesalahan Pembacaan Sensor dengan Pembacaan Meter Ukur

Kesalahan Pembacaan Sensor Dengan Pembacaan Meter Ukur Tanpa Gangguan			
Pengambilan Data	Tanpa Filter	Median	Moving Average
1	11,1%	0,0%	11,1%
2	15,4%	15,4%	30,8%
3	0,0%	11,8%	41,2%
4	4,8%	9,5%	42,9%
5	4,2%	8,3%	41,7%
6	3,6%	7,1%	39,3%
7	0,0%	6,5%	29,0%
8	0,0%	2,9%	26,5%
9	0,0%	0,0%	20,0%
10	0,0%	0,0%	13,9%
<b>Rata-Rata</b>	<b>3,9%</b>	<b>6,2%</b>	<b>29,6%</b>

Pada pengujian tanpa gangguan, berdasarkan hasil pengujian di Tabel 6, rata-rata kesalahan pembacaan terendah dihasilkan oleh pembacaan tanpa filter sebesar 3,9%, diikuti dengan pembacaan dengan filter median sebesar 6,2%, dan terakhir adalah pembacaan dengan filter moving average sebesar 29,6%

Pengujian kedua dilakukan dengan adanya gangguan terhadap sensor ultrasonik. Gangguan dilakukan dengan cara menghalangi sensor ultrasonik dengan tangan selama beberapa detik, dan menggoyang *plant* simulasi. Gangguan dilakukan untuk menyimulasikan keadaan dimana

terkadang antara sensor dengan permukaan air bisa terhalangi sesuatu, dan gelombang air.

Tabel 7. Pengujian Pembacaan Ketinggian Air Dengan Gangguan

Pengujian Pembacaan Ketinggian Air Dengan Gangguan				
Pengambilan data	Tanpa filter (cm)	Median (cm)	Moving Average (cm)	Meter Ukur (cm)
1	10	9	8	10
2	37	12	13	15
3	17	17	15	18
4	21	20	17	22
5	15	23	18	26
6	28	28	21	29
7	32	31	25	32
8	34	34	25	34
9	35	35	28	35
10	35	35	30	35

Tabel 8. Kesalahan Pembacaan Sensor dengan Pembacaan Meter Ukur

Kesalahan Pembacaan Sensor Dengan Pembacaan Meter Ukur Dengan Gangguan			
Pengambilan Data	Tanpa Filter	Median	Moving Average
1	0,0%	10,0%	20,0%
2	146,7%	20,0%	13,3%
3	5,6%	5,6%	16,7%
4	4,5%	9,1%	22,7%
5	42,3%	11,5%	30,8%
6	3,4%	3,4%	27,6%
7	0,0%	3,1%	21,9%
8	0,0%	0,0%	26,5%
9	0,0%	0,0%	20,0%
10	0,0%	0,0%	14,3%
<b>Rata-Rata</b>	<b>20,3%</b>	<b>6,3%</b>	<b>21,4%</b>

Pada pengujian dengan gangguan, berdasarkan hasil pengujian di Tabel 8, rata-rata kesalahan pembacaan terendah dihasilkan oleh pembacaan dengan filter median sebesar 6,3%, diikuti dengan pembacaan tanpa filter sebesar 20,3%, dan terakhir adalah pembacaan dengan filter moving average sebesar 21,4%

### B. Komunikasi Antara Raspberry Pi dengan Ubidots

Saat pengujian, dilakukan *ping test* ke alamat [things.ubidots.com](https://things.ubidots.com), dan hasilnya adalah sebagai berikut:

```
rtt min/avg/max/mdev =
209.131/214.224/217.716/3.702 ms
```

Pengujian dilakukan sebanyak 10 kali dengan interval lebih kurang 10 detik. Data yang dikirim oleh Raspberry Pi berupa angka acak dari 0-100.

Titik ukur dari pengujian ini adalah keberhasilan Raspberry Pi untuk mengirim data ke

Ubidots dengan protokol MQTT, dan kesesuaian data yang dikirim oleh Raspberry Pi dengan data yang diterima Ubidots.

Tabel 9. Pengujian Pengiriman Data dari Raspberry Pi ke Ubidots

<b>Pengujian Pengiriman Data dari Raspberry Pi ke Ubidots</b>		
<b>Percobaan</b>	<b>Diterima Ubidots</b>	<b>Kesalahan Data</b>
1	Berhasil	0
2	Berhasil	0
3	Berhasil	0
4	Berhasil	0
5	Berhasil	0
6	Berhasil	0
7	Berhasil	0
8	Berhasil	0
9	Berhasil	0
10	Berhasil	0
<b>Rata-rata</b>	100%	0%

Berdasarkan hasil pengujian di Tabel 9, Raspberry Pi memiliki tingkat keberhasilan 100% untuk mengirim data ke Ubidots dan tidak ditemukan kesalahan data yang dikirim selama pengujian dengan syarat nilai ping maksimal 217,716ms.

**C. Komunikasi Antara Aplikasi Android dengan Ubidots**

Saat pengujian, dilakukan *ping test* ke alamat things.ubidots.com, dan hasilnya adalah sebagai berikut:

```
rtt min/avg/max/mdev =
209.518/213.467/224.113/4.912 ms
```

Pengujian pertama adalah pengiriman data dari aplikasi Android ke Ubidots menggunakan Protokol HTTP. Pengujian dilakukan sebanyak 10 kali dengan interval lebih kurang 10 detik. Data yang dikirim oleh aplikasi berupa angka acak dari 0-100.

Titik ukur dari pengujian ini adalah keberhasilan aplikasi Android dalam berkomunikasi dengan Ubidots menggunakan protokol HTTP untuk mengirim data, dan kesesuaian data yang dikirim oleh aplikasi dengan data yang diterima Ubidots.

Tabel 10. Pengujian Pengiriman Data dari Aplikasi Android ke Ubidots Menggunakan Protokol HTTP

<b>Pengujian Pengiriman Data dari Aplikasi Android ke Ubidots Menggunakan Protokol HTTP</b>		
<b>Percobaan</b>	<b>Diterima Ubidots</b>	<b>Kesalahan Data</b>
1	Berhasil	0
2	Berhasil	0
3	Berhasil	0
4	Berhasil	0

5	Berhasil	0
6	Berhasil	0
7	Berhasil	0
8	Berhasil	0
9	Berhasil	0
10	Berhasil	0
<b>Rata-Rata</b>	100%	0%

Berdasarkan hasil pengujian di Tabel 10, aplikasi Android memiliki tingkat keberhasilan 100% untuk mengirim data ke Ubidots dan tidak ditemukan kesalahan data yang dikirim selama pengujian menggunakan protokol HTTP dengan syarat nilai ping maksimal 224,113ms.

Pengujian kedua adalah pengambilan data dari Ubidots oleh aplikasi Android menggunakan Protokol MQTT. Pengujian dilakukan sebanyak 10 kali dengan interval lebih kurang 10 detik. Data yang dikirim dari Ubidots berupa angka acak dari 0-100.

Titik ukur dari pengujian ini adalah keberhasilan aplikasi Android dalam berkomunikasi dengan Ubidots menggunakan protokol MQTT untuk mengambil data, dan kesesuaian data yang diambil oleh aplikasi dengan data yang dikirim Ubidots

Tabel 11. Pengujian Pengambilan Data dari Ubidots oleh Aplikasi Android Menggunakan Protokol MQTT

<b>Pengujian Pengambilan Data dari Ubidots oleh Aplikasi Android Menggunakan Protokol MQTT</b>		
<b>Percobaan</b>	<b>Diterima Aplikasi</b>	<b>Kesalahan Data</b>
1	Berhasil	0
2	Berhasil	0
3	Berhasil	0
4	Berhasil	0
5	Berhasil	0
6	Berhasil	0
7	Berhasil	0
8	Berhasil	0
9	Berhasil	0
10	Berhasil	0
<b>Rata-Rata</b>	100%	0%

Berdasarkan hasil pengujian di Tabel 11, aplikasi Android memiliki tingkat keberhasilan 100% untuk mengambil data dari Ubidots dan tidak ditemukan kesalahan data yang diambil selama pengujian menggunakan protokol MQTT dengan syarat nilai ping maksimal 224,113ms.

Pengujian ketiga adalah pengambilan data dari Ubidots oleh aplikasi Android menggunakan Protokol HTTP. Pengujian dilakukan sebanyak 10 kali dengan interval lebih kurang 10 detik. Data yang dikirim dari Ubidots berupa angka acak dari 0-100.

Titik ukur dari pengujian ini adalah keberhasilan aplikasi Android dalam

berkomunikasi dengan Ubidots menggunakan protokol HTTP untuk mengambil data, dan kesesuaian data yang diambil oleh aplikasi dengan data yang dikirim Ubidots

Tabel 12. Pengujian Pengambilan Data dari Ubidots oleh Aplikasi Android Menggunakan Protokol HTTP

Pengujian Pengambilan Data dari Ubidots oleh Aplikasi Android Menggunakan Protokol HTTP		
Percobaan	Diterima Aplikasi	Kesalahan Data
1	Berhasil	0
2	Berhasil	0
3	Berhasil	0
4	Berhasil	0
5	Berhasil	0
6	Berhasil	0
7	Berhasil	0
8	Berhasil	0
9	Berhasil	0
10	Berhasil	0
<b>Rata-Rata</b>	100%	0%

Berdasarkan hasil pengujian di Tabel 12, aplikasi Android memiliki tingkat keberhasilan 100% untuk mengambil data dari Ubidots dan tidak ditemukan kesalahan data yang diambil selama pengujian menggunakan protokol HTTP dengan syarat nilai ping maksimal 224,113ms.

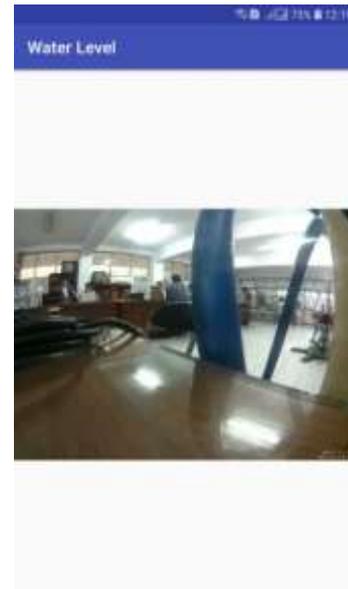
**D. Streaming Pi Camera**

Pengujian dilakukan sebanyak 10 kali dengan interval lebih kurang 10 detik. Titik ukur dari pengujian ini adalah keberhasilan aplikasi Android melakukan *streaming* Pi Camera.

Tabel 13. Pengujian Streaming Pi Camera

Percobaan	Streaming Berhasil
1	Berhasil
2	Berhasil
3	Berhasil
4	Berhasil
5	Berhasil
6	Berhasil
7	Berhasil
8	Berhasil
9	Berhasil
10	Berhasil
<b>Rata-Rata</b>	100%

Berdasarkan hasil pengujian di Tabel 13, aplikasi Android memiliki tingkat keberhasilan 100% untuk melakukan *streaming* Pi Camera.



Gambar 12. Aplikasi Melakukan *Streaming* Pi Camera

**IV. KESIMPULAN**

Kesimpulan yang didapat dari pembuatan dan pengujian sistem ini adalah sebagai berikut:

1. Raspberry Pi dengan HC-SR04 memerlukan filter data supaya pembacaan ketinggian air pada saat terjadi gangguan pada sensor tidak terlalu kacau.
2. Dibandingkan dengan tanpa filter, dan filter moving average, filter median memiliki kinerja terbaik untuk filter data.
3. Pembacaan ketinggian air dengan menggunakan filter median menghasilkan persentase rata-rata kesalahan pembacaan tanpa gangguan sebesar 6,2%, dan dengan gangguan sebesar 6,3%
4. Raspberry Pi dengan koneksi MQTT dapat mengirim data ke Ubidots tanpa ada kesalahan data dengan syarat maksimal ping 217,716ms.
5. Raspberry Pi dengan Pi Camera berfungsi dengan baik sebagai *streaming server*.
6. Aplikasi Android dengan koneksi MQTT dan HTTP dapat melakukan komunikasi dengan Ubidots untuk mengirim dan mengambil data tanpa ada kesalahan data dengan syarat maksimal ping 224,113ms.

**DAFTAR PUSTAKA**

[1] R. Davies, "The internet of things opportunities and challenges", *European Parliamentary Research Service*, vol. PE 557.012, Mei 2015.