

# UTS AI ENABLED INTERNET OF THINGS 2021 – 1

Otniel Abiezer – IF-42-GAB01 – 1301180469

## Soal Ujian Bagian Nomor 1 : Teori MQTT

### 1. Standar protokol IoT dengan MQTT

- (a) Protokol komunikasi yang akan digunakan adalah Wifi.

Alasannya karena wifi memiliki bandwidth yang lebih besar, jangkauan wilayah wifi yang lebih luas, dan memiliki keamanan yang lebih baik daripada Bluetooth. Selain itu, data yang dikirim dari microcontroller dapat disimpan ke dalam cloud.

- (b) Data temperatur dan data water level dapat dikirimkan secara bersamaan.

Caranya, dari publisher (pada kasus ini microcontroller yang berperan sebagai publisher) memasukan data temperature dan data water level ke dalam satu Topic yang sama. Jadi, kita bisa melakukan konfigurasi agar dapat mengirimkan kedua data tersebut ke dalam satu Topic sehingga dapat dikirimkan kepada MQTT client sebagai subscriber.

### 2. Melewati X

- (a) X merupakan Broker. Broker adalah penghubung transaksi data antara publisher (microcontroller) dengan subscriber (MQTT client). MQTT dapat mengenali pengelompokan yang disebut Topic tersebut sehingga dapat handle data dari berbagai device.

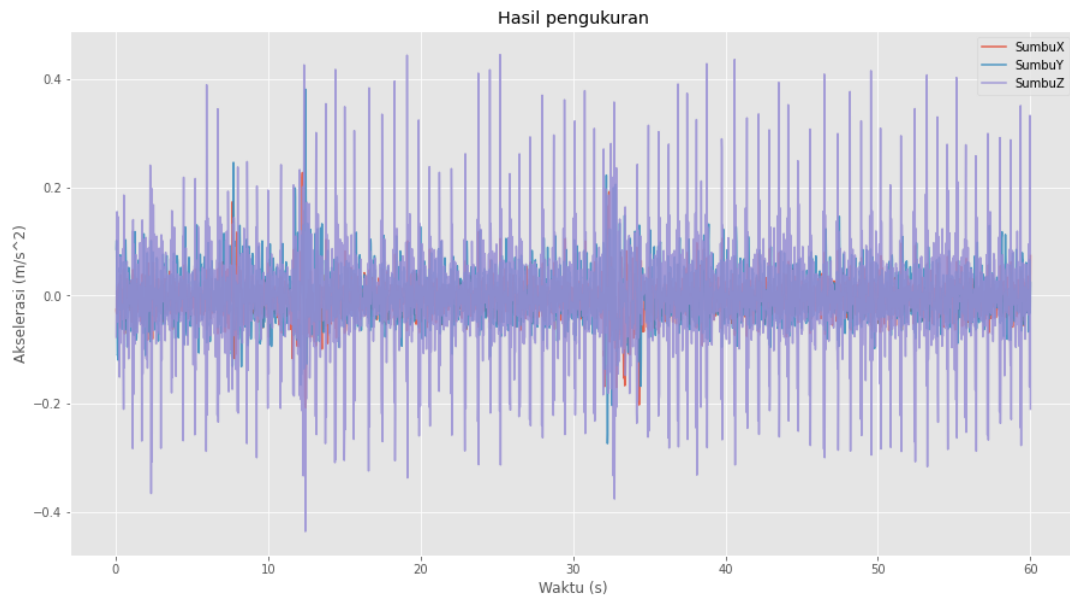
- (b) Broker (tanda X) dapat berperan menjadi subscriber dan publisher sekaligus.

X menjadi subscriber saat menerima data dari publisher (microcontroller) dan X menjadi publisher saat mengirim data kepada subscriber (MQTT client).

## Soal Ujian Bagian Nomor 2 : Eksperimen IoT dengan Smartphone

### 3. Direct method

#### (a) Hasil plot



Selama waktu 60 s, pernafasan diukur dengan menggunakan 3 sumbu yaitu Sumbu X, Sumbu Y, dan Sumbu Z. Interval dari akselerasi ketiga sumbu tersebut dari -0.4 sampai 0.4 m/s<sup>2</sup>. Sumbu yang paling banyak noise, adalah Sumbu Y.

#### (b) Program

Membaca file excel berupa pengukuran dari Phyphox

Mendapatkan Dataset, hasil pengukuran dari Phyphox

```
url = 'https://drive.google.com/uc?id=1AxN3P1ImJ9s6m2zYzBdXXh-uohtH7jN'  
df_nafas = pd.read_excel(url)
```

Mengubah nama kolom, untuk mempermudah pekerjaan ke depannya dan menampilkan contohnya

Mengganti nama kolom untuk mempermudah

```
[ ] df_nafas.rename(columns={'Time (s)': 'Time', 'Linear Acceleration x (m/s^2)': 'SumbuX', 'Linear Acceleration y (m/s^2)': 'SumbuY', 'Linear Acceleration z (m/s^2)': 'SumbuZ'})  
df_nafas.head(3)
```

	Time	SumbuX	SumbuY	SumbuZ
0	0.054410	-0.026112	0.085231	-0.077026
1	0.069333	-0.031871	0.101178	0.010769
2	0.084256	-0.033066	0.008082	0.128082

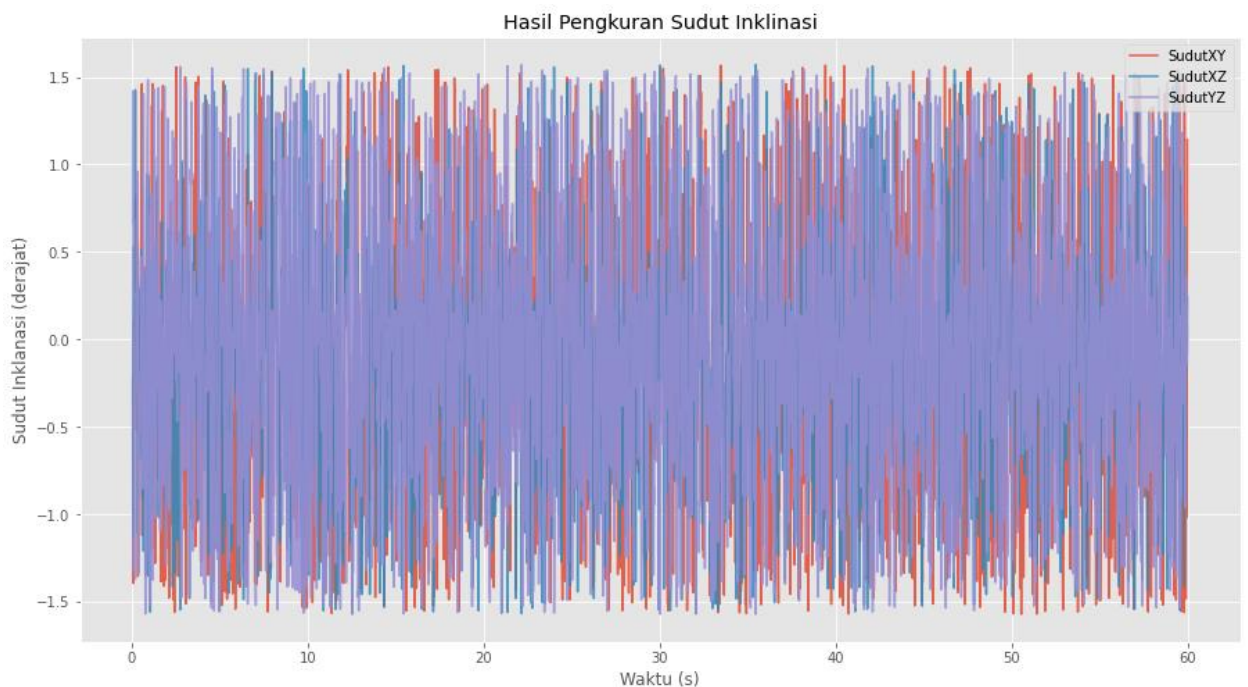
Melakukan plot

Visualisasi hasil pengukuran langsung dari Phyphox yang memiliki 3 sumbu (X,Y,Z)

```
df_nafas.plot(x='Time', figsize=(15,8), alpha=0.85)  
plt.title("Hasil pengukuran")  
plt.xlabel("Waktu (s)")  
plt.ylabel("Akselerasi (m/s^2)")  
plt.show()
```

#### 4. Inklinasi

(a) Hasil plot

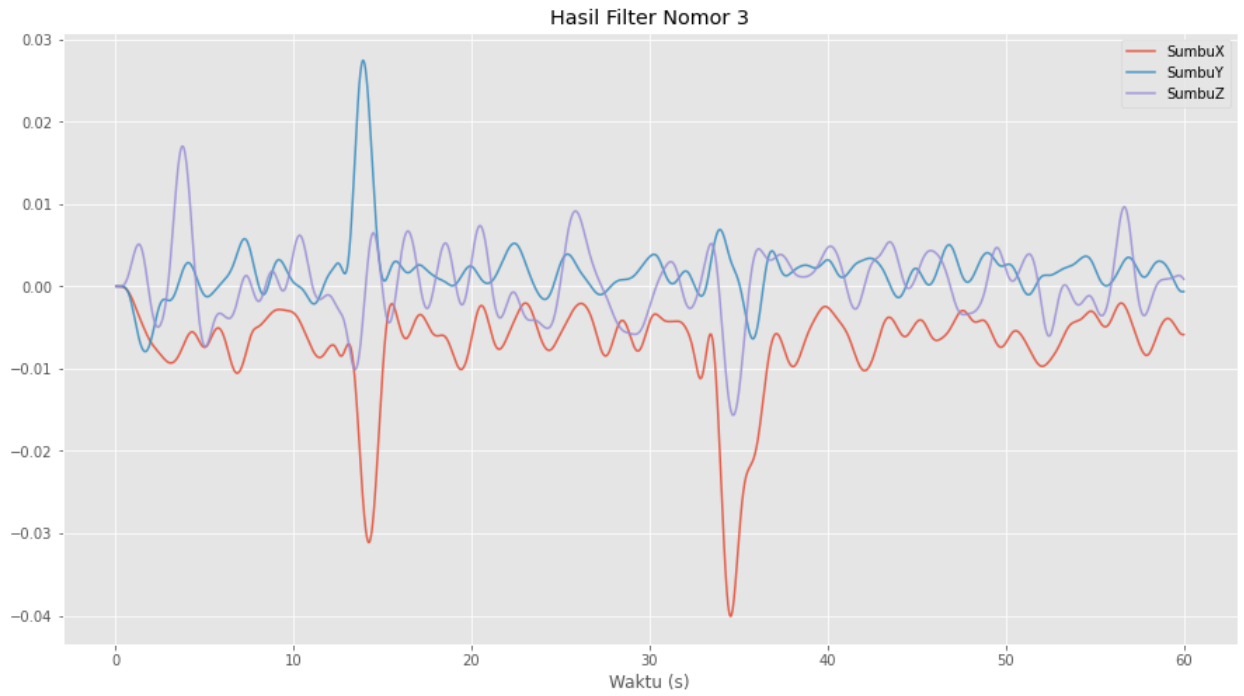


Plot menggunakan 3 sudut yaitu antara Sumbu X dan Y (dinamakan Sudut XY), Sumbu X dan Z (dinamakan Sudut XZ), dan Sumbu Y dan Z (dinamakan Sudut YZ). Besarnya sudut inklinasi berkisar antara -1.5 dan 1.5 (dalam satuan derajat). Sudut yang paling noise adalah Sudut YZ, menyesuaikan juga dengan yang paling noise di nomor 3 adalah Sumbu Y dan Sumbu Z.

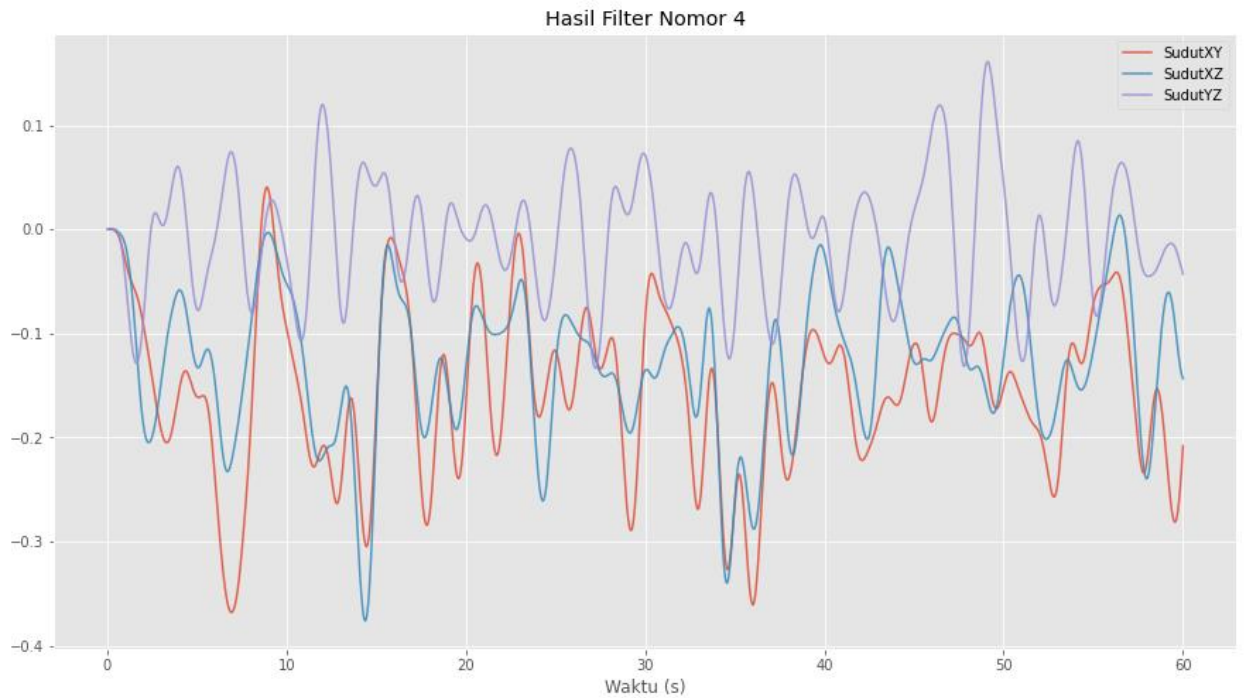
(b) Kedua grafik agak berbeda, tetapi memiliki kemiripan pada Sumbu dan Sudut yang dominan (seperti dijelaskan di atas) yang sama-sama memiliki noise paling besar.

## 5. Menghitung Laju Respirasi

### Hasil LPF nomor 3



### Hasil LPF nomor 4



(a) Mendeteksi peak

```
[ ] peaks_x, _ = find_peaks(df_filter_nomor3['SumbuX'])
peaks_y, _ = find_peaks(df_filter_nomor3['SumbuY'])
peaks_z, _ = find_peaks(df_filter_nomor3['SumbuZ'])
peaks_xy, _ = find_peaks(df_filter_nomor4['SudutXY'])
peaks_xz, _ = find_peaks(df_filter_nomor4['SudutXZ'])
peaks_yz, _ = find_peaks(df_filter_nomor4['SudutYZ'])

print('\nTitik puncak Sumbu X pada nomor 3 : ', peaks_x)
print('\nTitik puncak Sumbu Y pada nomor 3 : ', peaks_y)
print('\nTitik puncak Sumbu Z pada nomor 3 : ', peaks_z)
print('\nTitik puncak Sudut XY pada nomor 4 : ', peaks_xy)
print('\nTitik puncak Sudut XZ pada nomor 4 : ', peaks_xz)
print('\nTitik puncak Sudut YZ pada nomor 4 : ', peaks_yz)

Titik puncak Sumbu X pada nomor 3 : [ 286 384 614 815 875 1038 1144 1226 1375 1541 1750 1905 2029 2110
2237 2485 2669 2909 3026 3187 3260 3386 3669 3783 3958]

Titik puncak Sumbu Y pada nomor 3 : [ 20 182 272 484 613 835 930 1053 1141 1333 1499 1699 2024 2143
2272 2468 2594 2677 2837 3009 3135 3279 3359 3498 3649 3810 3926]

Titik puncak Sumbu Z pada nomor 3 : [ 85 250 390 490 588 691 798 968 1098 1239 1369 1495 1571 1729
2088 2240 2418 2500 2689 2911 3066 3315 3436 3570 3661 3794 3999]

Titik puncak Sudut XY pada nomor 4 : [ 289 355 594 806 908 1057 1254 1381 1535 1669 1789 1882 2032 2256
2359 2482 2635 2736 2915 3018 3162 3256 3373 3605 3769 3921]

Titik puncak Sudut XZ pada nomor 4 : [ 267 372 598 890 1046 1243 1377 1544 1710 1882 2013 2130 2247 2365
2495 2663 2916 3051 3162 3248 3404 3586 3781 3964]

Titik puncak Sudut YZ pada nomor 4 : [ 20 177 260 460 612 802 953 1031 1155 1283 1411 1552 1730 1897
2001 2159 2253 2395 2567 2670 2828 3109 3288 3482 3625 3789 3973]
```

Mendeteksi puncak dengan menggunakan find\_peaks.

Pada 6 baris teratas, untuk membuat variabel yang menampung peaks dari masing-masing sumbu dan sudut, yang ditampung hasil return nya dalam bentuk list berisi angka-angka dalam indeks yang merupakan titik puncak (peaks).

Pada 6 baris terakhir, menampilkan ke layer di titik mana saja yang merupakan puncak. Perlu diingat angka tersebut merupakan indeks, bukan waktu.

(b) Menghitung banyak puncak

```
▶ print("Pada Nomor 3, banyaknya puncak = ")
print(len(peaks_x), " pada sumbu X")
print(len(peaks_y), " pada sumbu Y")
print(len(peaks_z), " pada sumbu Z")

print("\nPada Nomor 4, banyaknya puncak = ")
print(len(peaks_xy), " pada sudut XY")
print(len(peaks_xz), " pada sudut XZ")
print(len(peaks_yz), " pada sudut YZ")

☐ Pada Nomor 3, banyaknya puncak =
25 pada sumbu X
27 pada sumbu Y
27 pada sumbu Z

Pada Nomor 4, banyaknya puncak =
26 pada sudut XY
24 pada sudut XZ
27 pada sudut YZ
```

Menghitung banyaknya puncak dengan menggunakan len() yaitu panjang atau banyaknya anggota dari list masing-masing.

Pada 4 baris teratas, menampilkan banyaknya puncak pada Sumbu X, Y, Z pada percobaan nomor 3 masing-masing 25, 27, 27.

Pada 4 baris terakhir, menampilkan banyaknya puncak pada Sudut XY, XZ, YZ pada percobaan nomor 4 masing-masing 26, 24, 27.

(c) Bandingkan hasil perhitungan.

Jika dihitung manual, jumlah pernafasan per menit sebanyak 11.

Untuk cara nomor 3 (direct method) memiliki rata-rata 26.3333

Untuk cara nomor 4 (inklinasi) memiliki rata-rata 25.66667.

Metode yang lebih baik adalah inklinasi karena lebih mendekati nilai aktualnya.

Program lengkap ada di : [https://github.com/Otniel113/loT\\_Pernafasan](https://github.com/Otniel113/loT_Pernafasan)