

# Sistem *Counter Chin-Up* dan *Pull-Up* Berbasis Sensor MPU6050 dan Mikrokontroler ESP32 dengan Pemantauan *Real-Time* melalui Website

1<sup>st</sup> Ela Diranda Pahira  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

eladirandapahira@student.telkomuniversity.ac.id

2<sup>nd</sup> Inung Wijayanto  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

iwijayanto@telkomuniversity.ac.id

3<sup>rd</sup> Sugondo Hadiyoso  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

sugondo@telkomuniversity.ac.id

**Abstrak** — Pada era digital saat ini, penggunaan teknologi berbasis sensor dan mikrokontroler untuk monitoring aktivitas fisik semakin populer. Penelitian ini bertujuan untuk mengembangkan sebuah sistem counter chin-up dan pull-up berbasis sensor MPU6050 dan mikrokontroler ESP32 yang memungkinkan pemantauan gerakan secara real-time melalui website. Sensor MPU6050 digunakan untuk mendeteksi sudut dan gerakan tubuh saat melakukan chin-up dan pull-up, sedangkan mikrokontroler ESP32 berfungsi sebagai pengolah data serta pengirim hasil pengukuran ke server menggunakan protokol Bluetooth Low Energy (BLE). Data yang diterima dari sensor diproses oleh ESP32 dan kemudian dikirimkan ke server untuk diolah lebih lanjut dan ditampilkan pada website menggunakan framework Django. Sistem ini memungkinkan pengguna untuk memantau performa latihan mereka secara langsung melalui antarmuka web yang responsif. Hasil penelitian menunjukkan bahwa sistem ini mampu mendeteksi gerakan dengan akurat dan menampilkan data secara real-time, sehingga dapat membantu pengguna dalam meningkatkan kualitas latihan fisik chin-up dan pull-up

**Kata kunci:** Chin-up, Pull-up, Sensor MPU6050, Mikrokontroler ESP32, BLE, Django

## I. PENDAHULUAN

Dalam kegiatan kebugaran dan proses seleksi fisik TNI dan Polri, tes *chin-up* dan *pull-up* merupakan tes yang digunakan untuk mengevaluasi kekuatan dan daya tahan tubuh. Namun, metode perhitungan yang umumnya dilakukan secara manual sering kali tidak konsisten dan rawan terhadap kesalahan manusia. Pada penelitian ini, dirancang sebuah sistem *counter* berbasis sensor MPU6050 dan mikrokontroler ESP32 yang dapat mendeteksi dan menghitung jumlah gerakan *chin-up* dan *pull-up* secara otomatis. Data yang dikumpulkan oleh sensor MPU6050, dikirimkan ke mikrokontroler ESP32, diproses, dan kemudian ditampilkan secara *real-time* melalui sebuah *website* berbasis Django.

## II. KAJIAN TEORI

### A. Sensor MPU6050

MPU6050 adalah sensor yang menggabungkan *accelerometer* dan *gyroscope*, digunakan untuk mendeteksi orientasi dan gerakan dalam tiga sumbu (X, Y, Z). Sensor ini dapat mengukur percepatan linear dan kecepatan rotasi, yang penting untuk berbagai aplikasi seperti pengukuran sudut dalam penelitian ini

### B. Mikrokontroler ESP32

ESP32 adalah mikrokontroler dengan kemampuan komunikasi Bluetooth Low Energy (BLE) dan Wi-Fi, serta performa komputasi yang cukup tinggi. ESP32 digunakan untuk memproses data dari sensor dan mengirimkan hasilnya ke server secara *real-time*.

### C. Backend Website dengan Django

Django adalah *framework* web yang digunakan untuk mengembangkan *backend* aplikasi. Django Channels mendukung komunikasi antara mikrokontroler dengan web, dan melalui WebSockets, pemantauan data dapat ditampilkan secara langsung dan *real-time* di *website*.

## III. METODE

### A. Sistem Counter Menggunakan MPU6050 dan ESP32

Sistem *counter* ini menggunakan sensor MPU6050 yang terhubung dengan ESP32 untuk mendeteksi dan menghitung gerakan *pull-up* dan *chin-up*. Metode yang digunakan dalam pengolahan data sensor adalah sebagai berikut:

#### 1. Pengukuran Sudut Menggunakan Accelerometer:

Data percepatan dari *accelerometer* pada sumbu Y dan Z digunakan untuk memperkirakan sudut pada sumbu X dengan rumus:

$$\text{accelAngleX} = \arctan\left(\frac{\text{accelY}}{\text{accelZ}}\right) \times \frac{180}{\pi} \quad (1)$$

Rumus ini digunakan untuk menghitung sudut kemiringan pada sumbu X.

## IV. HASIL DAN PEMBAHASAN

### 2. Pengukuran Sudut Menggunakan *Gyroscope*:

*Gyroscope* memberikan data kecepatan rotasi yang diintegrasikan terhadap waktu untuk menghitung sudut:

$$\text{angleX} += \text{gyroX} \times dt \quad (2)$$

Di mana  $\text{gyroX}$  adalah kecepatan rotasi pada sumbu X, dan  $dt$  adalah perubahan waktu antara dua pengukuran.

### 3. Penggabungan Data dengan *Complementary Filter*:

Data dari *accelerometer* dan *gyroscope* digabungkan menggunakan *complementary filter* untuk menghasilkan sudut yang lebih stabil:

$$\text{angleX} = 0.98 \times \text{angleX} + 0.02 \times \text{accelAngleX} \quad (3)$$

Filter ini memberikan bobot lebih besar pada data dari *gyroscope* untuk mengurangi noise.

### 4. Pengukuran *Roll* dan *Pitch*:

Sudut *roll* dan *pitch* dihitung menggunakan data percepatan dari *accelerometer*:

$$\text{roll} = \arctan\left(\frac{\text{accelY}}{\text{accelZ}}\right) \times \frac{180}{\pi} \quad (4)$$

$$\text{pitch} = \arctan\left(-\text{accelX}, \sqrt{\text{accelY}^2 + \text{accelZ}^2}\right) \times \frac{180}{\pi} \quad (5)$$

Rumus ini digunakan untuk menghitung orientasi sensor dalam ruang tiga dimensi.

### 5. Implementasi Logika Penghitungan:

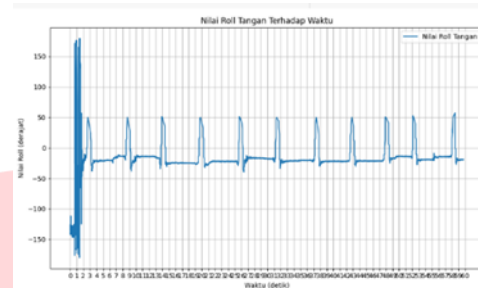
Dalam sistem ini, terdapat dua *client*: *client 1* bertugas mendeteksi gerakan tangan, dan *client 2* bertugas mendeteksi gerakan kaki. ESP32 bertanggung jawab sebagai pusat pengolahan data, di mana data yang diperoleh dari *client 1* dan *client 2* dihitung dan dianalisis untuk menentukan apakah gerakan memenuhi kriteria sebagai *pull-up* atau *chin-up*. Jika kriteria terpenuhi, *counter* akan bertambah, dan hasilnya akan dikirimkan ke server *website* Django melalui Bluetooth Low Energy (BLE).

### B. Implementasi *Backend Website*

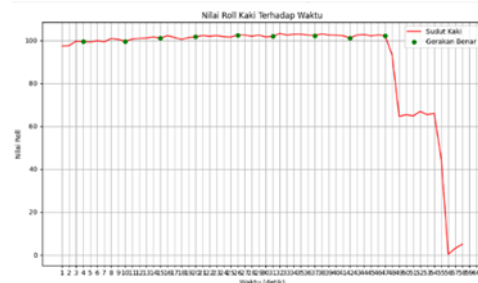
Dalam implementasi sistem ini, Django digunakan sebagai *framework backend* untuk mengelola komunikasi data secara *real-time* antara sensor dan aplikasi web. Alamat MAC sensor dan UUID BLE dari mikrokontroler dan sensor harus disimpan di dalam file Django agar server dapat secara tepat mengidentifikasi dan menghubungkan perangkat sensor yang sesuai. Fungsi seperti `async def send_command(command)` digunakan untuk mengirim perintah dari server ke sensor melalui koneksi BLE, dan kemudian sensor akan menerima dan memproses instruksi dengan benar. Kemudian, untuk mendapatkan koneksi yang *real-time* digunakan *WebSocket*, yang memungkinkan server untuk berkomunikasi langsung dengan klien web tanpa perlu me-refresh halaman, sehingga data dari sensor yang diterima melalui BLE dapat diperbarui secara langsung pada antarmuka web.

### A. Hasil Pembacaan Sudut Oleh Sensor

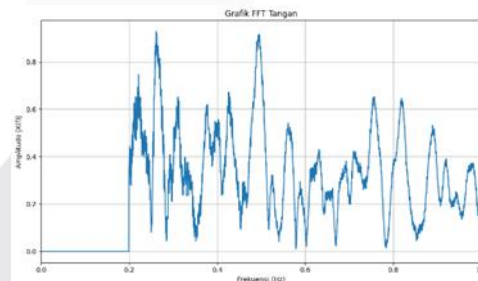
Pada pengujian, sensor terpasang di tangan dan kaki dan melakukan pembacaan sudut *roll* dan *pitch*. Pengujian dilakukan dalam beberapa skenario percobaan gerakan. Grafik di bawah ini menunjukkan hasil pembacaan sudut *roll* dan *pitch* dari salah satu skenario pengujian, yaitu gerakan *pull-up* dengan interval waktu setiap 4 detik selama 1 menit.



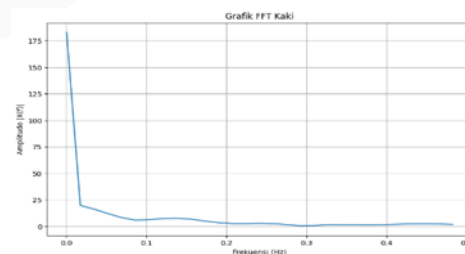
GAMBAR 1  
(Grafik Pembacaan Sudut Tangan)



GAMBAR 2  
(Grafik Pembacaan Sudut Kaki)



GAMBAR 3  
(Grafik FFT Sudut Tangan)



GAMBAR 4  
(Grafik FFT Sudut Kaki)

Hasil pengujian menunjukkan bahwa pembacaan sudut yang dilakukan oleh sensor sudah akurat, terlihat dari sudut

sudut yang tercapai pada grafik. *Marker* hijau pada gambar 1 dan gambar 2 menunjukkan bahwa gerakan tangan dan kaki sudah mencapai sudut *threshold* yang telah ditetapkan. Pada beberapa titik waktu tertentu, saat sudut yang terbaca oleh sensor mencapai *threshold*, sistem akan mendeteksi gerakan tersebut sebagai gerakan yang benar. Sebaliknya, jika sudut tidak memenuhi *threshold*, gerakan tersebut akan teridentifikasi sebagai gerakan yang salah. Pada gambar 2, terdapat gelombang dengan fluktuasi besar di detik-detik terakhir pergerakan yang menunjukkan bahwa kaki berada di luar *threshold* yang telah ditetapkan.

Gambar 3 dan 4 di atas menunjukkan bagaimana distribusi frekuensi dari sudut *roll* dan *pitch* selama pergerakan. Pada gambar 3 yang menunjukkan hasil FFT untuk sudut tangan, terlihat adanya variasi frekuensi yang signifikan dalam rentang sekitar 0,2 hingga 0,6 Hz. Nilai amplitudo yang mencapai hingga 0,8 pada frekuensi sekitar 0,25 Hz menunjukkan bahwa gerakan tangan cukup bervariasi dan berulang dengan pola tertentu sesuai dengan interval waktu pergerakan yang telah ditentukan. Sebaliknya, pada gambar 4 yang menampilkan FFT untuk sudut kaki, sebagian besar energi sinyal terpusat pada frekuensi sangat rendah di bawah 0,1 Hz yang disebabkan oleh *noise* pada awal persiapan gerakan. Setelahnya, gelombang menunjukkan bahwa gerakan kaki cenderung lebih stabil dan memiliki fluktuasi frekuensi yang lebih sedikit dibandingkan tangan. Hal ini sesuai dengan aturan bahwa kaki tidak boleh terlalu berayun selama pergerakan, sehingga grafik kaki lebih stabil. Secara keseluruhan, pergerakan kaki tampak lebih terkontrol dan kurang bervariasi, sementara tangan mengalami perubahan frekuensi yang lebih dinamis.

#### B. Hasil Perhitungan Sistem dan Manual

Pengujian selanjutnya dilakukan dengan melakukan beberapa percobaan gerakan dan kemudian membandingkan hasil perhitungan yang terbaca oleh sistem yang ada di website dengan perhitungan yang dilakukan secara manual. Dalam pengujian ini, responden melakukan *chin-up* dan *pull-up* dengan beberapa scenario gerakan.

TABEL 1  
(Hasil Perhitungan *Chin-Up* oleh Sistem dan Manual)

Gerakan <i>Chin-Up</i>				
Percobaan	Frekuensi Waktu Gerakan	Hitungan Sistem	Hitungan Manual	Error (Perbedaan)
Percobaan 1	2 detik selama 1 menit	19	19	0
Percobaan 2	4 detik selama 1 menit	10	11	-1
Percobaan 3	Bebas/sebanyak-banyaknya selama 1 menit	29	29	0
Percobaan 4	Bebas/sebanyak-banyaknya selama 30 detik	25	25	0
Percobaan 5	Bebas/sebanyak-banyaknya selama 30 detik	20	20	0
Percobaan 6	Bebas/sebanyak-banyaknya selama 30 detik	40	40	0
Percobaan 7	Bebas/sebanyak-banyaknya selama 30 detik	25	25	0
Percobaan 8	Bebas/sebanyak-banyaknya selama 30 detik	33	33	0

TABEL 2  
(Hasil Perhitungan *Pull-Up* oleh Sistem dan Manual)

Gerakan <i>Pull-Up</i>				
Percobaan	Frekuensi Waktu Gerakan	Hitungan Sistem	Hitungan Manual	Error (Perbedaan)
Percobaan 1	4 detik selama 1 menit	9	9	0
Percobaan 2	6 detik selama 1 menit	7	7	0
Percobaan 3	Bebas/Sebanyak-banyaknya selama 1 menit	12	13	-1
Percobaan 4	Bebas/Sebanyak-banyaknya selama 30 detik	5	5	0
Percobaan 5	Bebas/Sebanyak-banyaknya selama 30 detik	1	1	0
Percobaan 6	Bebas/Sebanyak-banyaknya selama 30 detik	6	6	0
Percobaan 7	Bebas/Sebanyak-banyaknya selama 30 detik	6	6	0
Percobaan 8	Bebas/Sebanyak-banyaknya selama 30 detik	7	7	0

Hasil perbandingan antara perhitungan gerakan *chin-up* dan *pull-up* menggunakan sistem sensor dan perhitungan manual menunjukkan bahwa sistem sensor yang digunakan memiliki tingkat akurasi yang baik. Hanya terdapat 1 perbedaan pada percobaan *chin-up* dan *pull-up*. Pada gerakan *chin-up*, percobaan dengan interval waktu 2 detik dan percobaan dengan interval waktu bebas mencatat jumlah gerakan yang lebih banyak dibandingkan dengan percobaan yang menggunakan interval waktu 4 detik. Hal ini menunjukkan bahwa sistem sensor mampu mendeteksi gerakan dengan baik, bahkan ketika gerakan dilakukan dengan frekuensi yang lebih cepat.

Hal yang serupa juga terlihat pada gerakan *pull-up*. Percobaan dengan interval waktu 4 detik dan percobaan dengan interval waktu bebas detik menunjukkan jumlah gerakan yang lebih banyak dibandingkan dengan percobaan yang menggunakan interval waktu 6 detik. Sistem sensor tetap mampu mengikuti kecepatan gerakan yang lebih tinggi dan mencatat jumlah gerakan dengan akurasi yang sama dengan perhitungan manual. Secara keseluruhan, hasil perbandingan ini mengindikasikan bahwa sistem sensor memiliki kemampuan yang cukup akurat dalam mendeteksi gerakan *chin-up* dan *pull-up*, dengan performa yang tetap baik meskipun gerakan dilakukan dalam interval waktu lebih cepat, di mana jumlah gerakan yang terdeteksi cenderung lebih banyak.

#### V. KESIMPULAN

Sensor MPU6050 yang digunakan berhasil mendeteksi sudut-sudut pergerakan dengan tepat dan akurat menggunakan kombinasi data *accelerometer* dan *gyroscope*. Sensor juga dapat mendeteksi sudut-sudut dalam berbagai kondisi pergerakan, baik cepat maupun lambat. Data tersebut juga berhasil dikirimkan ke ESP32 dan kemudian diteruskan ke web Django. Namun, dari pengujian yang telah dilakukan, terdapat satu perhitungan yang terbaca oleh sensor dan ESP32 tetapi tidak tersampaikan ke sistem *website* melalui BLE. Hal ini menunjukkan bahwa meskipun sensor dan ESP32 bekerja dengan baik, terdapat potensi kendala dalam

transmisi data ke *website* yang perlu diperhatikan dan diatasi. Meskipun demikian, secara keseluruhan, alat ini telah berfungsi dengan baik dalam mendeteksi dan menghitung gerakan *chin-up* dan *pull-up* secara *real-time*.

#### REFERENSI

- [1] A. H. Kurniawan and M. Rivai, "Sistem Stabilisasi Naman Menggunakan IMU Sensor dan Arduino Nano," *Jurnal Teknik ITS*, pp. 270–275, 2018.
- [2] InvenSense Inc. (2013). *MPU-6000 and MPU-6050 Product Specification* (Revision 3.4). Diambil dari <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [3] Andrianto, H. & Darmawan, A. (2017). *Arduino Belajar Cepat dan Pemrograman*. Bandung: Informatika
- [4] Nugroho, H. (2023). Perancangan Node Sensor Nirkabel Bluetooth Low Energy Bertenaga Baterai menggunakan ESP32 untuk Aplikasi Pertanian Cerdas. *TELEKONTRAN*, 11(1), 1-10.