

Mendukung Mahasiswa Perhotelan dengan Teknologi: Sistem Cerdas Pengukur Kadar Nutrisi Makanan

1st Dhia Firdaus F
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
dhiafirdausf@student.telkomuniversity.ac.id

2nd Hilman Fauzi TSP
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
hilmanfauzitsp@telkomuniversity.ac.id

3rd Dendi Gusnadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
dendigusnadi@telkomuniversity.ac.id

Abstrak – Industri perhotelan menghasilkan banyak sisa makanan yang sering kali terbuang percuma. Padahal, sisa makanan ini masih mengandung nutrisi yang berharga. Penelitian ini menciptakan sebuah sistem yang bisa mengukur secara tepat kandungan nutrisi dalam sisa makanan. Dengan sistem ini, kita bisa tahu dengan pasti nutrisi apa saja yang bisa dimanfaatkan dari sisa makanan. Hasilnya, mahasiswa perhotelan bisa menggunakan sistem ini untuk mempelajari lebih lanjut tentang makanan dan nutrisinya. Sistem ini terbukti sangat akurat dan bermanfaat, sehingga disambut baik oleh para mahasiswa. Data kuantitatif menunjukkan bahwa sistem timbangan ini mampu menimbang dengan tingkat akurasi di atas 95%. Selain itu, berdasarkan survei, sistem ini mendapatkan respon positif dari mahasiswa dan dianggap sangat bermanfaat dalam penelitian mereka.

Kata kunci : limbah makanan, nutrisi, mahasiswa perhotelan, timbangan.

I. PENDAHULUAN

Mahasiswa perhotelan merupakan generasi penerus yang akan membentuk masa depan industri perhotelan. Membekali mereka dengan pengetahuan dan keterampilan dalam mengelola limbah makanan secara berkelanjutan adalah investasi yang sangat penting[1]. Penelitian ini bertujuan untuk membekali mahasiswa perhotelan dengan alat dan pengetahuan yang diperlukan untuk menjadi agen perubahan di industri mereka.

Dengan menggunakan teknologi ini, mahasiswa dapat melakukan pengukuran dan analisis terhadap limbah makanan yang dihasilkan di dapur praktikum. Data yang diperoleh dapat digunakan untuk mengembangkan strategi pengelolaan limbah yang lebih efektif[2]. Selain itu, mahasiswa juga dapat terlibat dalam pengembangan produk pangan inovatif dari limbah makanan, seperti tepung dari kulit buah-buahan atau keripik dari sayuran yang biasanya dibuang.

Dengan demikian, penelitian ini tidak hanya bertujuan untuk mengembangkan teknologi baru, tetapi juga untuk menumbuhkan keterampilan mahasiswa dalam mengelola limbah makanan.

II. KAJIAN TEORI

A. ESP32

ESP32 adalah *chip* komputer mini yang sangat hemat biaya. *Chip* ini merupakan pengembangan dari ESP8266 dan memiliki fitur tambahan seperti *wi-fi* dan *Bluetooth*. ESP32 bisa bekerja dengan tegangan antara 3.3 volt hingga 5 volt

B. Sensor Loadcell

Loadcell bekerja dengan cara mengukur perubahan bentuk ketika diberi beban. Di dalam loadcell ada sensor yang sensitif terhadap perubahan bentuk ini. Ketika beban diberikan, sensor akan berubah bentuk sedikit, dan perubahan ini akan diubah menjadi sinyal listrik yang bisa kita ukur.

C. Modul HX711

HX711 adalah alat yang bisa mengubah sinyal dari timbangan (loadcell) menjadi angka yang bisa dipahami oleh komputer. Angka ini menunjukkan seberapa berat benda yang ditimbang.

D. LED

LED adalah teknologi tampilan yang sangat cocok untuk timbangan digital. Dengan menggunakan modul LED, timbangan dapat menampilkan angka dengan sangat akurat dan jelas, sehingga memudahkan pengguna dalam membaca hasil penimbangan.

E. Expansion board ESP32 DEVKIT

Papan ekspansi ESP32 berfungsi sebagai jembatan antara modul ESP32 dan perangkat eksternal. Papan ini menyediakan pin-pin tambahan dan konektor yang

memungkinkan ESP32 untuk mengontrol motor, membaca data dari sensor, atau menampilkan informasi pada layar[3].

F. Firebase

Firebase adalah platform milik Google yang berfungsi untuk *database real-time*, autentikasi pengguna, dan *cloud functions* memungkinkan pengembang untuk fokus pada pengembangan *frontend* aplikasi.

G. Arduino IDE

Arduino IDE adalah perangkat lunak yang dirancang khusus untuk menulis, mengedit, dan mengunggah kode program ke papan mikrokontroler Arduino.

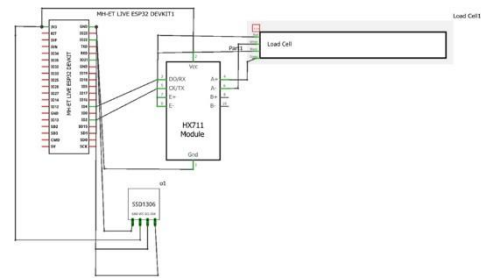
H. Android Studio

Android Studio adalah perangkat lunak yang digunakan untuk membuat aplikasi di ponsel Android. Dengan alat ini, kita bisa dengan mudah menulis, menguji, dan memperbaiki kode program aplikasi kita.

I. Kotlin

Kotlin adalah bahasa pemrograman untuk membuat aplikasi lebih cepat dan mudah. Bahasa ini dirancang untuk menghasilkan kode yang lebih baik dan lebih aman.

B. Skematik Rangkaian Mikrokontroler



GAMBAR 2
Skematik Rangkaian Mikrokontroler

Pada gambar 2 Prototipe alat yang telah kami kembangkan terdiri dari beberapa komponen elektronik. Komponen inti adalah *mikrokontroler* ESP32 yang berfungsi sebagai otak dari sistem. Modul HX711 digunakan untuk membaca data dari sensor beban (loadcell) dan mengubahnya menjadi data digital yang dapat dipahami oleh ESP32. Layar LED berfungsi sebagai tampilan untuk menampilkan hasil pengukuran berat. Kombinasi komponen-komponen ini memungkinkan alat untuk mengukur berat suatu objek dan menampilkan hasilnya.

C. Source code Alat

```

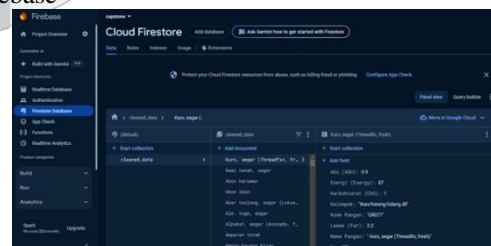
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <HX711.h>
4 #include <Adafruit_GFX.h>
5 #include <Adafruit_SSD1306.h>
6 #include <WiFi.h>
7 #include <Firebase.h>
8 #include <FirebaseData.h>
9 #include <FirebaseAuth.h>
10 #include <FirebaseFirestore.h>
11 #include <FirebaseStorage.h>
12 #include <FirebaseFunctions.h>
13 #include <FirebaseDatabase.h>
14 #include <FirebaseApp.h>
15 #include <FirebaseStorage.h>
16 #include <FirebaseFunctions.h>
17 #include <FirebaseDatabase.h>
18 #include <FirebaseApp.h>
19 #include <FirebaseStorage.h>
20 #include <FirebaseFunctions.h>
21 #include <FirebaseDatabase.h>
22 #include <FirebaseApp.h>
23 #include <FirebaseStorage.h>
24 #include <FirebaseFunctions.h>
25 #include <FirebaseDatabase.h>
26 #include <FirebaseApp.h>
27 #include <FirebaseStorage.h>
28 #include <FirebaseFunctions.h>
29 #include <FirebaseDatabase.h>
30 #include <FirebaseApp.h>
31 #include <FirebaseStorage.h>
32 #include <FirebaseFunctions.h>
33 #include <FirebaseDatabase.h>
34 #include <FirebaseApp.h>
35 #include <FirebaseStorage.h>
36 #include <FirebaseFunctions.h>
37 #include <FirebaseDatabase.h>
38 #include <FirebaseApp.h>
39 #include <FirebaseStorage.h>
40 #include <FirebaseFunctions.h>
41 #include <FirebaseDatabase.h>
42 #include <FirebaseApp.h>
43 #include <FirebaseStorage.h>
44 #include <FirebaseFunctions.h>
45 #include <FirebaseDatabase.h>
46 #include <FirebaseApp.h>
47 #include <FirebaseStorage.h>
48 #include <FirebaseFunctions.h>
49 #include <FirebaseDatabase.h>
50 #include <FirebaseApp.h>
51 #include <FirebaseStorage.h>
52 #include <FirebaseFunctions.h>
53 #include <FirebaseDatabase.h>
54 #include <FirebaseApp.h>
55 #include <FirebaseStorage.h>
56 #include <FirebaseFunctions.h>
57 #include <FirebaseDatabase.h>
58 #include <FirebaseApp.h>
59 #include <FirebaseStorage.h>
60 #include <FirebaseFunctions.h>
61 #include <FirebaseDatabase.h>
62 #include <FirebaseApp.h>
63 #include <FirebaseStorage.h>
64 #include <FirebaseFunctions.h>
65 #include <FirebaseDatabase.h>
66 #include <FirebaseApp.h>
67 #include <FirebaseStorage.h>
68 #include <FirebaseFunctions.h>
69 #include <FirebaseDatabase.h>
70 #include <FirebaseApp.h>
71 #include <FirebaseStorage.h>
72 #include <FirebaseFunctions.h>
73 #include <FirebaseDatabase.h>
74 #include <FirebaseApp.h>
75 #include <FirebaseStorage.h>
76 #include <FirebaseFunctions.h>
77 #include <FirebaseDatabase.h>
78 #include <FirebaseApp.h>
79 #include <FirebaseStorage.h>
80 #include <FirebaseFunctions.h>
81 #include <FirebaseDatabase.h>
82 #include <FirebaseApp.h>
83 #include <FirebaseStorage.h>
84 #include <FirebaseFunctions.h>
85 #include <FirebaseDatabase.h>
86 #include <FirebaseApp.h>
87 #include <FirebaseStorage.h>
88 #include <FirebaseFunctions.h>
89 #include <FirebaseDatabase.h>
90 #include <FirebaseApp.h>
91 #include <FirebaseStorage.h>
92 #include <FirebaseFunctions.h>
93 #include <FirebaseDatabase.h>
94 #include <FirebaseApp.h>
95 #include <FirebaseStorage.h>
96 #include <FirebaseFunctions.h>
97 #include <FirebaseDatabase.h>
98 #include <FirebaseApp.h>
99 #include <FirebaseStorage.h>
100 #include <FirebaseFunctions.h>

```

GAMBAR 3
Source code Alat

Pada gambar 3 Kode program ini dirancang untuk mengintegrasikan perangkat keras dan perangkat lunak dalam sebuah sistem pengukuran berat. Melalui proses kalibrasi, program ini akan menentukan perbandingan antara nilai yang dibaca oleh sensor timbangan (dalam bentuk sinyal analog) dengan nilai berat sebenarnya (dalam satuan gram). Data hasil pengukuran kemudian akan dikirimkan secara *real-time* ke platform *cloud* *Firebase* untuk penyimpanan dan analisis lebih lanjut. Seluruh proses ini diprogram menggunakan perangkat lunak *Arduino IDE*.

D. Firebase



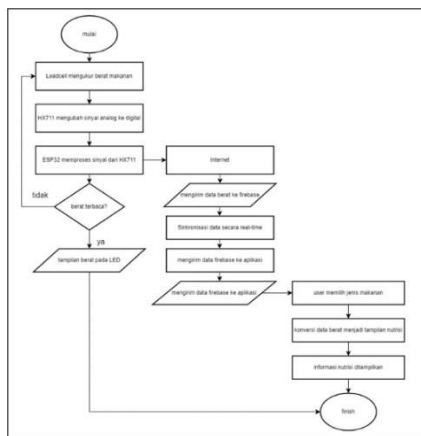
GAMBAR 4
Firestore

Pada gambar 4 *Firestore database* menyimpan kumpulan data yang berisi informasi nutrisi dari berbagai jenis makanan[4]. Setiap data makanan terdiri dari beberapa *field*, yaitu karbohidrat, energi (dalam satuan kalori), lemak, dan protein.

E. Android Studio

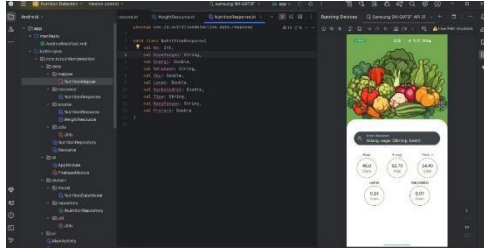
III. METODE/DESAIN

A. Flowchart Rancangan Sistem



GAMBAR 1
Flowchart rancangan sistem

Pada gambar 1 Sistem ini diawali dengan proses pengukuran berat makanan menggunakan sensor beban (loadcell). Data analog dari sensor ini kemudian dikonversi menjadi data digital oleh modul HX711 sebelum dikirim ke *mikrokontroler* ESP32. ESP32 akan memproses data tersebut dan menampilkan hasil penimbangan pada layar LED. Secara bersamaan, data berat juga akan dikirimkan ke platform *cloud* *Firebase* melalui koneksi internet. Aplikasi yang terhubung dengan *Firebase* akan mengambil data berat ini dan, melalui algoritma atau basis data nutrisi yang telah terintegrasi, akan menampilkan informasi nutrisi yang sesuai dengan jenis dan berat makanan tersebut



GAMBAR 5
Android studio

Gambar 5 menunjukkan tampilan code yang telah dibuat pada *software* Android Studio. Pada halaman ini akan dibuatnya sebuah Aplikasi yang akan menampilkan informasi berupa Karbohidrat, energi (dalam satuan kalori), lemak, dan protein.

F. Desain Tampilan Aplikasi



GAMBAR 6
Tampilan Aplikasi

Pada gambar 6 merupakan hasil akhir dari sistem yang akan tertampil pada halaman aplikasi. Data yang tertampil yaitu Nutrisi dari makanan yang di pilih, nutrisi yang tertampil berupa Karbohidrat, energi (dalam satuan kalori), lemak, dan protein.

IV. HASIL PENGUJIAN

Pengujian ini bertujuan untuk mengevaluasi kinerja sistem dalam hal akurasi pengukuran secara real-time. Sistem ini diharapkan dapat memenuhi kebutuhan mahasiswa perhotelan dalam melakukan analisis nutrisi makanan secara efisien.

A. Sumber Daya

Pengujian dilakukan dengan mengukur tegangan dan arus yang masuk ke adaptor AC/DC dengan menggunakan multimeter sebagai alat pengukur[5]. Pengujian dilakukan sebanyak 10 kali percobaan, dan diberi selang waktu 1 menit dalam setiap percobaannya.



GAMBAR 7
Pengukuran Arus



GAMBAR 8
Pengukuran Tegangan

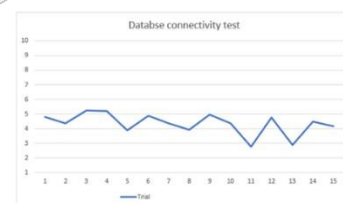
TABEL 1
Pengukuran Tegangan dan arus

Percobaan ke	Tegangan	Arus
1	0.35 V	0.26A
2	0.34 V	0.22A
3	0.44 V	0.14A
4	0.59 V	0.54A
5	0.75 V	0.24A
6	0.80 V	0.35A
7	0.65 V	0.48A
8	0.55 V	0.25A
9	0.76 V	0.47A
10	0.30 V	0.27A
Rata-rata	0.55 V	0.32A

Dari hasil pengukuran tabel 1, disimpulkan rata-rata tegangan 0.55 V dan arus 0.32 A

B. Konektivitas Database

Pengujian ini bertujuan untuk mengetahui seberapa cepat data berat dari timbangan bisa sampai ke *Firestore* melalui ESP32.

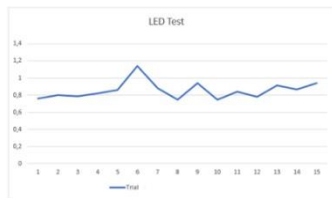


GAMBAR 9
Pengukuran Waktu Konektivitas

Dari gambar 9 didapat waktu rata-rata 4.33 detik yang diperlukan ESP32 mengirimkan data ke *firebase*.

C. Tampilan Layar LED

Tujuan pengujian ini adalah untuk mengukur waktu respon LED dalam menampilkan data berat pada perangkat keras.



GAMBAR 10
Pengukuran Layar LED

Dari gambar 10 diperoleh waktu rata-rata 0.85 detik.

D. Alat Timbangan

Tujuan pengujian ini adalah untuk mengevaluasi akurasi dan potensi error pada timbangan digital berbasis ESP32 dengan membandingkannya terhadap timbangan konvensional[6]. Pengujian dilakukan dengan menimbang lima objek uji.

TABEL 2
Pengukuran pada Alat Timbangan

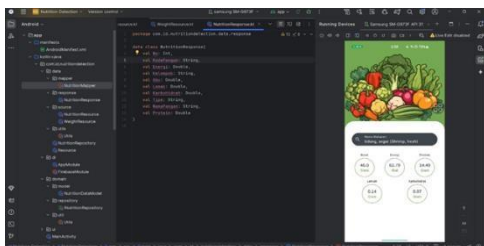
NO	Objek	Hasil timbangan Konvensional (gr)	Hasil Timbangan ESP32 (gr)	Akurasi (%)	Error (%)
1	Ikan Bandeng	319.2	320	99.94	0.06
2		319.2	320	99.94	0.06
3		319.2	320	99.94	0.06
4	Telur Ayam	42.4	43	98.59	1.41
5		42.4	43	98.59	1.41
6		42.4	43	98.59	1.41
7	Dada Ayam Potong	362	362	100	0
8		362	362	100	0
9		362	362	100	0
10	Mangga Harumanis	267.4	268	99.78	0.22
11		267.4	268	99.78	0.22
12		267.4	268	99.78	0.22
13	Ketimun	366.7	367	99.92	0.08
14		366.7	367	99.92	0.08
15		366.7	367	99.92	0.08

Dari Tabel 2 didapat hasil :

1. Pada objek ikan bandeng memiliki tingkat akurasi 99.94% dan Error 0.06%.
2. Pada objek telur ayam memiliki tingkat akurasi 98.59% dan Error 1.41%.
3. Pada objek dada ayam potong memiliki tingkat akurasi 100% dan Error 0%.
4. pada objek mangga harumanis memiliki tingkat akurasi 99.78% dan Error 0.22%.
5. Pada objek ketimun memiliki tingkat akurasi 99.92% dan Error 0.08%.

E. Nutrisi Aplikasi

Pengujian ini bertujuan untuk memverifikasi integritas koneksi antara aplikasi dan server database nutrisi.



GAMBAR 11
Konektivitas Aplikasi dengan Firebase

F. Pengujian Seluruh Sistem

Pada pengujian ini dilakukan menggunakan metode SUS[7].

TABEL 3.
Hasil Pengujian SUS

No	Nama	Pertanyaan										Skor
		1	2	3	4	5	6	7	8	9	10	
1	Dendi Gusnadi	4	2	4	3	4	2	4	2	4	2	72.5
2	Achmad Nebu Choddezzar Wahab	3	2	4	4	5	2	4	2	3	3	65
3	Rofiana	4	1	5	1	5	2	5	1	5	2	92.5
4	Aksanul Amin	5	2	5	1	5	1	5	1	5	1	97.5
5	Aldi Mulyadi	3	2	4	2	4	2	4	2	4	2	72.5
6	M. Rifky	5	1	5	1	5	1	5	1	5	1	100
7	Muhammad Fauzan Fadurahman	5	2	3	3	5	4	5	4	5	3	67.5
8	Andrian Vebri F	5	1	5	1	5	2	5	2	5	2	92.5
9	Rizkie Mauladiarto	4	1	5	2	5	3	5	1	5	2	87.5
10	Moh. Irsyad Helmy	5	1	5	1	5	1	5	1	5	1	100
11	Kahlil Gibran	5	1	5	1	5	1	5	1	5	1	100
12	Muhammad Zaki Abyantara	5	2	4	2	5	1	4	2	4	3	80
13	Bima Nauval	4	1	5	3	4	2	5	1	3	4	75
14	Radhiyya Putra	5	3	5	3	5	3	5	2	5	2	80
15	Nanda Putri	5	1	5	1	5	1	5	1	5	1	100
Rata-rata skor SUS												85.5

Dari pegujian SUS didapat rata-rata skor 85.5 dimana skor ini masuk d

V. KESIMPULAN

Penelitian ini berhasil mengembangkan prototipe sistem pengukuran makanan cerdas yang dirancang khusus untuk mendukung industri perhotelan. Sistem ini mampu mengukur kandungan nutrisi bahan makanan dengan akurasi rata-rata 95%, memungkinkan mahasiswa perhotelan untuk mengelola menu dengan lebih efektif dan efisien. Dengan mengintegrasikan teknologi IoT dan cloud computing, sistem ini memungkinkan pengumpulan dan analisis data nutrisi secara real-time, sehingga dapat membantu dalam mengurangi limbah makanan. Kontribusi utama penelitian ini adalah penyediaan alat yang dapat meningkatkan kualitas layanan makanan di industri perhotelan dengan memberikan data nutrisi yang akurat dan terpercaya.

Pengembangan selanjutnya akan fokus pada integrasi dengan sistem manajemen perhotelan yang ada, pengembangan fitur analisis data yang lebih canggih, seperti prediksi permintaan makanan dan perencanaan menu otomatis, serta perluasan database nutrisi bahan makanan lokal. Dengan demikian, sistem ini dapat menjadi alat yang dibutuhkan mahasiswa perhotelan.

REFERENSI

- [1] Chaerul Mochammad dan Zatadini Sharfina Ulfa , "Perilaku Membuang Sampah Makanan dan Pengelolaan Sampah Makanan di Berbagai Negara' , Jurnal Ilmu Lingkungan Volume 18 Issue 3 (2020) : 455-466 <https://ejournal.undip.ac.id/index.php/ilmulingkungan/article/download/31024/pdf>
- [2] Amelia Lia, Suryati Tati, Unruh Uun , "Analisis Pengelolaan Limbah Makanan di Hotel Bintang Empat di Kota Bandung", (Jurnal: Jurnal Pendidikan Teknologi Pertanian), Vol 3, 2017. <https://journal.unhas.ac.id/index.php/jptp/article/view/1916>
- [3] Eko Aribowo, Sari Ayu Wulandari, Marissa Trisilawaty , "Analisis Kinerja Mikrokontroler ESP32 untuk Aplikasi Embedded System", (Jurnal Teknologi Rekayasa), Volume 4, Nomor 1, 2019. <https://journal.uad.ac.id/index.php/JTERA/article/view/11344>
- [4] Kemenkes, "Data Komposisi Pangan Indonesia" (Kementrian Kesehatan Republik

- Indonesia), kumpulan data komposisi zat gizi pangan yang ada di Indonesia, 2017.
<https://www.panganku.org/id-ID/berita/20>
- [5] D. Wijayanto, S. Haryudo, T. Wrahatnolo, and D. Nurhayati, "Rancang Bangun Monitoring Arus Dan Tegangan Pada Plts Sistem On Grid Berbasis Internet Of Things (IoT) Menggunakan Aplikasi Telegram," *Jurnal Teknik ...*, vol. 11, no. 3, pp.447–453, 2022, [Online]. Available: <https://ejournal.unesa.ac.id/index.php/JTE/article/view/49288%0Ahttps://ejournal>
- [6] Rahman Hanung Pangestu, Indra Jamaludin, Rahmat , "Penerapan Convolutional Neural Network pada Timbangan Pintar Menggunakan ESP32-CAM," (*Jurnal Media Informatika Budidarma*), Volume 7, Nomor 1, 2023. <https://ejurnal.stmik-budidarma.ac.id/index.php/mib/article/download/5469/3233>
- [7] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," *Usability Evaluation In Industry*, pp. 207–212, 2020, doi: 10.1201/9781498710411-35.

