

# Desain dan Implementasi Aplikasi Seluler untuk Monitoring Kondisi Internal Batang Pohon dengan *Acoustic Tomography* Menggunakan Flutter

Tifany Saulina Nababan  
Electrical Engineering  
Bandung Institute of Technology  
Bandung, Indonesia  
13218070@std.stei.itb.ac.id

Indra Sihar  
Electrical Engineering  
Bandung Institute of Technology  
Bandung, Indonesia  
indra.sihar@itb.ac.id

Akhmadi Surawijaya  
Electrical Engineering  
Bandung Institute of Technology  
Bandung, Indonesia  
asurawijaya@itb.ac.id

**Abstrak**— Pemantauan kondisi internal batang pohon secara langsung memerlukan metode yang dapat memberikan informasi terkait struktur dan kesehatan pohon tanpa menimbulkan kerusakan. Salah satu solusi yang efektif adalah dengan menggunakan metode *Acoustic Tomography*, yang mampu mengevaluasi variasi kepadatan pohon sehingga dapat mendeteksi adanya kerusakan struktural, pembusukan, atau retakan internal secara nondestruktif. Untuk mempermudah pengguna dalam mengakses dan menganalisis data, dibutuhkan alat yang memadai, seperti aplikasi seluler dengan antarmuka yang ramah pengguna dan mudah digunakan. Penelitian ini memaparkan pengembangan sebuah aplikasi seluler berbasis Android menggunakan Flutter, yang memungkinkan pengumpulan data pohon, pemrosesan data *Time of Flight* (ToF) yang diterima dari Bluetooth oleh *mainboard*, sampai menampilkan data yang sudah diproses menjadi data kecepatan. Aplikasi ini tidak hanya berfungsi sebagai alat untuk memonitor kondisi internal batang pohon, tetapi juga sebagai antarmuka yang menyajikan data secara visual dengan cara yang mudah dipahami pengguna.

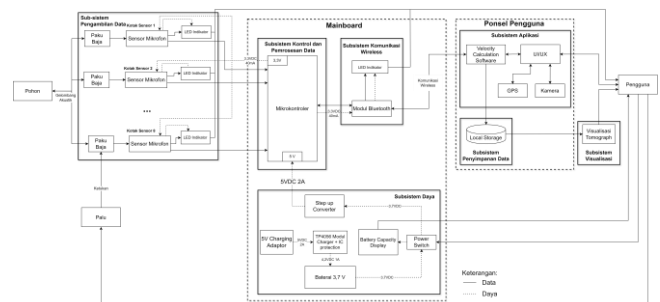
**Keywords**—Acoustic Tomography, aplikasi seluler, Flutter, Time of Flight.

## I. PENDAHULUAN

Kesehatan pohon memiliki peran penting dalam keberlanjutan lingkungan, perencanaan tata kota, dan pengelolaan kehutanan. Pohon berkontribusi terhadap peningkatan kualitas udara, penyerapan karbon, dan pelestarian lingkungan, serta menjadi komponen vital dalam ruang hijau perkotaan [1][2]. Namun, pemantauan kondisi internal batang pohon sering kali sulit dilakukan karena kerusakan seperti pembusukan atau masalah struktural lainnya yang tidak terdeteksi secara visual. Metode *acoustic tomography* memberikan pendekatan nondestruktif dengan memanfaatkan propagasi gelombang suara untuk mendeteksi anomali struktur di dalam batang pohon[3]. Penelitian ini berfokus pada pengembangan aplikasi seluler berbasis Flutter yang memanfaatkan *acoustic tomography* untuk memantau kesehatan internal pohon secara akurat. Aplikasi dirancang untuk mendukung pengumpulan data pohon dan analisis data akustik dari sensor yang terhubung ke *mainboard* melalui koneksi Bluetooth. Fungsi utama aplikasi mencakup perhitungan kecepatan gelombang akustik berdasarkan data *Time of Flight* (ToF), pengambilan data pohon, serta tampilan hasil pengukuran.

Aplikasi yang dibuat dirancang untuk mendukung berbagai fitur utama pengukuran yang dilakukan pengguna, seperti memasukkan data pohon (spesies, lokasi, dan foto), mencatat informasi geometri (jumlah sensor, keliling batang,

dan tinggi pengukuran), serta menampilkan jarak penempatan sensor di sekitar batang pohon. Selain itu, aplikasi memproses data ToF yang diterima dari *mainboard* untuk menghitung kecepatan gelombang dan menghasilkan visualisasi tomogram kondisi internal batang pohon. Data hasil pengukuran dapat disimpan secara lokal pada perangkat pengguna. Adapun pengembangan fungsi visualisasi dan penyimpanan data dilakukan oleh anggota tim lainnya, seperti yang terlihat pada Gambar 1. Tujuan utama proyek ini adalah menyediakan alat bantu yang dapat diakses oleh arboris, dokter pohon, dan peneliti untuk memantau kondisi internal batang pohon secara nondestruktif. Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan pemantauan kondisi internal batang pohon sekaligus mendukung implementasi praktik perencanaan dan pengelolaan hutan berbasis teknologi.



Gambar 1. Desain Arsitektur Alat *Acoustic Tomography*

## II. TINJAUAN PUSTAKA

### A. Flutter Framework

Flutter adalah Software Development Kit (SDK) yang dikembangkan oleh Google untuk membangun aplikasi *cross platform* untuk Android dan iOS, serta sistem operasi Google Fuchsia. Dirilis pada tahun 2017, Flutter menggunakan bahasa pemrograman Dart dan Skia sebagai *graphics engine*. Dengan menggunakan widget sebagai komponen utama, Flutter memungkinkan pengembangan aplikasi dengan Material Design untuk Android dan Cupertino untuk iOS, yang dapat dikompilasi untuk kedua platform tanpa perubahan pada emulator [4].

Salah satu keunggulan utama dari Flutter adalah kemampuannya untuk menghasilkan aplikasi dengan ukuran file yang lebih kecil, waktu buka aplikasi yang lebih cepat, serta penggunaannya yang terus meningkat di kalangan developer. Bahasa pemrograman Flutter, yaitu Dart, mudah dipelajari bagi pengembang yang sudah familiar dengan bahasa pemrograman berorientasi objek seperti C#, C++,

atau Java, menjadikan Flutter lebih mudah digunakan dibandingkan dengan React Native yang lebih bergantung pada bahasa JavaScript yang sudah lebih umum, namun memerlukan pengembang untuk menguasai lebih banyak library pihak ketiga untuk mengakses fitur-fitur perangkat seperti Bluetooth atau Wi-Fi [4 dan 5].

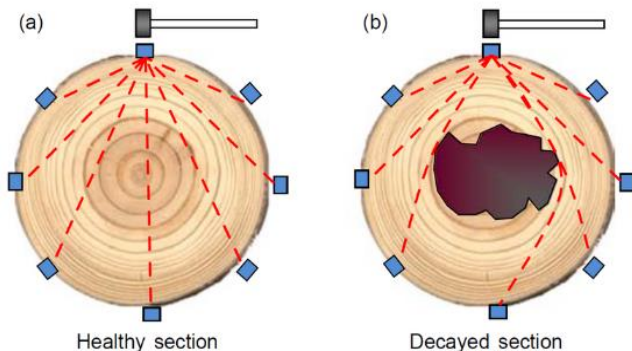
### B. Acoustic Tomography

*Acoustic tomography* merupakan teknik nondestruktif yang digunakan untuk memetakan dan memonitor kondisi internal benda padat dengan menggunakan gelombang akustik. Prinsip dasar dari *acoustic tomography* adalah dengan mengirimkan gelombang suara melalui material, mengukur waktu perjalanan gelombang, kemudian membandingkan hasilnya dengan referensi untuk mendapatkan gambaran tentang keadaan internal material. Kecepatan gelombang akustik dapat dihitung menggunakan jarak antara dua transduser ( $L$ ) sebagai:

$$V = \frac{L}{t} \quad (1)$$

Metode ini mengandalkan perbedaan kecepatan gelombang suara yang merambat melalui material yang memiliki kepadatan atau sifat elastisitas yang berbeda.

Dalam konteks pemantauan kondisi internal batang pohon, teknik ini digunakan untuk mendeteksi perubahan atau kerusakan pada bagian internal pohon yang kemungkinan tidak terlihat dari luar, digunakan untuk investigasi batang dan akar pohon, mulai dari deteksi pelapukan, retakan, rongga ataupun lubang [6], hingga karakterisasi untuk mengevaluasi kualitas kayu [7]. Pengukuran dilakukan seperti pada Gambar 2, di mana 8 buah transduser diletakkan di batang pohon dengan selisih jarak yang sama.



Gambar 2. Ilustrasi metode *Acoustic Tomography* pada batang pohon[8]

Ketukan yang diberikan pengguna akan menghasilkan gelombang akustik yang merambat melalui batang pohon, dengan jalur lintas menyesuaikan kepadatan kayu batang.

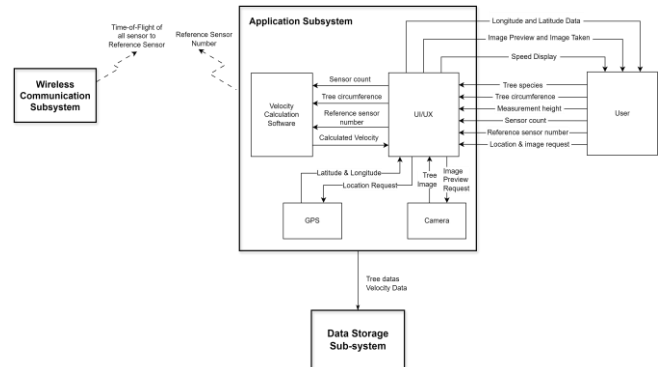
## III. DESAIN

### A. Data dan Alur Navigasi Aplikasi

Untuk mendukung pengukuran alat *acoustic tomography* yang dibuat sebagai alat yang digunakan untuk monitoring kondisi batang pohon, aplikasi dirancang untuk menampung berbagai macam data seperti:

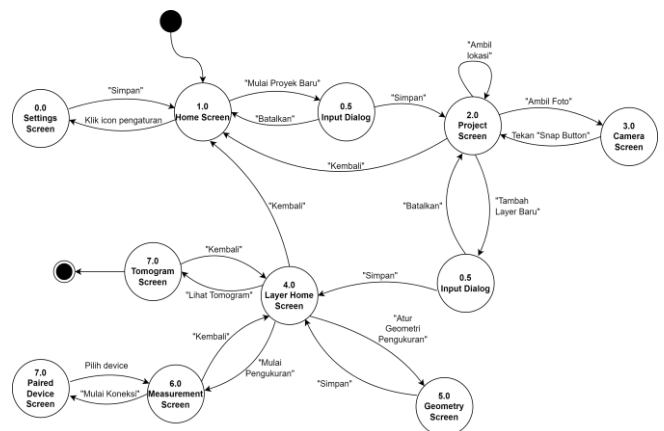
- Data proyek, seperti nama proyek dan nama layer dari pengukuran
- Data pohon, seperti spesies pohon, koordinat GPS, dan foto pohon
- Data geometri dari layer batang pohon yang diukur, seperti keliling lingkaran batang, jumlah sensor yang digunakan, dan ketinggian pengukuran
- Data ToF tiap sensor yang dikirimkan dari *mainboard*

Pengiriman nomor sensor yang diketuk dari aplikasi ke *mainboard* juga akan dilakukan untuk memastikan perhitungan pengambilan data di *mainboard* sesuai. Diagram fungsi aplikasi dapat dilihat pada Gambar 3.



Gambar 3. Dekomposisi Fungsi Sub-sistem Aplikasi

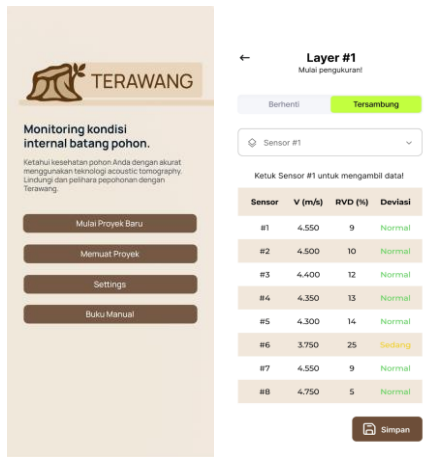
Beberapa *screen* dibuat untuk memastikan *User Interface* (UI) dan *User Experience* (UX) yang baik, dengan navigasi aplikasi seperti Gambar 4 di bawah.



Gambar 4. State Diagram Desain Aplikasi

### B. Desain User Interface

Desain UI dikembangkan menggunakan Figma dengan pendekatan yang mempertimbangkan fungsionalitas aplikasi dalam menerima masukan data dari pengguna dan menampilkannya. Pemilihan warna didasarkan pada tema utama aplikasi, yaitu coklat yang merepresentasikan warna batang pohon. Gambar 5 di bawah menunjukkan tangkapan layar beberapa *screen* yang dirancang untuk aplikasi ini.



Gambar 5. Desain UI di Figma

Desain tersebut merupakan gambaran awal dari tampilan dan alur aplikasi yang menjadi panduan utama dalam proses implementasi menggunakan Flutter.

### C. Software Development Kit dan Library

Dalam mengimplementasikan aplikasi seluler, diperlukan SDK yang dapat memenuhi kebutuhan pengembangan dengan fleksibilitas tinggi. Target awal aplikasi adalah mendukung platform Android dengan tujuan akhir yang juga dapat mendukung platform iOS, sehingga dipilih SDK *cross-platform* untuk mempermudah ekspansi ke platform lain di masa mendatang. SDK yang dipilih harus memiliki komunitas yang besar, dokumentasi yang lengkap, dan ekosistem library yang kaya untuk mempermudah pengembangan aplikasi. Flutter dipilih sebagai SDK utama karena menawarkan *base component* yang sudah siap digunakan, serta kemampuan untuk menghasilkan antarmuka yang konsisten di berbagai perangkat. Dibandingkan SDK lain seperti React Native, Flutter memberikan pendekatan *widget-first* yang lebih mudah dipahami, sehingga mengurangi waktu pengembangan fitur utama aplikasi.

Untuk memenuhi kebutuhan aplikasi, dipilih beberapa *package* tambahan berdasarkan kebutuhan teknis, stabilitas, dan tingkat dukungan komunitasnya:

- Camera: dipilih untuk fitur pengambilan foto pohon karena menawarkan stabilitas yang tinggi, kemampuan integrasi yang baik dengan perangkat keras kamera, serta dokumentasi yang lengkap sehingga mempermudah proses debugging dan pengembangan dibandingkan *package* Flutter lainnya.
- Geolocator: digunakan untuk memperoleh data lokasi pohon dengan akurasi tinggi. Dibandingkan dengan *package* lain, geolocator lebih sering diperbarui dan memiliki kompatibilitas tinggi dengan versi terbaru Flutter.
- Url\_launcher: *package* ini digunakan untuk mengarahkan pengguna ke Google Maps untuk melihat lokasi pohon dengan lebih spesifik. Dibanding *package* lain, implementasi dengan url\_launcher lebih sederhana dan lebih stabil ketika diintegrasikan dengan aplikasi pihak ketiga.
- Flutter\_bluetooth\_serial: untuk melakukan komunikasi Bluetooth, flutter\_bluetooth\_serial dipilih karena kompatibel dengan modul Bluetooth Classic

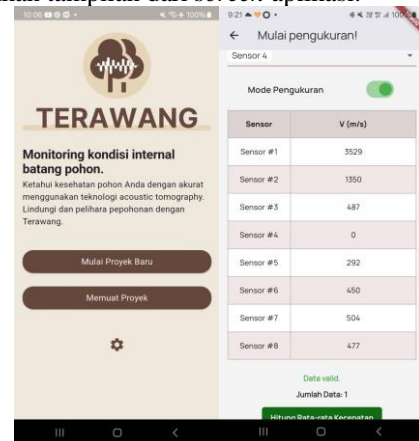
HC-05 yang digunakan sebagai modul Bluetooth pada alat *Acoustic Tomography* yang dibuat.

- Permission\_handler: digunakan untuk mengelola izin aplikasi seperti akses kamera, lokasi, dan Bluetooth pada HP.
- Provider: digunakan untuk mempermudah pengelolaan *state* secara global dalam aplikasi, terutama dalam pengaturan ukuran font, karena sederhana dalam membagikan *state* antar-widget tanpa harus melewati hierarki widget secara manual, sehingga mengurangi kompleksitas kode.

## IV. IMPLEMENTASI DAN PENGUJIAN

### A. UI/UX

Menyesuaikan desain yang telah dibuat, UI diimplementasikan dengan Flutter menyesuaikan tampilan dan navigasi layar yang dirancang sebelumnya. Gambar 6 menunjukkan tampilan dari *screen* aplikasi.



Gambar 6. Tampilan aplikasi di HP

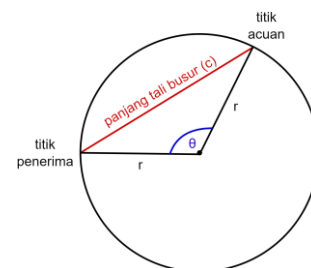
### B. Perhitungan Jarak Sensor

Jarak antara sensor dihitung berdasarkan konfigurasi geometris sensor yang dipasang pada batang pohon. Setiap sensor diposisikan secara melingkar dengan sudut yang sama antar-sensor. Dengan asumsi bahwa lingkaran batang pohon yang diukur adalah lingkaran, jarak antar sensor adalah busur lingkaran yang dihitung dengan persamaan (2),

$$c = 2r \sin \frac{\theta}{2}$$

(2)

dimana  $c$  adalah jarak busur lingkaran,  $r$  jari-jari lingkaran, dan  $\theta$  sudut dari kedua sensor seperti yang terlihat pada Gambar 7.

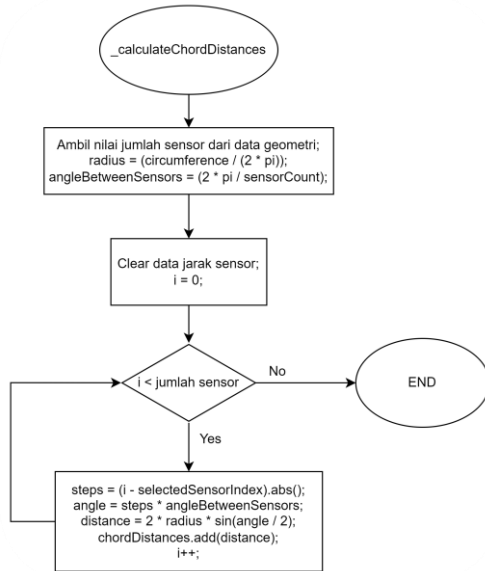


Gambar 7. Perhitungan tali busur lingkaran sebagai jarak antar sensor

Sudut antara sensor acuan yang diketuk dengan sensor lainnya dihitung dengan persamaan (3),

$$\theta = |i - \text{indeks sensor acuan}| \frac{2\pi}{\text{jumlah sensor}} \quad (3)$$

dimana  $i$  adalah indeks sensor yang nilainya akan dicari. Berdasarkan kedua persamaan tersebut, kode perhitungan jarak antar sensor dibuat dengan proses seperti Gambar 8.



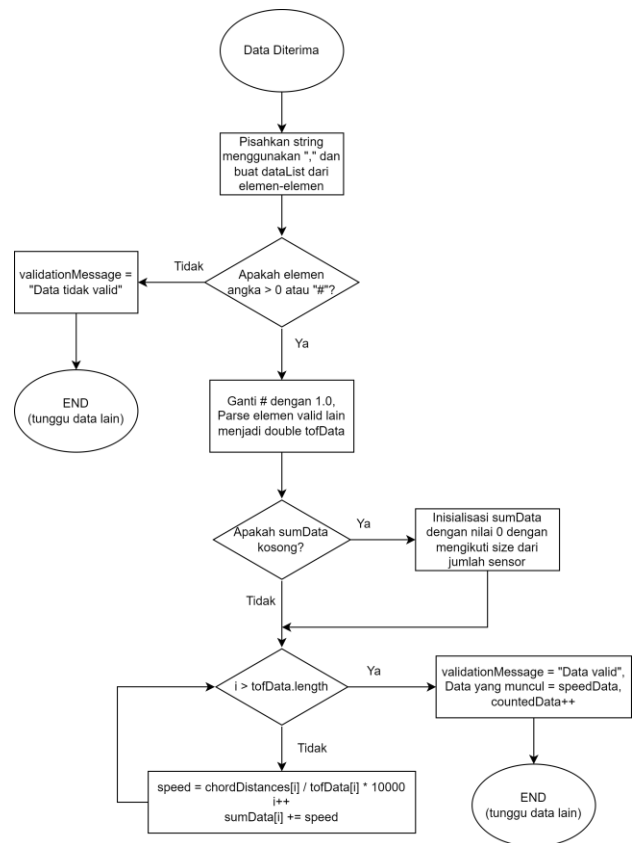
Gambar 8. Flowchart perhitungan jarak sensor

### C. Pengolahan Data ToF

Data ToF tiap sensor dikirimkan melalui koneksi Bluetooth dari *mainboard* ke aplikasi secara bersamaan. Pengiriman data dilakukan dengan menggabungkan nilai dari kedelapan sensor menjadi satu rangkaian kalimat, dengan menggunakan tanda koma (,) sebagai pemisah antar elemen dalam array, dan nilai ToF sensor acuan sebagai "#". Aplikasi kemudian menerima data tersebut, memisahkan setiap elemen, dan mengasosiasikan nilai ToF yang diterima dengan data sensor yang bersangkutan. Selanjutnya, kecepatan dihitung dengan membagi jarak tali busur antara pasangan sensor dengan nilai ToF yang terkait untuk masing-masing pasangan sensor tersebut. Karena data ToF yang dikirimkan dari *mainboard* memiliki satuan mikrosekon dan jarak yang dihitung memiliki satuan cm, perhitungan kecepatan dalam m/s lakukan dengan mengalikan hasil dengan 10000 seperti pers. (4).

$$\text{speed} = \frac{\text{chordDistances}[i]}{\text{tofData}[i]} \times 10000 \quad (4)$$

Data kecepatan yang diperoleh ditampilkan secara *real-time* di aplikasi dan akan dihitung rata-ratanya setelah pengukuran di sensor acuan selesai dilakukan. Gambar 9 menunjukkan diagram alir pengolahan data ToF ini.



Gambar 9. Flowchart pengolahan data ToF

### D. Pengujian Fitur

Pengujian dilakukan secara langsung dengan mengupload aplikasi Flutter ke beberapa jenis HP Android. Berikut adalah hasil pengujian dari fitur-fitur aplikasi.

#### Pengaturan Font

Pengujian dilakukan secara langsung dengan mengubah font pada SettingScreen, seperti yang terlihat pada Tabel I.

TABEL I. PENGUJIAN PENGATURAN FONT

Ukuran	Tampilan di Aplikasi	Ukuran	Tampilan di Aplikasi
Font Body 12		Font Body 24	



Aplikasi mampu mengubah font utama sesuai dengan masukan yang diberikan oleh pengguna, sehingga dapat menyesuaikan preferensi pengguna, membuktikan bahwa aplikasi memenuhi spesifikasi yaitu teks tetap mudah dibaca pada jarak pandang sekitar 41-46 cm.



## Lokasi dan Google Maps

Untuk mendapatkan lokasi dengan akurasi yang tinggi, data lokasi didapatkan dengan mengambil koordinat geografis berupa latitude dan longitude yang diperoleh secara langsung melalui GPS pada HP pengguna. Verifikasi lokasi dilakukan dengan memanfaatkan fitur button dalam aplikasi yang mengarahkan pengguna ke Google Maps berdasarkan koordinat yang didapatkan, dengan hasil seperti pada Tabel II.

TABEL II. PENGUJIAN LOKASI

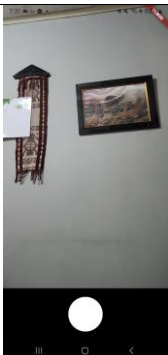
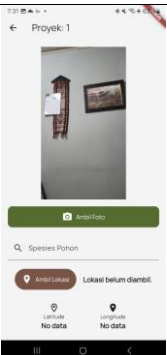
Tipe HP	Koordinat	Lokasi di Google Maps
Samsung S22 Ultra (Android 14)		

Pengujian juga telah dilakukan pada HP Oppo A5 2020 (Android 11) dan HP Poco M4 Pro (Android 13). Ketiga pengujian menunjukkan bahwa kamera telah berfungsi dengan baik dan terverifikasi fungsinya.

## Kamera

Tabel III menunjukkan hasil pengujian fungsi kamera. Kamera sudah berfungsi dengan baik untuk menunjukkan *preview* dan memunculkan hasil foto.

TABEL III. PENGUJIAN KAMERA

Tipe HP	Preview Kamera	Gambar di ProjectScreen
Samsung S22 Ultra (Android 14)		

Pengujian yang sama juga dilakukan di HP Oppo A5 2020 (Android 11) dan HP Samsung A25 (Android 14) dan kamera sudah berfungsi dengan baik.

## Koneksi Bluetooth


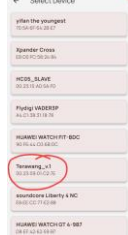
Pengujian koneksi Bluetooth dilakukan dengan mencoba koneksi dengan dua buah Bluetooth HC-05 yang terkoneksi dengan mikrokontroler, seperti pada Gambar 10.



Gambar 10. Pengujian dengan menggunakan HC-05 yang tersambung dengan Arduino Uno

Hasil pengujian dapat dilihat di Tabel IV.

TABEL IV. PENGUJIAN KONEKSI BLUETOOTH

Device	Tampilan di Aplikasi	Keterangan
HC-05 1 (terkoneksi dengan Arduino Uno)		Tersambung
HC-05 2 (terkoneksi dengan Teensy 4.1)		Tersambung

## Perhitungan Jarak

Pengujian perhitungan jarak dilakukan dengan melakukan koneksi Bluetooth dengan Arduino Uno yang tersambung dengan HC-05 dan mengirimkan data melalui serial monitor. Untuk memverifikasi bahwa perhitungan jarak sudah benar, data yang dikirimkan adalah satu untuk setiap sensor.

Input: 1,1,1,1,1,1,1 (nilai # untuk sensor acuan)

Output yang muncul dapat dilihat pada Tabel V.

TABEL V. PENGUJIAN PERHITUNGAN JARAK DENGAN KELILING LINGKAR BATANG 120 CM

Sensor Acuan	Sensor Tujuan				
	1	3	5	6	8
1	0	27.01	38.20	35.29	14.62
2	14.62	14.62	35.29	38.20	27.01
3	27.01	0	27.01	35.29	35.29
4	35.29	14.62	14.62	27.01	38.20
5	38.2	27.01	0	14.62	35.29
6	35.29	35.29	14.62	0	27.01
7	27.01	38.20	27.01	14.62	14.62
8	14.62	35.29	35.29	27.01	0

Hasil perhitungan jarak sudah berhasil terverifikasi dengan benar.

### Pengolahan Data ToF Menjadi Kecepatan

Sama seperti perhitungan jarak, pengujian validasi dan perhitungan kecepatan masih dilakukan dengan melakukan koneksi Bluetooth dengan Arduino Uno yang tersambung dengan HC-05 dan mengirimkan data melalui serial monitor. Namun, data yang dikirimkan lebih bervariasi untuk memproses validasi data. Tabel VI menunjukkan hasil pengujian perhitungan kecepatan.

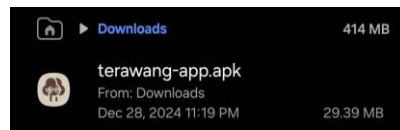
TABEL VI. PENGUJIAN PERHITUNGAN KECEPATAN DENGAN KELILING LINGKAR BATANG 120 CM

Input	Hasil Pengukuran		Input	Hasil Pengukuran 1	
	Sensor	Nilai		Sensor	Nilai
100,200,300,#,500,,700,800	1	Data tidak Valid.	100,200,300,#,500,600,700,800	1	3529
	2			2	1350
	3			3	487
	4			4	0
	5			5	292
	6			6	450
	7			7	504
	8			8	477
Input	Hasil Pengukuran		Input	Hasil Pengukuran 2	
	Sensor	Nilai		Sensor	Nilai
100,200,300,#,500,0,700,800	1	Data tidak Valid.	50,100,150,#,250,300,350,400	1	7058
	2			2	2701
	3			3	974
	4			4	0
	5			5	585
	6			6	900
	7			7	1008
	8			8	955
Input	Hasil Pengukuran 3		Input	Perhitungan Rata-rata	
	Sensor	Nilai		Sensor	Nilai
110,120,130,#,150,160,170,180	1	3208	“Hitung Rata-rata Kecepatan”	1	4598
	2	2251		2	2101
	3	1124		3	862
	4	0		4	0
	5	974		5	617
	6	1688		6	1013
	7	2076		7	1196
	8	2122		8	1185

Berdasarkan hasil pengujian di atas, pemrosesan dan perhitungan kecepatan sudah terverifikasi dengan benar.

### Instalasi Aplikasi Melalui APK

Pengujian aplikasi (versi *release*) dilakukan dengan mengunduh aplikasi secara manual dari file APK Terawang yang terlihat seperti Gambar 11.



Gambar 11. File APK Terawang

Hasil instalasi dapat dilihat pada Tabel VII.

TABEL VII. PENGUJIAN INSTALASI APK

Device	Tampilan Icon	Keterangan
Samsung S22 Ultra (Android 14)		

Pengujian juga dilakukan pada HP Oppo A5 2020 (Android 11), HP Poco M4 Pro (Android 13), HP Samsung A25 (Android 14) dan aplikasi sudah berhasil diunduh.

### V. KESIMPULAN

Hasil dari desain dan implementasi dari Aplikasi untuk alat *Acoustic Tomography* menggunakan Flutter menunjukkan bahwa aplikasi sudah berfungsi dengan sesuai harapan, dimana aplikasi dapat menerima input yang diberikan oleh user kemudian menampilkannya kembali di layar, melakukan fungsi utama aplikasi seperti pengaturan font sehingga dapat dibaca pada jarak 41-46 cm, dapat mengambil foto dengan kamera, mendapatkan koordinat lokasi dan membuka Google Maps, melakukan koneksi Bluetooth, menghitung jarak dan kecepatan dengan benar, dan berhasil diunduh ke HP dari bentuk file APK.

### REFERENCES

- [1] D. J. Nowak, D. E. Crane, and J. C. Stevens, "Air pollution removal by urban trees and shrubs in the United States," *Urban Forestry & Urban Greening*, vol. 4, no. 3-4, pp. 115-123, Feb. 2006, doi: 10.1016/j.ufug.2006.01.007.
- [2] L. Tyrväinen, S. Pauleit, K. Seeland, and S. De Vries, "Benefits and uses of urban forests and trees," in *Springer eBooks*, 2005, pp. 81-114. doi: 10.1007/3-540-27684-x\_5.
- [3] A. K. Buza, B. Goncz, J. Bodig, and A. Act, "Comparison of Trees and NDT Methods," *Wood Research*, vol. 60, no. 1, pp. 45-58, Jan. 2015.
- [4] E. Gülcüoğlu, A. B. Ustun, and N. Seyhan, "Comparison of flutter and React native platforms," *Journal of Internet Applications and Management*, Dec. 2021, doi: 10.34231/iuyd.888243.
- [5] F. Cheng, *Flutter Recipes: mobile development solutions for iOS and Android*. 2019, doi: 10.1007/978-1-4842-4982-6
- [6] L. Lantini, "Health Monitoring of Trees and Investigation of Tree Root Systems using Ground Penetrating Radar (GPR)," 2021, doi: 10.13140/RG.2.2.14148.14724.
- [7] V. Bucur, *Urban Forest Acoustics*. 2006. doi: 10.1007/3-540-30789-3.
- [8] E.-C. Leong, D. C. Burcham, and Y.-K. Fong, "A purposeful classification of tree decay detection tools," *Arboricultural Journal*, vol. 34, no. 2, pp. 91-115, Jun. 2012, doi: 10.1080/03071375.2012.701430.