

Laporan Penugasan Teknologi Berkembang
Human-Machine Interface - Bluetooth Low Energy (BLE)



Disusun Oleh:
Ihsan Kamil Al Khozi
5026211117

Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2024

Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) adalah teknologi komunikasi nirkabel yang dirancang untuk memungkinkan perangkat saling berkomunikasi dengan konsumsi daya yang sangat rendah. Dikembangkan sebagai bagian dari standar Bluetooth 4.0 dan versi-versi selanjutnya, BLE ditujukan untuk aplikasi yang memerlukan transmisi data secara periodik dengan efisiensi energi tinggi, sehingga ideal untuk perangkat yang mengandalkan baterai dan memerlukan masa pakai yang panjang.

Fitur Utama Bluetooth Low Energy

Berikut merupakan beberapa fitur utama dari BLE:

Fitur	Penjelasan
Konsumsi Daya Rendah	BLE dioptimalkan untuk penggunaan energi minimal, memungkinkan perangkat seperti sensor dan gadget wearable beroperasi dalam jangka waktu yang lama tanpa perlu sering mengganti atau mengisi ulang baterai.
Kecepatan Transmisi Data yang Efisien	Meskipun konsumsi dayanya rendah, BLE mampu mentransmisikan data dengan cepat dan efisien, cocok untuk aplikasi yang membutuhkan pengiriman data kecil secara berkala.
Jangkauan yang Memadai	BLE memiliki jangkauan sekitar 10 meter, yang cukup untuk banyak aplikasi seperti komunikasi antar perangkat dalam satu ruangan atau area terbatas.
Kompatibilitas Luas	Banyak perangkat modern seperti smartphone, tablet, dan komputer sudah mendukung BLE, memudahkan integrasi dan konektivitas antar berbagai perangkat.
Topologi Jaringan Fleksibel	BLE mendukung berbagai topologi jaringan seperti star, mesh, dan piconet, memungkinkan fleksibilitas dalam pengaturan jaringan perangkat.

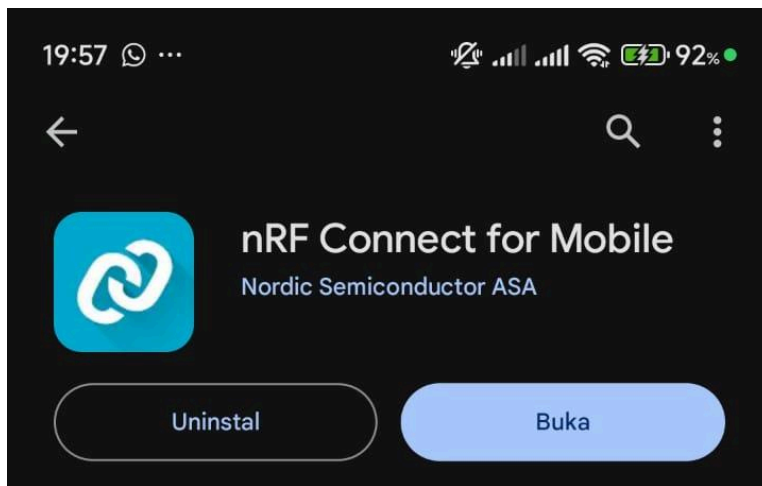
Percobaan Bluetooth Low Energy

Pada percobaan BLE ini, saya menggunakan beberapa hal di bawah ini:

- Smartphone dengan aplikasi "nRF Connect" yang terunduh di dalamnya
- Python dengan versi 3.13.0
- Library Python "Bleak"

Berikut adalah langkah - langkah percobaan menggunakan BLE:

1. Mengunduh Aplikasi "nRF Connect" Pada Smartphone



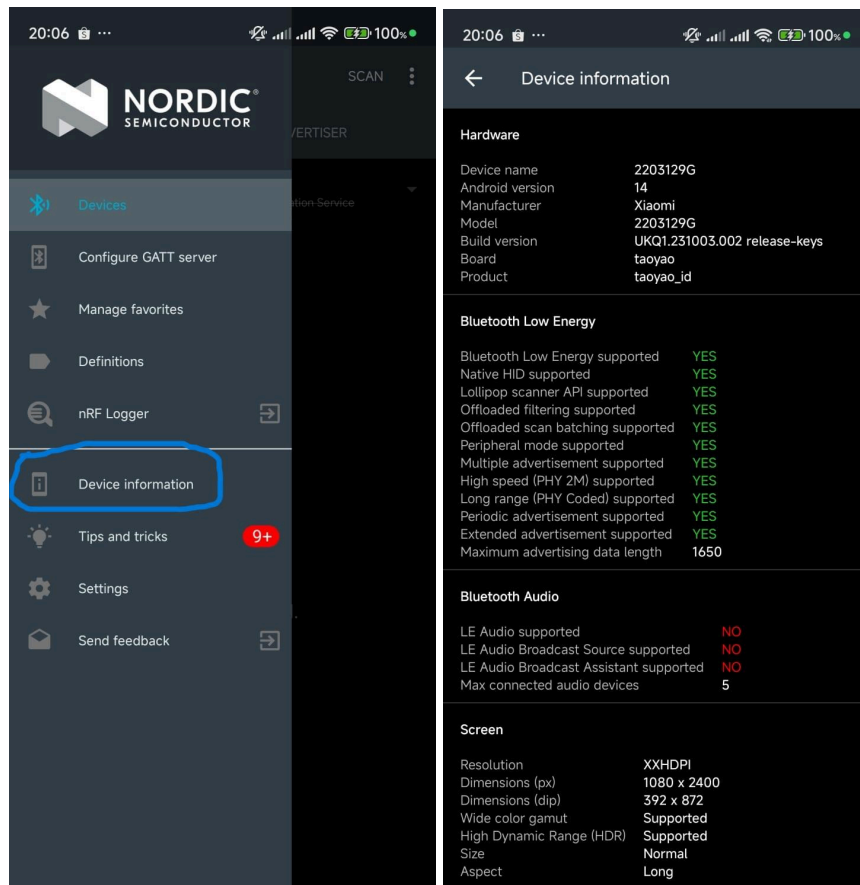
Aplikasi nRF Connect akan digunakan sebagai alat untuk berinteraksi dengan perangkat Bluetooth Low Energy.

2. Menginstall Library Python "Bleak"

```
Administrator: Prompt Perintah
C:\Windows\System32>pip install bleak
Collecting bleak
  Downloading bleak-0.22.3-py3-none-any.whl.metadata (5.4 kB)
Collecting winrt-Windows.Devices.Bluetooth<3,>=2 (from bleak)
  Downloading winrt-Windows.Devices.Bluetooth-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.6 kB)
Collecting winrt-Windows.Devices.Bluetooth.Advertisement<3,>=2 (from bleak)
  Downloading winrt-Windows.Devices.Bluetooth.Advertisement-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.3 kB)
Collecting winrt-Windows.Devices.Bluetooth.GenericAttributeProfile<3,>=2 (from bleak)
  Downloading winrt-Windows.Devices.Bluetooth.GenericAttributeProfile-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.5 kB)
Collecting winrt-Windows.Devices.Enumeration<3,>=2 (from bleak)
  Downloading winrt-Windows.Devices.Enumeration-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.5 kB)
Collecting winrt-Windows.Foundation<3,>=2 (from bleak)
  Downloading winrt-Windows.Foundation-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.0 kB)
Collecting winrt-Windows.Foundation.Collections<3,>=2 (from bleak)
  Downloading winrt-Windows.Foundation.Collections-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.1 kB)
Collecting winrt-Windows.Storage.Streams<3,>=2 (from bleak)
  Downloading winrt-Windows.Storage.Streams-2.2.0-cp313-cp313-win_amd64.whl.metadata (1.3 kB)
Collecting winrt-runtime<3,>=2 (from bleak)
  Downloading winrt_runtime-2.2.0-cp313-cp313-win_amd64.whl.metadata (733 bytes)
Downloading bleak-0.22.3-py3-none-any.whl (142 kB)
Downloading winrt_runtime-2.2.0-cp313-cp313-win_amd64.whl (213 kB)
Downloading winrt-Windows.Devices.Bluetooth-2.2.0-cp313-cp313-win_amd64.whl (104 kB)
Downloading winrt-Windows.Devices.Bluetooth.Advertisement-2.2.0-cp313-cp313-win_amd64.whl (83 kB)
Downloading winrt-Windows.Devices.Bluetooth.GenericAttributeProfile-2.2.0-cp313-cp313-win_amd64.whl (179 kB)
Downloading winrt-Windows.Devices.Enumeration-2.2.0-cp313-cp313-win_amd64.whl (125 kB)
Downloading winrt-Windows.Foundation-2.2.0-cp313-cp313-win_amd64.whl (100 kB)
Downloading winrt-Windows.Foundation.Collections-2.2.0-cp313-cp313-win_amd64.whl (60 kB)
Downloading winrt-Windows.Storage.Streams-2.2.0-cp313-cp313-win_amd64.whl (108 kB)
Installing collected packages: winrt-runtime, winrt-Windows.Storage.Streams, winrt-Windows.Foundation.Collections, winrt-Windows.Foundation, winrt-Windows.Devices.Enumeration, winrt-Windows.Devices.Bluetooth.GenericAttributeProfile, winrt-Windows.Devices.Bluetooth.Advertisement, winrt-Windows.Devices.Bluetooth, bleak
Successfully installed bleak-0.22.3 winrt-Windows.Devices.Bluetooth-2.2.0 winrt-Windows.Devices.Bluetooth.Advertisement-2.2.0 winrt-Windows.Devices.Bluetooth.GenericAttributeProfile-2.2.0 winrt-Windows.Devices.Enumeration-2.2.0 winrt-Windows.Foundation-2.2.0 winrt-Windows.Foundation.Collections-2.2.0 winrt-Windows.Storage.Streams-2.2.0 winrt-runtime-2.2.0
C:\Windows\System32>
```

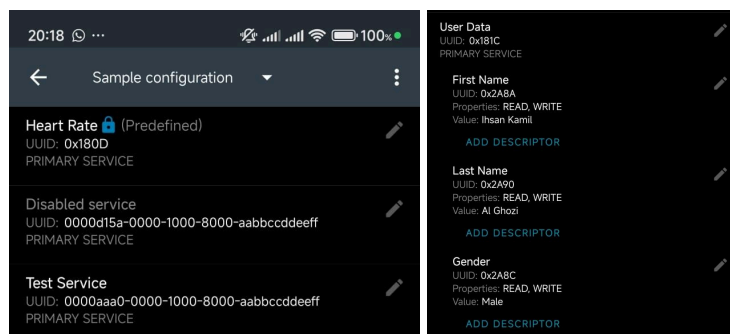
Bleak adalah sebuah pustaka (library) Python yang dirancang untuk memudahkan komunikasi dengan perangkat Bluetooth Low Energy (BLE). Dengan Bleak, pengembang dapat membuat aplikasi yang dapat terhubung, membaca, menulis, dan berinteraksi dengan berbagai perangkat BLE secara efisien dan sederhana.

3. Membuka Device Information Pada Aplikasi "nRF Connect"



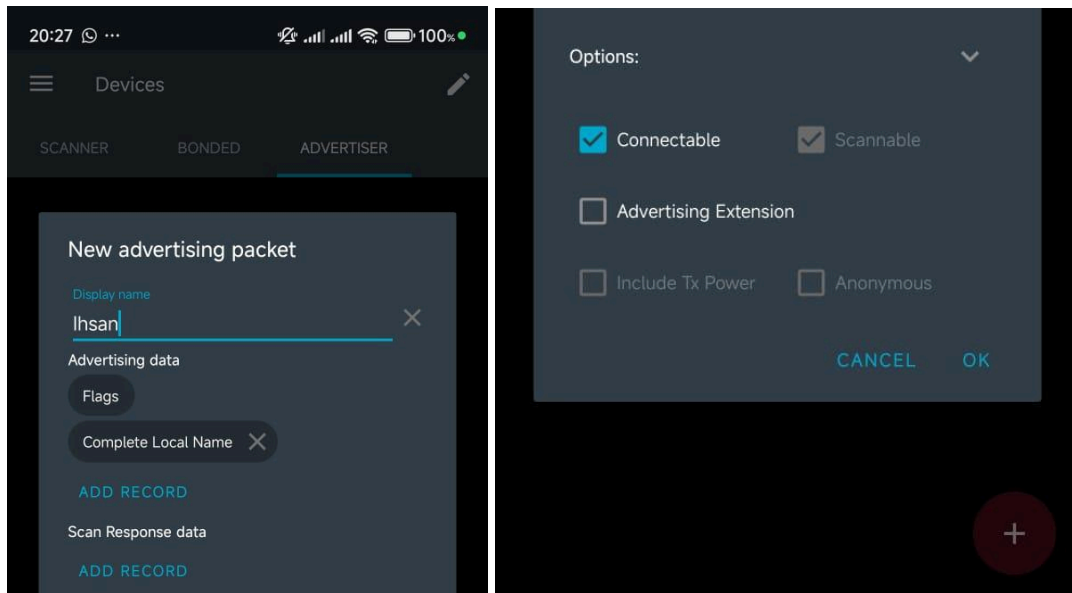
Pada menu ini akan ditampilkan berbagai informasi dari smartphone yang digunakan. Pastikan bahwa smartphone yang akan digunakan support dengan BLE.

4. Melakukan Konfigurasi Server Generic Attribute (GATT)



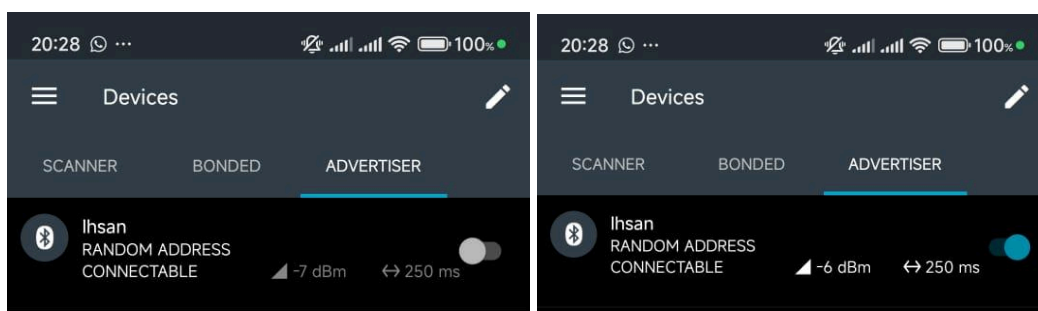
Halaman ini menampilkan konfigurasi contoh dari GATT Server yang akan kita gunakan. GATT Server ini berperan sebagai server simulasi untuk Bluetooth Low Energy (BLE). Kita bisa mengubah isian dari **User Data** jika diperlukan, seperti yang saya cantumkan pada gambar di atas.

5. Membuat Advertiser



Pada langkah ini kita akan membuat sebuah Advertiser baru yang akan digunakan untuk menghubungkan perangkat dengan target. Disini yang perlu diperhatikan adalah menambahkan record **Complete Local Name** pada bagian Advertising data dan mencentang **Connectable** pada options.

6. Menyalakan Advertiser



Setelah Advertiser dinyalakan, maka perangkat akan dapat terdeteksi.

7. Mencari MAC Address dari Smartphone menggunakan Bleak

```
scan-ble.py > ...
1  import asyncio
2  from bleak import BleakScanner
3
4  async def main():
5      devices = await BleakScanner.discover()
6      for d in devices:
7          print(d)
8
9  asyncio.run(main())
```

```
D:\Kuliah\Sem7\Tekber>scan-ble.py
7D:92:F9:25:97:4E: None
59:18:AC:5C:12:79: None
7A:4D:21:7D:4A:BB: Ihsantuyaja
6F:23:7D:7B:33:90: None
```

Dapat dilihat disini bahwa MAC Address dari perangkat saya (Ihsantuyaja) dapat ditemukan.

8. Melakukan Koneksi ke GATT Server

```
connect-services.py > ...
1  import asyncio
2  from bleak import BleakClient
3  import sys
4
5  async def get_services(mac):
6      async with BleakClient(mac) as client:
7          print(f"Connected: {client.is_connected}")
8
9          svcs = await client.get_services()
10         print("Services:", svcs)
11         for service in client.services:
12             print("Service: ")
13             print(service)
14
15             print("\nCharacteristics: ")
16             for char in service.characteristics:
17                 print(char)
18                 print("\nProperties: ")
19                 print(char.properties)
20
21         await client.disconnected()
22
23  try:
24      asyncio.run(get_services("79:66:FE:D9:CE:CF"))
25  except KeyboardInterrupt:
26      print("User stopped the program")
27      sys.exit(0)
```

```

Service:
0000181c-0000-1000-8000-00805f9b34fb (Handle: 85): User Data

Characteristics:
00002a8a-0000-1000-8000-00805f9b34fb (Handle: 86): First Name
|nProperties:
['read', 'write']
00002a90-0000-1000-8000-00805f9b34fb (Handle: 88): Last Name
|nProperties:
['read', 'write']
00002a8c-0000-1000-8000-00805f9b34fb (Handle: 90): Gender
|nProperties:
['read', 'write']

D:\Kuliah\Sem7\Tekber>

```

Disini terlihat bahwa laptop saya telah berhasil terhubung ke smartphone saya ketika file **connect-services.py** dijalankan. Hasil dari file tersebut berupa informasi terkait karakteristik, layanan, dan properti. Terlihat juga bahwa output yang dihasilkan ditampilkan dalam UUID.

9. Modifikasi Kode Program Untuk Menampilkan Karakter UUID

```

connect-services.py > ...
1  import asyncio
2  from bleak import BleakClient
3  import sys
4
5  uuid_first_name = "00002a8a-0000-1000-8000-00805f9b34fb"
6  uuid_last_name = "00002a90-0000-1000-8000-00805f9b34fb"
7  uuid_gender = "00002a8c-0000-1000-8000-00805f9b34fb"
8
9  async def get_services(mac):
10     async with BleakClient(mac) as client:
11         print(f"Connected: {client.is_connected}")
12
13         # svcs = await client.get_services()
14         # print("Services:", svcs)
15         # for service in client.services:
16         #     print("Service: ")
17         #     print(service)
18
19         #     print("\nCharacteristics: ")
20         #     for char in service.characteristics:
21         #         print(char)
22         #         print("\nProperties: ")
23         #         print(char.properties)
24
25         first_name = await client.read_gatt_char(uuid_first_name)
26         print("First Name: " + first_name.decode("utf-8"))
27         last_name = await client.read_gatt_char(uuid_last_name)
28         print("Last Name: " + last_name.decode("utf-8"))
29         gender = await client.read_gatt_char(uuid_gender)
30         print("Gender: " + gender.decode("utf-8"))
31
32         await client.disconnect()
33
34     try:
35         asyncio.run(get_services("75:33:E5:18:D1:BD"))
36     except KeyboardInterrupt:
37         print("User stopped the program")
38         sys.exit(0)

```

```
D:\Kuliah\Sem7\Tekber>connect-services.py
Connected: True
First Name: Ihsan Kamil
Last Name: Al Ghazi
Gender: Male
```

Setelah program dimodifikasi, kini saya dapat melihat informasi yang ditulis menggunakan UUID oleh sistem sebelumnya. Terlihat informasi nama dan gender yang tertera.

10. Melakukan Modifikasi Advertiser via Python

```
connect-services.py > ...
1  import asyncio
2  from bleak import BleakClient
3  import sys
4
5  uuid_first_name = "00002a8a-0000-1000-8000-00805f9b34fb"
6  uuid_last_name = "00002a90-0000-1000-8000-00805f9b34fb"
7  uuid_gender = "00002a8c-0000-1000-8000-00805f9b34fb"
8
9  async def get_services(mac):
10     async with BleakClient(mac) as client:
11         print(f"Connected: {client.is_connected}")
12
13         # svcs = await client.get_services()
14         # print("Services:", svcs)
15         # for service in client.services:
16         #     print("Service: ")
17         #     print(service)
18
19         #     print("\nCharacteristics: ")
20         #     for char in service.characteristics:
21         #         print(char)
22         #         print("\nProperties: ")
23         #         print(char.properties)
24
25         await client.write_gatt_char(uuid_first_name, "Wanda".encode())
26         await client.write_gatt_char(uuid_last_name, "Armadianti".encode())
27         await client.write_gatt_char(uuid_gender, "Female".encode())
28
29         first_name = await client.read_gatt_char(uuid_first_name)
30         print("First Name: " + first_name.decode("utf-8"))
31         last_name = await client.read_gatt_char(uuid_last_name)
32         print("Last Name: " + last_name.decode("utf-8"))
33         gender = await client.read_gatt_char(uuid_gender)
34         print("Gender: " + gender.decode("utf-8"))
35
36         await client.disconnect()
37
38     try:
39         asyncio.run(get_services("64:3B:50:7F:51:04"))
40     except KeyboardInterrupt:
41         print("User stopped the program")
42     sys.exit(0)
```



```
D:\Kuliah\Sem7\Tekber>connect-services.py
Connected: True
First Name: Wanda
Last Name: Armadianti
Gender: Female
```

Percobaan perubahan data Advertiser dapat dilakukan karena setelan dari first name, last name, dan gender berupa **Read and Write**. Dengan demikian, melalui koneksi GATT, dapat dilakukan modifikasi.