



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202246976, 22 Juli 2022

Pencipta

Nama : **Novie Theresia Br. Pasaribu, S.T., M.T., Dr. Erwani Merry Sartika, S.T., M.T. dkk**
Alamat : Kompleks Puri Budi Asri E11 Cihanjuang, Parongpong , Bandung Barat, JAWA BARAT, 40559
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Universitas Kristen Maranatha**
Alamat : Jl. Suria Sumantri No. 65, Bandung, JAWA BARAT, 40164
Kewarganegaraan : Indonesia
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Program Komputer Monitoring Tekanan Lidah Berbasis Aplikasi Android & Web**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 1 Juli 2022, di Bandung

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000362705

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

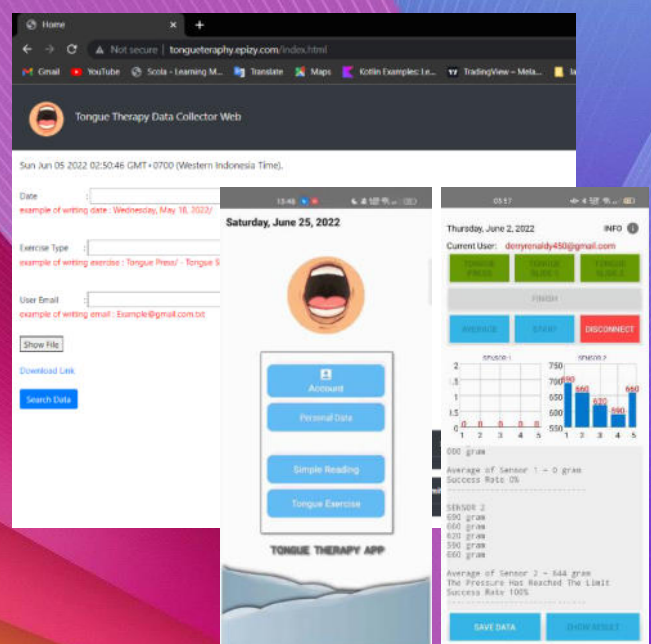
Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Novie Theresia Br. Pasaribu, S.T., M.T.	Kompleks Puri Budi Asri E11 Cihanjuang, Parongpong
2	Dr. Erwani Merry Sartika, S.T., M.T.	Jl. Pulolaut No. 6
3	Derry Renaldy	Jl. Jati Utama Ruko A3 No. 34



2022



Program Komputer Monitoring Tekanan Lidah berbasis Aplikasi Android & Web

Novie Theresia Br. Pasaribu

Erwani Merry Sartika

Derry Renaldy

URAIAN CIPTAAN

Program Komputer “Monitoring Tekanan Lidah berbasis Android & Web” merupakan hasil karya yang berisi program komputer untuk yang dapat membantu pengguna untuk memantau hasil pembacaan tekanan lidah (menampilkan data identitas pengguna, pembacaan tekanan, dan repetisi gerakan) agar pengguna mengetahui progress pembacaan tekanan lidah yang telah dilakukannya dengan menggunakan aplikasi berbasis Android. Kemudian data hasil pembacaan keluaran sensor dimasukkan ke dalam database server. Aplikasi berbasis web yang dirancang dapat mengakses semua data identitas pasien serta hasil pembacaan keluaran sensor yang ada pada database server dengan tujuan untuk mempermudah proses monitoring pembacaan tekanan lidah yang dipantau oleh dokter.

a. Perancangan Gerakan Lidah



Semua gerakan dilakukan selama periode waktu 10 detik dan dalam 5 repetisi

b. Perancangan Antarmuka Aplikasi Android

Aplikasi android yang dibuat memiliki 8 buah tampilan antarmuka, yang terdiri dari :

Tampilan *Splash Screen*

Tampilan dan Proses Kerja Menu *Login dan Register*

Tampilan dan Proses Kerja Menu *List Bluetooth*

Tampilan dan Proses Kerja Menu *Tongue Therapy*

Tampilan Menu Utama

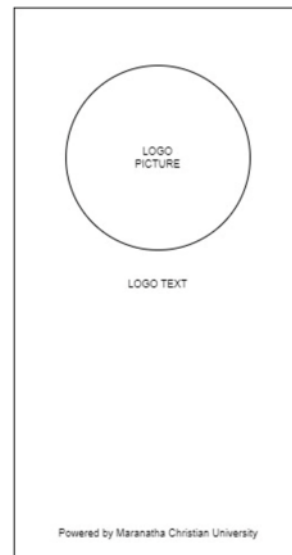
Tampilan dan Proses Kerja Menu *Reset Password*

Tampilan Menu *Simple Reading*

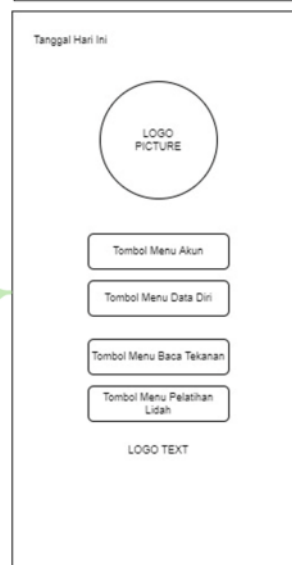
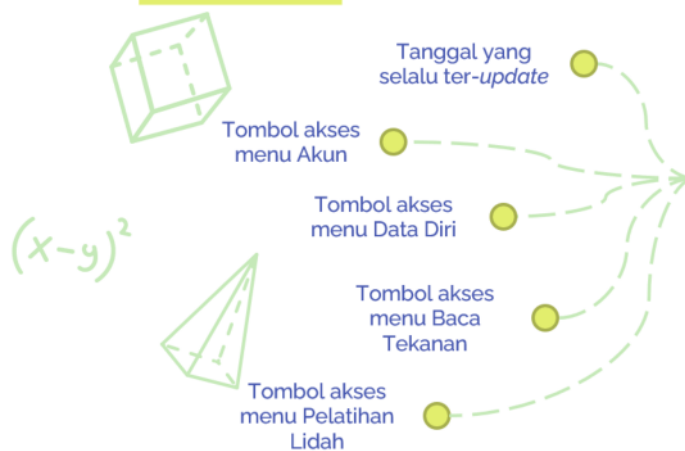
Tampilan Menu *Info*

1. RANCANGAN ANTARLUKA SPLASH SCREEN

Splash Screen merupakan tampilan pertama dari suatu program yang muncul untuk beberapa saat, sebelum akhirnya pengguna dapat memasuki fitur menu utama



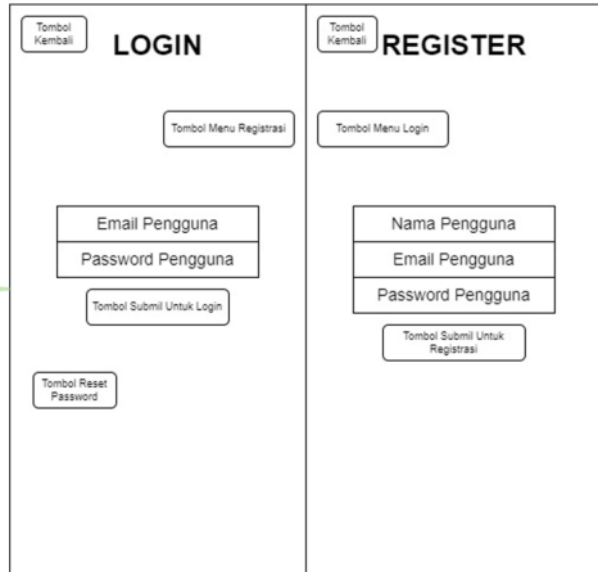
2. RANCANGAN ANTARLUKA MENU UTAMA



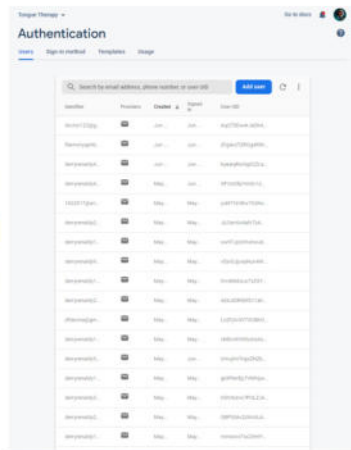
3. RANCANGAN ANTARMUKA MENU LOGIN DAN REGISTER

Berisi form untuk mengisi nama lengkap, email, dan password

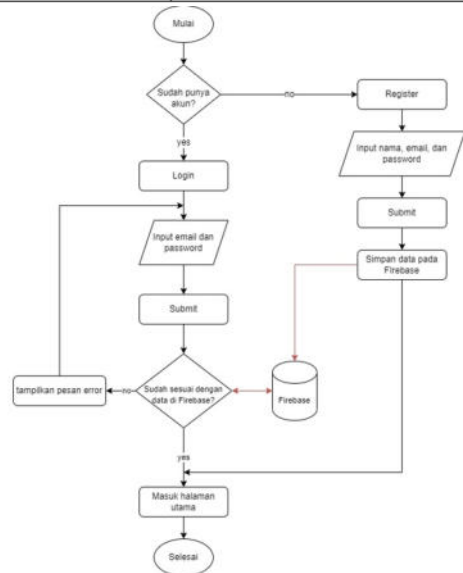
Tombol akses menu Reset Password pada menu login



FLOWCHART ALUR KERJA MENU LOGIN DAN REGISTER



DATABASE AUTHENTICATION FIREBASE



4. RANCANGAN ANTARMUKA MENU RESET PASSWORD

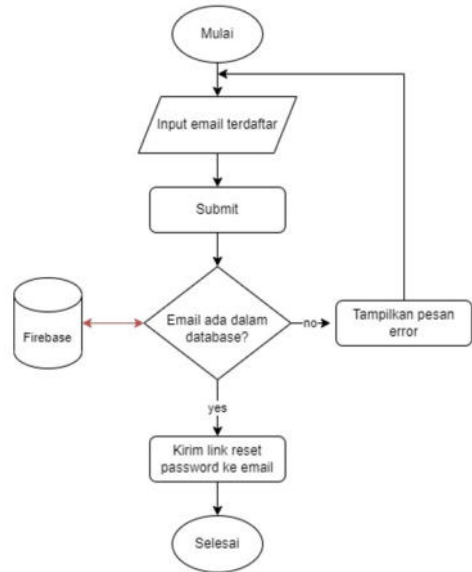
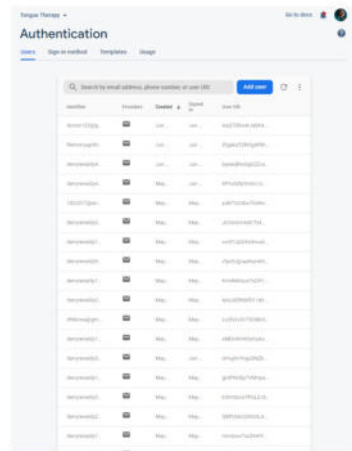
Form untuk memasukkan email pengguna

Tombol submit

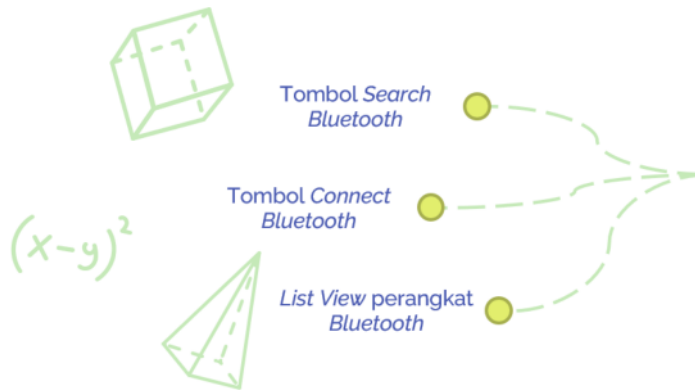


FLOWCHART ALUR KERJA MENU RESET PASSWORD

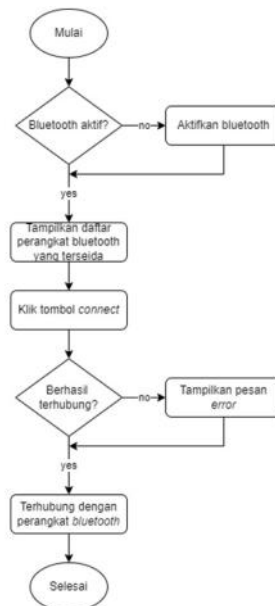
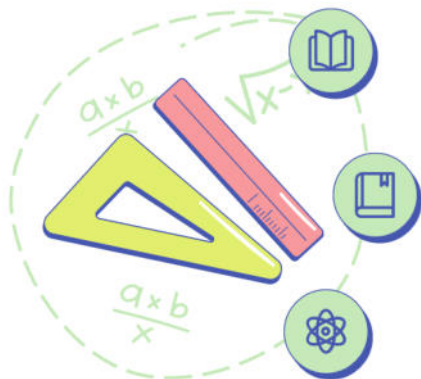
DATABASE AUTHENTICATION FIREBASE



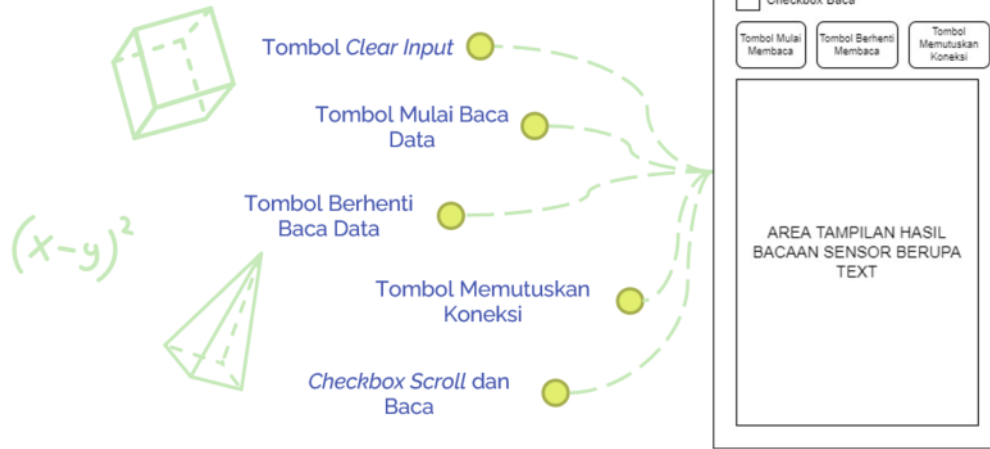
5. RANCANGAN ANTARMUKA MENU KONEKSI BLUETOOTH



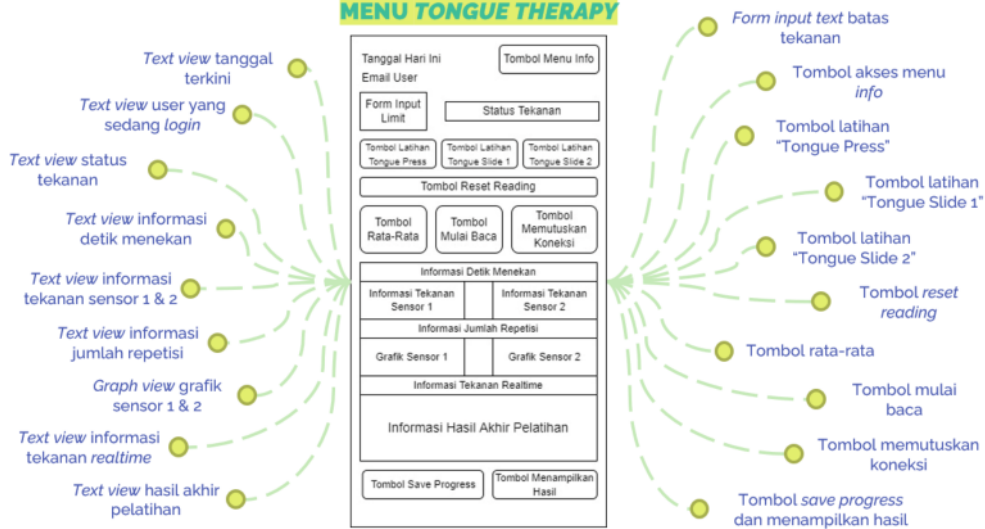
FLOWCHART ALUR KERJA MENU KONEKSI BLUETOOTH



6. RANCANGAN ANTARMUKA MENU SIMPLE READING

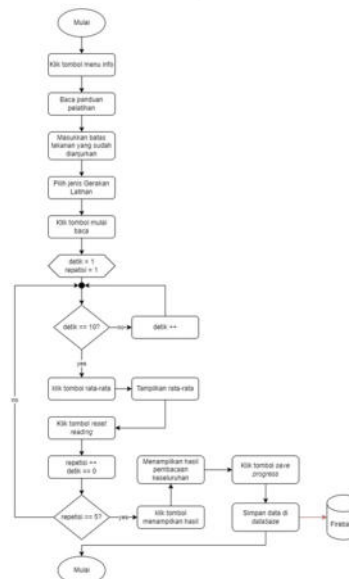
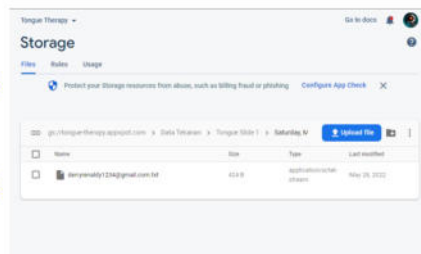


7. RANCANGAN ANTARMUKA MENU TONGUE THERAPY



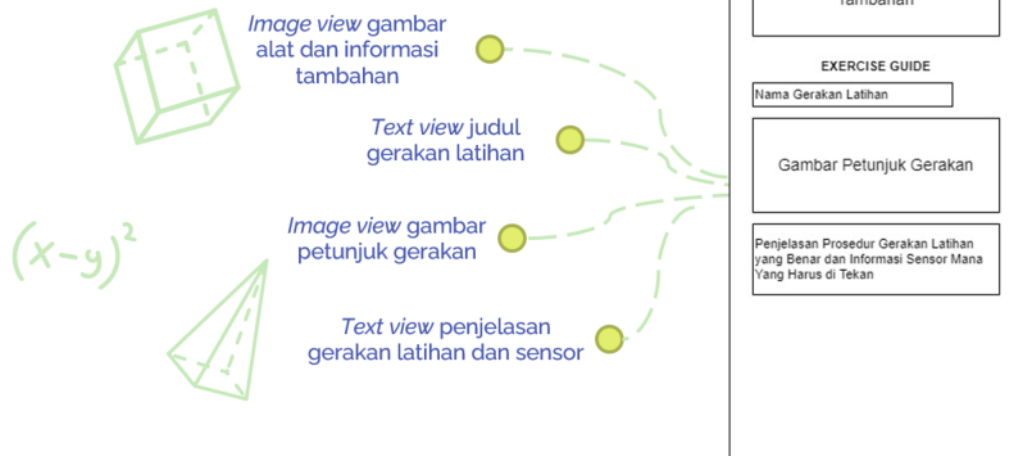
FLOWCHART ALUR KERJA MENU TONGUE THERAPY

DATABASE STORAGE FIREBASE



8. RANCANGAN ANTARMUKA

MENU INFO



c. Perancangan Antarmuka Aplikasi Web

APLIKASI WEB

Aplikasi Web yang dibuat memiliki 2 buah tampilan antarmuka, yang terdiri dari :

Tampilan dan Proses Kerja Menu Login dan Register

Tampilan Halaman Utama

1. RANCANGAN ANTARMUKA HALAMAN LOGIN DAN REGISTER

Berisi form untuk mengisi username, email, dan password

Tongue Therapy Website For Doctor

Sign Up

Username Pengguna

Email Pengguna

Password Pengguna

Tombol Register

Tombol Menu Login

Tongue Therapy Website For Doctor

Sign In

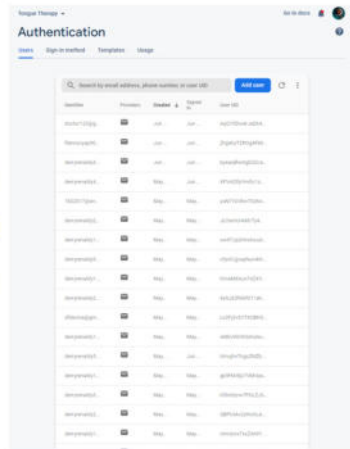
Email Pengguna

Password Pengguna

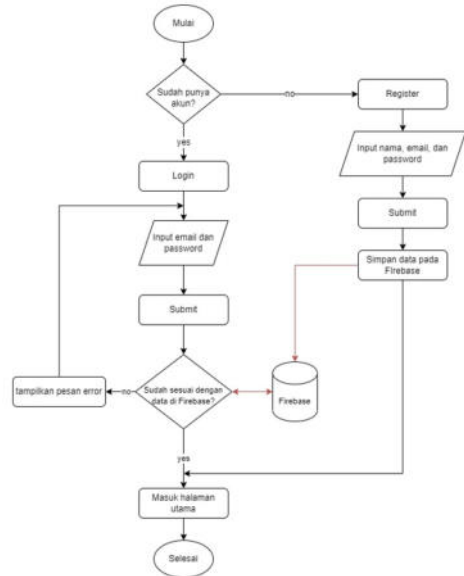
Tombol Login

Tombol Menu Register

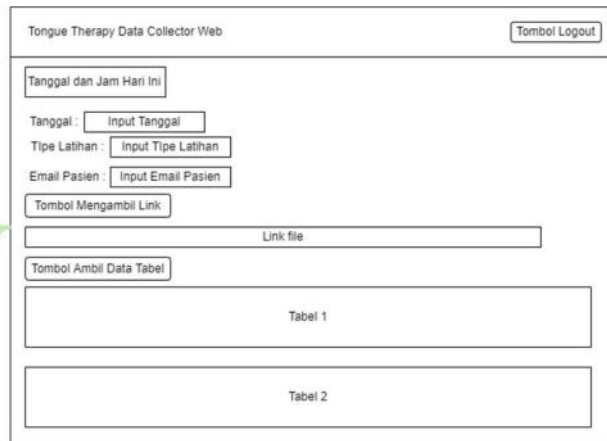
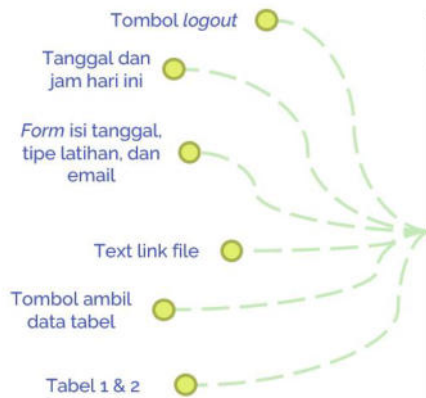
FLOWCHART ALUR KERJA HALAMAN LOGIN DAN REGISTER



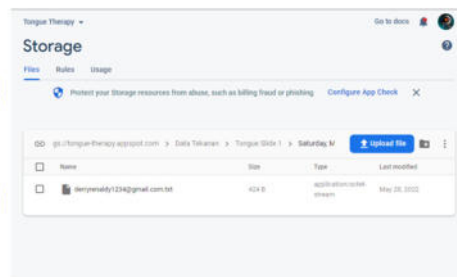
DATABASE AUTHENTICATION FIREBASE



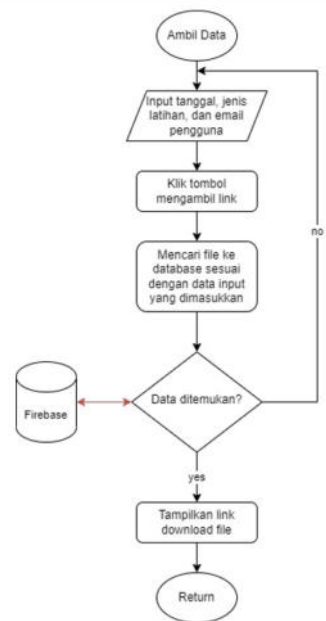
2. RANCANGAN ANTARMUKA HALAMAN UTAMA



FLOWCHART ALUR KERJA HALAMAN UTAMA

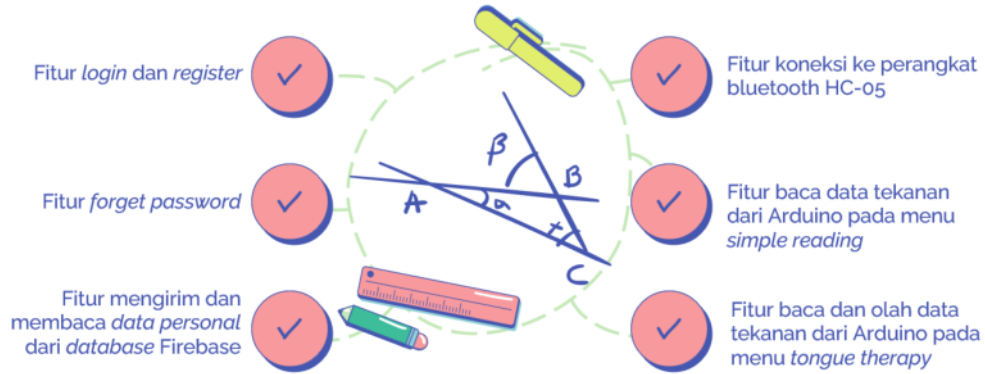


DATABASE STORAGE FIREBASE



d. Uji Coba Aplikasi Android

SKENARIO UJI COBA APLIKASI ANDROID



HASIL UJI COBA FUNGSIONALITAS

Hasil uji coba semua fitur yang telah dijabarkan sebelumnya akan dilakukan dengan **metode kotak hitam**



Metode pengujian kotak hitam atau **black box testing** adalah pengujian yang dilakukan untuk **mengamati hasil input dan output** dari perangkat lunak **tanpa mengetahui structural kode** dari perangkat lunak yang diujikan



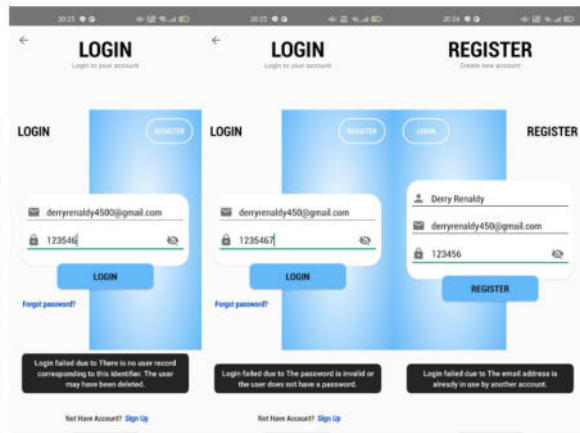
UJI COBA FITUR LOGIN DAN REGISTER

ID	UJI-001
Nama	Uji coba fitur <i>login</i> dan <i>register</i>
Tujuan uji coba	Melakukan pendaftaran akun baru dan <i>login</i> dengan akun yang sudah dibuat
Skenario 1	Aplikasi menampilkan menu profil dan menampilkan email user yang terdaftar
Kondisi awal	Pengguna berada pada menu <i>login</i> atau <i>register</i>
Masukan	Username, Email dan <i>Password</i>
Keluaran yang diharapkan	Menu profil ditampilkan pada layar <i>smartphone</i>
Hasil uji coba	Berhasil
Kondisi akhir	Menu profil berhasil ditampilkan pada layar <i>smartphone</i>



UJI COBA FITUR LOGIN DAN REGISTER

ID	UJI-002
Nama	Uji coba kondisi <i>error</i> fitur <i>login</i> dan <i>register</i>
Tujuan uji coba	Menampilkan pesan <i>error</i> yang sesuai dengan kesalahan yang terjadi
Skenario 1	Aplikasi menampilkan pesan <i>error</i> pada menu <i>login</i> dan <i>register</i>
Kondisi awal	Belum ada <i>error</i>
Masukan	Username, Email dan <i>Password</i>
Keluaran yang diharapkan	Pesan <i>error</i> tertampil pada menu <i>login</i> dan <i>register</i>
Hasil uji coba	Berhasil
Kondisi akhir	Pesan <i>error</i> berhasil tertampil pada menu <i>login</i> dan <i>register</i>



Error karna email tidak terdaftar

Error karna password salah

Error karna email yang digunakan sudah terdaftar

UJI COBA FITUR LOGIN DAN REGISTER

ID	UJI-003
Nama	Uji coba <i>database</i> fitur <i>login</i> dan <i>register</i>
Tujuan uji coba	Data berhasil masuk ke <i>database</i>
Skenario 1	<i>Database</i> menampilkan data yang baru saja di <i>input</i>
Kondisi awal	Belum ada data akun
Masukan	Username, Email dan <i>Password</i>
Keluaran yang diharapkan	Data masuk ke <i>database</i> <i>Firebase</i>
Hasil uji coba	Berhasil
Kondisi akhir	Data berhasil masuk ke <i>database</i> <i>Firebase</i>

DATABASE AUTHENTICATION FIREBASE

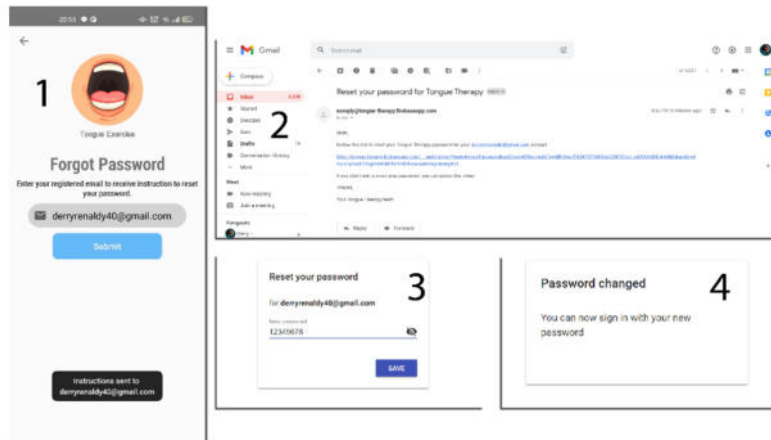
Search by email address, phone number, or user UID					Add user		
Identifier	Providers	Created ↓	Signed In	User UID			
derryrenaldy450@gmail.com	📧	Jun 1, 2022	Jun 1, 2022	byeanjRwGgS2Zca2hoaBkx24XLZ2			

UJI COBA FITUR FORGET PASSWORD

ID	UJI-004
Nama	Uji coba melakukan <i>reset password</i>
Tujuan uji coba	Mengubah <i>password</i> pengguna yang telah terdaftar di <i>database</i>
Skenario 1	<i>Password</i> berhasil di <i>reset</i>
Kondisi awal	<i>Password</i> belum di <i>reset</i>
Masukan	Email
Keluaran yang diharapkan	<i>Password</i> di <i>reset</i>
Hasil uji coba	Berhasil
Kondisi akhir	<i>Password</i> berhasil di <i>reset</i>

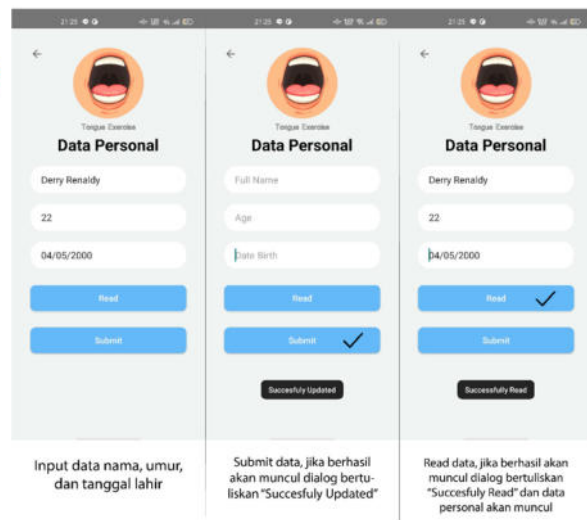
UJI COBA FITUR FORGET PASSWORD

1. Masukkan email yang sudah terdaftar.
2. Cek email, klik link yang diberikan untuk melakukan *reset password*.
3. Masukkan *password* yang baru.
4. *Password* berhasil di ubah.



UJI COBA FITUR MENGIRIM DAN MEMBACA DATA PERSONAL DARI DATABASE FIREBASE

ID	UJI-005
Nama	Uji coba mengirim dan membaca <i>data personal</i> dari <i>database</i> Firebase
Tujuan uji coba	Menyimpan informasi <i>data personal</i> pada <i>database</i> dan menampilkannya kembali di aplikasi
Skenario 1	Aplikasi menampilkan <i>progress dialog</i> dan <i>data personal</i> yang sudah di <i>input</i> pada <i>form</i>
Kondisi awal	<i>Data personal</i> belum di <i>input</i> ke <i>database</i>
Masukan	Nama lengkap, umur, tanggal lahir
Keluaran yang diharapkan	Data masuk ke <i>database</i> , <i>progress dialog</i> muncul, data bisa ditampilkan kembali pada aplikasi
Hasil uji coba	Berhasil
Kondisi akhir	Data berhasil masuk ke <i>database</i> , <i>progress dialog</i> berhasil muncul, dan data berhasil ditampilkan kembali pada aplikasi



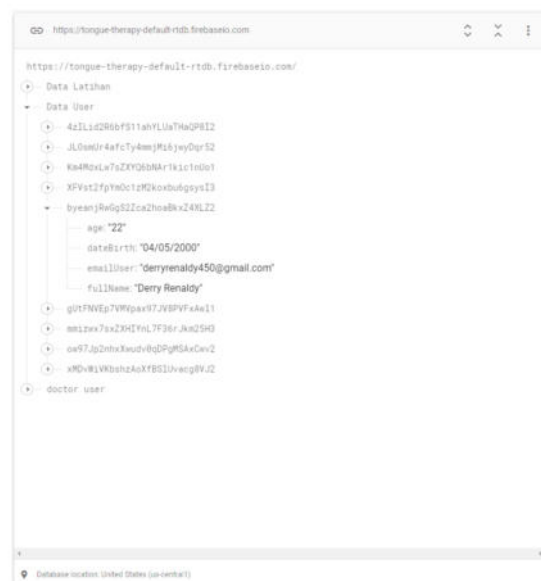
Input data nama, umur, dan tanggal lahir

Submit data, jika berhasil akan muncul dialog bertuliskan "Successfully Updated"

Read data, jika berhasil akan muncul dialog bertuliskan "Successfully Read" dan data personal akan muncul

UJI COBA FITUR MENGIRIM DAN MEMBACA DATA PERSONAL DARI DATABASE FIREBASE

ID	UJI-005
Nama	Uji coba mengirim dan membaca <i>data personal</i> dari <i>database</i> Firebase
Tujuan uji coba	Menyimpan informasi <i>data personal</i> pada <i>database</i> dan menampilkannya kembali di aplikasi
Skenario 1	Aplikasi menampilkan <i>progress dialog</i> dan <i>data personal</i> yang sudah di <i>input</i> pada <i>form</i>
Kondisi awal	<i>Data personal</i> belum di <i>input</i> ke <i>database</i>
Masukan	Nama lengkap, umur, tanggal lahir
Keluaran yang diharapkan	Data masuk ke <i>database</i> , <i>progress dialog</i> muncul, data bisa ditampilkan kembali pada aplikasi
Hasil uji coba	Berhasil
Kondisi akhir	Data berhasil masuk ke <i>database</i> , <i>progress dialog</i> berhasil muncul, dan data berhasil ditampilkan kembali pada aplikasi



UJI COBA FITUR KONEKSI KE PERANGKAT BLUETOOTH HC-05

ID	UJI-006
Nama	Uji coba fitur koneksi ke perangkat bluetooth HC-05
Tujuan uji coba	Menghubungkan <i>smartphone</i> ke perangkat bluetooth HC-05
Skenario 1	Aplikasi menampilkan progress dialog yang menunjukkan <i>smartphone</i> telah terhubung dengan perangkat bluetooth HC-05
Kondisi awal	<i>Smartphone</i> belum terhubung dengan perangkat bluetooth HC-05
Masukan	-
Keluaran yang diharapkan	<i>Smartphone</i> terhubung dengan perangkat bluetooth HC-05
Hasil uji coba	Berhasil
Kondisi akhir	<i>Smartphone</i> berhasil terhubung dengan perangkat bluetooth HC-05

UJI COBA FITUR KONEKSI KE PERANGKAT BLUETOOTH HC-05

List View

Tampilan awal, list view masih kosong

HC-05
00:21:10:01:15:7E
M10BT
12:11:48:26:38:7A

List view berisi daftar semua perangkat bluetooth yang pernah terhubung dengan *smartphone* ketika tombol search ditekan

CLEAR INPUT

Scroll
 Read

START PRESSING STOP PRESSING DISCONNECT PAIRING

Message shows up here ...

Connected to device

SAVE DATA

Progress dialog muncul ketika *smartphone* berhasil terhubung dengan perangkat bluetooth HC-05

HC-05
00:21:10:01:15:7E
M10BT
12:11:48:26:38:7A

Could not connect to device. Is it a Serial device? Also check if the UUID is correct in the settings

Progress dialog muncul ketika *smartphone* gagal terhubung dengan perangkat bluetooth HC-05

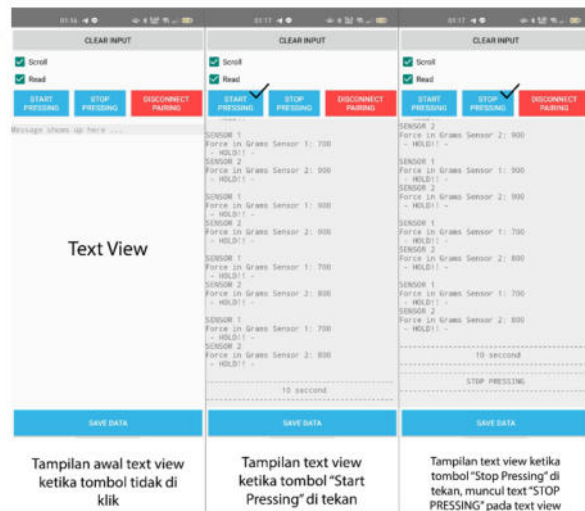
UJI COBA FITUR BACA DATA TEKANAN DARI ARDUINO PADA MENU SIMPLE READING

Pada menu simple reading terdapat beberapa tombol control yang dapat digunakan oleh pengguna yang dapat dijabarkan sebagai berikut :

- Tombol **"Clear Input"** = Digunakan untuk menghapus semua hasil data bacaan yang muncul pada aplikasi.
- Tombol **"Start Pressing"** = Digunakan untuk mengirimkan sinyal state ke Arduino yang akan menjalankan fungsi untuk membaca nilai tekanan yang dihasilkan sensor dan kemudian ditampilkan pada aplikasi.
- Tombol **"Stop Pressing"** = Digunakan untuk mengirimkan sinyal state ke Arduino yang akan menjalankan fungsi untuk memberhentikan proses membaca output sensor dan menampilkan text **"STOP PRESSING"** pada aplikasi.
- Tombol **"Disconnect Pairing"** = Digunakan untuk memutuskan koneksi dengan perangkat bluetooth HC-05 yang terhubung.
- Check Box **"Scroll"** = Page text view pada aplikasi akan melakukan scroll secara otomatis setiap ada data baru yang masuk.
- Check Box **"Read"** = Memberhentikan proses menulis data yang di terima ke dalam *text view*.

UJI COBA FITUR BACA DATA TEKANAN DARI ARDUINO PADA MENU SIMPLE READING

ID	UJI-007
Nama	Uji coba tombol "Start Pressing" dan tombol "Stop Pressing"
Tujuan uji coba	Mengirim perintah ke arduino untuk menjalankan fungsi baca dan berhenti baca melalui tombol pada <i>smartphone</i>
Skenario 1	Aplikasi menampilkan hasil pembacaan sensor dari Arduino ke <i>text view</i> yang dapat dilihat pada layar <i>smartphone</i>
Kondisi awal	<i>Text view</i> masih kosong
Masukan	-
Keluaran yang diharapkan	Data bacaan sensor dari Arduino ditampilkan pada <i>text view</i>
Hasil uji coba	Berhasil
Kondisi akhir	Data bacaan sensor dari Arduino berhasil ditampilkan pada <i>text view</i>



UJI COBA FITUR BACA DAN OLAH DATA TEKANAN DARI ARDUINO PADA MENU TONGUE THERAPY

Pada menu simple reading terdapat beberapa tombol control yang dapat digunakan oleh pengguna yang dapat dijabarkan sebagai berikut :

- Tombol **"Info"** = Membuka halaman menu *info*.
- Tombol **"Start"** = Digunakan untuk mengirimkan sinyal state ke Arduino yang akan menjalankan fungsi untuk membaca nilai tekanan yang dihasilkan sensor dan kemudian ditampilkan pada aplikasi.
- Tombol **"Average"** = Digunakan untuk menghitung rata-rata hasil pembacaan sensor selama 10 detik.
- Tombol **"Reset Reading"** = Digunakan untuk mengulang pembacaan selama 10 detik dan menaikkan perhitungan repetisi.
- Tombol **"Disconnect"** = Digunakan untuk memutuskan koneksi dengan perangkat bluetooth HC-05 yang terhubung.
- Tombol **"Show Result"** = Digunakan untuk menampilkan seluruh hasil pembacaan selama 5 repetisi.
- Tombol **"Save Data"** = Digunakan untuk menyimpan data laporan pelatihan ke dalam *database*.

**UJI COBA FITUR
BACA DAN OLAH DATA TEKANAN DARI ARDUINO PADA
MENU TONGUE THERAPY**

ID	UJI-008
Nama	Uji coba fitur pelatihan pada menu <i>tongue therapy</i>
Tujuan uji coba	Mendapat laporan akhir hasil latihan dan mengirim datanya ke <i>database</i>
Skenario 1	Aplikasi menampilkan laporan akhir latihan pada <i>text view</i> dan dapat mengirim nya ke dalam <i>database</i>
Kondisi awal	Belum dilakukan pelatihan
Masukan	<i>Limit</i> tekanan, jenis gerakan latihan
Keluaran yang diharapkan	<i>Text view</i> menampilkan hasil akhir laporan dan data terkirim ke <i>database</i>
Hasil uji coba	Berhasil
Kondisi akhir	<i>Text view</i> berhasil menampilkan hasil akhir laporan dan data berhasil terkirim ke <i>database</i>

**UJI COBA FITUR
BACA DAN OLAH
DATA TEKANAN
DARI ARDUINO
PADA MENU
TONGUE THERAPY**

00:04 00:06 00:08 00:17

Thursday, June 2, 2022 Thursday, June 2, 2022 Thursday, June 2, 2022 Thursday, June 2, 2022

Current User: *deryrenaldy450@gmail.com* Current User: *deryrenaldy450@gmail.com* Current User: *deryrenaldy450@gmail.com* Current User: *deryrenaldy450@gmail.com*

400 ✓ 400 400 400

Enough Pressure Enough Pressure Enough Pressure Enough Pressure

RESET READING RESET READING RESET READING FINISH

AVERAGE START DISCONNECT AVERAGE START DISCONNECT AVERAGE START DISCONNECT FINISH START DISCONNECT

not reading reading not finished (3) 10 Kali Pembacaan 10 Kali Pembacaan

average pressure: average pressure: average pressure: average pressure:

result sensor 1 result sensor 2 result sensor 1 result sensor 2 result sensor 1 result sensor 2 result sensor 1 result sensor 2

000 000 000 000

000 800 000 000

SAVE DATA SHOW RESULT SAVE DATA SHOW RESULT SAVE DATA SHOW RESULT SAVE DATA SHOW RESULT

Untuk input awal pengguna perlu memasukkan limit tekanan dan memilih jenis gerakan latihan

Tombol "Start" akan memulai proses pelatihan. Pengguna harus terus menekan lidah ke sensor ketika indikator masih bertuliskan "reading not finished"

Tombol "Average" akan menampilkan hasil rata-rata pembacaan selama 10 detik. Tombol "Reset Reading" akan mengulang pembacaan 10 detik dan repetisi meningkat

Saat periode latihan selesai pengguna akan diperintahkan untuk men-klik tombol "Show Result"

**UJI COBA FITUR
BACA DAN OLAH DATA TEKANAN DARI
ARDUINO PADA MENU TONGUE
THERAPY**

00:57 00:57

Thursday, June 2, 2022 Thursday, June 2, 2022

Current User: *deryrenaldy450@gmail.com* Current User: *deryrenaldy450@gmail.com*

FINISH FINISH

AVERAGE START DISCONNECT AVERAGE START DISCONNECT

SENSOR 1 SENSOR 2 SENSOR 1 SENSOR 2

000 gram 000 gram

Average of Sensor 1 = 0 gram Average of Sensor 1 = 0 gram

Success Rate 0% Success Rate 0%

SENSOR 2 SENSOR 2

690 gram 690 gram

660 gram 660 gram

620 gram 620 gram

590 gram 590 gram

660 gram 660 gram

Average of Sensor 2 = 644 gram Average of Sensor 2 = 644 gram

The Pressure Has Reached The Limit The Pressure Has Reached The Limit

Success Rate 100% Success Rate 100%

Upload successful!

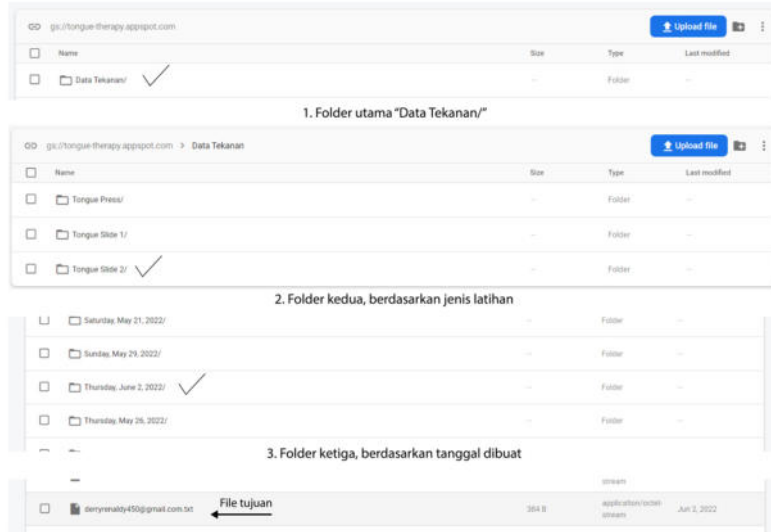
SAVE DATA SHOW RESULT SAVE DATA SHOW RESULT

Hasil latihan akan di tampilkan pada *text view* dan grafik rata-rata tiap repetisi akan muncul ketika pengguna menekan tombol "Show Result"

Progress dialog akan muncul ketika data berhasil dikirim ke dalam *database* storage *Firestore*

**UJI COBA FITUR
BACA DAN OLAH
DATA TEKANAN
DARI ARDUINO
PADA MENU
TONGUE THERAPY**

**DATABASE
STORAGE
FIREBASE**



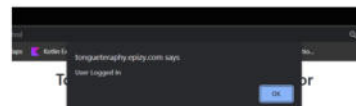
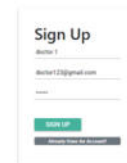
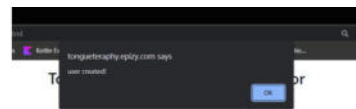
e. Uji Coba Aplikasi Web

SKENARIO UJI COBA APLIKASI WEB

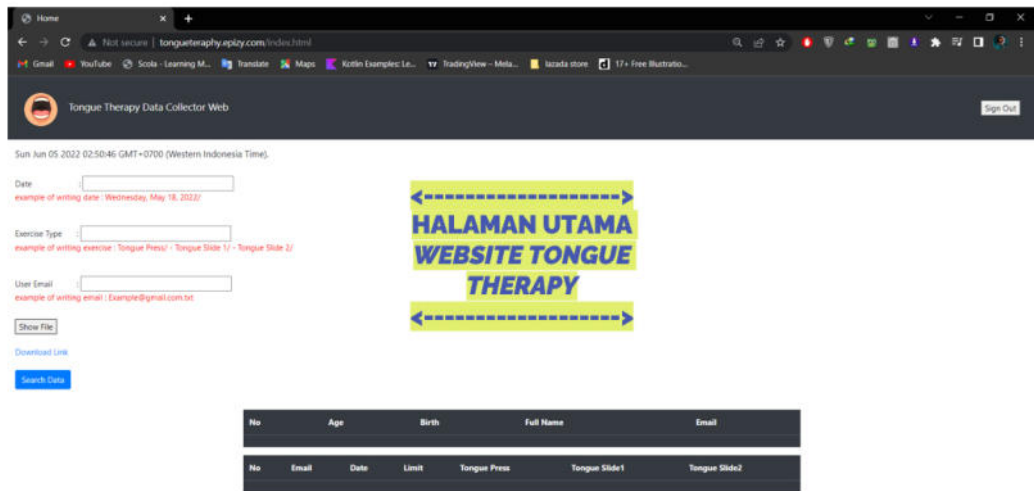


**UJI COBA FITUR
LOGIN DAN REGISTER**

ID	UJI-009
Nama	Uji coba fitur login dan register
Tujuan uji coba	Melakukan pendaftaran akun baru dan login dengan akun yang sudah dibuat
Skenario 1	Website menampilkan notifikasi berhasil melakukan pendaftaran ataupun login dan beralih ke halaman utama ketika berhasil login
Kondisi awal	Pengguna berada pada halaman login atau register
Masukan	Username, Email dan Password
Keluaran yang diharapkan	Halaman login atau register menampilkan notifikasi dan berpindah ke halaman utama ketika berhasil login
Hasil uji coba	Berhasil
Kondisi akhir	Halaman login atau register berhasil menampilkan notifikasi dan berpindah ke halaman utama ketika berhasil login



UJI COBA FITUR LOGIN DAN REGISTER



UJI COBA FITUR AMBIL DATA PASIEN DAN DATA LATIHAN

ID	UJI-010
Nama	Uji coba fitur ambil data pasien dan data latihan dari <i>realtime database</i> Firebase
Tujuan uji coba	Menampilkan data pasien dan data pelatihan yang sudah tersimpan dalam <i>realtime database</i>
Skenario 1	Website menampilkan table yang berisi data pasien dan data pelatihan
Kondisi awal	Tabel masih dalam keadaan kosong
Masukan	-
Keluaran yang diharapkan	Tabel menampilkan data pasien dan juga data pelatihan
Hasil uji coba	Berhasil
Kondisi akhir	Tabel berhasil menampilkan data pasien dan juga data pelatihan

**UJI COBA FITUR
AMBIL DATA PASIEN
DAN DATA LATIHAN**

Sun Jun 05 2022 03:05:46 GMT+0700 (Western Indonesia Time)

Date
example of writing date: Wednesday, May 18, 2022

Exercise Type
example of writing exercise: Tongue Press / Tongue Slide 1 / Tongue Slide 2

User Email
example of writing email: Example@gmail.com

No	Age	Birth	Full Name	Email		
No	Email	Date	Limit	Tongue Press	Tongue Slide 1	Tongue Slide 2

KONDISI AWAL

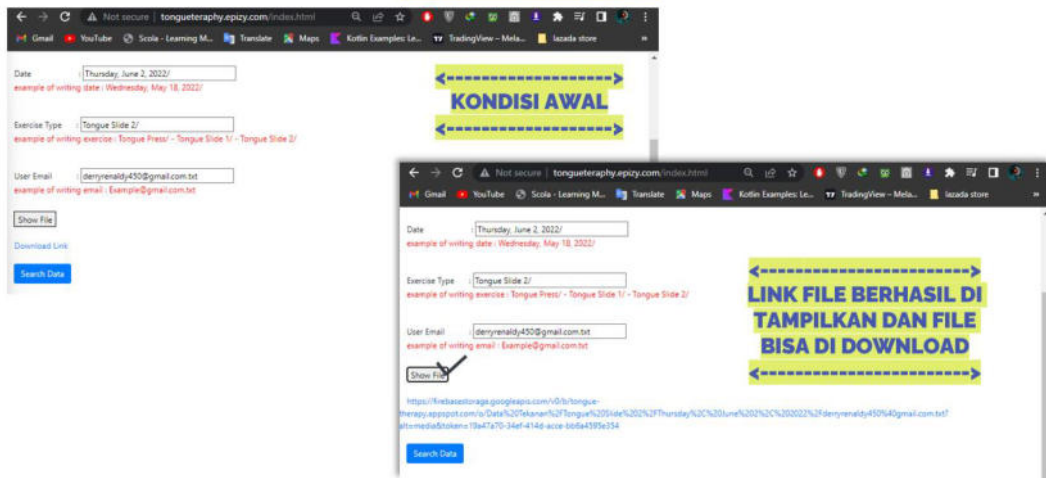
**UJI COBA FITUR
AMBIL DATA PASIEN
DAN DATA LATIHAN**

**ISI DATA DALAM
DATABASE (DATA
PENGGUNA DAN
DATA PELATIHAN)**

**UJI COBA FITUR
AMBIL DAN UNDUH FILE PELATIHAN DARI DATABASE
STORAGE FIREBASE**

ID	UJI-011
Nama	Uji coba fitur ambil dan unduh file pelatihan dari <i>database storage</i> Firebase
Tujuan uji coba	Menampilkan link yang bisa digunakan untuk meng-unduh file pelatihan yang berada dalam <i>database storage</i> Firebase
Skenario 1	Website menampilkan link download file
Kondisi awal	Link download belum muncul
Masukan	-
Keluaran yang diharapkan	Website menampilkan link download file
Hasil uji coba	Berhasil
Kondisi akhir	Website berhasil menampilkan link download file

**UJI COBA FITUR
AMBIL DAN UNDUH FILE PELATIHAN DARI DATABASE
STORAGE FIREBASE**



**UJI COBA FITUR
AMBIL DAN UNDUH FILE
PELATIHAN DARI DATABASE
STORAGE FIREBASE**

**FILE TEXT YANG DI
DOWNLOAD DARI
WEBSITE (BERUPA
FILE TXT)**

$$\frac{a \times b}{x}$$

$$\frac{a \times b}{x}$$

$\sqrt{x-y}$

```

derryrenaldy450@gmail.com.txt - Notepad
File Edit View

Tongue Slide 2
Thursday, June 2, 2022
User = derryrenaldy450@gmail.com

Limit = 400
SENSOR 1
000 gram
000 gram
000 gram
000 gram
000 gram

Average of Sensor 1 = 0 gram
Success Rate 0%
-----

SENSOR 2
690 gram
660 gram
620 gram
590 gram
660 gram

Average of Sensor 2 = 644 gram
The Pressure Has Reached The Limit
Success Rate 100%
-----

Ln 1, Col 1      100%      Unix (LF)      UTF-8
    
```

Program Android Studio

a. Login

```
package com.example.tonguetherapy.common

import android.app.ProgressDialog
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.TextUtils
import android.util.Patterns
import android.widget.Button
import android.widget.Toast
import com.example.tonguetherapy.R
import
com.example.tonguetherapy.databinding.ActivityFormLoginBinding
import com.google.firebase.auth.FirebaseAuth

class FormLogin : AppCompatActivity() {

    //viewBinding
    private lateinit var binding: ActivityFormLoginBinding

    //ActionBar
    //private lateinit var actionBar: ActionBar

    //progressDialog
    private lateinit var progressDialog : ProgressDialog

    //FirebaseAuth
    private lateinit var firebaseAuth : FirebaseAuth
    private var email = ""
    private var password = ""

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityFormLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //configureActionBar
        //actionBar = supportActionBar!!
        // actionBar.title = "Login"

        //configureProgressDialog
        progressDialog = ProgressDialog(this)
        progressDialog.setTitle("Please Wait")
        progressDialog.setMessage("Logging In...")
        progressDialog.setCancelableOnTouchOutside(false)

        //initFirebaseAuth
        firebaseAuth = FirebaseAuth.getInstance()
        checkUser()

        //handle click, open form registrasi
        binding.noAccount.setOnClickListener {
            startActivity(Intent(this, FormRegistrasi::class.java))
            overridePendingTransition(R.anim.slide_from_right,
```

```

R.anim.slide_to_left)
    }

    //handle click, begin login
    binding.buttonLogin.setOnClickListener {
        //before logging in, Validate data
        validateData()
    }

    //back button
    binding.backBtn.setOnClickListener {
        onBackPressed()
    }

    binding.forgotPass.setOnClickListener {
        startActivity(Intent(this, ForgotPassword::class.java))
        overridePendingTransition(R.anim.slide_from_right,
R.anim.slide_to_left)
    }

    val b = findViewById<Button>(R.id.btnRegLogin)
    b.setOnClickListener {
        startActivity(Intent(this,
FormRegistrasi::class.java))
        overridePendingTransition(R.anim.slide_from_right,
R.anim.slide_to_left)
    }
}

private fun validateData() {
    //get data
    email = binding.email.text.toString().trim()
    password = binding.password.text.toString().trim()

    //validate data
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        //invalid email format
        binding.email.error = "Invalid email format"
    }

    else if (TextUtils.isEmpty(password)) {
        //no Password Enter
        binding.password.error = "Please enter password"
    }

    else {
        //data is validated, begin login
        firebaseLogin()
    }
}

private fun firebaseLogin() {
    //show progress
    progressDialog.show()
    firebaseAuth.signInWithEmailAndPassword(email, password)
        .addOnSuccessListener {
            //login success
            progressDialog.dismiss()

```

```

        val firebaseUser = firebaseAuth.currentUser
        val email = firebaseUser!!.email
        Toast.makeText(this, "Logged in as
$email", Toast.LENGTH_SHORT).show()

        //open profile
        startActivity(Intent(this,
ActivityProfile::class.java))
        finish()
    }
    .addOnFailureListener { e->
        //login failed
        progressDialog.dismiss()
        Toast.makeText(this, "Login failed due to
${e.message}", Toast.LENGTH_SHORT).show()
    }
}

private fun checkUser() {
    //if user is already logged in go to profile activity
    //get current user
    val firebaseUser = firebaseAuth.currentUser
    if (firebaseUser != null){
        //user is already legged in
        startActivity(Intent(this,
ActivityProfile::class.java))
        finish()
    }
}
}
}

```

b. Register

```

package com.example.tonguetherapy.common

import android.app.ProgressDialog
import android.content.Intent
import android.os.Bundle
import android.text.TextUtils
import android.util.Patterns
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.tonguetherapy.R
import
com.example.tonguetherapy.databinding.ActivityFormRegistrasiBinding
import com.google.firebase.auth.FirebaseAuth

class FormRegistrasi : AppCompatActivity() {

    //ViewBinding
    private lateinit var binding: ActivityFormRegistrasiBinding

    //ActionBar
    //private lateinit var actionBar: ActionBar
}

```

```

//ProgressDialog
private lateinit var progressDialog : ProgressDialog

//FirebaseAuth
private lateinit var firebaseAuth: FirebaseAuth
private var email = ""
private var password = ""

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding =
    ActivityFormRegistrasiBinding.inflate(layoutInflater)
    setContentView(binding.root)

    //configure actionBar
    //actionBar = supportActionBar!!
    //actionBar.title = "Sign Up"
    //actionBar.setDisplayHomeAsUpEnabled(true)
    //actionBar.setDisplayHomeAsUpEnabled(true)

    //configure progress dialog
    progressDialog = ProgressDialog(this)
    progressDialog.setTitle("Please Wait")
    progressDialog.setMessage("Creating Account In...")
    progressDialog.setCanceledOnTouchOutside(false)

    //init firebase auth
    firebaseAuth = FirebaseAuth.getInstance()

    //handle click, begin sign up
    binding.buttonRegister.setOnClickListener {
        //validate data
        validateData()
    }

    val a = findViewById<Button>(R.id.btnLogRegister)
    a.setOnClickListener{
        onBackPressed()
    }
}

private fun validateData() {
    //get data
    email = binding.email.text.toString().trim()
    password = binding.password.text.toString().trim()

    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        //invalid email format
        binding.email.error = "Invalid email format"
    }

    else if (TextUtils.isEmpty(password)) {
        //no Password Enter
        binding.password.error = "Please enter password"
    }

    else if (password.length <6) {
        //password length <6
        binding.password.error = "Password must atleast 6
character long"

```



```

    }
    else{
        //data is validated, begin login
        firebaseLogin()
    }
}

private fun firebaseLogin() {
    //show progress
    progressDialog.show()

    //create account
    firebaseAuth.createUserWithEmailAndPassword(email,
password)
        .addOnSuccessListener {
            //login success
            progressDialog.dismiss()
            val firebaseUser = firebaseAuth.currentUser
            val email = firebaseUser!!.email
            Toast.makeText(this, "Account created with email
$email", Toast.LENGTH_SHORT).show()

            //open profile
            startActivity(Intent(this,
ActivityProfile::class.java))
            finish()
        }
        .addOnFailureListener { e->
            //login failed
            progressDialog.dismiss()
            Toast.makeText(this, "Login failed due to
${e.message}", Toast.LENGTH_SHORT).show()
        }
    }

    override fun onSupportNavigateUp(): Boolean {
        onBackPressed() //go back to previous activity, when
button of actionbar clicked
        return super.onSupportNavigateUp()
    }

    override fun onBackPressed() {
        super.onBackPressed()
    }

    overridePendingTransition(R.anim.slide_from_left, R.anim.slide_to_r
ight)
}
}

```

c. Forgot Password

```

package com.example.tonguetherapy.common

import android.app.ProgressDialog
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Patterns

```

```

import android.widget.Toast
import
com.example.tonguetherapy.databinding.ActivityForgotPasswordBinding
import com.example.tonguetherapy.mainMenu.MainActivity
import com.google.firebase.auth.FirebaseAuth

class ForgotPassword : AppCompatActivity() {

    //viewBinding
    private lateinit var binding: ActivityForgotPasswordBinding

    //FirebaseAuth
    private lateinit var firebaseAuth: FirebaseAuth
    private var email = ""

    //progress dialog
    private lateinit var progressDialog: ProgressDialog

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding =
ActivityForgotPasswordBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //back button
        binding.backBtn.setOnClickListener {
            startActivity(Intent(this, MainActivity::class.java))
        }

        //init firebaseAuth
        firebaseAuth = FirebaseAuth.getInstance()

        //init/setup progress dialog
        progressDialog = ProgressDialog(this)
        progressDialog.setTitle("Please Wait")
        progressDialog.setCanceledOnTouchOutside(false)

        //handle click, begin password recovery process
        binding.submitBtn.setOnClickListener {
            validateData()
        }
    }

    private fun validateData() {
        //get data
        email = binding.emailEt.text.toString().trim()

        //validateData
        if(email.isEmpty()) {
            Toast.makeText(this, "Please enter your email
address", Toast.LENGTH_SHORT).show()
        }
        else if
(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
            Toast.makeText(this, "Invalid email format",
Toast.LENGTH_SHORT).show()
        }
        else {

```

```

        recoverPassword()
    }
}

private fun recoverPassword() {
    //Show Progress
    progressDialog.setMessage("Sending password reset
Instructions to $email")
    progressDialog.show()

    firebaseAuth.sendPasswordResetEmail(email)
        .addOnSuccessListener {
            //sent
            progressDialog.dismiss()
            Toast.makeText(this,"Instructions sent to
\n$email",Toast.LENGTH_SHORT).show()
        }
        .addOnFailureListener { e->
            //failed
            progressDialog.dismiss()
            Toast.makeText(this,"Failed to send due to
${e.message}",Toast.LENGTH_SHORT).show()
        }
    }
}
}

```

d. Personal Data

```

package com.example.tonguetherapy.personalData;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import com.example.tonguetherapy.R;
import com.example.tonguetherapy.databinding.ActivityMainBinding;
import
com.example.tonguetherapy.databinding.ActivityPersonalDataBinding;
import com.example.tonguetherapy.mainMenu.MainActivity;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class PersonalData extends AppCompatActivity {

```

```

ActivityPersonalDataBinding binding;
String emailUser, fullName, age, dateBirth, uid;

private TextView mTvDataDiri;

private String CurrentUserID;

//Firebase Auth
FirebaseAuth firebaseAuth;
FirebaseUser firebaseUser;

//Realtime Database
DatabaseReference reference;
FirebaseDatabase db;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding =
ActivityPersonalDataBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    binding.btnReadData.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            FirebaseUser currentUser =
FirebaseAuth.getInstance().getCurrentUser();
            if (currentUser!=null){
                String userId = currentUser.getId();
                reference =
FirebaseDatabase.getInstance().getReference("Data User");

reference.child(userId).get().addOnCompleteListener(new
OnCompleteListener<DataSnapshot>() {
                    @Override
                    public void onComplete(@NonNull
Task<DataSnapshot> task) {
                        if (task.isSuccessful()){
                            if (task.getResult().exists()){

Toast.makeText(PersonalData.this,"Successfully
Read",Toast.LENGTH_SHORT).show();

                            dataSnapshot =
task.getResult();

                            String emailUser =
String.valueOf(dataSnapshot.child("emailUser").getValue());
                            String fullName =
String.valueOf(dataSnapshot.child("fullName").getValue());
                            String age =
String.valueOf(dataSnapshot.child("age").getValue());
                            String dateBirth =
String.valueOf(dataSnapshot.child("dateBirth").getValue());

binding.etEmail.setText(emailUser);

binding.etFullName.setText(fullName);

                            binding.etAge.setText(age);

```

```

binding.etDateBirth.setText(dateBirth);
                                }else {

Toast.makeText(PersonalData.this,"User Doesn't
Exist",Toast.LENGTH_SHORT).show();
                                }

                                }else {

Toast.makeText(PersonalData.this,"Failed to
read",Toast.LENGTH_SHORT).show();
                                }
                                }
                                });

                                } else {
                                Toast.makeText(PersonalData.this, "No User
Signed In", Toast.LENGTH_SHORT).show();
                                }
                                }
                                });

                                binding.backBtn.setOnClickListener(new
View.OnClickListener() {
                                @Override
                                public void onClick(View v) {
                                startActivity(new Intent(PersonalData.this,
MainActivity.class));
                                }
                                });

                                binding.registerBtn.setOnClickListener(new
View.OnClickListener() {
                                @Override
                                public void onClick(View v) {

                                FirebaseAuth currentUser =
FirebaseAuth.getInstance().getCurrentUser();

                                fullName =
binding.etFullName.getText().toString();
                                age = binding.etAge.getText().toString();
                                dateBirth =
binding.etDateBirth.getText().toString();
                                uid = binding.tvUid.getText().toString();

                                if (!fullName.isEmpty() && !age.isEmpty()
&& !dateBirth.isEmpty()){

                                if (currentUser != null) {

                                String email =
currentUser.getEmail();
                                binding.etEmail.setText(email);
                                emailUser =
binding.etEmail.getText().toString();

```



```

import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.tonguetherapy.R;
import com.example.tonguetherapy.Bluetooth2.MainActivityBt2;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;
import java.util.UUID;

public class MainActivityBt extends AppCompatActivity {

    private Button search;
    private Button connect;
    private ListView listView;
    private BluetoothAdapter mBTAdapter;
    private static final int BT_ENABLE_REQUEST = 10; // This is
the code we use for BT Enable
    private static final int SETTINGS = 20;
    private static final int REQUEST_DISCOVER_BT = 1;
    private UUID mDeviceUUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");
    private int mBufferSize = 50000; //Default
    public static final String DEVICE_EXTRA =
"com.example.ansensormonitoring.SOCKET";
    public static final String DEVICE_UUID =
"com.example.ansensormonitoring.uuid";
    private static final String DEVICE_LIST =
"com.example.ansensormonitoring.devicelist";
    private static final String DEVICE_LIST_SELECTED =
"com.example.ansensormonitoring.devicelistselected";
    public static final String BUFFER_SIZE =
"com.example.ansensormonitoring.buffersize";
    private static final String TAG = "BlueTest5-MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_bt);

        search = (Button) findViewById(R.id.search);
        connect = (Button) findViewById(R.id.connect);

        listView = (ListView) findViewById(R.id.listview);

        if (savedInstanceState != null) {
            ArrayList<BluetoothDevice> list =
savedInstanceState.getParcelableArrayList(DEVICE_LIST);
            if (list != null) {
                initList(list);
            }
        }
    }
}

```

```

        MyAdapter adapter = (MyAdapter)
listView.getAdapter();
        int selectedIndex =
savedInstanceState.getInt(DEVICE_LIST_SELECTED);
        if (selectedIndex != -1) {
            adapter.setSelectedIndex(selectedIndex);
            connect.setEnabled(true);
        }
    } else {
        initList(new ArrayList<BluetoothDevice>());
    }

} else {
    initList(new ArrayList<BluetoothDevice>());
}
search.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        mBTAdapter = BluetoothAdapter.getDefaultAdapter();

        if (mBTAdapter == null) {
            Toast.makeText(getApplicationContext(),
"Bluetooth not found", Toast.LENGTH_SHORT).show();
        } else if (!mBTAdapter.isEnabled()) {
            Intent enableBT = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBT,
BT_ENABLE_REQUEST);
        } else {
            new SearchDevices().execute();
        }
    }
});

connect.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {

        BluetoothDevice device = ((MyAdapter)
(listView.getAdapter())).getSelectedItem();
        Intent intent = new
Intent(getApplicationContext(), MonitoringScreen.class);
        intent.putExtra(DEVICE_EXTRA, device);
        intent.putExtra(DEVICE_UUID,
mDeviceUUID.toString());
        intent.putExtra(BUFFER_SIZE, mBufferSize);
        startActivity(intent);
    }
});

}

protected void onPause() {
// TODO Auto-generated method stub

```



```

        super.onPause();
    }

    @Override
    protected void onStop() {
// TODO Auto-generated method stub
        super.onStop();
    }

    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        switch (requestCode) {
            case BT_ENABLE_REQUEST:
                if (resultCode == RESULT_OK) {
                    msg("Bluetooth Enabled successfully");
                    new SearchDevices().execute();
                } else {
                    msg("Bluetooth couldn't be enabled");
                }

                break;
            case SETTINGS: //If the settings have been updated
                SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(this);
                String uuid = prefs.getString("prefUuid", "Null");
                mDeviceUUID = UUID.fromString(uuid);
                Log.d(TAG, "UUID: " + uuid);
                String bufSize = prefs.getString("prefTextBuffer",
"Null");
                mBufferSize = Integer.parseInt(bufSize);

                String orientation =
prefs.getString("prefOrientation", "Null");
                Log.d(TAG, "Orientation: " + orientation);
                if (orientation.equals("Landscape")) {
                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE)
;
                } else if (orientation.equals("Portrait")) {
                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
                } else if (orientation.equals("Auto")) {
                    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_FULL_SENSOR);
                }
                break;
            default:
                break;
        }
        super.onActivityResult(requestCode, resultCode, data);
    }

    /**
     * Quick way to call the Toast
     * @param str
     */
    private void msg(String str) {

```

```

        Toast.makeText(getApplicationContext(), str,
Toast.LENGTH_SHORT).show();
    }

    /**
     * Initialize the List adapter
     * @param objects
     */
    private void initList(List<BluetoothDevice> objects) {
        final MyAdapter adapter = new
MyAdapter(getApplicationContext(), R.layout.list_item,
R.id.lstContent, objects);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                adapter.setSelectedIndex(position);
                connect.setEnabled(true);
            }
        });
    }

    /**
     * Searches for paired devices. Doesn't do a scan! Only
     * devices which are paired through Settings->Bluetooth
     * will show up with this. I didn't see any need to re-build
     * the wheel over here
     * @author ryder
     */
    private class SearchDevices extends AsyncTask<Void, Void,
List<BluetoothDevice>> {

        @Override
        protected List<BluetoothDevice> doInBackground(Void...
params) {
            Set<BluetoothDevice> pairedDevices =
mBTAdapter.getBondedDevices();
            List<BluetoothDevice> listDevices = new
ArrayList<BluetoothDevice>();
            for (BluetoothDevice device : pairedDevices) {
                listDevices.add(device);
            }
            return listDevices;
        }

        @Override
        protected void onPostExecute(List<BluetoothDevice>
listDevices) {
            super.onPostExecute(listDevices);
            if (listDevices.size() > 0) {
                MyAdapter adapter = (MyAdapter)
listView.getAdapter();
                adapter.replaceItems(listDevices);
            } else {

```

```

        msg("No paired devices found, please pair your
serial BT device and try again");
    }
}

/**
 * Custom adapter to show the current devices in the list.
 * This is a bit of an overkill for this
 * project, but I figured it would be good learning
 * Most of the code is lifted from somewhere but I can't find
 * the link anymore
 * @author ryder
 */
private class MyAdapter extends ArrayAdapter<BluetoothDevice>
{
    private int selectedIndex;
    private Context context;
    private int selectedColor = Color.parseColor("#abcdef");
    private List<BluetoothDevice> myList;

    public MyAdapter(Context ctx, int resource, int
textViewResourceId, List<BluetoothDevice> objects) {
        super(ctx, resource, textViewResourceId, objects);
        context = ctx;
        myList = objects;
        selectedIndex = -1;
    }

    public void setSelectedIndex(int position) {
        selectedIndex = position;
        notifyDataSetChanged();
    }

    public BluetoothDevice getSelectedItem() {
        return myList.get(selectedIndex);
    }

    @Override
    public int getCount() {
        return myList.size();
    }

    @Override
    public BluetoothDevice getItem(int position) {
        return myList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    private class ViewHolder {
        TextView tv;
    }
}

```

```

        public void replaceItems(List<BluetoothDevice> list) {
            myList = list;
            notifyDataSetChanged();
        }

        public List<BluetoothDevice> getEntireList() {
            return myList;
        }

        @Override
        public View getView(int position, View convertView,
ViewGroup parent) {
            View vi = convertView;
            ViewHolder holder;
            if (convertView == null) {
                vi =
LayoutInflater.from(context).inflate(R.layout.list_item, null);
                holder = new ViewHolder();

                holder.tv = (TextView)
vi.findViewById(R.id.lstContent);

                vi.setTag(holder);
            } else {
                holder = (ViewHolder) vi.getTag();
            }

            if (selectedIndex != -1 && position == selectedIndex)
{
                holder.tv.setBackgroundColor(selectedColor);
            } else {
                holder.tv.setBackgroundColor(Color.WHITE);
            }
            BluetoothDevice device = myList.get(position);
            holder.tv.setText(device.getName() + "\n " +
device.getAddress());

            return vi;
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is
present.
//getMenuInflater().inflate(R.menu.homescreen, menu);
return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.action_settings:
                Intent intent = new Intent(MainActivityBt.this,
com.example.tonguetherapy.Bluetooth.PreferencesActivity.class);
                startActivityForResult(intent, SETTINGS);
                break;
        }
    }
}

```

```

        return super.onOptionsItemSelected(item);
    }
}

```

f. Menu “Simple Reading”

```

package com.example.tonguetherapy.Bluetooth;

import android.Manifest;
import android.app.ProgressDialog;
import android.Bluetooth.BluetoothAdapter;
import android.Bluetooth.BluetoothDevice;
import android.Bluetooth.BluetoothSocket;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.tonguetherapy.R;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.series.DataPoint;
import com.jjoe64.graphview.series.LineGraphSeries;

import java.io.IOException;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import java.util.UUID;

public class MonitoringScreen extends AppCompatActivity {

    Button btn1, btn3, btnSave, btnDis;

```

```

private static final String TAG = "BlueTest5-MainActivity";
private int mMaxChars = 50000;//Default
private UUID mDeviceUUID;
private BluetoothSocket mBTSocket;
private ReadInput mReadThread = null;
StorageReference reference;

private boolean mIsUserInitiatedDisconnect = false;

// All controls here
private TextView mTxtReceive;
private Button mBtnClearInput;
private ScrollView scrollView;
private CheckBox chkScroll;
private boolean mIsBluetoothConnected = false;
private BluetoothDevice mDevice;
private ProgressDialog progressDialog;
private CheckBox chkReceiveText;

//Realtime Database
DatabaseReference mDatabase;
DatabaseReference mDataGraph;

int sum = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_monitoring_screen);

    if
(ContextCompat.checkSelfPermission(MonitoringScreen.this,
Manifest.permission.BLUETOOTH_CONNECT) ==
PackageManager.PERMISSION_DENIED) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
ActivityCompat.requestPermissions(MonitoringScreen.this, new
String[]{Manifest.permission.BLUETOOTH_CONNECT}, 10);
            return;
        }
    }

    //id declaration
    btn1 = findViewById(R.id.button2);
    btn3 = findViewById(R.id.button5);
    btnDis = findViewById(R.id.button4);
    btnSave = findViewById(R.id.buttonSend);
    reference =
FirebaseStorage.getInstance().getReference().child("Document");

    ActivityHelper.initialize(this);

    Intent intent = getIntent();
    Bundle b = intent.getExtras();
    mDevice = b.getParcelable(MainActivityBt.DEVICE_EXTRA);
    mDeviceUUID =
UUID.fromString(b.getString(MainActivityBt.DEVICE_UUID));
    mMaxChars = b.getInt(MainActivityBt.BUFFER_SIZE);
    Log.d(TAG, "Ready");
}

```

```

        mTxtReceive = (TextView) findViewById(R.id.txtReceive);
        chkScroll = (CheckBox) findViewById(R.id.chkScroll);
        chkReceiveText = (CheckBox)
findViewById(R.id.chkReceiveText);
        scrollView = (ScrollView) findViewById(R.id.viewScroll);
        mBtnClearInput = (Button)
findViewById(R.id.btnClearInput);
        mTxtReceive.setMovementMethod(new
ScrollingMovementMethod());

        if
(ContextCompat.checkSelfPermission(MonitoringScreen.this,
Manifest.permission.BLUETOOTH_CONNECT) ==
PackageManager.PERMISSION_DENIED) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
ActivityCompat.requestPermissions(MonitoringScreen.this, new
String[]{Manifest.permission.BLUETOOTH_CONNECT}, 10);
                return;
            }
        }

        //Get a reference to our firebase realtime database
        mDatabase = FirebaseDatabase.getInstance().getReference();
        mDataGraph = mDatabase.child("pembacaan_sensor");

        mDataGraph.push().child("sensor
1").child("tekanan").setValue(500);

        //button control
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View v) {
                sendSignal("0");
            }
        });

        btn3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View v) {
                sendSignal("3");
            }
        });

        btnDis.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View v) {
                Disconnect();
            }
        });

        mBtnClearInput.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                mTxtReceive.setText("");
            }
        });

        btnSave.setOnClickListener(new OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            //progressDialog =
            ProgressDialog.show(MonitoringScreen.this, "Upload File",
            "Uploading text file...");
            String data= mTxtReceive.getText().toString();

            reference.child("file.txt").putBytes(data.getBytes()).addOnSuccess
            Listener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot
            taskSnapshot) {
                    //progressDialog.dismiss();
                    Toast.makeText(MonitoringScreen.this,
            "Upload successful!", Toast.LENGTH_SHORT).show();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    //exception.printStackTrace();
                    //progressDialog.dismiss();
                    Toast.makeText(MonitoringScreen.this,
            e.toString(), Toast.LENGTH_SHORT).show();
                }
            });
        });
    }

    //Button Function
    private void sendSignal ( String number ) {
        if ( mBTSocket!= null ) {
            try {
                mBTSocket.getOutputStream().write(number.toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }

    private void Disconnect () {
        if ( mBTSocket!=null ) {
            try {
                mBTSocket.close();
            } catch(IOException e) {
                msg("Error");
            }
        }
    }

    finish();
}

//New Threads
private class ReadInput implements Runnable {
    private boolean bStop = false;

```



```

private Thread t;

public ReadInput() {
    t = new Thread(this, "Input Thread");
    t.start();
}

public boolean isRunning() {
    return t.isAlive();
}

@Override
public void run() {
    InputStream inputStream;

    try {
        inputStream = mBTSocket.getInputStream();
        while (!bStop) {
            byte[] buffer = new byte[256];
            if (inputStream.available() > 0) {
                inputStream.read(buffer);
                int i = 0;
                /*
                 * This is needed because new
String(buffer) is taking the entire buffer i.e. 256 chars on
Android 2.3.4 http://stackoverflow.com/a/8843462/1287554
                 */
                for (i = 0; i < buffer.length &&
buffer[i] != 0; i++) {
                }
                final String strInput = new String(buffer,
0, i);

                /*
                 * If checked then receive text, better
design would probably be to stop thread if unchecked and free
resources, but this is a quick fix
                 */

                if (chkReceiveText.isChecked()) {
                    mTxtReceive.post(new Runnable() {
                        @Override
                        public void run() {
                            mTxtReceive.append(strInput);

                            int txtLength =
mTxtReceive.getEditableText().length();
                            if (txtLength > mMaxChars) {
                                mTxtReceive.getEditableText().delete(0, txtLength - mMaxChars);
                            }

                            if (chkScroll.isChecked())
                                { // Scroll only if this is checked
                                    scrollView.post(new
Runnable() { // Snippet from
http://stackoverflow.com/a/4612082/1287554
                                        @Override

```

```

public void run() {
scrollView.fullScroll(View.FOCUS_DOWN);
}
});
}
});
}
}
Thread.sleep(500);
}
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (InterruptedException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}

public void stop() {
bStop = true;
}
}

private class DisconnectBT extends AsyncTask<Void, Void, Void>
{

@Override
protected void onPreExecute() {
}

@Override
protected Void doInBackground(Void... params) {

if (mReadThread != null) {
mReadThread.stop();
while (mReadThread.isRunning())
; // Wait until it stops
mReadThread = null;
}

try {
mBTSocket.close();
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

return null;
}

@Override

```

```

        protected void onPostExecute(Void result) {
            super.onPostExecute(result);
            mIsBluetoothConnected = false;
            if (mIsUserInitiatedDisconnect) {
                finish();
            }
        }

    }

    private void msg(String s) {
        Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onPause() {
        if (mBTSocket != null && mIsBluetoothConnected) {
            new DisconnectBT().execute();
        }
        Log.d(TAG, "Paused");
        super.onPause();
    }

    @Override
    protected void onResume() {
        if (mBTSocket == null || !mIsBluetoothConnected) {
            new ConnectBT().execute();
        }
        Log.d(TAG, "Resumed");
        super.onResume();
    }

    @Override
    protected void onStop() {
        Log.d(TAG, "Stopped");
        super.onStop();
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
// TODO Auto-generated method stub
        super.onSaveInstanceState(outState);
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void> {
        private boolean mConnectSuccessful = true;

        @Override
        protected void onPreExecute() {
            progressDialog =
ProgressDialog.show(MonitoringScreen.this, "Hold on",
"Connecting");// http://stackoverflow.com/a/11130220/1287554
        }

        @Override

```

```

        protected Void doInBackground(Void... devices) {

            try {
                if (mBTSocket == null || !mIsBluetoothConnected) {
                    mBTSocket =
mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);

//BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                    mBTSocket.connect();
                }
            } catch (IOException e) {
// Unable to connect to device
                e.printStackTrace();
                mConnectSuccessful = false;
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            super.onPostExecute(result);

            if (!mConnectSuccessful) {
                Toast.makeText(getApplicationContext(), "Could not
connect to device. Is it a Serial device? Also check if the UUID
is correct in the settings", Toast.LENGTH_LONG).show();
                finish();
            } else {
                msg("Connected to device");
                mIsBluetoothConnected = true;
                mReadThread = new ReadInput(); // Kick off input
reader
            }

            progressDialog.dismiss();
        }
    }
}

```

g. Menu “Tongue Therapy”

```

package com.example.tonguetherapy.Bluetooth2;

import android.app.ProgressDialog;
import android.Bluetooth.BluetoothAdapter;
import android.Bluetooth.BluetoothDevice;
import android.Bluetooth.BluetoothSocket;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.media.AudioManager;
import android.media.ToneGenerator;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;

```

```

import android.text.Editable;
import android.text.SpannableString;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.text.method.ScrollingMovementMethod;
import android.text.style.ForegroundColorSpan;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.tonguetherapy.R;
import com.example.tonguetherapy.common.ActivityProfile;
import com.example.tonguetherapy.personalData.PersonalData;
import com.example.tonguetherapy.personalData.Users;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.jjoe64.graphview.GraphView;
import com.jjoe64.graphview.series.BarGraphSeries;
import com.jjoe64.graphview.series.DataPoint;
import com.jjoe64.graphview.series.LineGraphSeries;

import java.io.IOException;
import java.io.InputStream;
import java.text.DateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.UUID;

public class MonitoringScreen2 extends AppCompatActivity {

    private Vibrator vibrator;
    Button btnSave, btnDis, btnGraph, btnAverage, btnTpExercise,
    btnTsExercise1, btnTsExercise2, btnInfo;
    String dataTanggal, dataStatus1, dataStatus2, dataStatus3,
    dataLimit, dataEmail;

    private static final String TAG = "BlueTest5-MainActivity";

```

```

private int mMaxChars = 50000;//Default
private UUID mDeviceUUID;
private BluetoothSocket mBTSocket;
private ReadInput mReadThread = null;
StorageReference reference;

private boolean mIsUserInitiatedDisconnect = false;

//graph
GraphView graphView, graphView2;

//FirebaseAuth
FirebaseAuth mAuth;

// All controls here
private TextView mTxtBantuan, mTvString1, mTvString2,
mTvStringResult1, mTvStringResult2, mTvRepetisi,
mTvTanggal, mTvStatusPembacaan, mTvUser, mTvReport,
mTvRataTekanan, mTvButtonExercise, mTvStatusTekanan,
mTvBatasTekanan,
mTvDataStatus1, mTvDataStatus2, mTvDataStatus3;

private ScrollView scrollView2;
private EditText mTxtReceive,mEtBatas;
private Button mBtnClearInput, mBtnReport;
private boolean mIsBluetoothConnected = false;
private BluetoothDevice mDevice;
private ProgressDialog progressDialog;

//Realtime Database
DatabaseReference references;
FirebaseDatabase db;

int sum = 0;
int sum2 = 0;
int i = 0;
int r = 1;
int x = 0;
int y = 0;
String[] myStringArray = new String[6];
String[] myStringArray2 = new String[6];

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_monitoring_screen2);

    //Vibrate
    vibrator = (Vibrator) getSystemService(VIBRATOR_SERVICE);

    //Grafik
    graphView = findViewById(R.id.graph);
    graphView2 = findViewById(R.id.graph2);

    //FirebaseAuth
    mAuth = FirebaseAuth.getInstance();

    //id declaration
    btnDis = findViewById(R.id.button4);

```

```

        btnSave = findViewById(R.id.buttonSend);
        btnGraph = findViewById(R.id.button3);
        btnAverage = findViewById(R.id.buttonAverage);
        mBtnReport = findViewById(R.id.buttonReport);
        btnTpExercise = findViewById(R.id.btnTonguePress);
        btnTsExercise1 = findViewById(R.id.btnTongueSlide1);
        btnTsExercise2 = findViewById(R.id.btnTongueSlide2);
        btnInfo = findViewById(R.id.btnInfo);

        //-----FOLDER
SIMPANAN-----//
        reference =
        FirebaseStorage.getInstance().getReference().child("Data
Tekanan");

        ActivityHelper2.initialize(this);

        Intent intent = getIntent();
        Bundle b = intent.getExtras();
        mDevice = b.getParcelable(MainActivityBt2.DEVICE_EXTRA);
        mDeviceUUID =
        UUID.fromString(b.getString(MainActivityBt2.DEVICE_UUID));
        mMaxChars = b.getInt(MainActivityBt2.BUFFER_SIZE);
        Log.d(TAG, "Ready");
        mTxtReceive = (EditText) findViewById(R.id.txtReceive);
        mEtBatas = (EditText) findViewById(R.id.etBatas);
        mTxtBantuan = (TextView) findViewById(R.id.textBantuan);
        mTvString1 = (TextView) findViewById(R.id.tvString1);
        mTvString2 = (TextView) findViewById(R.id.tvString2);
        mTvStringResult1 = (TextView)
        findViewById(R.id.tvStringResult1);
        mTvStringResult2 = (TextView)
        findViewById(R.id.tvStringResult2);
        mTvRepetisi = (TextView) findViewById(R.id.tvRepetisi);
        mTvTanggal = (TextView) findViewById(R.id.tvTanggal);
        mTvUser = (TextView) findViewById(R.id.tvUser);
        mTvReport = (TextView) findViewById(R.id.tvReport);
        mTvRataTekanan = (TextView)
        findViewById(R.id.tvRataTekanan);
        mTvButtonExercise = (TextView)
        findViewById(R.id.tvButtonExercise);
        mTvStatusTekanan = (TextView)
        findViewById(R.id.tvStatusTekanan);
        mTvDataStatus1 = (TextView)
        findViewById(R.id.tvDataStatus1);
        mTvDataStatus2 = (TextView)
        findViewById(R.id.tvDataStatus2);
        mTvDataStatus3 = (TextView)
        findViewById(R.id.tvDataStatus3);
        scrollView2 = (ScrollView) findViewById(R.id.viewScroll2);
        mTvStatusPembacaan = (TextView)
        findViewById(R.id.tvStatusPembacaan);
        mTvBatasTekanan = (TextView)
        findViewById(R.id.tvBatasTekanan);
        mBtnClearInput = (Button)
        findViewById(R.id.btnClearInput);
        mTvReport.setMovementMethod(new
        ScrollingMovementMethod());

```

```

//-----INITIAL-----
-----//
mTvReport.setVisibility(View.GONE);
btnSave.setVisibility(View.GONE);
btnAverage.setEnabled(false);
btnGraph.setEnabled(false);
mBtnClearInput.setEnabled(false);
graphView.setVisibility(View.GONE);
graphView2.setVisibility(View.GONE);
//-----BUTTON
INFO-----//
btnInfo.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        MyDialogFragment myDialogFragment = new
MyDialogFragment();
myDialogFragment.show(getSupportFragmentManager(), "MyFragment");
    }
});

//-----GET USER-----
-----//

FirebaseUser user = mAuth.getCurrentUser();
if (user!=null){
    String userEmail = user.getEmail();

    mTvUser.setText(userEmail);
} else {
    mTvUser.setText("No User Signed In");
}

//-----Date-----
-----//

Date currentTime = Calendar.getInstance().getTime();
String formattedDate =
DateFormat.getDateInstance(DateFormat.FULL).format(currentTime);
mTvTanggal.setText(formattedDate);

//-----BUTTON
EXERCISE-----//
btnTpExercise.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        btnAverage.setEnabled(true);
        btnGraph.setEnabled(true);
        mBtnClearInput.setEnabled(true);
        btnTsExercise1.setEnabled(false);
        btnTsExercise2.setEnabled(false);

        String tpExercise =
btnTpExercise.getText().toString();
        mTvButtonExercise.setText(tpExercise);
    }
});

```



```

        btnTsExercise1.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                btnAverage.setEnabled(true);
                btnGraph.setEnabled(true);
                mBtnClearInput.setEnabled(true);
                btnTpExercise.setEnabled(false);
                btnTsExercise2.setEnabled(false);

                String tsExercise1 =
btnTsExercise1.getText().toString();
                mTvButtonExercise.setText(tsExercise1);
            }
        });

        btnTsExercise2.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                btnAverage.setEnabled(true);
                btnGraph.setEnabled(true);
                mBtnClearInput.setEnabled(true);
                btnTpExercise.setEnabled(false);
                btnTsExercise1.setEnabled(false);

                String tsExercise2 =
btnTsExercise2.getText().toString();
                mTvButtonExercise.setText(tsExercise2);
            }
        });

        //-----BUTTON-----
        //-----//

        //graph function
        btnGraph.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick (View v) {

mTvBatasTekanan.setText(mEtBatas.getText().toString());
                String setBatas = mEtBatas.getText().toString();
                if(TextUtils.isEmpty(setBatas)) {
                    mEtBatas.setError("This Field Can't Be
Empty");
                }
                return;
            }
            else{
                //disable button
                btnGraph.setEnabled(false);
                btnSave.setEnabled(false);
                mBtnReport.setEnabled(false);
                //send signal
                sendSignal("4");
                //inputGrafik
                mTxtReceive.addTextChangedListener(new
TextWatcher() {
                    @Override
                    public void beforeTextChanged(CharSequence
s, int start, int count, int after) {

```

```

    }

    @Override
    public void onTextChanged(CharSequence s,
int start, int before, int count) {
        String data =
mTxtReceive.getEditableText().toString();
        mTxtBantuan.setText(data);
        if (r < 6) {
            mTvRepetisi.setText("Repetition ("
+ r + ")");

//mTvRepetisi.setTextColor(Color.parseColor("#FF0000"));
            if (i < 10) {

mBtnClearInput.setEnabled(false);

mTvStatusPembacaan.setText("reading not finished " + "(" + i +
")");

                //SENSOR 1
                char ch1 =
mTxtBantuan.getText().toString().charAt(0);

mTvString1.setText(String.valueOf(ch1));
                sum = sum +
Integer.parseInt(mTvString1.getText().toString());

mTvStringResult1.setText(Integer.toString(sum) + "00");

                //SENSOR 2
                char ch2 =
mTxtBantuan.getText().toString().charAt(5);

mTvString2.setText(String.valueOf(ch2));
                sum2 = sum2 +
Integer.parseInt(mTvString2.getText().toString());

mTvStringResult2.setText(Integer.toString(sum2) + "00");

                i++;
            } else if (i == 10) {
                mTvStatusPembacaan.setText("10
Kali Pembacaan");
            }
        } else if (r > 4) {
            mBtnClearInput.setText("FINISH");
            mTvRepetisi.setText("Please Click
the Show Result Button");

            mBtnReport.setEnabled(true);
            btnAverage.setEnabled(false);

mTvRepetisi.setTextColor(Color.parseColor("#FF0000"));
        }
    }

//-----STATUS TEKANAN-----
//
    int batass =
Integer.parseInt(mEtBatas.getText().toString());

```

```

        if (batass > 0) {
            if
(mTvButtonExercise.getText().toString().equals("Tongue Press")) {
mTvStringResult2.setTextColor(Color.parseColor("#008000"));
                int pembedingtp =
Integer.parseInt(mTvString2.getText().toString()) * 100;
                int pembedingBatas =
Integer.parseInt(mEtBatas.getText().toString());

                if (pembedingtp >=
pembedingBatas) {
mTvStatusTekanan.setText("Enough Pressure");
mTvStatusTekanan.setTextColor(Color.parseColor("#008000"));
                }
                else {
mTvStatusTekanan.setText("Not Enough Pressure!");
mTvStatusTekanan.setTextColor(Color.parseColor("#FF0000"));

                ToneGenerator toneGen1 =
new ToneGenerator(AudioManager.STREAM_MUSIC, 100);
toneGen1.startTone(ToneGenerator.TONE_CDMA_PIP, 150);
                }
            }
            else if
(mTvButtonExercise.getText().toString().equals("Tongue Slide
1")) ){
mTvStringResult1.setTextColor(Color.parseColor("#008000"));
                int pembedingts1 =
Integer.parseInt(mTvString1.getText().toString()) * 100;
                int pembedingBatas =
Integer.parseInt(mEtBatas.getText().toString());

                if (pembedingts1 >=
pembedingBatas) {
mTvStatusTekanan.setText("Enough Pressure");
mTvStatusTekanan.setTextColor(Color.parseColor("#008000"));
                }
                else {
mTvStatusTekanan.setText("Not Enough Pressure!");
mTvStatusTekanan.setTextColor(Color.parseColor("#FF0000"));

                ToneGenerator toneGen1 =
new ToneGenerator(AudioManager.STREAM_MUSIC, 100);
toneGen1.startTone(ToneGenerator.TONE_CDMA_PIP, 150);
                }
            }
        }
    }
}

```

```

                else if
(mTvButtonExercise.getText().toString().equals("Tongue Slide
2")) ){

mTvStringResult2.setTextColor(Color.parseColor("#008000"));
                int pembedingts2 =
Integer.parseInt(mTvString2.getText().toString()) * 100;
                int pembedingBatas =
Integer.parseInt(mEtBatas.getText().toString());

                if(pembedingts2 >=
pembedingBatas){

mTvStatusTekanan.setText("Enough Pressure");

mTvStatusTekanan.setTextColor(Color.parseColor("#008000"));
                }
                else{

mTvStatusTekanan.setText("Not Enough Pressure!");

mTvStatusTekanan.setTextColor(Color.parseColor("#FF0000"));

                ToneGenerator toneGen1 =
new ToneGenerator(AudioManager.STREAM_MUSIC, 100);

toneGen1.startTone(ToneGenerator.TONE_CDMA_PIP,150);
                }
            }
        }

        @Override
        public void afterTextChanged(Editable s) {

        }

    });

}

});

//button control
btnAverage.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick (View v) {
        Average();
        btnAverage.setEnabled(false);
        mBtnClearInput.setEnabled(true);
    }
});

btnDis.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick (View v) {

```

```

        Disconnect();
        r = 1;
    }
});

mBtnClearInput.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        //Average();
        sum = 0;
        sum2 = 0;
        i = 0;
        r++;
        mTvStringResult1.setText("0");

mTvStringResult1.setTextColor(Color.parseColor("#FF000000"));
        mTvStringResult2.setText("0");

mTvStringResult2.setTextColor(Color.parseColor("#FF000000"));
        btnAverage.setEnabled(true);
    }
});

mBtnReport.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        mEtBatas.setText("0");
        btnTsExercise2.setEnabled(false);
        btnTsExercise1.setEnabled(false);
        btnTpExercise.setEnabled(false);
        Report();
        btnSave.setEnabled(true);
        btnAverage.setEnabled(false);
        mBtnReport.setEnabled(false);
        mEtBatas.setVisibility(View.GONE);
        mTxtReceive.setVisibility(View.GONE);
        mTvReport.setVisibility(View.VISIBLE);
        mTvStatusPembacaan.setVisibility(View.GONE);
        mTvRataTekanan.setVisibility(View.GONE);
        mTvStringResult1.setVisibility(View.GONE);
        mTvStringResult2.setVisibility(View.GONE);
        mTvRepetisi.setVisibility(View.GONE);
        mTvStatusTekanan.setVisibility(View.GONE);
        graphView.setVisibility(View.VISIBLE);
        graphView2.setVisibility(View.VISIBLE);
        btnSave.setVisibility(View.VISIBLE);
        btnTpExercise.setVisibility(View.GONE);
        btnTsExercise1.setVisibility(View.GONE);
        btnTsExercise2.setVisibility(View.GONE);
    }
});

btnSave.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        //progressDialog =

```

```

ProgressDialog.show(MonitoringScreen.this, "Upload File",
"Uploading text file...");
        btnAverage.setEnabled(false);
        btnSave.setEnabled(false);

        String latihan =
mTvButtonExercise.getText().toString();
        String data = mTvReport.getText().toString();
        String user = mTvUser.getText().toString();
        String tanggal = mTvTanggal.getText().toString();
        reference.child(latihan).child(tanggal).child(user
+ ".txt").putBytes(data.getBytes()).addOnSuccessListener(new
OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot
taskSnapshot) {
                //progressDialog.dismiss();
                Toast.makeText(MonitoringScreen2.this,
"Upload successful!", Toast.LENGTH_SHORT).show();

                //dataLatihan
                mEtBatas.setText("0");
                FirebaseAuth currentUser = FirebaseAuth.getInstance().getCurrentUser();
                dataTanggal =
mTvTanggal.getText().toString();
                dataStatus1 =
mTvDataStatus1.getText().toString();
                dataStatus2 =
mTvDataStatus2.getText().toString();
                dataStatus3 =
mTvDataStatus3.getText().toString();
                dataLimit =
mTvBatasTekanan.getText().toString();
                dataEmail = mTvUser.getText().toString();

                if (currentUser!=null) {

                }else {
                    Toast.makeText(MonitoringScreen2.this,
"User Not Signed In", Toast.LENGTH_SHORT).show();
                }

                DataLatihan dataLatihan = new
DataLatihan(dataTanggal, dataStatus1, dataStatus2, dataStatus3,
dataLimit, dataEmail);
                db = FirebaseDatabase.getInstance();
                references = db.getReference("Data
Latihan");

                references.child(currentUser.getId()).setValue(dataLatihan).addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull
Task<Void> task) {
                        mTvTanggal.setText(dataTanggal);
                        mTvDataStatus1.setText(dataStatus1);

```

```

mTvDataStatus2.setText (dataStatus2);

mTvDataStatus3.setText (dataStatus3);
        mEtBatas.setText (dataLimit);
        mTvUser.setText (dataEmail);
    }
    });
    }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            //exception.printStackTrace();
            //progressDialog.dismiss();
            Toast.makeText(MonitoringScreen2.this,
e.toString(), Toast.LENGTH_SHORT).show();
        }
    });
    }

});

}

//-----FUNCTION-----
-----//

private void sendSignal ( String number ) {
    if ( mBTSocket!= null ) {
        try {
mBTSocket.getOutputStream().write(number.toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
}

private void Disconnect () {
    if ( mBTSocket!=null ) {
        try {
            mBTSocket.close();
        } catch (IOException e) {
            msg("Error");
        }
    }

    finish();
}

private void Average () {

    sum = sum*100 / 10;
    sum2 = sum2*100 / 10;
    mTvStringResult1.setText(sum + " gram");

mTvStringResult1.setTextColor(Color.parseColor("#FF0000"));

    mTvStringResult2.setText(sum2 + " gram");
}

```

```

mTvStringResult2.setTextColor(Color.parseColor("#FF0000"));

    if(r < 6){
        String value = String.valueOf(sum);
        String value2 = String.valueOf(sum2);

        myStringArray[r] = value;
        myStringArray2[r] = value2;
    }else {
        //Toast.makeText(MonitoringScreen2.this, "Done",
Toast.LENGTH_SHORT).show();
        mBtnClearInput.setEnabled(false);
    }
}

private void Report() {

    r = 1;
    mTvRepetisi.setTextColor(Color.parseColor("#FF000000"));

    //Graphic 1
    BarGraphSeries<DataPoint> series = new
BarGraphSeries<DataPoint>(new DataPoint[] {
        new DataPoint(1,
Double.parseDouble(myStringArray[1])),
        new DataPoint(2,
Double.parseDouble(myStringArray[2])),
        new DataPoint(3,
Double.parseDouble(myStringArray[3])),
        new DataPoint(4,
Double.parseDouble(myStringArray[4])),
        new DataPoint(5,
Double.parseDouble(myStringArray[5]))
    });
    graphView.setTitle("SENSOR 1");
    graphView.setTitleColor(R.color.purple_200);
    graphView.setTitleTextSize(32);
    graphView.addSeries(series);
    series.setDrawValuesOnTop(true);
    series.setValuesOnTopColor(Color.RED);
    series.setSpacing(20);

    //Graphic 2
    BarGraphSeries<DataPoint> series2 = new
BarGraphSeries<DataPoint>(new DataPoint[] {
        new DataPoint(1,
Double.parseDouble(myStringArray2[1])),
        new DataPoint(2,
Double.parseDouble(myStringArray2[2])),
        new DataPoint(3,
Double.parseDouble(myStringArray2[3])),
        new DataPoint(4,
Double.parseDouble(myStringArray2[4])),
        new DataPoint(5,
Double.parseDouble(myStringArray2[5]))
    });
    graphView2.setTitle("SENSOR 2");
    graphView2.setTitleColor(R.color.purple_200);

```



```

graphView2.setTitleTextSize(32);
graphView2.addSeries(series2);
series2.setDrawValuesOnTop(true);
series2.setValuesOnTopColor(Color.RED);
series2.setSpacing(20);

String latihan = mTvButtonExercise.getText().toString();
String tanggal = mTvTanggal.getText().toString();
String user = mTvUser.getText().toString();
String batas = mTvBatasTekanan.getText().toString();
mTvReport.append(latihan + "\n");
mTvReport.append(tanggal + "\n");
mTvReport.append("User = " + user + "\n" + "\n");
mTvReport.append("Limit = " + batas + "\n");
mTvReport.append("SENSOR 1" + "\n");

int[] jumlah = new int[myStringArray.length];
int batasTekanan =
Integer.parseInt(mTvBatasTekanan.getText().toString());
for (i = 1; i < myStringArray.length; i++){
    jumlah[i] = Integer.parseInt(myStringArray[i]);
    if (jumlah[i] > batasTekanan){
        x++;
    }
    if(myStringArray[i].length() > 2){
        mTvReport.append(myStringArray[i] + " gram" +
"\n");
    }
    else if (myStringArray[i].length() == 2){
        mTvReport.append("0" + myStringArray[i] + " gram"
+ "\n");
    }
    else{
        mTvReport.append(myStringArray[i] + "00 gram" +
"\n");
    }
}

int rata =
(jumlah[1]+jumlah[2]+jumlah[3]+jumlah[4]+jumlah[5])/5;
String sRata = String.valueOf(rata);
mTvReport.append("\n");
mTvReport.append("Average of Sensor 1 = " + sRata + "
gram" );

int sRataIntBatas =
Integer.parseInt(mTvBatasTekanan.getText().toString());
int sRataInt = Integer.parseInt(sRata);

//-----STATUS TEKANAN-----
//

if (mTvButtonExercise.getText().toString().equals("Tongue
Slide 1" ) ){
    if (sRataInt < sRataIntBatas){
        mTvReport.append("\n");
        mTvReport.append("The Pressure Hasn't Reached The
Limit");
        mTvDataStatus2.setText("The Pressure Hasn't

```

```

Reached The Limit");
    } else {
        mTvReport.append("\n");
        mTvReport.append("The Pressure Has Reached The
Limit");
        mTvDataStatus2.setText("The Pressure Has Reached
The Limit");
    }
}

if(x==0){
    mTvReport.append("\n");
    mTvReport.append("Success Rate 0%" + "\n");
    mTvReport.append("-----" +
"\n");
}
else if (x==1){
    mTvReport.append("\n");
    mTvReport.append("Success Rate 20%" + "\n");
    mTvReport.append("-----" +
"\n");
}
else if (x==2){
    mTvReport.append("\n");
    mTvReport.append("Success Rate 40%" + "\n");
    mTvReport.append("-----" +
"\n");
}
else if (x==3){
    mTvReport.append("\n");
    mTvReport.append("Success Rate 60%" + "\n");
    mTvReport.append("-----" +
"\n");
}
else if (x==4){
    mTvReport.append("\n");
    mTvReport.append("Success Rate 80%" + "\n");
    mTvReport.append("-----" +
"\n");
}
else {
    mTvReport.append("\n");
    mTvReport.append("Success Rate 100%" + "\n");
    mTvReport.append("-----" +
"\n");
}

mTvReport.append("\n");
mTvReport.append("SENSOR 2" + "\n");
int[] jumlah2 = new int[myStringArray.length];

for (i = 1; i < myStringArray2.length; i++){
    jumlah2[i] = Integer.parseInt(myStringArray2[i]);
    if (jumlah2[i] > batasTekanan){
        y++;
    }
    if(myStringArray2[i].length() > 2){
        mTvReport.append(myStringArray2[i] + " gram" +

```

```

"\n");
    }
    else if (myStringArray2[i].length() == 2){
        mTvReport.append("0" + myStringArray2[i] + " gram"
+ "\n");
    }
    else{
        mTvReport.append(myStringArray2[i] + "00 gram" +
"\n");
    }
}
int rata2 =
(jumlah2[1]+jumlah2[2]+jumlah2[3]+jumlah2[4]+jumlah2[5])/5;
String sRata2 = String.valueOf(rata2);
mTvReport.append("\n");
mTvReport.append("Average of Sensor 2 = " + sRata2 + "
gram");

int sRataInt2 = Integer.parseInt(sRata2);

//-----STATUS TEKANAN-----
//
if (mTvButtonExercise.getText().toString().equals("Tongue
Press")){
    if (sRataInt2 < sRataIntBatas){
        mTvReport.append("\n");
        mTvReport.append("The Pressure Hasn't Reached The
Limit");
        mTvDataStatus1.setText("The Pressure Hasn't
Reached The Limit");
    } else {
        mTvReport.append("\n");
        mTvReport.append("The Pressure Has Reached The
Limit");
        mTvDataStatus1.setText("The Pressure Has Reached
The Limit");
    }
}

else if
(mTvButtonExercise.getText().toString().equals("Tongue Slide
2")) ){
    if (sRataInt2 < sRataIntBatas){
        mTvReport.append("\n");
        mTvReport.append("The Pressure Hasn't Reached The
Limit");
        mTvDataStatus3.setText("The Pressure Hasn't
Reached The Limit");
    } else {
        mTvReport.append("\n");
        mTvReport.append("The Pressure Has Reached The
Limit");
        mTvDataStatus3.setText("The Pressure Has Reached
The Limit");
    }
}

if (y==0){
    mTvReport.append("\n");

```

```

        mTvReport.append("Success Rate 0%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
    else if (y==1){
        mTvReport.append("\n");
        mTvReport.append("Success Rate 20%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
    else if (y==2){
        mTvReport.append("\n");
        mTvReport.append("Success Rate 40%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
    else if (y==3){
        mTvReport.append("\n");
        mTvReport.append("Success Rate 60%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
    else if (y==4){
        mTvReport.append("\n");
        mTvReport.append("Success Rate 80%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
    else {
        mTvReport.append("\n");
        mTvReport.append("Success Rate 100%" + "\n");
        mTvReport.append("-----" +
"\n");
    }
}

//-----NEW THREAD-----
//

private class ReadInput implements Runnable {

    private boolean bStop = false;
    private Thread t;

    public ReadInput() {
        t = new Thread(this, "Input Thread");
        t.start();
    }

    public boolean isRunning() {
        return t.isAlive();
    }

    @Override
    public void run() {
        InputStream inputStream;

        try {
            inputStream = mBTSocket.getInputStream();

```

```

        while (!bStop) {
            byte[] buffer = new byte[256];
            if (inputStream.available() > 0) {
                inputStream.read(buffer);
                int i = 0;
                /*
                 * This is needed because new
String(buffer) is taking the entire buffer i.e. 256 chars on
Android 2.3.4 http://stackoverflow.com/a/8843462/1287554
                 */
                for (i = 0; i < buffer.length &&
buffer[i] != 0; i++) {
                }
                final String strInput = new String(buffer,
0, i);

                /*
                 * If checked then receive text, better
design would probably be to stop thread if unchecked and free
resources, but this is a quick fix
                 */

                mTxtReceive.post(new Runnable() {
                    @Override
                    public void run() {
                        mTxtReceive.setText(strInput);

                        int txtLength =
mTxtReceive.getEditableText().length();
                        if (txtLength > mMaxChars) {
mTxtReceive.getEditableText().delete(0, txtLength - mMaxChars);
                        }
                    }
                });

                Thread.sleep(500);
            }
        } catch (IOException e) {
// TODO Auto-generated catch block
            e.printStackTrace();
        } catch (InterruptedException e) {
// TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public void stop() {
        bStop = true;
    }
}

private class DisConnectBT extends AsyncTask<Void, Void, Void>

```

```

{

    @Override
    protected void onPreExecute() {
    }

    @Override
    protected Void doInBackground(Void... params) {

        if (mReadThread != null) {
            mReadThread.stop();
            while (mReadThread.isRunning())
                ; // Wait until it stops
            mReadThread = null;
        }

        try {
            mBTSocket.close();
        } catch (IOException e) {
// TODO Auto-generated catch block
            e.printStackTrace();
        }

        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        super.onPostExecute(result);
        mIsBluetoothConnected = false;
        if (mIsUserInitiatedDisconnect) {
            finish();
        }
    }

}

private void msg(String s) {
    Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_SHORT).show();
}

@Override
protected void onPause() {
    if (mBTSocket != null && mIsBluetoothConnected) {
        new DisconnectBT().execute();
    }
    Log.d(TAG, "Paused");
    super.onPause();
}

@Override
protected void onResume() {
    if (mBTSocket == null || !mIsBluetoothConnected) {
        new ConnectBT().execute();
    }
    Log.d(TAG, "Resumed");
    super.onResume();
}

```

```

    }

    @Override
    protected void onStop() {
        Log.d(TAG, "Stopped");
        super.onStop();
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
// TODO Auto-generated method stub
        super.onSaveInstanceState(outState);
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void> {
        private boolean mConnectSuccessful = true;

        @Override
        protected void onPreExecute() {
            progressDialog =
ProgressDialog.show(MonitoringScreen2.this, "Hold on",
"Connecting");// http://stackoverflow.com/a/11130220/1287554
        }

        @Override
        protected Void doInBackground(Void... devices) {

            try {
                if (mBTSocket == null || !mIsBluetoothConnected) {
                    mBTSocket =
mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);
//BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                    mBTSocket.connect();
                }
            } catch (IOException e) {
// Unable to connect to device
                e.printStackTrace();
                mConnectSuccessful = false;
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            super.onPostExecute(result);

            if (!mConnectSuccessful) {
                Toast.makeText(getApplicationContext(), "Could not
connect to device. Is it a Serial device? Also check if the UUID
is correct in the settings", Toast.LENGTH_LONG).show();
                finish();
            } else {
                msg("Connected to device");
                mIsBluetoothConnected = true;
                mReadThread = new ReadInput(); // Kick off input
reader
            }
        }
    }

```

```
        progressDialog.dismiss();  
    }
```


Program Aplikasi Web

a. Login

```
<html>
<head>
<title>Tongue Therapy</title>
<link rel="stylesheet" href="cssFile.css">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstra
p.min.css" integrity="sha384-
Vko08x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-
js/4.1.1/crypto-js.min.js"></script>
</head>

<body>

  <h1 style="text-align: center; padding-top: 20px;">Tongue Therapy
Website For Doctor</h1>

<div id="login-box" class="mainLog-div">
  <div class="left">
    <h1>Sign In</h1>

    <input type="text" id="email" name="email" placeholder="E-mail"
/>
    <input type="password" id="password" name="password"
placeholder="Password" />

    <input type="submit" id='signIn' name="login" value="Login"
/><br>
    <a href="register3.html" class="badge badge-secondary py-1 w-100
mt-3">Don't Have an Account?</a>
  </div>
</div>
</body>

<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-app.js";
  import { getDatabase, set, ref, update } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-database.js";
```

```

import { getAuth, createUserWithEmailAndPassword,
signInWithEmailAndPassword, onAuthStateChanged, signOut,
sendPasswordResetEmail } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-auth.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyA-uBrFAW00_oVi1sILfWl961-e4L_Ap9w",
  authDomain: "tongue-therapy.firebaseio.com",
  databaseURL: "https://tongue-therapy-default-
rtdb.firebaseio.com",
  projectId: "tongue-therapy",
  storageBucket: "tongue-therapy.appspot.com",
  messagingSenderId: "317678834724",
  appId: "1:317678834724:web:72ec39dce1fcca4115b4b1",
  measurementId: "G-XQB8G6XYEF"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const database = getDatabase(app);
const auth = getAuth();

//-----SIGN IN TO FIREBASE-----//

signIn.addEventListener('click', (e) => {

var email = document.getElementById('email').value;
var password = document.getElementById('password').value;
//var username = document.getElementById('username').value;

signInWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {

    // Signed in
    const user = userCredential.user;

    const dt = new Date();
    update(ref(database, 'doctor user/' + user.uid), {
      last_login: dt,
    })

    alert('User Logged In');
    window.location = "index.html"
    // ...

```

```

    })
    .catch((error) => {
      const errorCode = error.code;
      const errorMessage = error.message;

      alert(errorMessage);
      // ..
    });
  });
</script>
</html>

```

b. Register

```

<html>
<head>
<title>Tongue Therapy</title>
<link rel="stylesheet" href="cssFile.css">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
</head>

<body>

  <h1 style="text-align: center; padding-top: 20px;">Tongue Therapy Website For Doctor</h1>

  <div id="register-box">
    <div class="left">
      <h1>Sign Up</h1>
      <input type="text" id="username" name="username" placeholder="Username" />
      <input type="text" id="email" name="email" placeholder="E-mail" />
      <input type="password" id="password" name="password" placeholder="Password" />
      <input type="submit" id='signUp' name="signup_submit" value="Sign Up" /><br>

```

```

    <a href="login3.html" class="badge badge-secondary py-1 w-100
mt-3">Already Have An Account?</a>

  </div>

</div>
</body>

<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-app.js";
  import { getDatabase, set, ref, update } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-database.js";
  import { getAuth, createUserWithEmailAndPassword,
signInWithEmailAndPassword, onAuthStateChanged, signOut } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-auth.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyA-uBrFAW00_oVi1sILfWl961-e4L_Ap9w",
    authDomain: "tongue-therapy.firebaseio.com",
    databaseURL: "https://tongue-therapy-default-
rtdb.firebaseio.com",
    projectId: "tongue-therapy",
    storageBucket: "tongue-therapy.appspot.com",
    messagingSenderId: "317678834724",
    appId: "1:317678834724:web:72ec39dce1fcca4115b4b1",
    measurementId: "G-XQB8G6XYEF"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const database = getDatabase(app);
  const auth = getAuth();

  //-----SIGN UP TO FIREBASE-----//

  signUp.addEventListener('click', (e) => {

    var email = document.getElementById('email').value;
    var password = document.getElementById('password').value;
    var username = document.getElementById('username').value;

    createUserWithEmailAndPassword(auth, email, password)

```

```

.then((userCredential) => {
  // Signed in
  const user = userCredential.user;

  set(ref(database, 'doctor user/' + user.uid),{
    username: username,
    email: email
  })

  alert('user created!');
  // ...
})
.catch((error) => {
  const errorCode = error.code;
  const errorMessage = error.message;

  alert(errorMessage);
  // ..
});
});
</script>
</html>

```

c. Index

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Home</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstra
p.min.css" integrity="sha384-
Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
  <script src="https://www.gstatic.com/firebasejs/8.1.1/firebase-
app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.1.1/firebase-
storage.js"></script>

```

```

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.
js"></script>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstra
p.min.css" integrity="sha384-
Vtkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">

</head>
<style>
.tab1 {
    display: inline-block;
    margin-left: 20px;
}

.tab2 {
    display: inline-block;
    margin-left: 40px;
}

.tab4 {
    display: inline-block;
    margin-left: 85px;
}

.center {
    display: block;
    margin-left: auto;
    margin-right: auto;
}
</style>
<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="#">  Tongue Therapy Data Collector Web </a>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <ul class="navbar-nav mr-auto">

                </ul>
                <form class="form-inline my-2 my-lg-0">

                    <ul class="navbar-nav mr-auto">
                        <li class="nav-item">

```

```

        <a id = "userlink" class="nav-link m-2"
href="">Username</a>
    </li>

    <li class="nav-item">
        <a id = "signoutlink" class="nav-link m-2" href="">Log
In</a>
    </li>

    <li class="nav-item m-3">
        <input type="submit" id='logout' name="signup_submit"
value="Sign Out" />
    </li>

</ul>
</form>
</div>
</nav>

<p style="font-size: large; margin: 15px;"> <span id='date-
time'></span>.</p>

<label for="fileNameInput" style="margin: 15px; width: fit-
content;">Date <span class="tab4"></span> :
    <input type="text" id='fileNameInput' style="width: 300px;">
    <p style="color: red;"> example of writing date : Wednesday, May
18, 2022/</p>
</label><br>

<label for="exerciseName" style="margin: 15px; width: fit-
content;">Exercise Type <span class="tab1"></span> :
    <input type="text" id='exerciseName' style="width: 300px;">
    <p style="color: red;"> example of writing exercise : Tongue
Press/ - Tongue Slide 1/ - Tongue Slide 2/</p>
</label><br>

<label for="userNameInput" style="margin: 15px; width: fit-
content;">User Email <span class="tab2"></span> :
    <input type="text" id='userNameInput' style="width: 300px;">
    <p style="color: red;"> example of writing email :
Example@gmail.com.txt</p>
</label><br>

```

```

<button id='btnFile' style="margin-left: 15px;">Show
File</button><br><br>
<a id="linkFile" href="#" target="_blank" style="margin:
15px;">Download Link</a><br><br>

<button type="submit" id="getData" class="btn btn-primary"
style="margin-left: 15px;">Search Data</button><br><br>
<div class="container mt-3">
  <table class="table table-dark" id='dataTbl'>
    <thead>
      <tr>
        <th scope="col">No</th>
        <th scope="col">Age</th>
        <th scope="col">Birth</th>
        <th scope="col">Full Name</th>
        <th scope="col">Email</th>

      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </tbody>
  </table>

  <table class="table table-dark" id='dataTb2'>
    <thead>
      <tr>
        <th scope="col">No</th>
        <th scope="col">Email</th>
        <th scope="col">Date</th>
        <th scope="col">Limit</th>
        <th scope="col">Tongue Press</th>
        <th scope="col">Tongue Slide1</th>
        <th scope="col">Tongue Slide2</th>

      </tr>
    </thead>
    <tbody>
      <tr>
        <td></td>

```



```

        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
</div>

<script type="module">
    // Import the functions you need from the SDKs you need
    import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-app.js";
    import { getDatabase, set, ref, update, push, child,
onValue} from "https://www.gstatic.com/firebasejs/9.6.10/firebase-
database.js";
    import { getAuth, createUserWithEmailAndPassword,
signInWithEmailAndPassword, onAuthStateChanged, signOut } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-auth.js";
    import { getStorage, ref as sRef, listAll } from
"https://www.gstatic.com/firebasejs/9.6.10/firebase-storage.js";

    // TODO: Add SDKs for Firebase products that you want to use
    // https://firebase.google.com/docs/web/setup#available-
libraries

    // Your web app's Firebase configuration
    const firebaseConfig = {
        apiKey: "AIzaSyA-uBrFAW00_oVi1sILfWl961-e4L_Ap9w",
        authDomain: "tongue-therapy.firebaseio.com",
        databaseURL: "https://tongue-therapy-default-
rtdb.firebaseio.com",
        projectId: "tongue-therapy",
        storageBucket: "tongue-therapy.appspot.com",
        messagingSenderId: "317678834724",
        appId: "1:317678834724:web:72ec39dce1fcca4115b4b1",
        measurementId: "G-XQB8G6XYEF"
    };

    // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const database = getDatabase(app);
    const auth = getAuth();
    const storage = getStorage();
    //-----THE REFERENCES-----//

```

```

let userlink = document.getElementById('userlink');
let signoutlink = document.getElementById('signoutlink');
let header = document.getElementById('hh')
//var currentUser = null;

//-----FUNCTION-----//

logout.addEventListener('click', (e) => {

  signOut(auth).then(() => {
    // Sign-out successful.
    alert('user loged out');
  }).catch((error) => {
    // An error happened.
    const errorCode = error.code;
    const errorMessage = error.message;

    alert(errorMessage);
  });

});

//-----TIME-----//
var dt = new Date();
document.getElementById('date-time').innerHTML=dt;

//-----FUNCTION KEEP LOGIN-----//
//

const user = auth.currentUser;
onAuthStateChanged(auth, (user) => {

  if (user) {
    const uid = user.uid;

    document.getElementById("userlink").style.display =
"none";
    document.getElementById("signoutlink").style.display =
"none";

  } else {

    document.getElementById("logout").style.display =
"none";

    userlink.innerText="Create New Account";
    userlink.classList.replace("nav-link", "btn");
  }
});

```

```

        userlink.classList.add("btn-primary");
        userlink.href = "register3.html";

        signoutlink.innerText = "Login";
        signoutlink.classList.replace("nav-link", "btn");
        signoutlink.classList.add("btn-success");
        signoutlink.href = "login3.html";

    }

});

//-----BUTTON GET DATA-----//
firebase.initializeApp(firebaseConfig);

$('#btnFile').on('click', function() {
    const storage = firebase.storage();

    storage.ref('Data Tekanan/' + $('#exerciseName').val() +
$('#fileNameInput').val() +
$('#userNameInput').val() ).getDownloadURL()
        .then((url) => {
            $("#linkFile").attr("href" , url);
            $('#linkFile').text(url);
            console.log(url);
        })
    });

//-----TABLE-----//
// read data
getData.addEventListener('click', (e) => {

    $('#dataTb1 td').remove();
    $('#dataTb2 td').remove();
    var rowNum = 0;
    var rowNum2 = 0;
    const dbRef = ref(database, 'Data User/', 'DataLatihan/');
    const dbRef2 = ref(database, 'Data Latihan/');

    onValue(dbRef, (snapshot) => {
        snapshot.forEach((childSnapshot) => {
            const childKey = childSnapshot.key;
            const childData = childSnapshot.val();
            // ...
            rowNum += 1;
            var row = "<tr><td>" + rowNum + "</td><td>" +
childData.age + "</td><td>" + childData.dateBirth + "</td><td>" +

```

```

childData.fullName + "</td><td>" + childData.emailUser +
"</td></tr>"

    $(row).appendTo('#dataTb1');

    });
}, {
    onlyOnce: true
});

onValue(dbRef2, (snapshot) => {
    snapshot.forEach((childSnapshot) => {
        const childKey = childSnapshot.key;
        const childData = childSnapshot.val();
        // ...
        rowNum2 += 1;
        var row2 = "<tr><td>" + rowNum2 + "</td><td>" +
childData.dataEmail + "</td><td>" + childData.dataTanggal +
"</td><td>" + childData.dataLimit + "</td><td>" +
childData.dataStatus1 + "</td><td>" + childData.dataStatus2 +
"</td><td>" + childData.dataStatus3 + "</td><tr>"
        $(row2).appendTo('#dataTb2');

    });
}, {
    onlyOnce: true
});
</script>
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-
J6qa4849b1E2+poT4WhyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.m
in.js" integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap
.min.js" integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6"
crossorigin="anonymous"></script>
</body>
</html>

```