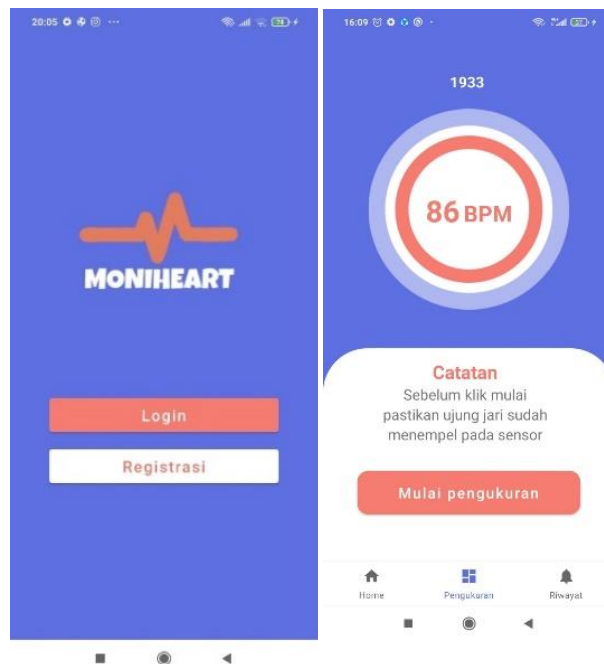


MANUAL SISTEM

IMPLEMENTASI IOT PADA APLIKASI MONITORING KONDISI KESEHATAN DENYUT JANTUNG MENGGUNAKAN SENSOR MAX30100 BERBASIS ANDROID



OLEH

1. Gugum Saefulah Zidni
2. Ikrimach S.Kom., M.Cs.

(NIDN 0506128401)

UNIVERSITAS TEKNOLOGI YOGYAKARTA
TAHUN 2023

DAFTAR ISI

JUDUL.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	ii
1. TAMPILAN DAN KODE PROGRAM.....	3
1.1 Halaman Login dan registrasi	3
1.1.1. Tampilan halaman login	3
1.1.2. Potongan Source Code Login	3
1.1.3. Tampilan Halaman Registrasi	5
1.1.4. Potongan Source Code Registrasi.....	5
1.2 Halaman Home	8
1.2.1. Tampilan Home	8
1.2.2. Source Code Home	8
1.3 Halaman Pengukuran dan Hasil Pengukuran.....	10
1.3.1. Tampilan Halaman Pengukuran	10
1.3.2. Source Code Pengukuran.....	10
1.3.3. Tampilan Halaman Hasil Pengukuran	12
1.3.4. Source Code Hasil Pengukuran	12
1.4 Halaman Riwayat.....	15
1.4.1 Tampilan Halaman Riwayat	15
1.4.2 Source Code Halaman Riwayat	15
1.5 Halaman Profile	17
1.5.1 Tampilan Halaman Profile.....	17
1.5.2 Source Code Profile.....	17
1.6 Racangan sensor dab Arduino IDE.....	18
2. PENGGUNAAN PROGRAM.....	21
2.1 Langkah Melakukan Pengukuran Denyut Jantung	21
2.2 Hasil Pengukuran	22
2.3 Riwayat	22
2.4 Halaman Home	23

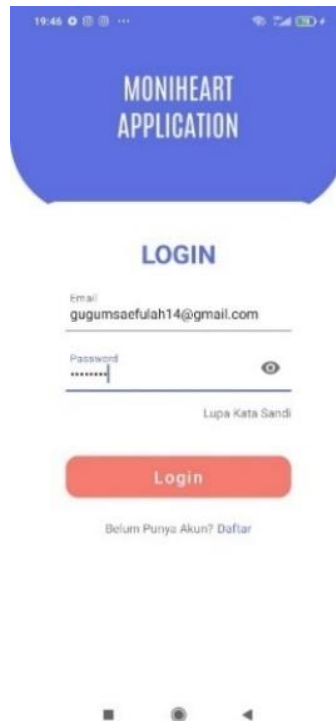
DAFTAR GAMBAR

Gambar 1 - Halaman login	3
Gambar 2 - Halaman Registrasi.....	5
Gambar 3 - Halaman Home.....	8
Gambar 4 - Halaman Pengukuran	10
Gambar 5 - Halaman Hasil Pengukuran	12
Gambar 6 - Halaman Riwayat	15
Gambar 7 – Halaman <i>Profile</i>	17
Gambar 8 - Rancangan sensor	18
Gambar 9 – Melakukan Pengukuran	21
Gambar 10 – Hasil Pengukuran.....	22
Gambar 11 - Riwayat.....	23
Gambar 12 - Home	23

1. TAMPILAN DAN KODE PROGRAM

1.1 Halaman Login dan registrasi

1.1.1. Tampilan halaman login



Gambar 1 - Halaman login

1.1.2. Potongan Source Code Login

```
class RegisterActivity : AppCompatActivity() {
    private var _binding: ActivityRegisterBinding? = null
    private val binding get() = _binding!!

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        _binding = ActivityRegisterBinding.inflate(layoutInflater)
        setContentView(binding.root)
        var editnama = binding.editNama
        var editemail = binding.editEmail
        var editjk = binding.editJk
        var editumur = binding.editEditUmur
        var editalamat = binding.editAlamat
        var editpass = binding.editPassword

        fun register() {
            if(editnama.text!!.isEmpty()) {
                editnama.error = "Kolom tidak boleh kosong"
                editnama.requestFocus()
                return
            }

            if(editemail.text!!.isEmpty()) {
                editemail.error = "Kolom tidak boleh kosong"
                editemail.requestFocus()
                return
            }
        }
    }
}
```

```

    }

    if(editjk.text!!.isEmpty()){
        editjk.error = "Kolom tidak boleh kosong"
        editjk.requestFocus()
        return
    }

    if(editumur.text!!.isEmpty()){
        editumur.error = "Kolom tidak boleh kosong"
        editumur.requestFocus()
        return
    }

    if(editalamat.text!!.isEmpty()){
        editalamat.error = "Kolom tidak boleh kosong"
        editalamat.requestFocus()
        return
    }

    if(editpass.text!!.isEmpty()){
        editpass.error = "Kolom tidak boleh kosong"
        editpass.requestFocus()
        return
    }
}

```

```

ApiConfig.ApiConfig.instanceRetrofit.register(editnama.text.toString(),ed
itemail.text.toString(), editjk.text.toString(),
editumur.text.toString(), editalamat.text.toString(),
editpass.text.toString()).enqueue( object : Callback<ResponModel>{
    override fun onResponse(call: Call<ResponModel>,
response: Response<ResponModel>) {

```

```

        if (response.isSuccessful ) {
            var respon = response.body()!!
            if (respon.code == 200) {
                // berhasil
                Toast.makeText(
                    this@RegisterActivity,
                    "Success:" + respon.success+ " " +
respon.message,
                    Toast.LENGTH_SHORT
                ).show()
            }else if(respon.code == 400){
                Toast.makeText(
                    this@RegisterActivity,
                    "Error:" + respon.message,
                    Toast.LENGTH_SHORT
                ).show()
            }
        } else if (response.body() == null){
            var respon = response.body()
            Toast.makeText(
                this@RegisterActivity,
                "Gagal : " + "Email sudah ada",
                Toast.LENGTH_SHORT
            ).show()
        }
    }
}

```



```

var editalamat = binding.editAlamat
var editpass = binding.editPassword

fun register(){
    if(editnama.text!!.isEmpty()){
        editnama.error = "Kolom tidak boleh kosong"
        editnama.requestFocus()
        return
    }

    if(editemail.text!!.isEmpty()){
        editemail.error = "Kolom tidak boleh kosong"
        editemail.requestFocus()
        return
    }

    if(editjk.text!!.isEmpty()){
        editjk.error = "Kolom tidak boleh kosong"
        editjk.requestFocus()
        return
    }

    if(editumur.text!!.isEmpty()){
        editumur.error = "Kolom tidak boleh kosong"
        editumur.requestFocus()
        return
    }

    if(editalamat.text!!.isEmpty()){
        editalamat.error = "Kolom tidak boleh kosong"
        editalamat.requestFocus()
        return
    }

    if(editpass.text!!.isEmpty()){
        editpass.error = "Kolom tidak boleh kosong"
        editpass.requestFocus()
        return
    }
}

```

```

ApiConfig.ApiConfig.instanceRetrofit.register(editnama.text.toString(),ed
itemail.text.toString(), editjk.text.toString(),
editumur.text.toString(), editalamat.text.toString(),
editpass.text.toString()).enqueue( object : Callback<ResponModel>{
    override fun onResponse(call: Call<ResponModel>,
response: Response<ResponModel>) {

        if (response.isSuccessful ) {
            var respon = response.body()!!
            if (respon.code == 200) {
                // berhasil
                Toast.makeText(
                    this@RegisterActivity,
                    "Success:" + respon.success+ " " +
respon.message,
                    Toast.LENGTH_SHORT
                ).show()
            }else if(respon.code == 400){
                Toast.makeText(

```



```

        this@RegisterActivity,
        "Error:" + respon.message,
        Toast.LENGTH_SHORT
    ).show()
    }
} else if (response.body() == null){
    var respon = response.body()
    Toast.makeText(
        this@RegisterActivity,
        "Gagal : " + "Email sudah ada",
        Toast.LENGTH_SHORT
    ).show()
}
}

override fun onFailure(call: Call<ResponModel>, t:
Throwable) {
    Toast.makeText(this@RegisterActivity, "Error:" +
t.message, Toast.LENGTH_SHORT).show()
}

})

}

binding.btnRegister.setOnClickListener {
    register()
}

}

}

```

1.2 Halaman Home

1.2.1. Tampilan Home



Gambar 3 - Halaman Home

1.2.2. Source Code Home

```
class HomeFragment : Fragment() {  
  
    private var _binding: FragmentHomeBinding? = null  
    private val binding get() = _binding!!  
    lateinit var s: prefs  
  
    lateinit var rvArtikel: RecyclerView  
  
    override fun onCreateView(  
        inflater: LayoutInflater,  
        container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        val homeViewModel =  
            ViewModelProvider(this).get(HomeViewModel::class.java)  
  
        _binding = FragmentHomeBinding.inflate(inflater, container, false)  
        val root: View = binding.root  
  
        val textView: TextView = binding.textHome  
        homeViewModel.text.observe(viewLifecycleOwner) {  
            textView.text = it  
        }  
        s = prefs(requireActivity())  
  
        val tvnama: TextView = binding.tvNama  
        val ivprofil: ImageView = binding.ivProfil  
        val btndetail: Button = binding.btnDetail
```

```

val tvbpmakhir: TextView = binding.tvBpmakhir
tvnama.text = s.getString(s.nama)
rvArtikel = binding.rvartikel

ivprofil.setOnClickListener{
    val intent = Intent(activity, ProfileActivity::class.java)
    requireActivity().startActivity(intent)
}

val layoutManager = LinearLayoutManager(activity)
layoutManager.orientation = LinearLayoutManager.HORIZONTAL

rvArtikel.adapter = AdapterArtikel(arrArtikel)
rvArtikel.layoutManager = layoutManager

val prefs = prefs(requireActivity())
val id = prefs.getInt2("user_id")

ApiConfig.ApiConfig.instanceRetrofit.getriwayat(id).enqueue(
object :
    Callback<ResponModelRiwayat> {

        override fun onResponse(
            call: Call<ResponModelRiwayat>,
            response: Response<ResponModelRiwayat>
        ) {
            val res = response.body()
            if (res != null ) {

                if(res.riwayats.size != 0){
                    val bpmakhir = res.riwayats[res.riwayats.size -
1].bpm

                    tvbpmakhir.text = bpmakhir.toString()
                } else{
                    tvbpmakhir.text = "0"
                }

            }else{
                Log.d("coba", "onResponse: gagal")
            }
        }

        override fun onFailure(call: Call<ResponModelRiwayat>, t:
Throwable) {

            Log.d("coba", "onResponse: gagal")

        }
    })

return root
}

// recycle view artikel
val arrArtikel: ArrayList<Artikel>get(){
    var arr = ArrayList<Artikel>()
    val a1 = Artikel()
    a1.judul = "Cara Menjaga Kesehatan Jantung"
    a1.body = "... "
    a1.gambar = R.drawable.gambar_jantung

```

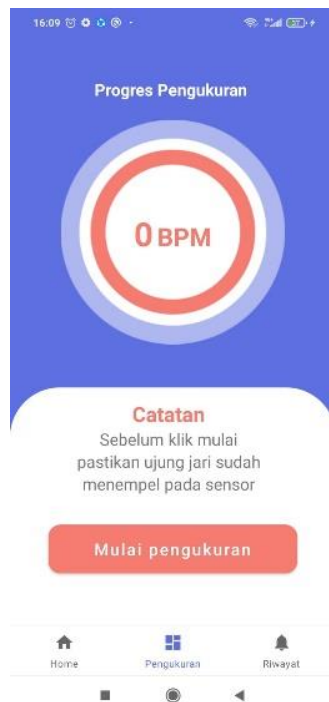
```

    val a2 = Artikel()
    a2.judul = "Jantung harus di monitoring"
    a2.body = "..."/>

```

1.3 Halaman Pengukuran dan Hasil Pengukuran

1.3.1. Tampilan Halaman Pengukuran



Gambar 4 - Halaman Pengukuran

1.3.2. Source Code Pengukuran

```

class PengukuranFragment : Fragment() {

    private var _binding: FragmentPengukuranBinding? = null
    private lateinit var timer: CountdownTimer
}

```

```

lateinit var ref:DatabaseReference
lateinit var btnPengukuran: Button
lateinit var tvPengukuran: TextView
lateinit var calendar: Calendar
lateinit var simpleDateFormat: SimpleDateFormat
lateinit var date: String
lateinit var tv_timer: TextView
var hasilbpm: Int = 0

private val binding get() = _binding!!

override fun onCreateView(
    inflater: LayoutInflater,
    container: ViewGroup?,
    savedInstanceState: Bundle?
): View {
    val dashboardViewModel =
ViewModelProvider(this).get(PengukuranViewModel::class.java)
    _binding = FragmentPengukuranBinding.inflate(inflater, container,
false)
    val root: View = binding.root
    val textView: TextView = binding.textDashboard
    dashboardViewModel.text.observe(viewLifecycleOwner) {
        textView.text = it
    }

    ref = FirebaseDatabase.getInstance().getReference("bpm")
    btnPengukuran = binding.btnPengukuran
    tvPengukuran = binding.tvPengukuran
    tv_timer = binding.tvTimer

    calendar = Calendar.getInstance()
    simpleDateFormat = SimpleDateFormat("dd-MM-yyyy")
    date = simpleDateFormat.format(calendar.time)

    btnPengukuran.setOnClickListener {
        ref.addValueEventListener(object: ValueEventListener{
            override fun onDataChange(snapshot: DataSnapshot) {
                if (snapshot.exists()){
                    var bpm = snapshot.getValue().toString()
                    if (bpm != null){
                        tvPengukuran.text = bpm
                        var kondisi = "Lemah"
                        hasilbpm = bpm.toInt()
                    }
                }
            }
            override fun onCancelled(error: DatabaseError) {
                TODO("Not yet implemented")
            }
        })

        timer = object: CountdownTimer(30000, 1){
            override fun onTick(remaining: Long) {
                tv_timer.text = remaining.toString()
            }

            override fun onFinish() {
                tv_timer.text = "Selesai"
            }
        }
    }
}

```

```

    }
}

timer.start()

Handler(Looper.getMainLooper()).postDelayed(object : Runnable
{
    override fun run() {
        val intent = Intent(activity,
HasilActivity::class.java)
        intent.putExtra("bpm", hasilbpm)
        //intent.putExtra("kondisi", kondisi)
        intent.putExtra("waktu", date)
        activity!!.startActivity(intent)
    }
}, 30000)
}

return root
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}
}
}

```

1.3.3. Tampilan Halaman Hasil Pengukuran



Gambar 5 - Halaman Hasil Pengukuran

1.3.4. Source Code Hasil Pengukuran

```

class HasilActivity : AppCompatActivity() {
    private var _binding: ActivityHasilBinding? = null
    private val binding get() = _binding!!
    lateinit var tvbpm: TextView

```

```

lateinit var tvkondisi: TextView
lateinit var tvwaktu: TextView
lateinit var btnhapus: Button
lateinit var btnsimpan: Button

lateinit var getkondisi: String

lateinit var s: prefs

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

    s = prefs(this)

    tvbpm = binding.tvBpm
    tvkondisi = binding.tvKondisi
    tvwaktu = binding.tvWaktu
    btnhapus = binding.btnHapus
    btnsimpan = binding.btnSimpan

    val items = listOf("istirahat", "olahraga", "duduk", "bekerja")
    val adapter = ArrayAdapter(this@HasilActivity,
R.layout.dropdown_item, items)
    binding.materialSpinner.setAdapter(adapter)

    val getuser_id = s.getInt2("user_id")
    val getbpm = intent.getIntExtra("bpm", 0)
    if (getbpm <60){
        getkondisi = "Rendah"
    }else if (getbpm > 60 && getbpm < 100){
        getkondisi = "Normal"
    }else{
        getkondisi = "Tinggi"
    }
    val getwaktu = intent.getStringExtra("waktu")

    tvbpm.setText(getbpm.toString())
    tvkondisi.setText(getkondisi)
    tvwaktu.setText(getwaktu)

    btnhapus.setOnClickListener{
        val intent = Intent(this@HasilActivity,
NavigationActivity::class.java)
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
        startActivity(intent)
        finish()
    }

    fun postriwayat(){

        val mySpinner = binding.materialSpinner as Spinner
        val catatan = mySpinner.selectedItem.toString()

ApiConfig.ApiConfig.instanceRetrofit.riwayat(getuser_id.toString(),
getbpm.toString(), getkondisi.toString(), getwaktu.toString(),
catatan).enqueue( object :
        Callback<ResponModelRiwayat> {

```

```

        override fun onResponse(call: Call<ResponModelRiwayat>,
response: Response<ResponModelRiwayat>) {

            if (response.isSuccessful ) {
                var respon = response.body()!!
                if (respon.code == 200) {
                    // berhasil
                    val intent = Intent(this@HasilActivity,
RiwayatFragment::class.java)

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)

                    Toast.makeText(
                        this@HasilActivity,
                        "Success:" + respon.success+ " " +
respon.message,
                        Toast.LENGTH_SHORT
                    ).show()
                } else if(respon.code == 400){
                    Toast.makeText(
                        this@HasilActivity,
                        "Error:" + respon.message,
                        Toast.LENGTH_SHORT
                    ).show()
                }
            } else if (response.body() == null){
                var respon = response.body()
                Toast.makeText(
                    this@HasilActivity,
                    "Gagal : " + "Email sudah ada",
                    Toast.LENGTH_SHORT

                ).show()
            }
        }

        override fun onFailure(call: Call<ResponModelRiwayat>, t:
Throwable) {
            Toast.makeText(this@HasilActivity, "Error:" +
t.message, Toast.LENGTH_SHORT).show()
        }
    })
}
btnsimpan.setOnClickListener{
    postr riwayat()
}
}
}
}

```


1.4 Halaman Riwayat

1.4.1 Tampilan Halaman Riwayat



Gambar 6 - Halaman Riwayat

1.4.2 Source Code Halaman Riwayat

```
class RiwayatFragment : Fragment() {
    private var _binding: FragmentRiwayatBinding? = null

    lateinit var rvriwayat: RecyclerView
    lateinit var tvmaxbpm: TextView
    lateinit var tvminbpm: TextView
    lateinit var tvratabpm: TextView
    var lisbpm = arrayListOf<Int>()

    // This property is only valid between onCreateView and
    // onDestroyView.
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        val notificationsViewModel =
            ViewModelProvider(this).get(RiwayatViewModel::class.java)

        _binding = FragmentRiwayatBinding.inflate(inflater, container,
false)
        val root: View = binding.root

        val textView: TextView = binding.textNotifications
        notificationsViewModel.text.observe(viewLifecycleOwner) {
            textView.text = it
        }

        rvriwayat = binding.rvriwayat
    }
}
```

```

    tvmaxbpm = binding.tvMaxbpm
    tvminbpm = binding.tvMinbpm
    tvratabpm = binding.tvRatabpm

    getriwayat()

    Log.d("array bpm = ", "onCreateView: "+lisbpm)
    return root
}

fun getriwayat(){

    val prefs = prefs(requireActivity())
    val id = prefs.getInt2("user_id")

    ApiConfig.ApiConfig.instanceRetrofit.getriwayat(id).enqueue(
object :
    Callback<ResponModelRiwayat> {

        override fun onResponse(
            call: Call<ResponModelRiwayat>,
            response: Response<ResponModelRiwayat>
        ) {
            val res = response.body()
            if (res != null ) {
                displayRiwayat(res.riwayats)

                for (i in res.riwayats){

                    lisbpm.add(i.bpm)

                }

                if (res.riwayats.size != 0){
                    tvmaxbpm.setText(lisbpm.max().toString())
                    tvminbpm.setText(lisbpm.min().toString())
                    tvratabpm.setText(lisbpm.average().toInt().toString())
                }else{
                    tvmaxbpm.setText("0")
                    tvminbpm.setText("0")
                    tvratabpm.setText("0")
                }

                }else{
                    Log.d("coba", "onResponse: gagal")
                }
            }

            override fun onFailure(call: Call<ResponModelRiwayat>, t:
Throwable) {

                Log.d("coba", "onResponse: gagal")

            }
        })
    }
}

```

```

fun displayRiwayat(riwayats: ArrayList<Riwayat>){
    val layoutManager = LinearLayoutManager(activity)
    layoutManager.orientation = LinearLayoutManager.VERTICAL

    rvriwayat.adapter = AdapterRiwayat(riwayats)
    rvriwayat.layoutManager = layoutManager
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}
}

```

1.5 Halaman Profile

1.5.1 Tampilan Halaman Profile



Gambar 7 – Halaman *Profile*

1.5.2 Source Code Profile

```

class ProfileActivity : AppCompatActivity() {
    private var _binding: ActivityProfileBinding? = null
    private val binding get() = _binding!!

    lateinit var s: prefs

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

```

```

s = prefs(this)
val tvnama: TextView = binding.tvNama
val tvemail: TextView = binding.tvEmail
val tvjk: TextView = binding.tvJk
val tvumur: TextView = binding.tvUmur
val tvalamat: TextView = binding.tvAlamat
val btnlogout: Button = binding.btnLogout

Log.d("nama", "onCreate: "+s.getString(s.email))

tvnama.text = s.getString(s.nama)
tvemail.text = s.getString(s.email)
tvjk.text = s.getString(s.jk)
tvumur.text = s.getString(s.umur)
tvalamat.text = s.getString(s.alamat)

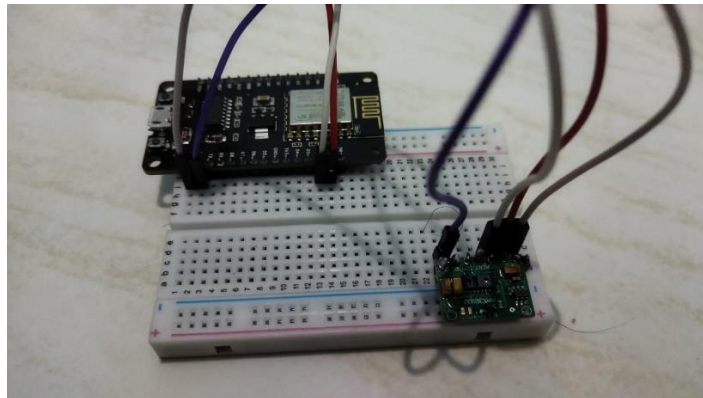
btnlogout.setOnClickListener() {
    s.setIsLogin(false)
    val intent = Intent(this@ProfileActivity,
AwalActivity::class.java)
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
    startActivity(intent)
    finish()
}

}
}

```

1.6 Rancangan sensor dab Arduino IDE

1.6.1 Rancangan sensor



Gambar 8 - Rancangan sensor

1.6.2 Source Code Arduino IDE

```

#include <dummy.h>
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include "Timer.h"

#if defined(ESP32)
#include <WiFi.h>

```

```

#include <FirebaseESP32.h>
#ifdef ESP8266
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#endif

// Provide the token generation process info.
#include <addons/TokenHelper.h>
// Provide the RTDB payload printing info and other helper functions.
#include <addons/RTDBHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "Xiaomi 12 Lite"
#define WIFI_PASSWORD "123456788"
/* 2. Define the API Key */
#define API_KEY "AIzaSyBolnpHdmhYtCl7C7I8YtxfKLLDrhOIWj8"
/* 3. Define the RTDB URL */
#define DATABASE_URL "moniheart-faf5f-default-rtdb.firebaseio.com"
/* 4. Define the user Email and password that already registered or added
in your project */
#define USER_EMAIL "Gugum57@gmail.com"
#define USER_PASSWORD "Jammengnong57"
// Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
MAX30105 particleSensor;
const byte RATE_SIZE = 12; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
Timer t;

void setup() {
  Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.println("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  config.database_url = DATABASE_URL;
  config.token_status_callback = tokenStatusCallback;

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
  Firebase.setDoubleDigits(5);
  Serial.println("Initializing...");
  // Initialize sensor

```

```

    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port,
    400kHz speed
    {
        Serial.println("MAX30105 was not found. Please check wiring/power. ");
        while (1);
    }
    Serial.println("Place your index finger on the sensor with steady
    pressure.");
    t.every(10000, kirimData);
    particleSensor.setup(); //Configure sensor with
    default settings
    particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to
    indicate sensor is running
    particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}

void kirimData() {
    Serial.print("ok");
    if (Firebase.ready()) {
        Serial.println();
        Serial.printf("Set string... %s\n", Firebase.setString(fbdo, F("bpm"),
        beatAvg) ? "upload success" : fbdo.errorReason().c_str());
        Serial.println();
    }
}

void loop() {
    t.update();
    long irValue = particleSensor.getIR();
    if (checkForBeat(irValue) == true) {
        //We sensed a beat!
        long delta = millis() - lastBeat;
        lastBeat = millis();
        beatsPerMinute = 60 / (delta / 1000.0);
        if (beatsPerMinute < 255 && beatsPerMinute > 20) {
            rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in
            the array
            rateSpot %= RATE_SIZE; //Wrap variable
            //Take average of readings
            beatAvg = 0;
            for (byte x = 0; x < RATE_SIZE; x++)
                beatAvg += rates[x];
            beatAvg /= RATE_SIZE;
        }
    }
    Serial.print("IR=");
    Serial.print(irValue);
    Serial.print(", BPM=");
    Serial.print(beatsPerMinute);
    Serial.print(", Avg BPM=");
    Serial.print(beatAvg);
    if (irValue < 50000){
        Serial.print(" No finger?");
        beatAvg = 0;
    }

    Serial.println();
}

```

2. PENGGUNAAN PROGRAM

Sistem Monitoring denyut jantung atau Moniheart merupakan system yang digunakan untuk melakukan monitoring kondisi Kesehatan denyut jantung secara realtime dan pengguna dapat menyimpan hasil pengukuran sebagai catatan pemeriksaan denyut jantung secara berkala. System ini terintegrasi dengan Internet of Things menggunakan sensor MAX30100 sebagai sensor denyut jantung.

2.1 Langkah Melakukan Pengukuran Denyut Jantung



Gambar 9 – Melakukan Pengukuran

Halaman Pengukuran merupakan halaman yang digunakan Ketika pengguna akan melakukan pengukuran denyut jantung mereka. Sebelum melakukan pengukuran, pengguna harus menyalakan alat terlebih dahulu kemudian menempelkan ujung jari pada sensor MAX30100. Jika sudah, maka pengguna bisa melakukan pengukuran dengan menekan tombol mulai pengukuran. Pengukuran akan dilakukan selama 1 menit dan hasilnya akan langsung ditampilkan setelah 1 menit pengukuran. Tampilan proses pengukuran dapat dilihat pada gambar 9.

2.2 Hasil Pengukuran



Gambar 10 – Hasil Pengukuran

Halaman Hasil pengukuran merupakan halaman yang berisi informasi hasil pengukuran denyut jantung yang telah dilakukan oleh pengguna. Pada halaman hasil pengukuran, informasi yang ditampilkan adalah BPM (beats per minute), Kondisi (Normal, Lemah, Tinggi), Waktu Pengukuran dan Catatan kegiatan saat pengukuran dilakukan. Jika pengguna ingin menyimpan hasil pengukuran denyut jantung, maka pengguna dapat menekan tombol simpan. Namun jika ingin menghapus hasil pengukuran, maka pengguna dapat menekan tombol hapus. Tampilan Hasil pengukuran dapat dilihat pada gambar 10.

2.3 Riwayat

Halaman riwayat merupakan halaman yang memuat informasi riwayat pengukuran yang disimpan oleh pengguna. Pada halaman riwayat, pengguna dapat menghapus riwayat yang ingin mereka hapus dengan menekan riwayat yang ingin dihapus kemudian klik hapus. Pada halaman riwayat, hasil pengukuran denyut jantung telah melalui proses Fuzzy yaitu dengan melakukan klasifikasi kondisi Kesehatan (Lemah, Normal, Tinggi) berdasarkan tiga parameter (Crips Input) yaitu Umur, Aktivitas dan BPM (Beats per minute). Tampilan halaman riwayat dapat dilihat pada gambar 11 berikut.



Gambar 11 - Riwayat

2.4 Halaman Home

Halaman Home merupakan halaman yang muncul pertama kali saat pengguna berhasil melakukan Login. Pada halaman home terdapat informasi nama pengguna, BPM terakhir pengguna dan artikel-artikel kesehatan tentang denyut jantung yang dapat dibaca oleh pengguna dengan melakukan klik pada artikel. Tampilan halaman Home dapat dilihat pada gambar 12 berikut.



Gambar 12 - Home