

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

N a m a : 1. Gugum Saefulah Zidni
Kewarganegaraan : Indonesia
Alamat : Jl.desa jatimulya, Rt.02, Rw.02, Blok Jatisema, Desa Jatimulya, Kec. Kasokandel,
Kab. Majalengka.
Kode Pos : 45141

N a m a : 2. Ikrimach, S.Kom., M.Cs.
Kewarganegaraan : Indonesia
Alamat : Jl. Bantul Pendowoharjo, Kab. Bantul, Yogyakarta
Kode Pos : 55186

N a m a : 3. Sutarman, S. Kom., M. Kom., Ph.D.
Kewarganegaraan : Indonesia
Alamat : Jl. Janti Karangjambe 130B, RT.01/RW.00, Banguntapan, Yogyakarta.
Kode Pos : 55198

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : IOT Pada Aplikasi Monitoring Kondisi Kesehatan Denyut Jantung Menggunakan Sensor Max30100 Berbasis Android
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:

- a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
- b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Yogyakarta, 12 Juli 2023



1. Gugum Saefulah Zidni
2. Ikrimach, S.Kom., M.Cs.
3. Sutarman, S.Kom., M. Kom., Ph.D.

Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai.

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

N a m a : Ikrimach, S.Kom., M.Cs.

Alamat : Jl. Bantul Pendowoharjo, Kab. Bantul, Yogyakarta. 55186

N a m a : Sutarman, S. Kom., M. Kom., Ph.D.

Alamat : Jl. Janti Karangjambe 130B, RT.01/RW.00, Banguntapan, Yogyakarta. 55198

N a m a : Gugum Saefulah Zidni

Alamat : Jl. Desa Jatimulya, RT.02/RW.02, Blok Jatisema, Desa Jatimulya, Kec. Kasokandel, Kab. Majalengka, Jawa Barat. 45451

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

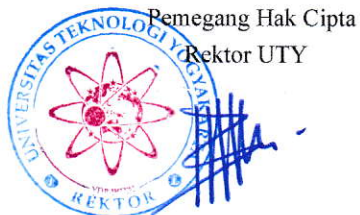
N a m a : Universitas Teknologi Yogyakarta

Alamat : Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I.Yogyakarta, 55285

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa IOT Pada Aplikasi Monitoring Kondisi Kesehatan Denyut Jantung Menggunakan Sensor MAX30100 Berbasis Android untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 13 Juli 2023



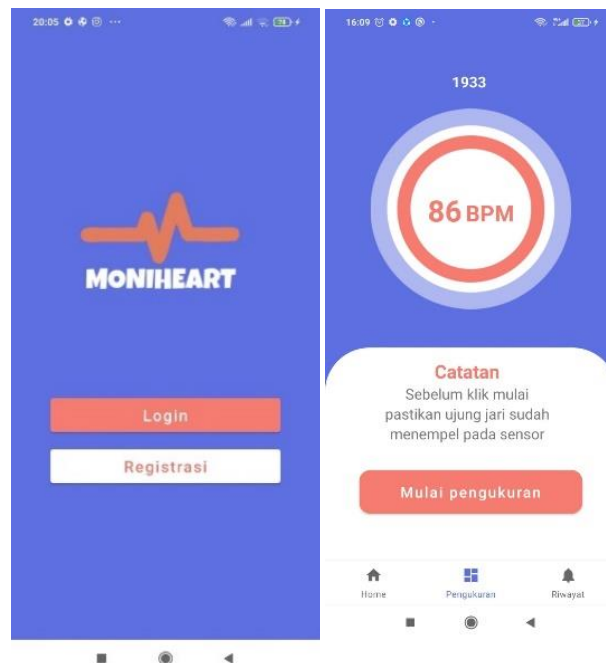
Dr. Bambang Moertono S, MM., Akt, CA.



Ikrimach, S.Kom., M.Cs.
Sutarman, S. Kom., M. Kom., Ph.D.
Gugum Saefulah Zidni

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.
3. Sutarman, S.Kom., M.Kom., Ph.D.

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

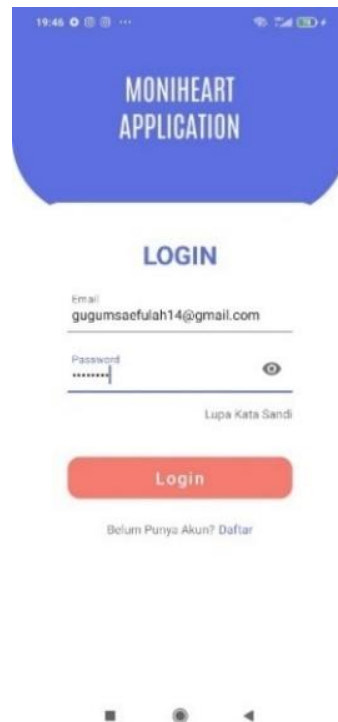
DAFTAR GAMBAR

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.materialSpiner.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.materialSpiner 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> {

```


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

    rvriwayat.adapter = AdapterRiwayat (riwayat)
    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)

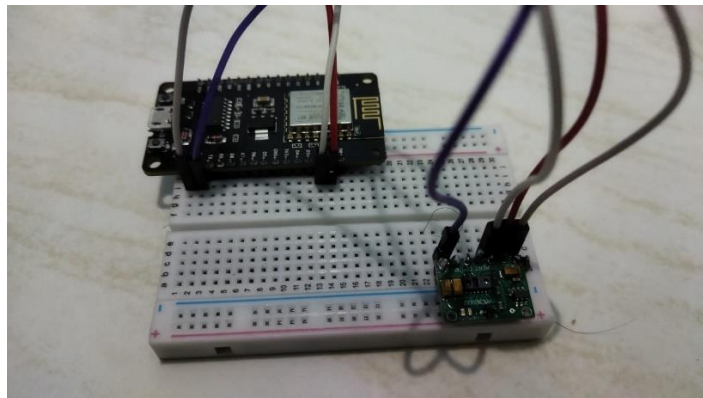
btnlogout.setOnClickListener() {
    s.setIsLogin(false)
    val intent = Intent(this@ProfileActivity,
AwalActivity::class.java)
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
    startActivity(intent)
    finish()
}

}
}

```

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.

3.3 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

REPUBLIC 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 : EC00202365835, 10 Agustus 2023

Pencipta

Nama : **Gugum Saefulah Zidni, Ikrimach, S.Kom., N4.Cs. dkk**

Alamat : Jl.desa Jatimulya, RT.02, RW.02, Blok Jatisema, Desa Jatimulya, Kec. Kasokandel, Kab. Majalengka, Jawa Barat , Kasokandel, Majalengka, Jawa Barat, 45141

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Universitas Teknologi Yogyakarta**

Alamat : Jl. Siliwangi (Ringroad Utara), Jombor, Sleman, D.I. Yogyakarta, Sleman, DI YOGYAKARTA 55285

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **IOT Pada Aplikasi Monitoring Kondisi Kesehatan Denyut Jantung Lvlenggunakan Sensor Max30100 Berbasis Android**

Tanggal dan tempat diumumkan untuk pertama kali : 10 Juli 2023, di Yogyakarta
di wilayah Indonesia atau di luar wilayah Indonesia

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

Nomor pencatatan : 000498781

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 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	Gugum Saefulah Zidni	Jl.desa Jatimulya, RT.02, RW.02, Blok Jatisema, Desa Jatimulya, Kec. Kasokandel, Kab. Majalengka, Jawa Barat
2	Ikrimach, S.Kom., N4.Cs.	Jl. Bantul Pendowoharjo, Kab. Bantul, Yogyakarta
3	Sutarman, S. Kom., M. Kom., Ph.D	Jl. Janti Karangjambe 1308, RT.01/RW.00, Banguntapan, Yogyakarta

