L A M P I R A N

DAFTAR LAMPIRAN

Lampiran- 1 Kode Program Aplikasi

```
@file:Suppress("SetTextI18n")
package your.package.name
import android.content.Intent
import android.os.Bundle
import android.view.inputmethod.EditorInfo
import android.widget.RadioButton
import android.widget.Toast
import androidx.activity.addCallback
import androidx.activity.viewModels
import
androidx.appcompat.app.AppCompatActivit
y
import
androidx.core.widget.addTextChangedListe
ner
import androidx.lifecycle.Lifecycle
import androidx.lifecycle.lifecycleScope
import androidx.lifecycle.repeatOnLifecycle
import
dagger.hilt.android.AndroidEntryPoint
import kotlinx.coroutines.launch
import
your.package.name.databinding.ActivityReg
isterBinding
import javax.inject.Inject

@AndroidEntryPoint
class RegisterActivity :
AppCompatActivity() {
   private lateinit var binding:
ActivityRegisterBinding
   private val viewModel:
RegisterViewModel by viewModels()
   @Inject lateinit var loading:
LoadingAlertDialog
   // Simpan pilihan gender terakhir
   private var gender: Gender? = null
   override fun onCreate(savedInstanceState:
Bundle?) {
      super.onCreate(savedInstanceState)
      binding =
ActivityRegisterBinding.inflate(layoutInflat
er)
      setContentView(binding.root)
      initUi()
      bindActions()
      observeUiState()
   }

   // -------------------- UI Setup ----------------
----
   private fun initUi() = with(binding) {
      // aksi tombol back (gesture sistem)

onBackPressedDispatcher.addCallback(this
@RegisterActivity) {
         navigateToLogin()
      }

      // bersihkan error saat user mengetik
ulang
      etNama.addTextChangedListener {
tilNama.error = null }
      etNomorHp.addTextChangedListener {
tilNomorHp.error = null }
      etAlamat.addTextChangedListener {
tilAlamat.error = null }
      etUsername.addTextChangedListener {
tilUsername.error = null }
      etPassword.addTextChangedListener {
tilPassword.error = null }

      // submit via keyboard (Done)
      etPassword.setOnEditorActionListener
{ _, actionId, _ ->
         if (actionId ==
EditorInfo.IME_ACTION_DONE) {
            submitIfValid()
            true
         } else false
      }
   }

   private fun bindActions() = with(binding)
{
      ivBack.setOnClickListener {
navigateToLogin() }


rgGender.setOnCheckedChangeListener { _,
checkedId ->
         gender = when (checkedId) {
            R.id.rbLakiLaki -> Gender.MALE
            R.id.rbPerempuan ->
Gender.FEMALE
            else -> null
         }
      }
```

```
        btnRegistrasi.setOnClickListener {
submitIfValid() }
    }

    // -------------------- Submit & Validation --
------------------

    private fun submitIfValid() {
        val payload =
validateAndBuildPayload() ?: return
        viewModel.postRegister(
            payload.nama,
            payload.nomorHp,
            payload.alamat,
            payload.username,
            payload.password
        )
    }

    private fun validateAndBuildPayload():
RegistrationPayload? = with(binding) {
        val nama =
etNama.text?.toString()?.trim().orEmpty()
        val nomorHp =
etNomorHp.text?.toString()?.trim().orEmpty
()
        val alamat =
etAlamat.text?.toString()?.trim().orEmpty()
        val username =
etUsername.text?.toString()?.trim().orEmpty
()
        val password =
etPassword.text?.toString()?.trim().orEmpty(
)

        // urutan validasi dibuat jelas dan
singkat
        when {
            nama.isBlank() -> {
                tilNama.error = "Nama wajib
diisi"
                etNama.requestFocus()
                return null
            }
            nomorHp.isBlank() -> {
                tilNomorHp.error = "Nomor HP
wajib diisi"
                etNomorHp.requestFocus()
                return null
            }
            alamat.isBlank() -> {
                tilAlamat.error = "Alamat wajib
diisi"
                etAlamat.requestFocus()
                return null
            }
            gender == null -> {

Toast.makeText(this@RegisterActivity,
"Silakan pilih jenis kelamin",
Toast.LENGTH_SHORT).show()
                return null
            }
            username.isBlank() -> {
                tilUsername.error = "Username
wajib diisi"
                etUsername.requestFocus()
                return null
            }
            password.isBlank() -> {
                tilPassword.error = "Password
wajib diisi"
                etPassword.requestFocus()
                return null
            }
        }

        RegistrationPayload(
            nama = nama,
            nomorHp = nomorHp,
            alamat = alamat,
            username = username,
            password = password,
            genderLabel = gender?.label ?: "" //
jika butuh dikirim ke server
        )
    }

    // -------------------- Observe State ----------
----------

    private fun observeUiState() {
        lifecycleScope.launch {

repeatOnLifecycle(Lifecycle.State.STARTE
D) {
            viewModel.getRegister().collect {
state ->
                when (state) {
                    is UIState.Loading -> {
                        setLoading(true)
                    }
                    is UIState.Success -> {
                        setLoading(false)

handleRegisterSuccess(state.data)
                    }
                    is UIState.Failure -> {
                        setLoading(false)
```

```
handleRegisterFailure(state.message)
                    }
                }
            }
        }
    }
}

// -------------------- Helpers --------------------

private fun setLoading(isLoading: Boolean) = with(binding) {
    if (isLoading) {
        loading.alertDialogLoading(this@RegisterActivity)
    } else {
        loading.alertDialogCancel()
    }
    btnRegistrasi.isEnabled = !isLoading
}

private fun handleRegisterSuccess(data: ResponseModel) {
    if (data.status == "0") {
        Toast.makeText(this, "Registrasi berhasil!", Toast.LENGTH_SHORT).show()
        navigateToLogin()
    } else {
        // tampilkan pesan dari server jika ada
        Toast.makeText(this, data.message_response, Toast.LENGTH_SHORT).show()
    }
}

private fun handleRegisterFailure(message: String) {
    Toast.makeText(this, message.ifBlank { "Terjadi kesalahan saat registrasi" }, Toast.LENGTH_SHORT).show()
}

private fun navigateToLogin() {
    startActivity(Intent(this, LoginActivity::class.java))
    finish()
}

// -------------------- Types --------------------

private enum class Gender(val label: String) {
    MALE("Laki-laki"),
    FEMALE("Perempuan")
}

private data class RegistrationPayload(
    val nama: String,
    val nomorHp: String,
    val alamat: String,
    val username: String,
    val password: String,
    val genderLabel: String
)
}
```

```
package com.abcd.monitoring_gym.ui.activity.login

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.inputmethod.EditorInfo
import android.widget.Toast
import androidx.activity.addCallback
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatActivity
import androidx.core.widget.addTextChangedListener
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject
import com.abcd.monitoring_gym.databinding.ActivityLoginBinding

@AndroidEntryPoint
class LoginActivity : AppCompatActivity() {

    private lateinit var binding: ActivityLoginBinding
    private val viewModel: LoginViewModel by viewModels()

    @Inject lateinit var loading: LoadingAlertDialog

    private lateinit var session: SharedPreferencesLogin
```

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

        session = SharedPreferencesLogin(this)

        initViews()
        wireEvents()
        observeLoginState()
    }

    // -------------------- UI Setup ----------------
----

    private fun initViews() = with(binding) {
        // bersihkan error saat user mengetik
ulang
        etUsername.addTextChangedListener {
etUsername.error = null }
        etPassword.addTextChangedListener {
etPassword.error = null }

        // tekan "Done" di keyboard untuk
submit
        etPassword.setOnEditorActionListener
{ _, actionId, _ ->
            if (actionId ==
EditorInfo.IME_ACTION_DONE) {
                submitLogin()
                true
            } else false
        }

        // tombol back sistem -> tutup activity

onBackPressedDispatcher.addCallback(this
@LoginActivity) { finish() }
    }

    private fun wireEvents() = with(binding)
{
        btnLogin.setOnClickListener {
submitLogin() }
        btnRegister.setOnClickListener {
navigateToRegister() }
    }

    // -------------------- Submit & Validation --
------------------

    private fun submitLogin() {
        val creds =
validateAndExtractCredentials() ?: return
        setLoading(true)
        viewModel.fetchUser(creds.username,
creds.password)
    }

    private fun
validateAndExtractCredentials():
Credentials? = with(binding) {
        val username =
etUsername.text?.toString()?.trim().orEmpty
()
        val password =
etPassword.text?.toString()?.trim().orEmpty(
)

        when {
            username.isBlank() -> {
                etUsername.error = "Username
harus diisi"
                etUsername.requestFocus()
                return null
            }
            password.isBlank() -> {
                etPassword.error = "Password
harus diisi"
                etPassword.requestFocus()
                return null
            }
        }
        return Credentials(username, password)
    }

    // -------------------- Observe State ----------
----------

    private fun observeLoginState() {

viewModel.getUser().observe(this@LoginA
ctivity) { state ->
            when (state) {
                is UIState.Loading ->
setLoading(true)
                is UIState.Success -> {
                    setLoading(false)
                    handleLoginSuccess(state.data)
                }
                is UIState.Failure -> {
                    setLoading(false)

handleLoginFailure(state.message)
                }
            }
        }
```

```
    }

    // -------------------- Handlers ----------------
----

    private fun handleLoginSuccess(data:
UserModel) {
        if (!isValidUser(data)) {
            showToast("Data tidak
ditemukan.\nPastikan Username dan
Password terdaftar.")
            return
        }

        // Persist dan arahkan
        try {
            persistSession(data)
            showToast("Berhasil masuk")
            routeByRole(data.sebagai)
        } catch (ex: Exception) {
            showToast(ex.message ?: "Gagal
menyimpan sesi pengguna")
            Log.d("LoginTAG", "persistSession
error: ${ex.message}")
        }
    }

    private fun handleLoginFailure(message:
String) {
        showToast("Data tidak
ditemukan.\nPastikan Username dan
Password terdaftar.")
        Log.d("LoginTAG", "login failure:
$message")
    }

    // -------------------- Helpers ----------------
---

    private fun isValidUser(user: UserModel):
Boolean = user.idUser != null

    private fun persistSession(user:
UserModel) {
        // gunakan nilai default aman untuk
menghindari NPE
        val idUser = user.idUser ?: throw
IllegalStateException("ID user kosong")
        val nama = user.nama ?: ""
        val nomorHp = user.nomor_hp ?: ""
        val alamat = user.alamat ?: ""
        val jk = user.jenis_kelamin ?: ""
        val uname = user.username ?: ""
        val pwd = user.password ?: ""
        val role = user.sebagai ?: ""

        session.setLogin(
            idUser, nama, nomorHp, alamat, jk,
uname, pwd, role
        )
    }

    private fun routeByRole(role: String?) {
        when (role?.lowercase()) {
            "user" -> {
                startActivity(Intent(this,
MainActivity::class.java))
                finish()
            }
            else -> {
                // fallback: tetap ke MainActivity
atau tampilkan info role
                startActivity(Intent(this,
MainActivity::class.java))
                finish()
            }
        }
    }

    private fun setLoading(isLoading:
Boolean) = with(binding) {
        if (isLoading)
loading.alertDialogLoading(this@LoginActi
vity)
        else loading.alertDialogCancel()
        btnLogin.isEnabled = !isLoading
        btnRegister.isEnabled = !isLoading
    }

    private fun navigateToRegister() {
        startActivity(Intent(this,
RegisterActivity::class.java))
    }

    private fun showToast(msg: String) =
        Toast.makeText(this, msg,
Toast.LENGTH_SHORT).show()

    // -------------------- Types -------------------
-

    private data class Credentials(val
username: String, val password: String)
}
```

```
package com.abcd.monitoring_gym.ui.home

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
```

```
import android.widget.Toast
import androidx.fragment.app.Fragment
import androidx.fragment.app.viewModels
import androidx.lifecycle.Lifecycle
import androidx.lifecycle.lifecycleScope
import androidx.lifecycle.repeatOnLifecycle
import
androidx.recyclerview.widget.LinearLayout
Manager
import
dagger.hilt.android.AndroidEntryPoint
import kotlinx.coroutines.launch
import
com.abcd.monitoring_gym.databinding.Frag
mentHomeBinding
import
com.abcd.monitoring_gym.ui.activity.main.
MainActivity

@AndroidEntryPoint
class HomeFragment : Fragment() {

    private var _binding:
FragmentHomeBinding? = null
    private val binding get() = _binding!!

    private val viewModel: HomeViewModel
by viewModels()
    private lateinit var session:
SharedPreferencesLogin

    // Jika adapter Anda butuh list awal,
siapkan di sini
    private lateinit var pesananAdapter:
PesananAdapter

    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        session =
SharedPreferencesLogin(requireContext())
    }

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding =
FragmentHomeBinding.inflate(inflater,
container, false)
        return binding.root
    }

    override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
        super.onViewCreated(view,
savedInstanceState)
        initRecycler()
        bindClicks()
        observePesanan()

        // fetch data setelah view siap
        val idUser = session.getIdUser()
        viewModel.fetchPesanan(idUser)
    }

    override fun onDestroyView() {
        super.onDestroyView()
        // hindari memory leak pada Fragment
        _binding = null
    }

    // ---------- UI Setup ----------

    private fun initRecycler() = with(binding)
{
        pesananAdapter =
PesananAdapter(arrayListOf())
        rvPesanan.apply {
            adapter = pesananAdapter
            layoutManager =
LinearLayoutManager(
                requireContext(),

LinearLayoutManager.VERTICAL,
                false
            )
            setHasFixedSize(true)
        }
    }

    private fun bindClicks() = with(binding) {
        btnPelatihan.setOnClickListener {
            (activity as?
MainActivity)?.clickPelatihan()
            // atau gunakan Navigation
Component bila ada:
            //
findNavController().navigate(R.id.action_ho
me_to_pelatihan)
        }
        btnAgenda.setOnClickListener {
            (activity as?
MainActivity)?.clickAgenda()
        }
        btnAkun.setOnClickListener {
```

```
        (activity as?
MainActivity)?.clickAccount()
    }
  }

  // -------------------- Observe & Render ----
---------------

  private fun observePesanan() {
    // Jika getPesanan adalah LiveData:

viewModel.getPesanan.observe(viewLifecyc
leOwner) { state ->
        when (state) {
          is UIState.Loading ->
renderLoading(true)
          is UIState.Success -> {
            renderLoading(false)
            renderData(state.data)
          }
          is UIState.Failure -> {
            renderLoading(false)
            renderError(state.message)
          }
        }
    }

    // Jika versi Flow, gunakan ini (ganti
yang di atas):
    /*

viewLifecycleOwner.lifecycleScope.launch
{

viewLifecycleOwner.lifecycle.repeatOnLife
cycle(Lifecycle.State.STARTED) {
          viewModel.getPesanan.collect {
state -> ... }
        }
    }
    */
  }

  private fun renderLoading(show:
Boolean) = with(binding) {
    if (show) startShimmer() else
stopShimmer()
  }

  private fun renderData(items:
ArrayList<PesananModel>) {
    if (items.isEmpty()) {
      toast("Tidak ada data")
    }
    // jika adapter Anda tidak punya
submitList, ganti seluruh data:
    pesananAdapter =
PesananAdapter(items)
    binding.rvPesanan.adapter =
pesananAdapter
  }

  private fun renderError(message: String) {
    toast(message.ifBlank { "Gagal
memuat data" })
  }

  // -------------------- Shimmer Helpers ------
--------------

  private fun startShimmer() =
with(binding) {
    smPesanan.startShimmer()
    smPesanan.visibility = View.VISIBLE
    rvPesanan.visibility = View.GONE
  }

  private fun stopShimmer() =
with(binding) {
    smPesanan.stopShimmer()
    smPesanan.visibility = View.GONE
    rvPesanan.visibility = View.VISIBLE
  }

  // -------------------- Utils --------------------

  private fun toast(msg: String) =
    Toast.makeText(requireContext(), msg,
Toast.LENGTH_SHORT).show()
}
```

```
@AndroidEntryPoint
class PelatihanFragment : Fragment() {
private lateinit var binding:
FragmentPelatihanBinding
private val viewModel:
PelatihanViewModel by viewModels()
private lateinit var adapterJenisPelatihan:
JenisPelatihanAdapter
private lateinit var adapterPelatihan:
PelatihanAdapter
private var checkJenisPelatihanFirst = true
private var nameJenisPelatihan = ""
override fun onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
viewModel.loadData()
}
override fun onCreateView(
```

```
inflater: LayoutInflater, container:
ViewGroup?,
savedInstanceState: Bundle?
): View {
binding =
FragmentPelatihanBinding.inflate(layoutInfl
ater)
return binding.root
}
override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
super.onViewCreated(view,
savedInstanceState)
setTopAppBar()
getJenisPelatihan()
getPelatihan()
}
override fun onResume() {
super.onResume()
viewModel.fetchPelatihan()
}
private fun setTopAppBar() {
binding.myAppBar.apply {
etSearch.addTextChangedListener(object :
TextWatcher {
    override fun beforeTextChanged(
        s: CharSequence?,
        start: Int,
        count: Int,
        after: Int
    ) {
    }
    override fun onTextChanged(s:
CharSequence?, start: Int, before: Int, count:
Int) {
    }
    override fun afterTextChanged(s:
Editable?) {

adapterPelatihan.searchData(s.toString())
        if(!checkJenisPelatihanFirst){

adapterPelatihan.searchJenisPelatihan(name
JenisPelatihan)
        }
    }
})}}
private fun getJenisPelatihan() {
viewModel.getJenisPelatihan.observe(viewL
ifecycleOwner){result->
when(result){
    is UIState.Loading->
setStartShimmerJenisPelatihan()
    is UIState.Failure->
setFailureJenisPelatihan(result.message)
    is UIState.Success->
setSuccessJenisPelatihan(result.data)
}}}
private fun
setFailureJenisPelatihan(message: String) {
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
setStopShimmerJenisPelatihan()
}
private fun setSuccessJenisPelatihan(data:
ArrayList<JenisPelatihanModel>) {
setStopShimmerJenisPelatihan()
val arrayJenisPelatihan :
ArrayList<JenisPelatihanModel> =
arrayListOf()
arrayJenisPelatihan.add(JenisPelatihanMode
l(0, "Semua"))
arrayJenisPelatihan.addAll(data)
if(data.isNotEmpty()){
setAdapterJenisPelatihan(arrayJenisPelatiha
n)
} else{
Toast.makeText(requireContext(), "Tidak
ada data", Toast.LENGTH_SHORT).show()
}}
private fun setAdapterJenisPelatihan(data:
ArrayList<JenisPelatihanModel>) {
adapterJenisPelatihan =
JenisPelatihanAdapter(data, object :
OnClickItem.ClickJenisPelatihan{
override fun
clickJenisPelatihan(jenisPelatihan:
JenisPelatihanModel) {
    checkJenisPelatihanFirst =
jenisPelatihan.id_jenis_pelatihan == 0
    if(checkJenisPelatihanFirst){

adapterPelatihan.searchJenisPelatihan("")
    } else{

adapterPelatihan.searchJenisPelatihan(jenisP
elatihan.jenis_pelatihan!!)
    }
    nameJenisPelatihan =
jenisPelatihan.jenis_pelatihan!!
}
})
binding.rvJenisPelatihan.apply {
adapter = adapterJenisPelatihan
layoutManager =
LinearLayoutManager(requireContext(),
LinearLayoutManager.HORIZONTAL,
false)
}}
private fun getPelatihan() {
```

```
viewModel.getPelatihan.observe(viewLifecy
cleOwner){result->
when(result){
   is UIState.Loading->
setStartShimmerPelatihan()
   is UIState.Failure->
setFailurePelatihan(result.message)
   is UIState.Success->
setSuccessPelatihan(result.data)
}
}}
private fun setFailurePelatihan(message:
String) {
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
setStopShimmerPelatihan()
}
private fun setSuccessPelatihan(data:
ArrayList<PelatihanModel>) {
setStopShimmerPelatihan()
if(data.isNotEmpty()){
setAdapterPelatihan(data)
} else{
Toast.makeText(requireContext(), "Tidak
ada data", Toast.LENGTH_SHORT).show()
}}
private fun setAdapterPelatihan(data:
ArrayList<PelatihanModel>) {
adapterPelatihan = PelatihanAdapter(data)
binding.rvPelatihan.apply {
adapter = adapterPelatihan
layoutManager =
LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
}}
private fun
setStartShimmerJenisPelatihan(){
binding.apply {
smJenisPelatihan.startShimmer()
smJenisPelatihan.visibility = View.VISIBLE
rvJenisPelatihan.visibility = View.GONE
}}
private fun
setStopShimmerJenisPelatihan(){
binding.apply {
smJenisPelatihan.stopShimmer()
smJenisPelatihan.visibility = View.GONE
rvJenisPelatihan.visibility = View.VISIBLE
}}
private fun setStartShimmerPelatihan(){
binding.apply {
smPelatihan.startShimmer()
smPelatihan.visibility = View.VISIBLE
rvPelatihan.visibility = View.GONE
}}
private fun setStopShimmerPelatihan(){
binding.apply {
smPelatihan.stopShimmer()
smPelatihan.visibility = View.GONE
rvPelatihan.visibility = View.VISIBLE
}}}
```

```
@AndroidEntryPoint
class AkunFragment : Fragment() {
private lateinit var binding:
FragmentAkunBinding
private val viewModel: AkunViewModel by
viewModels()
private lateinit var sharedPreferences:
SharedPreferencesLogin
private lateinit var userModel: UserModel
private lateinit var tempUserModel:
UserModel
private var loading = LoadingAlertDialog()
override fun onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
sharedPreferences =
SharedPreferencesLogin(requireContext())
sharedPreferences.apply {
userModel = UserModel(
   getIdUser(), getNama(), getNomorHp(),
getAlamat(), getJenisKelamin(),
   getUsername(), getPassword(),
getSebagai()
)}}
override fun onCreateView(
inflater: LayoutInflater, container:
ViewGroup?,
savedInstanceState: Bundle?
): View {
binding =
FragmentAkunBinding.inflate(layoutInflater
)
return binding.root}
override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
super.onViewCreated(view,
savedInstanceState)
setNavDrawer()
setDataUser()
setButton()
getUpdateData()
getRiwayatPelatihan()
}
private fun setNavDrawer() {
binding.myAppBar.apply {
ivNavDrawer.visibility = View.GONE
tvTitle.visibility = View.GONE
}}
private fun setDataUser() {
```

```
binding.apply {
tvNama.text = userModel.nama
tvNomorHp.text = userModel.nomor_hp
tvAlamat.text = userModel.alamat
tvJenisKelamin.text =
userModel.jenis_kelamin
tvUsername.text = userModel.username
tvPassword.text = userModel.password
}}
private fun setButton() {
binding.apply {
btnAkun.setOnClickListener {
   setAccountActive()
}
btnRiwayat.setOnClickListener {
   setHistoryActive()
}
btnUbahData.setOnClickListener {
   setShowEditProfile()
}
btnLogout.setOnClickListener {
   setLogout()
}}}
private fun setLogout() {
sharedPreferences.setLogout()
startActivity(Intent(requireContext(),
LoginActivity::class.java))}
private fun setShowEditProfile() {
val view =
AlertDialogAkunBinding.inflate(layoutInflat
er)
val alertDialog =
AlertDialog.Builder(requireContext())
alertDialog.setView(view.root)
.setCancelable(false)
val dialogInputan = alertDialog.create()
dialogInputan.show()
view.apply {
etEditNama.setText(sharedPreferences.getN
ama())
etEditNomorHp.setText(sharedPreferences.g
etNomorHp())
etEditAlamat.setText(sharedPreferences.get
Alamat())
etEditUsername.setText(sharedPreferences.g
etUsername())
etEditPassword.setText(sharedPreferences.g
etPassword())
when(sharedPreferences.getJenisKelamin())
{
   "Laki-laki"-> rbLakiLaki.isChecked =
true
   "Perempuan"-> rbPerempuan.isChecked =
true
}
var selectedGender = ""
rgGender.setOnCheckedChangeListener { _,
checkedId ->
   when (checkedId) {
      R.id.rbLakiLaki -> selectedGender =
"Laki-laki"
      R.id.rbPerempuan -> selectedGender =
"Perempuan"
   }
}
btnSimpan.setOnClickListener {
   var cek = false
   if(etEditNama.toString().isEmpty()){
      etEditNama.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditAlamat.toString().isEmpty()){
      etEditAlamat.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditNomorHp.toString().isEmpty()){
      etEditNomorHp.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditUsername.toString().isEmpty()){
      etEditUsername.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditPassword.toString().isEmpty()){
      etEditPassword.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(!cek){
      val nama = etEditNama.text.toString()
      val nomorHp =
etEditNomorHp.text.toString()
      val alamat =
etEditAlamat.text.toString()
      val jenisKelamin = selectedGender
      val username =
etEditUsername.text.toString()
      val password =
etEditPassword.text.toString()
      val usernameLama =
sharedPreferences.getUsername()
      tempUserModel = UserModel(
         sharedPreferences.getIdUser(),
         nama, nomorHp, alamat,
jenisKelamin,
         username, password, "user"
```

```
        )
        postUpdateData(
            sharedPreferences.getIdUser(),
nama, nomorHp, alamat,
            jenisKelamin, username, password,
usernameLama
        )
        dialogInputan.dismiss()
    }}
btnBatal.setOnClickListener {
    dialogInputan.dismiss()
}}}
private fun postUpdateData(
idUser: Int,
nama: String,
nomorHp: String,
alamat: String,
jenisKelamin: String,
username: String,
password: String,
usernameLama: String){
viewModel.postUpdateProgress(
idUser, nama, nomorHp, alamat,
jenisKelamin, username, password,
usernameLama
)}
private fun getUpdateData(){
viewModel.getUpdateProgress.observe(view
LifecycleOwner){result->
when(result){
    is UIState.Loading ->
loading.alertDialogLoading(requireContext(
))
    is UIState.Failure ->
setFailureUpdateData(result.message)
    is UIState.Success ->
setSuccessUpdateData(result.data)
}}}
private fun setFailureUpdateData(message:
String) {
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
loading.alertDialogCancel()
}
private fun setSuccessUpdateData(data:
ResponseModel) {
loading.alertDialogCancel()
if(data.status=="0"){
Toast.makeText(requireContext(), "Berhasil
Update Data",
Toast.LENGTH_SHORT).show()
userModel = tempUserModel
tempUserModel = UserModel()
setDataUser()
} else{
Toast.makeText(requireContext(),
data.message_response,
Toast.LENGTH_SHORT).show()
}}
private fun setAccountActive() {
binding.apply {
btnAkun.setBackgroundResource(R.drawabl
e.bg_card_active)
btnAkun.setTextColor(ContextCompat.getC
olor(binding.root.context, R.color.white))
llAccount.visibility = View.VISIBLE
btnRiwayat.setBackgroundResource(R.draw
able.bg_card)
btnRiwayat.setTextColor(ContextCompat.ge
tColor(binding.root.context, R.color.black))
llRiwayat.visibility = View.GONE
}}
private fun setHistoryActive() {
binding.apply {
btnAkun.setBackgroundResource(R.drawabl
e.bg_card)
btnAkun.setTextColor(ContextCompat.getC
olor(binding.root.context, R.color.black))
llAccount.visibility = View.GONE
btnRiwayat.setBackgroundResource(R.draw
able.bg_card_active)
btnRiwayat.setTextColor(ContextCompat.ge
tColor(binding.root.context, R.color.white))
llRiwayat.visibility = View.VISIBLE
fetchRiwayatPelatihan(sharedPreferences.ge
tIdUser())
}}
private fun fetchRiwayatPelatihan(idUser:
Int) {
viewModel.fetchRiwayatPelatihan(idUser)
}
private fun getRiwayatPelatihan(){
viewModel.getRiwayat.observe(viewLifecy
cleOwner){result->
when(result){
    is UIState.Loading -> {}
    is UIState.Failure ->
setFailureRiwayatPelatihan(result.message)
    is UIState.Success ->
setSuccessRiwayatPelatihan(result.data)
}}}
private fun
setFailureRiwayatPelatihan(message: String)
{
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
}
private fun
setSuccessRiwayatPelatihan(data:
ArrayList<PesananModel>) {
```

```
if(data.isNotEmpty()){
setAdapterPelatihan(data)
} else{
Toast.makeText(requireContext(), "Tidak
ada data", Toast.LENGTH_SHORT).show()
}}
private fun setAdapterPelatihan(data:
ArrayList<PesananModel>) {
val adapterPesanan = PesananAdapter(data)
binding.rvRiwayat.apply {
adapter = adapterPesanan
layoutManager =
LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
}}}
```

```
@AndroidEntryPoint
class AkunFragment : Fragment() {
private lateinit var binding:
FragmentAkunBinding
private val viewModel: AkunViewModel by
viewModels()
private lateinit var sharedPreferences:
SharedPreferencesLogin
private lateinit var userModel: UserModel
private lateinit var tempUserModel:
UserModel
private var loading = LoadingAlertDialog()
override fun onCreate(savedInstanceState:
Bundle?) {
super.onCreate(savedInstanceState)
sharedPreferences =
SharedPreferencesLogin(requireContext())
sharedPreferences.apply {
userModel = UserModel(
   getIdUser(), getNama(), getNomorHp(),
getAlamat(), getJenisKelamin(),
   getUsername(), getPassword(),
getSebagai()
)}}
override fun onCreateView(
inflater: LayoutInflater, container:
ViewGroup?,
savedInstanceState: Bundle?
): View {
binding =
FragmentAkunBinding.inflate(layoutInflater
)
return binding.root}
override fun onViewCreated(view: View,
savedInstanceState: Bundle?) {
super.onViewCreated(view,
savedInstanceState)
setNavDrawer()
setDataUser()
setButton()
getUpdateData()
getRiwayatPelatihan()
}
private fun setNavDrawer() {
binding.myAppBar.apply {
ivNavDrawer.visibility = View.GONE
tvTitle.visibility = View.GONE
}}
private fun setDataUser() {
binding.apply {
tvNama.text = userModel.nama
tvNomorHp.text = userModel.nomor_hp
tvAlamat.text = userModel.alamat
tvJenisKelamin.text =
userModel.jenis_kelamin
tvUsername.text = userModel.username
tvPassword.text = userModel.password
}}
private fun setButton() {
binding.apply {
btnAkun.setOnClickListener {
   setAccountActive()
}
btnRiwayat.setOnClickListener {
   setHistoryActive()
}
btnUbahData.setOnClickListener {
   setShowEditProfile()
}
btnLogout.setOnClickListener {
   setLogout()
}}}
private fun setLogout() {
sharedPreferences.setLogout()
startActivity(Intent(requireContext(),
LoginActivity::class.java))}
private fun setShowEditProfile() {
val view =
AlertDialogAkunBinding.inflate(layoutInflat
er)
val alertDialog =
AlertDialog.Builder(requireContext())
alertDialog.setView(view.root)
.setCancelable(false)
val dialogInputan = alertDialog.create()
dialogInputan.show()
view.apply {
etEditNama.setText(sharedPreferences.getN
ama())
etEditNomorHp.setText(sharedPreferences.g
etNomorHp())
etEditAlamat.setText(sharedPreferences.get
Alamat())
etEditUsername.setText(sharedPreferences.g
etUsername())
```

```
etEditPassword.setText(sharedPreferences.g
etPassword())
when(sharedPreferences.getJenisKelamin())
{
   "Laki-laki"-> rbLakiLaki.isChecked =
true
   "Perempuan"-> rbPerempuan.isChecked =
true
}
var selectedGender = ""
rgGender.setOnCheckedChangeListener { _,
checkedId ->
   when (checkedId) {
      R.id.rbLakiLaki -> selectedGender =
"Laki-laki"
      R.id.rbPerempuan -> selectedGender =
"Perempuan"
   }
}
btnSimpan.setOnClickListener {
   var cek = false
   if(etEditNama.toString().isEmpty()){
      etEditNama.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditAlamat.toString().isEmpty()){
      etEditAlamat.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditNomorHp.toString().isEmpty()){
      etEditNomorHp.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditUsername.toString().isEmpty()){
      etEditUsername.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(etEditPassword.toString().isEmpty()){
      etEditPassword.error = "Tidak Boleh
Kosong"
      cek = true
   }
   if(!cek){
      val nama = etEditNama.text.toString()
      val nomorHp =
etEditNomorHp.text.toString()
      val alamat =
etEditAlamat.text.toString()
      val jenisKelamin = selectedGender
      val username =
etEditUsername.text.toString()
      val password =
etEditPassword.text.toString()
      val usernameLama =
sharedPreferences.getUsername()
      tempUserModel = UserModel(
         sharedPreferences.getIdUser(),
         nama, nomorHp, alamat,
jenisKelamin,
         username, password, "user"
      )
      postUpdateData(
         sharedPreferences.getIdUser(),
nama, nomorHp, alamat,
         jenisKelamin, username, password,
usernameLama
      )
      dialogInputan.dismiss()
   }}
btnBatal.setOnClickListener {
   dialogInputan.dismiss()
}}}
private fun postUpdateData(
idUser: Int,
nama: String,
nomorHp: String,
alamat: String,
jenisKelamin: String,
username: String,
password: String,
usernameLama: String){
viewModel.postUpdateProgress(
idUser, nama, nomorHp, alamat,
jenisKelamin, username, password,
usernameLama
)}
private fun getUpdateData(){
viewModel.getUpdateProgress.observe(view
LifecycleOwner){result->
when(result){
   is UIState.Loading ->
loading.alertDialogLoading(requireContext(
))
   is UIState.Failure ->
setFailureUpdateData(result.message)
   is UIState.Success ->
setSuccessUpdateData(result.data)
}}}
private fun setFailureUpdateData(message:
String) {
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
loading.alertDialogCancel()
}
private fun setSuccessUpdateData(data:
ResponseModel) {
```

```
loading.alertDialogCancel()
if(data.status=="0"){
Toast.makeText(requireContext(), "Berhasil
Update Data",
Toast.LENGTH_SHORT).show()
userModel = tempUserModel
tempUserModel = UserModel()
setDataUser()
} else{
Toast.makeText(requireContext(),
data.message_response,
Toast.LENGTH_SHORT).show()
}}
private fun setAccountActive() {
binding.apply {
btnAkun.setBackgroundResource(R.drawabl
e.bg_card_active)
btnAkun.setTextColor(ContextCompat.getC
olor(binding.root.context, R.color.white))
llAccount.visibility = View.VISIBLE
btnRiwayat.setBackgroundResource(R.draw
able.bg_card)
btnRiwayat.setTextColor(ContextCompat.ge
tColor(binding.root.context, R.color.black))
llRiwayat.visibility = View.GONE
}}
private fun setHistoryActive() {
binding.apply {
btnAkun.setBackgroundResource(R.drawabl
e.bg_card)
btnAkun.setTextColor(ContextCompat.getC
olor(binding.root.context, R.color.black))
llAccount.visibility = View.GONE
btnRiwayat.setBackgroundResource(R.draw
able.bg_card_active)
btnRiwayat.setTextColor(ContextCompat.ge
tColor(binding.root.context, R.color.white))
llRiwayat.visibility = View.VISIBLE
fetchRiwayatPelatihan(sharedPreferences.ge
tIdUser())
}}
private fun fetchRiwayatPelatihan(idUser:
Int) {
viewModel.fetchRiwayatPelatihan(idUser)
}
private fun getRiwayatPelatihan(){
viewModel.getRiwayat.observe(viewLifecy
cleOwner){result->
when(result){
   is UIState.Loading -> {}
   is UIState.Failure ->
setFailureRiwayatPelatihan(result.message)
   is UIState.Success ->
setSuccessRiwayatPelatihan(result.data)
}}}
private fun
setFailureRiwayatPelatihan(message: String)
{
Toast.makeText(requireContext(), message,
Toast.LENGTH_SHORT).show()
}
private fun
setSuccessRiwayatPelatihan(data:
ArrayList<PesananModel>) {
if(data.isNotEmpty()){
setAdapterPelatihan(data)
} else{
Toast.makeText(requireContext(), "Tidak
ada data", Toast.LENGTH_SHORT).show()
}}
private fun setAdapterPelatihan(data:
ArrayList<PesananModel>) {
val adapterPesanan = PesananAdapter(data)
binding.rvRiwayat.apply {
adapter = adapterPesanan
layoutManager =
LinearLayoutManager(requireContext(),
LinearLayoutManager.VERTICAL, false)
}}}
```

```
<?php
session_start();
include_once '../koneksi.php';

date_default_timezone_set('Asia/Makassar');
$tanggalSaja = date('Y-m-d');
$waktuSaja = date('H:i:s');
$id_pelatih = "";
if(isset($_COOKIE['id_user'])){
$id_pelatih = $_COOKIE['id_user'];
} else if(isset($_SESSION['id_user'])){
$id_pelatih = $_SESSION['id_user'];
}

if (!isset($_SESSION['sebagai']) &&
!isset($_COOKIE['sebagai'])) {
header("Location: ../login/");
exit();
}?>
<div >
<div >
<h4 class="fw-bold py-3 mb-4"><span
></span> Pelatihan Terdaftar</h4>
<div class="card">
<h5 >Halaman Pelatihan Terdaftar</h5>
<div>
<table class="table">
   <tr >
      <th>Nomor</th>
      <th>Jenis Pelatihan</th>
      <th>Pelatihan</th>
```

```
        <th>Deskripsi</th>
        <th>Hari Khusus</th>
    </tr>
    <tbody>
    <?php
    $no = 1;
    $programQuery =
mysqli_query($koneksi, "SELECT * FROM
program WHERE id_pelatih='$id_pelatih'
");
    while ($row =
mysqli_fetch_assoc($programQuery)) {
        $id_program = $row['id_program'];
        $id_pelatihan = $row['id_pelatihan'];
        $querypelatihan =
mysqli_query($koneksi, "SELECT * FROM
pelatihan WHERE
id_pelatihan='$id_pelatihan' ");
        $rowpelatihan =
mysqli_fetch_assoc($querypelatihan);
        $id_jenis_pelatihan =
$rowpelatihan['id_jenis_pelatihan'];
        $pelatihan = $rowpelatihan['pelatihan'];
        $deskripsi = $rowpelatihan['deskripsi'];
        $hari_khusus =
$rowpelatihan['hari_khusus'];

        $queryjenis_pelatihan =
mysqli_query($koneksi, "SELECT * FROM
jenis_pelatihan WHERE
id_jenis_pelatihan='$id_jenis_pelatihan' ");
        $rowjenis_pelatihan =
mysqli_fetch_assoc($queryjenis_pelatihan);
        $jenis_pelatihan =
$rowjenis_pelatihan['jenis_pelatihan'];
    ?>
        <tr>
        <th ><?= $no ?></th>
        <td><?= $jenis_pelatihan ?></td>
        <td><?= $pelatihan ?></td>
        <td><?= $deskripsi ?></td>
        <td><?= $hari_khusus ?></td>
        </tr>
    <?php
        $no++;
    }
    ?>
    </tbody>
</table>
</div></div></div>
```

```
<?php
session_start();
include_once '../koneksi.php';

date_default_timezone_set('Asia/Makassar');
$tanggalSaja = date('Y-m-d');
$waktuSaja = date('H:i:s');

// Check if user is logged in
if (!isset($_SESSION['sebagai']) &&
!isset($_COOKIE['sebagai'])) {
header("Location: ../login/");
exit();
}
?>
<div >
    <h4><span >Pelatihan /</span> Jenis
Pelatihan</h4>
    <div class="card">
    <h5 class="card-header">Halaman Jenis
Pelatihan</h5>
<div >
<button
type="button"
data-bs-target="#tambahData"
data-bs-toggle="modal"
>
Tambah Data
</button>

<div tabindex="-1" id="tambahData" data-
bs-backdrop="static" >
<div >
<form action="jenis-pelatihan-post.php"
method="post">
<div >
<h5 id="tambahDataTitle">Tambah
Data</h5>
<button
data-bs-dismiss="modal"
aria-label="Close"
type="button"
></button>
</div>
<div >
<label for="jenis_pelatihan" >Jenis
Pelatihan</label>
<input type="link" id="jenis_pelatihan"
name="jenis_pelatihan "
placeholder="Masukkan Jenis Pelatihan"
required />
</div></div>
<div class="modal-footer">
<button type="button" data-bs-
dismiss="modal">
Close
</button>
<button type="submit"
name="tambah_data" >Save</button>
</div>
```

```
</form>
</div></div>
<table >
<thead>
   <tr >
      <th>No</th>
      <th>Jenis Pelatihan</th>
      <th>Actions</th>
   </tr>
</thead>
<tbody>
<?php
$no = 1;
$queryJenisPelatihan =
mysqli_query($koneksi, "SELECT * FROM
jenis_pelatihan ");
$row =
mysqli_num_rows($queryJenisPelatihan);
while ($row =
mysqli_fetch_assoc($queryJenisPelatihan))
{
$id_jenis_pelatihan =
$row['id_jenis_pelatihan'];
$jenis_pelatihan = $row['jenis_pelatihan'];
?>
<tr>
   <th scope="row"><?= $no ?></th>
   <td><?= $jenis_pelatihan ?></td>
   <td>
   <div class="dropdown">
   <button type="button" data-bs-
toggle="dropdown">
      <i class="bx bx-dots-vertical-
rounded"></i>
   </button>
   <div >
      <a href="" data-bs-toggle="modal"
data-bs-target="#edit<?=
$id_jenis_pelatihan ?>"><i class="bx bx-
edit-alt me-1"></i> Edit</a>
      <a href="" data-bs-toggle="modal"
data-bs-target="#hapus<?=
$id_jenis_pelatihan ?>"><i class="bx bx-
trash me-1"></i> Delete</a>
   </div></div>
   </td>
</tr>

<div class="modal fade" id="edit<?=
$id_jenis_pelatihan ?>" >
<div >
<div >
<form method="post" action="jenis-
pelatihan-post.php">
   <div>
      <h5 id="tambahDataTitle">Edit
Data</h5>
   <button
      data-bs-dismiss="modal"
      type="button"
      aria-label="Close"></button>
   </div>
   <div >
   <input type="text"
id="id_jenis_pelatihan"
name="id_jenis_pelatihan " value="<?=
$id_jenis_pelatihan ?>" hidden />
   <div >
      <div >
      <label for="jenis_pelatihan" >Jenis
Pelatihan</label>
      <input "
value="<?=$jenis_pelatihan?>"
placeholder="Masukkan Jenis Pelatihan"
name="jenis_pelatihan" type="link"
id="jenis_pelatihan/>
      </div>
   </div>
   <div >
   <button data-bs-dismiss="modal"
type="button" >
      Close
   </button>
   <button type="submit" name="edit_data"
>Save</button>
   </div>
</form>
</div></div>
<!-- Modal Hapus -->
<div id="hapus<?= $id_jenis_pelatihan ?>"
tabindex="-1" data-bs-backdrop="static" >
<div>
<div >
<form action="jenis-pelatihan-post.php"
method="post">
   <div >
   <h5 id="tambahDataTitle">Hapus
Data</h5>
   <button
      data-bs-dismiss="modal"
      type="button"
      aria-label="Close"></button>
   </div>
   <div >
   <input type="text"
id="id_jenis_pelatihan"
name="id_jenis_pelatihan"
placeholder="Masukkan id_jenis_pelatihan"
value="<?= $id_jenis_pelatihan ?>" hidden
/>
```

```
   <div class="row">
      <div class="col mb-3">
      <label for="judul" >Hapus Data
ini?</label>
      <label for="">Klik delete untuk
menghapus</label>
      </div>
   </div>
   </div>
   <div class="modal-footer">
   <button type="button" data-bs-
dismiss="modal">
      Close
   </button>
   <button type="submit"
name="hapus_data">Delete</button>
   </div>
</form>
</div>
<?php
$no++;
}
?>
</tbody>
</table>
</div>
```

```
<?php
session_start();
include_once '../koneksi.php';

date_default_timezone_set('Asia/Makassar');
$tanggalSaja = date('Y-m-d');
$waktuSaja = date('H:i:s');
$tanggalDanWaktu = date('Y-m-d H:i:s');

// Check if user is logged in
if (!isset($_SESSION['sebagai']) &&
!isset($_COOKIE['sebagai'])) {
header("Location: ../login/");
exit();
}
require_once "additionFunction.php";
?>
<div >
<div >
<h4 ><span></span> Riwayat Pesanan
</h4>
<div >
<h5 >Halaman Riwayat Pesanan</h5>
<div>
<button
type="button"
data-bs-toggle="modal"
data-bs-target="#tambahData">
Tambah Data
</button>
<!-- Modal -->
<div id="tambahData" data-bs-
backdrop="static" tabindex="-1">
<div >
<form class="modal-content"
action="riwayat-pesanan-post.php"
method="post" enctype="multipart/form-
data">
<div >
<h5 class="modal-title"
id="tambahDataTitle">Tambah Data</h5>
<button
type="button"
data-bs-dismiss="modal"
aria-label="Close"></button>
</div>
<div >
<div >
<div >
<label for="id_user" class="form-
label">user</label>
<select name="id_user" class="form-select"
id="id_user">
   <?php
   $query = mysqli_query($koneksi,
"SELECT * FROM user WHERE
sebagai='user' ");
   while ($row =
mysqli_fetch_assoc($query)) {
   $w_id_user = $row['id_user'];
   $nama = $row['nama'];
   ?>
   <option value="<?= $w_id_user ?>"><?=
$nama ?></option>
   <?php
   }
   ?>
</select>
</div>
</div>
<div class="row">
<div class="col mb-3">
<label for="id_pelatih" >pelatih</label>
<select name="id_pelatih" class="form-
select" id="id_pelatih">
   <?php
   $query = mysqli_query($koneksi,
"SELECT * FROM user WHERE
sebagai='pelatih' ");
   while ($row =
mysqli_fetch_assoc($query)) {
   $w_id_pelatih = $row['id_user'];
   $nama = $row['nama'];
   ?>
```

```
   <option value="<?= $w_id_pelatih
?>"><?= $nama ?></option>
   <?php
   }
   ?>
</select>
</div></div>
<div class="row">
<div class="col mb-3">
<label for="id_pelatihan
">Pelatihan</label>
<select name="id_pelatihan"
onchange="onClickSelected()"
id="id_pelatihan">
   <option value="">- Pilih Pelatihan -
</option>
   <?php
   $query = mysqli_query($koneksi,
"SELECT * FROM pelatihan ");
   while ($row =
mysqli_fetch_assoc($query)) {
   $w_id_pelatihan = $row['id_pelatihan'];
   $pelatihan = $row['pelatihan'];
   $id_jenis_pelatihan =
$row['id_jenis_pelatihan'];
   $deskripsi = $row['deskripsi'];
   $hari_khusus = $row['hari_khusus'];
   $harga = $row['harga'];
   $valueHarga = (strlen($harga) > 0) ?
$harga : "0";

   $queryJenisPelatihan =
mysqli_query($koneksi, "SELECT * FROM
jenis_pelatihan WHERE
id_jenis_pelatihan='$id_jenis_pelatihan' ");
   $dataJenisPelatihan =
mysqli_fetch_assoc($queryJenisPelatihan);
   $jenis_pelatihan =
$dataJenisPelatihan['jenis_pelatihan'];

   ?>
   <option value="<?= $w_id_pelatihan ?>"
      data-pelatihan="<?= $pelatihan ?>"
      data-jenis="<?= $jenis_pelatihan ?>"
      data-deskripsi="<?= $deskripsi ?>"
      data-hari="<?= $hari_khusus ?>"
      data-harga="<?= $harga ?>">
      <?= $pelatihan ?>
   </option>
   <?php
   }
   ?>
</select>
</div></div>
<div class="row">
<div class="col mb-3">
<label for="pelatihan">Pelatihan</label>
<input type="text" id="pelatihan"
name="pelatihan" class="form-control"
value="" />
</div></div>
<div class="row">
<div class="col mb-3">
<label for="jenis_pelatihan">Jenis
Pelatihan</label>
<input type="text" id="jenis_pelatihan"
name="jenis_pelatihan" value="" readonly
/>
</div></div>
<div >
<div >
<label for="deskripsi ">deskripsi</label>
<textarea id="deskripsi" name="deskripsi"
rows="3" placeholder="Masukkan
deskripsi" required></textarea>
</div></div>
<div class="row">
<div >
<label for="hari_khusus" >Hari
Khusus</label>
<select name="hari_khusus"
id="hari_khusus">
   <option value="">-</option>
   <option value="Senin" <?= ($hari_khusus
== 'Senin') ? 'selected' : "
?>>Senin</option>
   <option value="Selasa" <?=
($hari_khusus == 'Selasa') ? 'selected' : "
?>>Selasa</option>
   <option value="Rabu" <?= ($hari_khusus
== 'Rabu') ? 'selected' : " ?>>Rabu</option>
   <option value="Kamis" <?=
($hari_khusus == 'Kamis') ? 'selected' : "
?>>Kamis</option>
   <option value="Jumat" <?= ($hari_khusus
== 'Jumat') ? 'selected' : "
?>>Jumat</option>
   <option value="Sabtu" <?= ($hari_khusus
== 'Sabtu') ? 'selected' : "
?>>Sabtu</option>
   <option value="Minggu" <?=
($hari_khusus == 'Minggu') ? 'selected' : "
?>>Minggu</option>
</select>
</div></div>
<div class="row">
<div class="col mb-3">
<label for="harga">harga</label>
<input type="number" id="harga"
name="harga" value="" />
```

```
</div></div>
<div class="row">
<div “
<label for="tanggal" class="form-
label">tanggal</label>
<input type="date" id="tanggal"
name="tanggal" class="form-control"
value="<?= $tanggalSaja ?>" />
</div>
<div class="col mb-3">
<label for="waktu ">waktu</label>
<input type="time" id="waktu"
name="waktu" class="form-control"
value="<?= $waktuSaja ?>" />
</div></div></div>
<div class="modal-footer">
<button type="button"  data-bs-
dismiss="modal">
Close
</button>
<button type="submit"
name="tambah_data" ">Save</button>
</div>
</form>
</div></div>

<table >
<thead>
   <tr >
      <th>No</th>
      <th>User</th>
      <th>Pelatih</th>
      <th>Pelatihan</th>
      <th>Jenis Latihan</th>
      <th>Deskripsi</th>
      <th>Hari Khusus</th>
      <th>Harga</th>
      <th>Tanggal</th>
      <th>Waktu</th>
      <th>Actions</th>
   </tr>
</thead>
<tbody>
<?php
$no = 1;
$querypesanan = mysqli_query($koneksi,
"SELECT * FROM pesanan WHERE
selesai='1' ");
$row = mysqli_num_rows($querypesanan);
while ($row =
mysqli_fetch_assoc($querypesanan)) {
$id_pesanan = $row['id_pesanan'];
$id_user = $row['id_user'];
$id_pelatih = $row['id_pelatih'];
$id_pelatihan = $row['id_pelatihan'];
$pelatihan = $row['pelatihan'];
$jenis_pelatihan = $row['jenis_pelatihan'];
$deskripsi = $row['deskripsi'];
$hari_khusus = $row['hari_khusus'];
$harga = $row['harga'];
$selesai = $row['selesai'];
$tanggal = $row['tanggal'];
$waktu = $row['waktu'];
$valueHarga = (strlen($harga) > 0) ? $harga
: 0;

$queryuser = mysqli_query($koneksi,
"SELECT * FROM user WHERE
id_user='$id_user' ");
$datauser =
mysqli_fetch_assoc($queryuser);
$user = $datauser['nama'];

$querypelatih = mysqli_query($koneksi,
"SELECT * FROM user WHERE
id_user='$id_pelatih' ");
$datapelatih =
mysqli_fetch_assoc($querypelatih);
$pelatih = $datapelatih['nama'];
?>
<tr>
   <th scope="row"><?= $no ?></th>
   <td><?= $user ?></td>
   <td><?= $pelatih ?></td>
   <td><?= $pelatihan ?></td>
   <td><?= $jenis_pelatihan ?></td>
   <td><?= $deskripsi ?></td>
   <td><?= $hari_khusus ?></td>
   <td><?= formatRupiah($valueHarga)
?></td>
   <td><?= $tanggal ?></td>
   <td><?= $waktu ?></td>
   <td>
   <div class="dropdown">
   <button type="button" data-bs-
toggle="dropdown">
      <i class="bx bx-dots-vertical-
rounded"></i>
   </button>
   <div >
      <a href="" data-bs-toggle="modal"
data-bs-target="#edit<?= $id_pesanan
?>"><i class="bx bx-edit-alt me-1"></i>
Edit</a>
      <a class="dropdown-item" href=""
data-bs-toggle="modal" data-bs-
target="#hapus<?= $id_pesanan ?>"><i
class="bx bx-trash me-1"></i> Delete</a>
   </div>
   </div>
```

```
    </td>
</tr>

<!-- Modal Edit -->
<div id="edit<?= $id_pesanan ?>" data-bs-
backdrop="static" tabindex="-1">
<div >
<div >
<form action="riwayat-pesanan-post.php"
method="post" enctype="multipart/form-
data">
    <div class="modal-header">
    <h5 class="modal-title"
id="tambahDataTitle">Edit Data</h5>
    <button
        type="button"
        class="btn-close"
        data-bs-dismiss="modal"
        aria-label="Close"></button>
    </div>
    <div class="modal-body">
    <input type="text" id="id_pesanan"
name="id_pesanan" value="<?=
$id_pesanan ?>" hidden />
    <div class="row">
        <div class="col mb-3">
        <label for="id_user" class="form-
label">user</label>
        <select name="id_user" class="form-
select" id="id_user">
            <?php
            $query = mysqli_query($koneksi,
"SELECT * FROM user WHERE
sebagai='user' ");
            while ($row =
mysqli_fetch_assoc($query)) {
            $w_id_user = $row['id_user'];
            $nama = $row['nama'];
            ?>
            <option value="<?= $w_id_user ?>"
<?= ($w_id_user == $id_user) ? 'selected' : "
?>><?= $nama ?></option>
            <?php
            }
            ?>
        </select>
        </div>
    </div>
    <div class="row">
        <div class="col mb-3">
        <label for="id_pelatih"
>pelatih</label>
        <select name="id_pelatih"
class="form-select" id="id_pelatih">
            <?php
            $query = mysqli_query($koneksi,
"SELECT * FROM user WHERE
sebagai='pelatih' ");
            while ($row =
mysqli_fetch_assoc($query)) {
            $w_id_pelatih = $row['id_user'];
            $nama = $row['nama'];
            ?>
            <option value="<?= $w_id_pelatih
?>" <?= ($w_id_pelatih == $id_pelatih) ?
'selected' : " ?>><?= $nama ?></option>
            <?php
            }
            ?>
        </select>
        </div>
    </div>
    <div class="row">
        <div class="col mb-3">
        <label for="pelatihan" class="form-
label">Pelatihan</label>
        <input type="text" id="pelatihan"
name="pelatihan" class="form-control"
value="<?=$pelatihan?>" />
        </div>
    </div>
    <div class="row">
        <div class="col mb-3">
        <label for="jenis_pelatihan"
class="form-label">Jenis Pelatihan</label>
        <input type="text" id="jenis_pelatihan"
name="jenis_pelatihan" class="form-
control" value="<?=$jenis_pelatihan?>" />
        </div>
    </div>
    <div class="row">
        <div class="col mb-3">
        <label for="deskripsi"
>deskripsi</label>
        <textarea class="form-control"
id="deskripsi" name="deskripsi" rows="3"
placeholder="Masukkan deskripsi"
required><?=$deskripsi?></textarea>
        </div>
    </div>
    <div class="row">
        <div class="col mb-3">
        <label for="hari_khusus"> Hari
Khusus</label>
        <select name="hari_khusus"
class="form-control" id="hari_khusus">
            <option value="">-</option>
            <option value="Senin" <?=
($hari_khusus == 'Senin') ? 'selected' : "
?>>Senin</option>
```

```
        <option value="Selasa" <?=
($hari_khusus == 'Selasa') ? 'selected' : ''
?>>Selasa</option>
        <option value="Rabu" <?=
($hari_khusus == 'Rabu') ? 'selected' : ''
?>>Rabu</option>
        <option value="Kamis" <?=
($hari_khusus == 'Kamis') ? 'selected' : ''
?>>Kamis</option>
        <option value="Jumat" <?=
($hari_khusus == 'Jumat') ? 'selected' : ''
?>>Jumat</option>
        <option value="Sabtu" <?=
($hari_khusus == 'Sabtu') ? 'selected' : ''
?>>Sabtu</option>
        <option value="Minggu" <?=
($hari_khusus == 'Minggu') ? 'selected' : ''
?>>Minggu</option>
      </select>
      </div>
   </div>
   <div class="row">
      <div class="col mb-3">
      <label for="harga" >harga</label>
      <input type="number" id="harga"
name="harga" class="form-control"
value="<?=$harga?>" />
      </div>
   </div>
   <div class="row">
      <div class="col mb-3">
      <label for="tanggal">tanggal</label>
      <input type="date" id="tanggal"
name="tanggal" class="form-control"
value="<?=$tanggal?>" />
      </div>
      <div class="col mb-3">
      <label for="waktu”>waktu</label>
      <input type="time" id="waktu"
name="waktu" class="form-control"
value="<?= $waktu ?>" />
      </div>
   </div>

   </div>
   <div class="modal-footer">
   <button type="button"  data-bs-
dismiss="modal">
      Close
   </button>
   <button type="submit" name="edit_data"
class="btn btn-warning">Save</button>
   </div>
</form>
</div></div></div>

<!-- Modal Hapus -->
<div  id="hapus<?= $id_pesanan ?>" data-
bs-backdrop="static" tabindex="-1">
<div >
<div >
<form action="riwayat-pesanan-post.php"
method="post">
   <div >
   <h5 class="modal-title"
id="tambahDataTitle">Hapus Data</h5>
   <button
      type="button"
      class="btn-close"
      data-bs-dismiss="modal"
      aria-label="Close"></button>
   </div>
   <div class="modal-body">
   <input type="text" id="id_pesanan"
name="id_pesanan" class="form-control"
placeholder="Masukkan id_pesanan"
value="<?= $id_pesanan ?>" hidden />
   <div class="row">
      <div class="col mb-3">
      <label for="judul" class="form-
label">Hapus Data ini?</label>
      <label for="">Klik delete untuk
menghapus</label>
      </div>
   </div>
   </div>
   <div >
   <button type="button" class="btn btn-
outline-secondary" data-bs-
dismiss="modal">
      Close
   </button>
   <button type="submit"
name="hapus_data" class="btn btn-
danger">Delete</button>
   </div>
</form>
</div></div></div>

<?php
$no++;
}
?>
</tbody>
</table>
</div></div></div>
<script>
function onClickSelected() {
   var select =
document.getElementById("id_pelatihan");
```

```
    var selectedOption =
select.options[select.selectedIndex];

    var pelatihan =
selectedOption.getAttribute("data-
pelatihan");
    var jenis_pelatihan =
selectedOption.getAttribute("data-jenis");
    var deskripsi =
selectedOption.getAttribute("data-
deskripsi");
    var harga =
selectedOption.getAttribute("data-harga");

    // Masukkan ke input

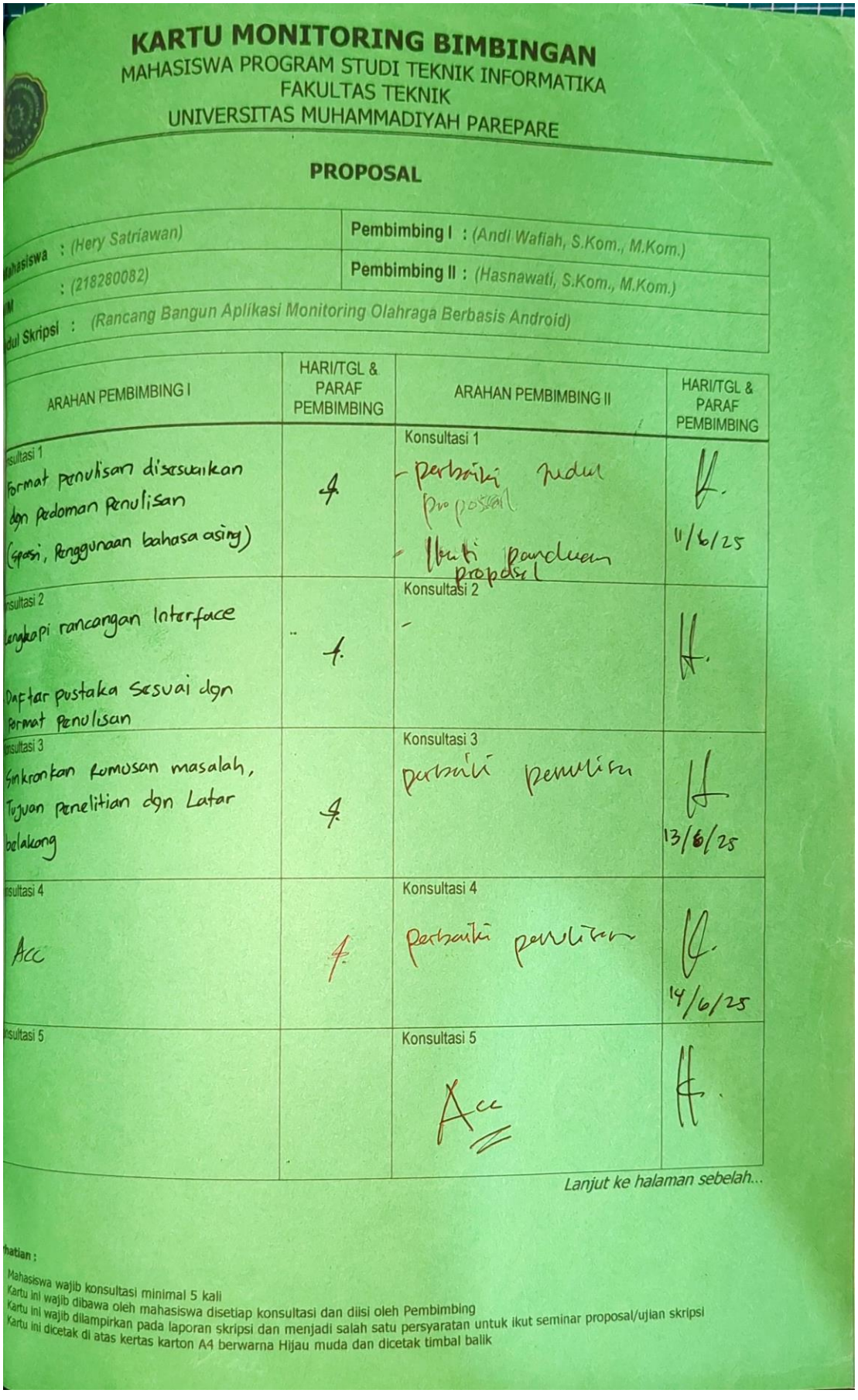
document.getElementById("pelatihan").valu
e = pelatihan;

document.getElementById("jenis_pelatihan"
).value = jenis_pelatihan;

document.getElementById("deskripsi").inne
rText = deskripsi;
    document.getElementById("harga").value
= harga;

    var hari =
selectedOption.getAttribute("data-hari");
    var hariSelect =
document.getElementById("hari_khusus");
    var hariValid = ["", "Senin", "Selasa",
"Rabu", "Kamis", "Jumat", "Sabtu",
"Minggu"];
    for (var i = 0; i < hariValid.length; i++) {
    if (hariValid[i] === hari) {
    hariSelect.selectedIndex = i;
    break;
}}}
</script>
```

Lampiran- 2 Kartu Monitoring Bimbingan Proposal



KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

ahasiswa : (Hery Satriawan)	Pembimbing I : (Andi Wafiah, S.Kom., M.Kom.)
M : (218280082)	Pembimbing II : (Hasnawati, S.Kom., M.Kom.)
dul Skripsi : (Rancang Bangun Aplikasi Monitoring Olahraga Berbasis Android)	

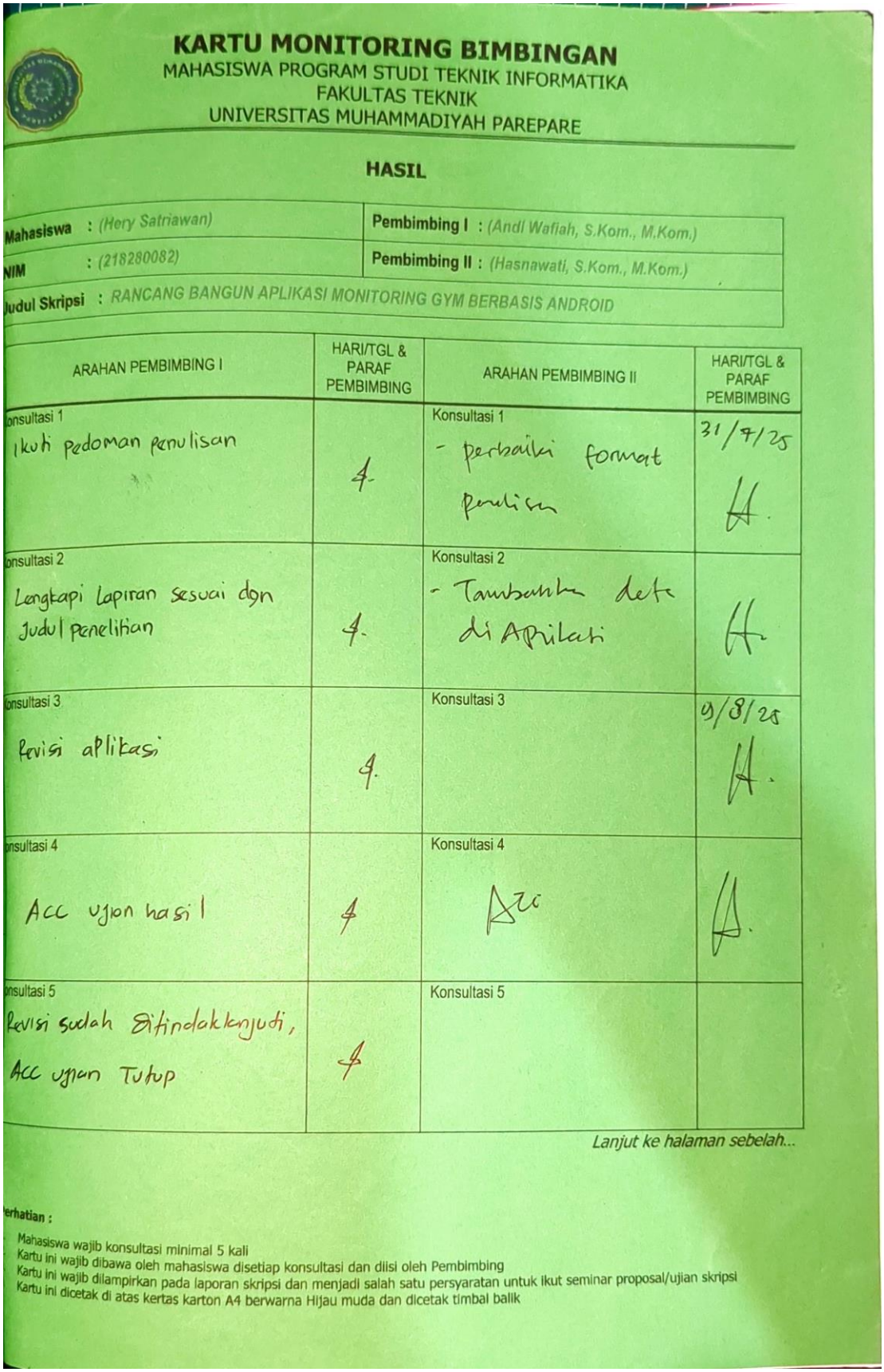
ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
nsultasi 1 Format penulisan disesuaikan dgn Pedoman Penulisan (Spasi, Penggunaan bahasa asing)	4	Konsultasi 1 - perbaiki judul proposal - lihat pandiran proposal	11/6/25
nsultasi 2 Lengkapi rancangan Interface Daftar pustaka sesuai dgn format penulisan	4	Konsultasi 2 -	
nsultasi 3 Sinkronkan rumusan masalah, Tujuan penelitian dgn Latar belakang	4	Konsultasi 3 perbaiki penulisan	13/6/25
nsultasi 4 Acc	4	Konsultasi 4 perbaiki penulisan	14/6/25
nsultasi 5		Konsultasi 5 Acc	

Lanjut ke halaman sebelah...

hatian :

Mahasiswa wajib konsultasi minimal 5 kali
Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

Lampiran- 3 Kartu Monitoring Bimbingan Skripsi



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : (Hery Satriawan)	**Pembimbing I** : (Andi Wafiah, S.Kom., M.Kom.)
NIM : (218280082)	**Pembimbing II** : (Hasnawati, S.Kom., M.Kom.)
Judul Skripsi : RANCANG BANGUN APLIKASI MONITORING GYM BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Ikuti pedoman penulisan		Konsultasi 1 - perbaiki format penulisan	31/7/25
Konsultasi 2 Lengkapi lapiran sesuai dgn judul penelitian		Konsultasi 2 - Tambahkan data di Aplikasi	
Konsultasi 3 Revisi aplikasi		Konsultasi 3	9/8/25
Konsultasi 4 Acc ujian hasil		Konsultasi 4	
Konsultasi 5 Revisi sudah ditindaklanjuti, Acc ujian Tutup		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

- Mahasiswa wajib konsultasi minimal 5 kali
- Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
- Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
- Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik