

DAFTAR PUSTAKA

- Abimanyu, A. (2025) "Development of an Information System for Blood Bag Screening As an Important Step in Blood Transfusion Handling in Indonesia," *Eastern-European Journal of Enterprise Technologies*, 1(9(133)), pp. 90–104. Available at: <https://doi.org/10.15587/1729-4061.2025.317259>.
- Ariyanti, D. (2019) "Perancangan dan Implementasi Aplikasi Pendaftaran Online PORKOTA Berbasis Client Server Menggunakan Web dan Android," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, 3(1), p. 35. Available at: <https://doi.org/10.30645/j-sakti.v3i1.94>.
- AryaRafa, D., Dyar Wahyuni, E. and Anjani Arifiyanti, A. (2024) "Rancang Bangun Aplikasi Donor Darah Darurat Donora Berbasis Android Dengan Konsep Gamifikasi Menggunakan Kotlin," *Jurnal Informatika dan Teknik Elektro Terapan*, 12(3), pp. 3009–3020. Available at: <https://doi.org/10.23960/jitet.v12i3.5025>.
- Aurellia, A., Nooriansyah, S. and Amrozi, Y. (2023) "PEMANFAATAN UML DALAM PERANCANGAN SISTEM INFORMASI PRODUK KREATIF DAUR ULANG SAMPAH BERBASIS WEB," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, 13(3), pp. 2303–0577.
- Ekarsih, D.P. et al. (2023) "Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) PERANCANGAN SISTEM INFORMASI PENERIMAAN PESERTA DIDIK BARU DI SDN 205 KOTA JAMBI BERBASIS WEB," *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)*, 3(2), pp. 684–694. Available at: <https://doi.org/https://doi.org/10.33998/jakakom.v3i2>.
- Ermawati Nopri (2020) "Sistem Informasi Unit Transfusi Darah Berbasis Android (Studi Kasus Rsud Kab.Kuansing)," *Jurnal Perencanaan,Sains,Teknologi, dan Komputer*, 3(2622–5980), pp. 1–9.
- Indriyanto, J., Huda, M. and Qirom, Q. (2022) "Sistem Informasi Berbasis Android dan SMS Gateway untuk Meningkatkan Pendor PMI Kota Tegal," *Syntax Literate ; Jurnal Ilmiah Indonesia*, 7(1), p. 300. Available at: <https://doi.org/10.36418/syntax-literate.v7i1.1872>.
- Irawan, D. et al. (2024) "Prototype Aplikasi Registrasi Donor Darah Palang Merah Indonesia(Pmi) Kota Bengkulu Menggunakan Metode Progressive Web App," *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(5), pp. 10124–10129. Available at: <https://doi.org/10.36040/jati.v8i5.10922>.
- Khairunnisa, A. (2022) "Perancangan Aplikasi Informasi Donor Darah

- Menggunakan Metode Waterfall,” *Jurnal SANTI (Sistem Informasi dan Teknologi Informasi)*, 2(2), pp. 98–112.
- Lassmayda, S. and Lumba, E. (2022) “Pengembangan Aplikasi Kegiatan Donor Darah Berbasis Android,” *Jurnal Mahasiswa Institut Teknologi dan Bisnis Kalbis*, 8(1), pp. 415–425. Available at: <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/2082>.
- Putri, D.U., Fachruddin and Jasmir (2025) “Implementasi Sistem Informasi Pendaftaran Online Berbasis Web Pada SMP Sinar Bijaksana Guang Ming,” *Jurnal Informatika Dan Rekayasa Komputer(JAKAKOM)*, 5(1), pp. 1384–1395. Available at: <https://doi.org/10.33998/jakakom.2025.5.1.2226>.
- Supono, S. and Armianti, S. (2021) “Aplikasi Web Service Aptimas Dengan Arsitektur Representational State Transfer (REST) API,” *Competitive*, 16(2), pp. 87–94. Available at: <https://doi.org/10.36618/competitive.v16i2.1572>.
- Syahputra, R.W. and Febriani, An. (2020) “SISTEM INFORMASI DONOR DARAH BERBASIS ANDROID PADA UNIT TRANSFUSI DARAH PALANG MERAH INDONESIA (UTD PMI) KOTA PEKANBARU,” *Jurnal Ilmu Komputer*, 9(1), pp. 11–19. Available at: <http://jik.hip.ac.id>.
- Tukan, Y.S., Deta, B. and Weking, A.N. (2025) “Perancangan Aplikasi Pencarian Pendorong Darah dengan Menggunakan Algoritma Dijkstra Berbasis Website,” *RIGGS: Journal of Artificial Intelligence and Digital Business*, 4(2), pp. 7700–7707. Available at: <https://doi.org/10.31004/riggs.v4i2.1903>.
- Wicaksono, A.S.R. and Priyambadha, B. (2025) “Implementasi Sistem Manajemen Basis Data Berbasis Web untuk Meningkatkan Efektifitas Kinerja Diskoperindag (Studi Kasus di Diskoperindag Kota Malang),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 9(11), pp. 1–7. Available at: <http://j-ptiik.ub.ac.id>.
- Wijayanti, E., Meimaharani, R. and Hidayati, H. (2023) “Desain Aplikasi Pendaftaran Donor Darah Korp Sukarela (KSR) Universitas Muria Kudus Berbasis Android,” *Jurnal Device*, 13(1), pp. 21–25. Available at: <https://doi.org/10.32699/device.v13i1.4097>.
- Yurindra, Y. et al. (2020) “Aplikasi Client Server Berbasis Android pada Barbershop The Barbega Menggunakan Model Multi Channel - Single Phase,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 9(1), pp. 138–143. Available at: <https://doi.org/10.32736/sisfokom.v9i1.837>.

LAMPIRAN

```

package com.example.ui.screens

import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Email
import androidx.compose.material.icons.filled.Lock
import androidx.compose.material.icons.filled.Visibility
import androidx.compose.material.icons.filled.VisibilityOff
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.testTag
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.R
import com.example.ui.theme.PmiRedPrimary
import com.example.viewmodel.PmiViewModel

@Composable
fun LoginScreen(
    viewModel: PmiViewModel,
    onLoginSuccess: () -> Unit,
    onNavigateToRegister: () -> Unit
) {
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var passwordVisible by remember { mutableStateOf(false) }

    var showChangeDialog by remember { mutableStateOf(false) }
    var accountIdentifier by remember { mutableStateOf("") }
    var newEmailInput by remember { mutableStateOf("") }
    var newPasswordInput by remember { mutableStateOf("") }
    var changeDialogMessage by remember { mutableStateOf<String?>(null) }
    var isSuccessMessage by remember { mutableStateOf(false) }

```

```

val authError by viewModel.authError.collectAsState()
val authSuccess by viewModel.authSuccess.collectAsState()

LaunchedEffect(authSuccess) {
    if (authSuccess) {
        onLoginSuccess()
    }
}

Box(
    modifier = Modifier
        .fillMaxSize()
        .background(MaterialTheme.colorScheme.background)
        .testTag("login_screen")
) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .verticalScroll(rememberScrollState())
            .padding(24.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Spacer(modifier = Modifier.height(40.dp))

        // Logo Box
        Surface(
            modifier = Modifier
                .size(90.dp)
                .clip(CircleShape),
            color = PmiRedPrimary.copy(alpha = 0.1f)
        ) {
            Image(
                painter = painterResource(id =
                    R.drawable.img_app_icon_1785945205753),
                contentDescription = "PMI Logo",
                modifier = Modifier
                    .fillMaxSize()
                    .padding(12.dp)
            )
        }

        Spacer(modifier = Modifier.height(16.dp))

        Text(
            text = "PMI KOTA BENGKULU",
            style = MaterialTheme.typography.titleLarge.copy(
                fontWeight = FontWeight.Bold,
                color = PmiRedPrimary
            )
        )

        Text(
            text = "Masuk ke Akun Pendorong / Petugas",

```

```

        style = MaterialTheme.typography.bodyMedium.copy(
            color = MaterialTheme.colorScheme.onSurfaceVariant
        ),
        modifier = Modifier.padding(top = 4.dp)
    )

    Spacer(modifier = Modifier.height(32.dp))

    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(20.dp),
        colors = CardDefaults.cardColors(containerColor =
MaterialTheme.colorScheme.surface),
        elevation = CardDefaults.cardElevation(defaultElevation = 2.dp)
    ) {
        Column(
            modifier = Modifier
                .fillMaxWidth()
                .padding(20.dp)
        ) {
            Text(
                text = "Login",
                style =
MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold)
            )

            Spacer(modifier = Modifier.height(16.dp))

            OutlinedTextField(
                value = email,
                onValueChange = { email = it },
                label = { Text("Email / Username") },
                leadingIcon = { Icon(Icons.Default.Email,
contentDescription = null) },
                modifier = Modifier
                    .fillMaxWidth()
                    .testTag("login_email_input"),
                singleLine = true,
                keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Email)
            )

            Spacer(modifier = Modifier.height(12.dp))

            OutlinedTextField(
                value = password,
                onValueChange = { password = it },
                label = { Text("Password") },
                leadingIcon = { Icon(Icons.Default.Lock,
contentDescription = null) },
                trailingIcon = {
                    IconButton(onClick = { passwordVisible =
!passwordVisible }) {

```

```

        Icon(
            imageVector = if (passwordVisible)
Icons.Default.Visibility else Icons.Default.VisibilityOff,
            contentDescription = "Toggle Password"
        )
    },
    visualTransformation = if (passwordVisible)
VisualTransformation.None else PasswordVisualTransformation(),
    modifier = Modifier
        .fillMaxWidth()
        .testTag("login_password_input"),
    singleLine = true,
    keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Password)
)

    Row(
        modifier = Modifier.fillMaxWidth(),
        horizontalArrangement = Arrangement.End
    ) {
        TextButton(
            onClick = {
                accountIdentifier = email
                newEmailInput = ""
                newPasswordInput = ""
                changeDialogMessage = null
                showChangeDialog = true
            },
            modifier =
Modifier.testTag("change_credentials_button")
        ) {
            Text(
                text = "Ubah Username / Password?",
                style =
MaterialTheme.typography.labelSmall.copy(color = PmiRedPrimary)
            )
        }
    }

    authError?.let { err ->
        Spacer(modifier = Modifier.height(4.dp))
        Text(
            text = err,
            style =
MaterialTheme.typography.bodySmall.copy(color = MaterialTheme.colorScheme.error)
        )
    }

    Spacer(modifier = Modifier.height(16.dp))

    Button(
        onClick = { viewModel.login(email, password) },

```

```

        modifier = Modifier
            .fillMaxWidth()
            .height(50.dp)
            .testTag("login_button"),
        shape = RoundedCornerShape(12.dp),
        colors = ButtonDefaults.buttonColors(containerColor =
PmiRedPrimary)
    ) {
        Text(
            text = "MASUK",
            style = MaterialTheme.typography.labelLarge.copy(
                fontWeight = FontWeight.Bold,
                color = Color.White
            )
        )
    }
}

Spacer(modifier = Modifier.height(24.dp))

// Quick Demo Login Cards
Text(
    text = "Atau Masuk Cepat (Demo Mode):",
    style = MaterialTheme.typography.labelMedium.copy(
        fontWeight = FontWeight.SemiBold,
        color = MaterialTheme.colorScheme.onSurfaceVariant
    )
)

Spacer(modifier = Modifier.height(10.dp))

Row(
    modifier = Modifier.fillMaxWidth(),
    horizontalArrangement = Arrangement.spacedBy(12.dp)
) {
    OutlinedButton(
        onClick = {
            email = "bibalhs011098@gmail.com"
            password = "donor"
            viewModel.loginAsDemoDonor()
        },
        modifier = Modifier
            .weight(1f)
            .testTag("demo_donor_button"),
        shape = RoundedCornerShape(12.dp)
    ) {
        Text("Pendonor Demo", fontSize = 13.sp)
    }

    OutlinedButton(
        onClick = {
            email = "admin@pmibengkulu.or.id"

```

```

        password = "admin"
        viewModel.loginAsAdmin()
    },
    modifier = Modifier
        .weight(1f)
        .testTag("demo_admin_button"),
    shape = RoundedCornerShape(12.dp)
) {
    Text("Admin Demo", fontSize = 13.sp)
}

Spacer(modifier = Modifier.height(8.dp))

Surface(
    color = MaterialTheme.colorScheme.surfaceVariant.copy(alpha =
0.5f),
    shape = RoundedCornerShape(10.dp),
    modifier = Modifier.fillMaxWidth()
) {
    Column(
        modifier = Modifier.padding(12.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Akun Pendonor: bibalhs011098@gmail.com | Pass:
donor",
            style =
MaterialTheme.typography.labelSmall.copy(fontSize = 11.sp, color =
MaterialTheme.colorScheme.onSurfaceVariant)
        )
        Text(
            text = "Akun Admin: admin@pmibengkulu.or.id | Pass:
admin",
            style =
MaterialTheme.typography.labelSmall.copy(fontSize = 11.sp, color =
MaterialTheme.colorScheme.onSurfaceVariant)
        )
    }
}

Spacer(modifier = Modifier.height(32.dp))

Row(
    verticalAlignment = Alignment.CenterVertically
) {
    Text(
        text = "Belum memiliki akun pendonor? ",
        style = MaterialTheme.typography.bodyMedium.copy(color =
MaterialTheme.colorScheme.onSurfaceVariant)
    )
    TextButton(
        onClick = onNavigateToRegister,

```

```

        modifier = Modifier.testTag("register_navigation_button")
    ) {
        Text(
            text = "Daftar Akun",
            style = MaterialTheme.typography.bodyMedium.copy(
                fontWeight = FontWeight.Bold,
                color = PmiRedPrimary
            )
        )
    }
}

if (showChangeDialog) {
    AlertDialog(
        onDismissRequest = { showChangeDialog = false },
        title = {
            Text(
                text = "Ubah Username / Password",
                style =
MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold)
            )
        },
        text = {
            Column(
                modifier = Modifier.fillMaxWidth(),
                verticalArrangement = Arrangement.spacedBy(10.dp)
            ) {
                Text(
                    text = "Masukkan Email/NIK/No HP akun Anda, lalu isi
username (email) baru dan/atau password baru.",
                    style =
MaterialTheme.typography.bodySmall.copy(color =
MaterialTheme.colorScheme.onSurfaceVariant)
                )

                OutlinedTextField(
                    value = accountIdIdentifier,
                    onValueChange = { accountIdIdentifier = it },
                    label = { Text("Email / NIK / No. HP Akun") },
                    singleLine = true,
                    modifier =
Modifier.fillMaxWidth().testTag("change_account_identifier_input")
                )

                OutlinedTextField(
                    value = newEmailInput,
                    onValueChange = { newEmailInput = it },
                    label = { Text("Username / Email Baru (Opsional)") }
                ),

                singleLine = true,
                modifier =
Modifier.fillMaxWidth().testTag("change_new_email_input")
            }
        )
    }
}

```

```

    )

    OutlinedTextField(
        value = newPasswordInput,
        onChange = { newPasswordInput = it },
        label = { Text("Password Baru (Opsional)") },
        singleLine = true,
        visualTransformation =
PasswordVisualTransformation(),
        modifier =
Modifier.fillMaxWidth().testTag("change_new_password_input")
    )

    changeDialogMessage?.let { msg ->
        Text(
            text = msg,
            style = MaterialTheme.typography.bodySmall.copy(
                color = if (isSuccessMessage)
Color(0xFF2E7D32) else MaterialTheme.colorScheme.error,
                fontWeight = FontWeight.Medium
            )
        )
    }
},
confirmButton = {
    Button(
        onClick = {
            viewModel.changeCredentials(
                accountIdIdentifier = accountIdIdentifier,
                newEmail = newEmailInput,
                newPassword = newPasswordInput
            ) { success, msg ->
                isSuccessMessage = success
                changeDialogMessage = msg
                if (success) {
                    if (newEmailInput.isNotBlank()) email =
newEmailInput.trim()

                    if (newPasswordInput.isNotBlank()) password
= newPasswordInput.trim()
                }
            }
        },
        colors = ButtonDefaults.buttonColors(containerColor =
PmiRedPrimary)
    ) {
        Text("Simpan Perubahan", color = Color.White)
    }
},
dismissButton = {
    OutlinedButton(onClick = { showChangeDialog = false }) {
        Text("Tutup")
    }
}

```

