

DAFTAR PUSTAKA

- [1] R. Hidayatullah, V. Asmawati, dan D. Wahyudi, “Aplikasi Manajemen Surat Masuk dan Keluar Surat Berbasis Mobile di Perum Bulog Subdivre Dumai,” *J. Inform. Sunan Kalijaga (JISKa)*, vol. 5, no. 2, pp. 66–80, 2020.
- [2] D. Nurdiana, “Implementasi Aplikasi Pengelolaan Surat Masuk dan Keluar Berbasis Web di Prodi Sistem Informasi,” *J. Teknol. dan Sist. Informasi (JURTEKSI)*, vol. 6, no. 2, pp. 135–144, 2020, doi: 10.33330/jurteksi.v6i2.437.
- [3] PCNU Cilacap, “PAC IPNU-IPPNU Kesugihan Gelar Sekolah Instruktur Pelajar,” *PCNU Cilacap*, 6 Juni 2023. [Online]. Tersedia: <https://pcnucilacap.com/pac-ipnu-ippnu-kesugihan-gelar-sekolah-instruktur-pelajar/>. [Diakses: 17-Apr-2025].
- [4] A. Wahyudin, A. Anisyah, dan D. Ahmaddifa, “Pengembangan Sistem Notifikasi Real-Time untuk Aplikasi Manajemen Persuratan Multiplatform menggunakan Firebase Cloud Messaging dan Application Programming Interface,” *J. Tek. Inform. Unika ST. Thomas (JTIUST)*, vol. 9, no. 2, pp. 284–300, 2024.
- [5] C. Yuliansyah, L. Andiani, D. Apriana, dan Dona, “Rancang Bangun Aplikasi Manajemen Surat Berbasis Android pada Kantor Desa Wisata Batu Ampar Kecamatan Merigi,” *Jurnal Komputer, Informasi dan Teknologi*, vol. 5, no. 1, pp. 1–11, 2025. [Online]. Tersedia: <https://penerbitadm.pubmedia.id/index.php/KOMITEK/article/view/2277>
- [6] Saifulloh, Hoiriyah, dan Bakir, “Sistem Informasi Manajemen Administrasi Ikatan Pelajar Nahdlatul Ulama (IPNU) Ikatan Pelajar Putri Nahdlatul Ulama (IPPNU) Berbasis Website,” *Jurnal Sistem Komputer dan Kecerdasan Buatan*, vol. 7, no. 3, pp. 268–275, 2024.
- [7] Yuliansyah, L. Andiani, D. Apriana, dan Dona, “Rancang Bangun Aplikasi Manajemen Surat Berbasis Android pada Kantor Desa Wisata Batu Ampar Kecamatan Merigi,” *Jurnal Komputer, Informasi dan Teknologi (KOMITEK)*, vol. 5, no. 1, pp. 1–11, 2025, doi: 10.53697/jkomitek.v5i1.2277.

- [8] E. Z. Arifin, *Komunikasi dalam Organisasi*. Jakarta: Rineka Cipta, 2011.
- [9] D. G. A. Candra, B. P. Putra, I. Meiditra, dan N. Afrianto, “Penerapan Aplikasi Manajemen Surat Berbasis Web Menggunakan Metode Waterfall,” *J. Tek. Inform. Unika ST. Thomas (JTIUST)*, vol. 8, no. 2, pp. 229–237, 2023.
- [10] H. Hasanah, “Teknik-Teknik Observasi (Sebuah Alternatif Metode Pengumpulan Data Kualitatif Ilmu-ilmu Sosial),” *J. At-Taqaddum*, vol. 8, no. 1, pp. 21–42, 2017, doi: 10.21580/at.v8i1.1163.
- [11] D. Buyens, *Mobile Application Development: Concept and Implementation*. Jakarta: Tech Publications, 2001. [Online]. Available: elib.unikom.ac.id.
- [12] A. Nikhlis dan B. Kurniawan, *Pengembangan Aplikasi Berbasis Android: Konsep dan Implementasi*. Jakarta: Tech Press, 2022.
- [13] M. N. Baharudin dan I. K. D. Nuryana, “Implementasi Algoritma Decision Tree untuk Klasifikasi Surat pada Aplikasi Mobile E-Surat Dinas Komunikasi dan Informatika Kota Kediri Berbasis Android,” *J. Inform. dan Comp. Sci. (JINACS)*, vol. 4, no. 1, pp. 76–85, 2022.
- [14] M. F. Santoso, “Implementasi Konsep dan Teknik UI/UX Dalam Rancang Bangun Layout Web dengan Figma,” *J. Infortech*, vol. 4, no. 2, pp. 156–163, 2022. [Online]. Tersedia: <http://ejournal.bsi.ac.id/ejurnal/index.php/infortech>.
- [15] K. N. Meiah, A. Arista, dan R. N. S. Amriza, “Implementasi Firebase Realtime Database pada Aplikasi FeedbackMe sebagai Penghubung Guru dan Orang Tua,” *Rekayasa Sist. dan Teknol. Informasi (J. RESTI)*, vol. 5, no. 2, pp. 327–334, 2021, doi: 10.29207/resti.v5i2.2909.
- [16] H. Jayusman dan F. Mahardika, “Perancangan Sistem Arsip Surat Masuk dan Surat Keluar Desa Wanutengah Kab. Temanggung Berbasis Mobile,” *BLEND SAINS Jurnal Teknik*, vol. 3, no. 1, pp. 74–84, 2024. [Online]. Available: <https://doi.org/10.56211/blendsains.v3i1.566>.
- [17] R. R. Siematauw dan M. A. I. Pakereng, “Aplikasi Arsip Surat Mobile Menggunakan Framework Flutter (Studi Kasus: PT. BPD Maluku, Malut),” *J. Tek. Inform. dan Sist. Informasi (JATISI)*, vol. 10, no. 1, pp. 204–216, 2023.

- [18] H. Helmy, F. Rahmasari, A. Nursyahid, T. A. Setyawan, dan A. S. Nugroho, "Analisis Kinerja Aplikasi Pemantauan dan Pengendalian Smart Agriculture Berbasis Android," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 11, no. 1, pp. 1–10, 2022.
- [19] Saryani, Harfizar, and R. Arianto, "Rancangan Sistem Informasi E-Arsip Surat Masuk dan Surat Keluar (Dinas Pariwisata dan Kebudayaan Kota Tangerang)," *Technomedia Journal (TMJ)*, vol. 4, no. 1, pp. 69–83, Aug. 2019. [Online]. Available: <https://jurnal.raharja.ac.id/index.php/technomedia/article/view/1848>
- [20] D. Andrian, "Penerapan metode Waterfall dalam perancangan sistem informasi pengawasan proyek berbasis web," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 1, no. 2, pp. 85–93, 2021. [Online]. Available: <https://jim.teknokrat.ac.id/index.php/informatika/article/view/729>. [Accessed: May 30, 2025].
- [21] D. Purnomo, E. Nugroho, dan S. Wibowo, "Analisis dan Perancangan Sistem Aplikasi E-Government," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 1, hlm. 45–54, 2020. [Daring]. Tersedia: <https://doi.org/10.29313/jti.v6i1.7115>
- [22] A. Ahmad, "Perancangan Sistem Informasi Pengelolaan Data Akademik Berbasis Web," *Jurnal Widya*, vol. 2, no. 1, pp. 12–20, 2020. [Online]. Available: <https://jurnal.amikwidyaloka.ac.id/index.php/awl/article/download/170/124>
- [23] Google Developers, "Dart Programming Language," Dart.dev, 2023. [Online]. Available: <https://dart.dev>. [Accessed: Jun. 01, 2025].

LAMPIRAN - LAMPIRAN

Lampiran I Tabel Pengujian

Pengujian Menu Login

No	Skenario Uji	Input	Ekspektasi	Hasil Aktual	Status
1	Halaman Login tampil	Buka aplikasi	Form login dengan input email dan password muncul	Form login muncul	Sesuai
2	Login berhasil	Email: admi@gmail.com Password: admin123	Berhasil login dan diarahkan ke Dashboard	Pengguna masuk ke Dashboard	Sesuai
3	Login gagal	Email: salah@gmail.com Password: 123456	Muncul pesan error “Akun tidak ditemukan”	Muncul pesan error	Sesuai

Pengujian Menu Arsip Surat Masuk

No	Skenario Uji	Input	Ekspektasi	Hasil Aktual	Status
1	Tambah Surat Masuk	Nomor Surat, Tanggal, Perihal, File	Surat berhasil ditambahkan dan tampil di daftar arsip	Surat tampil di daftar	Sesuai
2	Edit Surat Masuk	Ubah perihal surat	Perihal berubah dan tersimpan di Firebase	Data berhasil diperbarui	Sesuai
3	Hapus Surat Masuk	Tekan tombol hapus	Surat terhapus dari daftar dan Firebase	Surat hilang dari tampilan	Sesuai

4	Cari Surat Masuk	Kata kunci: undangan	Surat dengan kata “undangan” muncul	Surat muncul sesuai kata kunci	Sesuai
---	------------------	----------------------	-------------------------------------	--------------------------------	--------

Pengujian Menu Arsip Surat Keluar

No	Skenario Uji	Input	Ekspektasi	Hasil Aktual	Status
1	Tambah Surat Keluar	Nomor Surat, Tanggal, Perihal, File	Surat keluar ditambahkan ke daftar	Surat tampil di daftar arsip keluar	Sesuai
2	Edit Surat Keluar	Ubah tujuan surat	Tujuan berubah dan tersimpan di Firebase	Data berhasil diperbarui	Sesuai
3	Hapus Surat Keluar	Tekan tombol hapus	Surat keluar hilang dari sistem	Surat hilang dari daftar	Sesuai

Pengujian Laporan PDF Surat

No	Skenario Uji	Input	Ekspektasi	Hasil Aktual	Status
1	Laporan Arsip Masuk	Pilih bulan: Maret 2025	File PDF surat masuk berhasil dibuat	File PDF tersimpan dan bisa dibuka	Sesuai
2	Laporan Arsip Keluar	Pilih bulan: Maret 2025	File PDF surat keluar berhasil dibuat	File PDF bisa dilihat dan dicetak	Sesuai

Pengujian Fitur Logout

No	Skenario Uji	Input	Ekspektasi	Hasil Aktual	Status
1	Logout berhasil	Tekan tombol logout di dashboard	Pengguna diaahkan kembali ke halaman login	Halaman login muncul kembali	Sesuai

Lampiran II Kode program

Kode Login

```
import 'package:arsip_surat/services/auth_services.dart';
import 'package:flutter/material.dart';

class SignInPage extends StatefulWidget {
  const SignInPage({super.key});

  @override
  State<SignInPage> createState() => _SignInPageState();
}

class _SignInPageState extends State<SignInPage> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final AuthService _authService = AuthService();

  bool _obscurePassword = true;

  Future<void> _handleSignInWithEmail() async {
    final email = _emailController.text.trim();
    final password = _passwordController.text.trim();

    if (email.isEmpty || password.isEmpty) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Please enter email and password')),
      );
      return;
    }

    try {
      final user =
        await _authService.signInWithEmailAndPassword(email, password);
      if (user != null && mounted) {
        final role = await _authService.getUserRole(user.uid);

        if (role == 'admin') {
          Navigator.pushReplacementNamed(context, '/admin');
        } else if (role == 'ketua') {
          Navigator.pushReplacementNamed(context, '/ketua');
        }
      }
    }
  }
}
```

```

    } else if (role == 'user') {
      Navigator.pushReplacementNamed(context, '/user');
    } else {
      // Role tidak dikenali
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('Akun tidak memiliki peran yang dikenali')),
        );
    }
  }
} catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Login gagal: ${e.toString()}')),
  );
}
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: const Color(0xFFFF5F5F5),
    body: SafeArea(
      child: Stack(
        children: [
          // Background Wave Shape (simulasi)
          Container(
            decoration: const BoxDecoration(
              gradient: LinearGradient(
                colors: [
                  Color(0xFFFFFE08A),
                  Color(0xFFFF7F5F0),
                ],
                begin: Alignment.topLeft,
                end: Alignment.bottomRight,
              ),
            ),
          ),
          // Content
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 24),
            child: SingleChildScrollView(
              child: Center(

```

```

child: Padding(
  padding: const EdgeInsets.symmetric(vertical: 60),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      const SizedBox(height: 30),
      // Logo
      const Text(
        "Arsip Surat",
        style: TextStyle(
          fontSize: 32,
          fontWeight: FontWeight.bold,
          color: Colors.black,
        ),
      ),
      const SizedBox(height: 50),
      const Padding(
        padding: EdgeInsets.all(8.0),
        child: Text(
          "Login",
          style: TextStyle(
            fontSize: 32,
            fontWeight: FontWeight.bold,
            color: Colors.black,
          ),
        ),
      ),
      const SizedBox(height: 40),

      // Email Field
      TextFormField(
        controller: _emailController,
        keyboardType: TextInputType.emailAddress,
        decoration: InputDecoration(
          prefixIcon:
            const Icon(Icons.email, color: Colors.amber),
          hintText: 'Email',
          hintStyle: const TextStyle(color: Colors.amber),
          enabledBorder: const UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.black),
          ),
        ),
      ),
    ],
  ),
),

```

```

        focusedBorder: const UnderlineInputBorder(
          borderSide: BorderSide(color: Colors.black),
        ),
      ),
    ),
  ),
  const SizedBox(height: 24),

  // Password Field
  TextFormField(
    controller: _passwordController,
    obscureText: _obscurePassword,
    decoration: InputDecoration(
      prefixIcon:
        const Icon(Icons.lock, color: Colors.amber),
      hintText: 'Password',
      hintStyle: const TextStyle(color: Colors.amber),
      enabledBorder: const UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.black),
      ),
      focusedBorder: const UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.black),
      ),
      suffixIcon: IconButton(
        icon: Icon(
          _obscurePassword
            ? Icons.visibility_off
            : Icons.visibility,
          color: Colors.grey,
        ),
        onPressed: () {
          setState(() {
            _obscurePassword = !_obscurePassword;
          });
        },
      ),
    ),
  ),
  const SizedBox(height: 8),

  // Forgot Password
  Align(
    alignment: Alignment.centerRight,

```

```

        child: TextButton(
          onPressed: () {},
          child: const Text(
            "Forgot password?",
            style: TextStyle(color: Colors.black),
          ),
        ),
      ),
    ),
    const SizedBox(height: 20),

    // Login Button
    SizedBox(
      width: double.infinity,
      height: 50,
      child: ElevatedButton(
        onPressed: _handleSignInWithEmail,
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.amber,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
        ),
        child: const Text(
          "LOGIN",
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
            color: Colors.black,
          ),
        ),
      ),
    ),
    ),
    ),
    const SizedBox(height: 24),
  ],
),
),
),
),
),
],
),
),

```

Kode Dashborad

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

class Dashboard extends StatefulWidget {
  const Dashboard({super.key});

  @override
  State<Dashboard> createState() => _DashboardState();
}

class _DashboardState extends State<Dashboard> {
  String? role;
  bool isLoading = true;

  @override
  void initState() {
    super.initState();
    _loadRole();
  }

  Future<void> _loadRole() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user != null) {
      final snapshot = await FirebaseFirestore.instance
        .collection('users')
        .doc(user.uid)
        .get();
      setState(() {
        role = snapshot.data()?['role'] ?? 'user';
        isLoading = false;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    final menuItems = [
      {
        'label': 'Arsip Masuk',
```

```

      'icon': Icons.download,
      'route': '/arsipmasuk',
      'show': role == 'admin' || role == 'user',
    },
    {
      'label': 'Arsip Keluar',
      'icon': Icons.upload,
      'route': '/arsipkeluar',
      'show': role == 'admin' || role == 'user',
    },
    {
      'label': 'Export Arsip Masuk',
      'icon': Icons.picture_as_pdf,
      'route': '/exportmasuk',
      'show': role == 'admin' || role == 'ketua',
    },
    {
      'label': 'Export Arsip Keluar',
      'icon': Icons.picture_as_pdf_outlined,
      'route': '/exportkeluar',
      'show': role == 'admin' || role == 'ketua',
    },
    {
      'label': 'Tentang',
      'icon': Icons.info,
      'route': '/tentang',
      'show': true,
    },
    {
      'label': 'Log Out',
      'icon': Icons.logout,
      'route': '/sign_in',
      'show': true,
    },
  ],

  if (isLoading) {
    return const Scaffold(
      body: Center(child: CircularProgressIndicator()),
    );
  }

```

```

final visibleItems =
  menuItems.where((item) => item['show'] == true).toList();

return Scaffold(
  appBar: AppBar(
    title: const Text(
      'Dashboard',
      style: TextStyle(fontWeight: FontWeight.bold),
    ),
    backgroundColor: Colors.amber,
  ),
  body: Padding(
    padding: const EdgeInsets.all(14),
    child: Center(
      child: ConstrainedBox(
        constraints: const BoxConstraints(maxWidth: 600, maxHeight: 500),
        child: GridView.builder(
          shrinkWrap: true,
          itemCount: visibleItems.length,
          gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 24,
            mainAxisSpacing: 20,
            childAspectRatio: 1.3,
          ),
          itemBuilder: (context, index) {
            final item = visibleItems[index];
            return GestureDetector(
              onTap: () async {
                if (item['label'] == 'Log Out') {
                  final shouldLogout = await showDialog<bool>(
                    context: context,
                    builder: (_) => AlertDialog(
                      title: const Text('Konfirmasi Logout'),
                      content:
                        const Text('Apakah kamu yakin ingin keluar?'),
                      actions: [
                        TextButton(
                          onPressed: () => Navigator.pop(context, false),
                          child: const Text('Batal'),
                        ),
                        TextButton(

```

```

        onPressed: () => Navigator.pop(context, true),
        child: const Text('Keluar'),
      ),
    ],
  ),
);

if (shouldLogout == true) {
  await FirebaseAuth.instance.signOut();
  if (!context.mounted) return;
  Navigator.pushNamedAndRemoveUntil(
    context,
    item['route'] as String,
    (route) => false,
  );
}
} else {
  Navigator.pushNamed(context, item['route'] as String);
}
},
child: Container(
  decoration: BoxDecoration(
    color: Colors.amber[600],
    borderRadius: BorderRadius.circular(16),
  ),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Icon(item['icon'] as IconData,
        size: 48, color: Colors.white),
      const SizedBox(height: 12),
      Text(
        item['label'] as String,
        textAlign: TextAlign.center,
        style: const TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.bold,
          color: Colors.black87,
        ),
      ),
    ],
  ),
),

```

Kode Arsip Masuk

```
import 'package:arsip_surat/pages/liat_arsip_masuk.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:file_picker/file_picker.dart';
import 'dart:io';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:url_launcher/url_launcher.dart';

class ArsipMasuk extends StatefulWidget {
  const ArsipMasuk({super.key});

  @override
  State<ArsipMasuk> createState() => _ArsipMasukState();
}

class _ArsipMasukState extends State<ArsipMasuk> {
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;
  final FirebaseStorage _storage = FirebaseStorage.instance;
  List<DocumentSnapshot> _suratMasuk = [];

  @override
  void initState() {
    super.initState();
    _loadSuratMasuk();
  }

  void _loadSuratMasuk() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user != null) {
      final snapshot = await FirebaseFirestore.instance
        .collection('users')
        .doc(user.uid)
        .collection('surat_masuk')
        .orderBy('created_at', descending: true)
        .get();

      print('Dokumen ditemukan: ${snapshot.docs.length}');
      for (var doc in snapshot.docs) {
        print('Surat Masuk: ${doc.data()}');
      }
    }
  }
}
```

```

    }

    setState(() {
      _suratMasuk = snapshot.docs;
    });
  }
}

void _addOrEditSuratMasuk({DocumentSnapshot? existingSurat}) async {
  final user = FirebaseAuth.instance.currentUser;
  if (user == null) return;

  final isEdit = existingSurat != null;
  final data = existingSurat?.data() as Map<String, dynamic>? ?? {};

  final pengirimController =
    TextEditingController(text: data['pengirim'] ?? "");
  final tanggalSuratController = TextEditingController(
    text: (data['tanggal_surat'] != null &&
      data['tanggal_surat'] is Timestamp)
      ? (data['tanggal_surat'] as Timestamp)
        .toDate()
        .toString()
        .split(' ')[0]
      : data['tanggal_surat'] ?? "");
  final tanggalSuratMasukController = TextEditingController(
    text: (data['tanggal_surat_masuk'] != null &&
      data['tanggal_surat_masuk'] is Timestamp)
      ? (data['tanggal_surat_masuk'] as Timestamp)
        .toDate()
        .toString()
        .split(' ')[0]
      : data['tanggal_surat_masuk'] ?? "");
  final perihalController =
    TextEditingController(text: data['perihal'] ?? "");
  final kategoriController =
    TextEditingController(text: data['lampiran'] ?? "");
  final statusController = TextEditingController(text: data['status'] ?? "");
  final noSuratController =
    TextEditingController(text: data['no_surat'] ?? "");

  PlatformFile? pickedFile;

```

UploadTask? uploadTask;

```
await showDialog(
  context: context,
  builder: (context) => AlertDialog(
    backgroundColor: Colors.amber[50],
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
    title: Text(isEdit ? 'Edit Surat Masuk' : 'Tambah Surat Masuk'),
    content: SingleChildScrollView(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        children: [
          TextField(
            controller: noSuratController,
            decoration: const InputDecoration(labelText: 'No Surat')),
          TextField(
            controller: pengirimController,
            decoration: const InputDecoration(labelText: 'Pengirim')),
          TextField(
            controller: tanggalSuratController,
            keyboardType: TextInputType.datetime,
            decoration: const InputDecoration(
              labelText: 'Tanggal Surat (yyyy-MM-dd)'),
          TextField(
            controller: tanggalSuratMasukController,
            keyboardType: TextInputType.datetime,
            decoration: const InputDecoration(
              labelText: 'Tanggal Surat Masuk (yyyy-MM-dd)'),
          TextField(
            controller: perihalController,
            decoration: const InputDecoration(labelText: 'Perihal')),
          TextField(
            controller: kategoriController,
            decoration: const InputDecoration(labelText: 'Lampiran')),
          TextField(
            controller: statusController,
            decoration: const InputDecoration(labelText: 'Isi Surat')),
          const SizedBox(height: 10),
          ElevatedButton.icon(
            onPressed: () async {
              FilePickerResult? result =
                await FilePicker.platform.pickFiles();
```

```

        if (result != null) {
            pickedFile = result.files.first;
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text('File dipilih: ${pickedFile!.name}')),
                );
        }
    },
    icon: const Icon(Icons.attach_file),
    label: const Text('Pilih File'),
),
],
),
),
actions: [
    TextButton(
        onPressed: () => Navigator.pop(context),
        child: const Text('Batal'),
    ),
    TextButton(
        onPressed: () async {
            if ([
                noSuratController,
                pengirimController,
                tanggalSuratController,
                tanggalSuratMasukController,
                perihalController,
                kategoriController,
                statusController
            ].any((c) => c.text.isEmpty)) {
                ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Semua field harus diisi')),
                );
                return;
            }

            DateTime? tanggalSurat;
            DateTime? tanggalSuratMasuk;

            try {
                tanggalSurat = DateTime.parse(tanggalSuratController.text);
                tanggalSuratMasuk =

```

```

        DateTime.parse(tanggalSuratMasukController.text);
    } catch (_) {
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(
                content: Text('Format tanggal tidak valid (yyyy-MM-dd)'),
            );
        return;
    }

    showDialog(
        context: context,
        barrierDismissible: false,
        builder: (context) =>
            const Center(child: CircularProgressIndicator()),
    );

    String? fileUrl = data['file_url'];
    if (pickedFile != null) {
        try {
            final ref = _storage.ref().child(
                'surat_masuk/${user.uid}_${DateTime.now().millisecondsSinceEpoch}_${pickedFile!.name}',
            );
            uploadTask = ref.putFile(File(pickedFile!.path!));
            final snapshot = await uploadTask!.whenComplete(() {});
            fileUrl = await snapshot.ref.getDownloadURL();
        } catch (e) {
            Navigator.pop(context); // Tutup loading
            ScaffoldMessenger.of(context).showSnackBar(
                const SnackBar(content: Text('Upload file gagal.')),
            );
            return;
        }
    }

    final docId = isEdit
        ? existingSurat!.id
        : 'M-${DateTime.now().millisecondsSinceEpoch}';

    final docRef = _firestore
        .collection('users')
        .doc(user.uid)

```

```

        .collection('surat_masuk')
        .doc(docId);

    try {
      await docRef.set({
        'id': docId,
        'no_surat': noSuratController.text,
        'pengirim': pengirimController.text,
        'tanggal_surat': Timestamp.fromDate(tanggalSurat),
        'tanggal_surat_masuk': Timestamp.fromDate(tanggalSuratMasuk),
        'perihal': perihalController.text,
        'lampiran': kategoriController.text,
        'status': statusController.text,
        'file_url': fileUrl,
        'created_at': isEdit
          ? (data['created_at'] ?? FieldValue.serverTimestamp())
          : FieldValue.serverTimestamp(),
        'created_by': user.email,
      });

      Navigator.pop(context); // Tutup loading
      Navigator.pop(context); // Tutup form
      _loadSuratMasuk(); // Refresh data
    } catch (e) {
      Navigator.pop(context); // Tutup loading
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Gagal menyimpan data.')),
      );
    }
  },
  child: const Text('Simpan'),
),
],
),
);
}

void _deleteSuratMasuk(DocumentSnapshot surat) async {
  await surat.reference.delete();
  _loadSuratMasuk();
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Surat masuk berhasil dihapus')));
}

```

```

}

void _showDetailDialog(DocumentSnapshot surat) {
  final data = surat.data() as Map<String, dynamic>;
  final tanggalSurat = data['tanggal_surat'] is Timestamp
    ? (data['tanggal_surat'] as Timestamp).toDate().toString().split(' ')[0]
    : data['tanggal_surat'].toString();
  final tanggalMasuk = data['tanggal_surat_masuk'] is Timestamp
    ? (data['tanggal_surat_masuk'] as Timestamp)
      .toDate()
      .toString()
      .split(' ')[0]
    : data['tanggal_surat_masuk'].toString();
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text('Detail Surat Masuk'),
      backgroundColor: Colors.amber[50],
      content: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text('No Surat: ${data['no_surat']}'),
            Text('Pengirim: ${data['pengirim']}'),
            Text('Tanggal Surat: $tanggalSurat'),
            Text('Tanggal Masuk: $tanggalMasuk'),
            Text('Perihal: ${data['perihal']}'),
            Text('Lampiran: ${data['lampiran']}'),
            Text('Isi Surat: ${data['status']}'),
            if (data['file_url'] != null)
              TextButton(
                onPressed: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) =>
                        PdfViewerPage(pdfUrl: data['file_url']),
                    ),
                  );
                },
              ),
            child: const Text('Lihat Lampiran'),
          ],
        ),
      ),
    ),
  );
}

```

```

    ],
  ),
),
actions: [
  TextButton(
    onPressed: () => Navigator.pop(context),
    child: const Text('Tutup')),
  TextButton(
    onPressed: () => _addOrEditSuratMasuk(existingSurat: surat),
    child: const Text('Edit')),
],
),
);
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      automaticallyImplyLeading: false,
      title: const Text('Arsip Surat Masuk',
        style: TextStyle(fontWeight: FontWeight.bold)),
      backgroundColor: Colors.amber,
      actions: [
        Padding(
          padding: const EdgeInsets.all(10.0),
          child: IconButton(
            icon: const Icon(Icons.search),
            onPressed: () {
              TextEditingController searchController =
                TextEditingController();
              showDialog(
                context: context,
                builder: (context) => AlertDialog(
                  title: const Text('Cari Surat Masuk'),
                  content: TextField(
                    controller: searchController,
                    decoration: const InputDecoration(
                      labelText: 'Cari berdasarkan Pengirim atau Perihal',
                    ),
                  ),
                  onChanged: (value) {
                    setState(() {

```

```

        if (value.isEmpty) {
            _loadSuratMasuk();
        } else {
            _suratMasuk = _suratMasuk.where((surat) {
                final data = surat.data() as Map<String, dynamic>;
                final pengirim =
                    data['pengirim']?.toLowerCase() ?? "";
                final perihal =
                    data['perihal']?.toLowerCase() ?? "";
                return pengirim.contains(value.toLowerCase()) ||
                    perihal.contains(value.toLowerCase());
            }).toList();
        }
    });
},
),
actions: [
    TextButton(
        onPressed: () {
            setState(() {
                final value = searchController.text.trim();
                if (value.isEmpty) {
                    _loadSuratMasuk();
                } else {
                    _suratMasuk = _suratMasuk.where((surat) {
                        final data =
                            surat.data() as Map<String, dynamic>;
                        final pengirim =
                            data['pengirim']?.toLowerCase() ?? "";
                        final perihal =
                            data['perihal']?.toLowerCase() ?? "";
                        return pengirim.contains(value.toLowerCase()) ||
                            perihal.contains(value.toLowerCase());
                    }).toList();
                }
            });
            Navigator.pop(context);
        },
        child: const Text('Cari'),
    ),
],
),

```

```

        );
      },
    ),
  ),
],
),
body: ListView.builder(
  itemCount: _suratMasuk.length,
  itemBuilder: (context, index) {
    final surat = _suratMasuk[index];
    final data = surat.data() as Map<String, dynamic>;
    final tanggal = data['tanggal_surat'] is Timestamp
      ? (data['tanggal_surat'] as Timestamp)
        .toDate()
        .toString()
        .split(' ')[0]
      : data['tanggal_surat'].toString();
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: Container(
        decoration: BoxDecoration(
          color: Colors.amber[50],
          borderRadius: BorderRadius.circular(15),
        ),
        child: ListTile(
          title: Text(data['perihal'] ?? data['pengirim'] ?? ""),
          subtitle:
            Text('Pengirim: ${data['pengirim']} | Tanggal: $tanggal'),
          onTap: () => _showDetailDialog(surat),
          trailing: Row(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              IconButton(
                icon: const Icon(Icons.edit),
                onPressed: () =>
                  _addOrEditSuratMasuk(existingSurat: surat)),
              IconButton(
                icon: const Icon(Icons.delete),
                onPressed: () => _deleteSuratMasuk(surat)),
            ],
          ),
        ),
      ),
    ),
  ),
),

```

```
    ),  
    );  
  },  
),  
floatingActionButton: FloatingActionButton(  
  backgroundColor: Colors.amber,  
  onPressed: () => _addOrEditSuratMasuk(),  
  child: const Icon(Icons.add),  
),  
);  
}  
}
```

Kode Arsip Keluar

```
import 'package:arsip_surat/pages/liat_arsip_masuk.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:file_picker/file_picker.dart';
import 'dart:io';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:url_launcher/url_launcher.dart';

class ArsipKeluar extends StatefulWidget {
  const ArsipKeluar({super.key});

  @override
  State<ArsipKeluar> createState() => _ArsipKeluarState();
}

class _ArsipKeluarState extends State<ArsipKeluar> {
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;
  final FirebaseStorage _storage = FirebaseStorage.instance;
  List<DocumentSnapshot> _suratKeluar = [];

  @override
  void initState() {
    super.initState();
    _loadSuratKeluar();
  }

  void _loadSuratKeluar() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user != null) {
      final snapshot = await FirebaseFirestore.instance
        .collection('users')
        .doc(user.uid)
        .collection('surat_keluar')
        .orderBy('created_at', descending: true)
        .get();

      print('Dokumen ditemukan: ${snapshot.docs.length}');
      for (var doc in snapshot.docs) {
        print('Surat Keluar: ${doc.data()}');
      }
    }
  }
}
```

```

    }

    setState(() {
      _suratKeluar = snapshot.docs;
    });
  }
}

void _addOrEditSuratKeluar({DocumentSnapshot? existingSurat}) async {
  final user = FirebaseAuth.instance.currentUser;
  if (user == null) return;

  final isEdit = existingSurat != null;
  final data = existingSurat?.data() as Map<String, dynamic>? ?? {};

  final tujuanController = TextEditingController(text: data['tujuan'] ?? "");
  final perihalController =
    TextEditingController(text: data['perihal'] ?? "");
  final kategoriController =
    TextEditingController(text: data['lampiran'] ?? "");
  final statusController = TextEditingController(text: data['status'] ?? "");
  final noSuratController =
    TextEditingController(text: data['no_surat'] ?? "");

  final tanggalSuratController = TextEditingController(
    text:
      (data['tanggal_surat'] != null && data['tanggal_surat'] is Timestamp)
        ? (data['tanggal_surat'] as Timestamp)
          .toDate()
          .toString()
          .split(' ')[0]
          : "",
  );

  PlatformFile? pickedFile;
  UploadTask? uploadTask;

  await showDialog(
    context: context,
    builder: (context) => AlertDialog(
      backgroundColor: Colors.amber[50],
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),

```

```

title: Text(isEdit ? 'Edit Surat Keluar' : 'Tambah Surat Keluar'),
content: SingleChildScrollView(
  child: Column(
    children: [
      TextField(
        controller: noSuratController,
        decoration: const InputDecoration(labelText: 'No Surat'),
      ),
      TextField(
        controller: tujuanController,
        decoration: const InputDecoration(labelText: 'Tujuan'),
      ),
      TextField(
        controller: tanggalSuratController,
        decoration: const InputDecoration(
          labelText: 'Tanggal Surat (yyyy-MM-dd)',
        ),
      ),
      TextField(
        controller: perihalController,
        decoration: const InputDecoration(labelText: 'Perihal'),
      ),
      TextField(
        controller: kategoriController,
        decoration: const InputDecoration(labelText: 'Lampiran'),
      ),
      TextField(
        controller: statusController,
        decoration: const InputDecoration(labelText: 'Isi Surat'),
      ),
      const SizedBox(height: 10),
      ElevatedButton.icon(
        onPressed: () async {
          FilePickerResult? result =
            await FilePicker.platform.pickFiles();
          if (result != null) {
            pickedFile = result.files.first;
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text('File dipilih: ${pickedFile!.name}'),
              );
            }
          }
        ),
      ),
    ],
  ),
),

```

```

    },
    icon: const Icon(Icons.attach_file),
    label: const Text('Pilih File'),
  ),
],
),
),
actions: [
  TextButton(
    onPressed: () => Navigator.pop(context),
    child: const Text('Batal'),
  ),
  TextButton(
    onPressed: () async {
      if ([
        noSuratController,
        tujuanController,
        tanggalSuratController,
        perihalController,
        kategoriController,
        statusController
      ].any((c) => c.text.isEmpty)) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text('Semua field harus diisi')));
        return;
      }

      DateTime? parsedDate;
      try {
        parsedDate = DateTime.parse(tanggalSuratController.text);
      } catch (_) {
        ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
          content: Text('Format tanggal tidak valid (yyyy-MM-dd)'));
        return;
      }

      String? fileUrl = data['file_url'];
      if (pickedFile != null) {
        try {
          final ref = _storage.ref().child(
            'surat_keluar/${user.uid}_${DateTime.now().millisecondsSinceEpoch}_${pickedFile!.name}');

```

```

        uploadTask = ref.putFile(File(pickedFile!.path!));
        final snapshot = await uploadTask!.whenComplete(() {});
        fileUrl = await snapshot.ref.getDownloadURL();
    } catch (e) {
        Navigator.pop(context); // tutup loading
        ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('Upload file gagal.')));
        return;
    }
}

final docId = isEdit
    ? existingSurat!.id
    : 'K-${DateTime.now().millisecondsSinceEpoch}';
final docRef = _firestore
    .collection('users')
    .doc(user.uid)
    .collection('surat_keluar')
    .doc(docId);

try {
    await docRef.set({
        'id': docId,
        'no_surat': noSuratController.text,
        'tujuan': tujuanController.text,
        'tanggal_surat': Timestamp.fromDate(parsedDate),
        'perihal': perihalController.text,
        'lampiran': kategoriController.text,
        'status': statusController.text,
        'file_url': fileUrl,
        'created_at': isEdit
            ? (data['created_at'] ?? FieldValue.serverTimestamp())
            : FieldValue.serverTimestamp(),
        'created_by': user.email,
    });

    Navigator.pop(context); // tutup loading
    Navigator.pop(context); // tutup form
    _loadSuratKeluar();
} catch (e) {
    Navigator.pop(context); // tutup loading
    ScaffoldMessenger.of(context).showSnackBar(

```

```

        const SnackBar(content: Text('Gagal menyimpan data.'));
      },
      child: const Text('Simpan'),
    ),
  ],
),
);
}

void _deleteSuratKeluar(DocumentSnapshot surat) async {
  await surat.reference.delete();
  _loadSuratKeluar();
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Surat keluar berhasil dihapus')));
}

void _showDetailDialog(DocumentSnapshot surat) {
  final data = surat.data() as Map<String, dynamic>;
  final tanggalSurat = data['tanggal_surat'] is Timestamp
    ? (data['tanggal_surat'] as Timestamp).toDate().toString().split(' ')[0]
    : data['tanggal_surat'].toString();
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text('Detail Surat Keluar'),
      backgroundColor: Colors.amber[50],
      content: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text('No Surat: ${data['no_surat']}'),
            Text('Tujuan: ${data['tujuan']}'),
            Text('Tanggal Surat: $tanggalSurat'),
            Text('Perihal: ${data['perihal']}'),
            Text('Lampiran: ${data['lampiran']}'),
            Text('Isi Surat: ${data['status']}'),
            if (data['file_url'] != null)
              TextButton(
                onPressed: () {
                  Navigator.push(
                    context,

```

```

        MaterialPageRoute(
          builder: (context) =>
            PdfViewerPage(pdfUrl: data['file_url']),
        ),
      );
    },
    child: const Text('Lihat Lampiran'),
  ),
],
),
),
actions: [
  TextButton(
    onPressed: () => Navigator.pop(context),
    child: const Text('Tutup')),
  TextButton(
    onPressed: () => _addOrEditSuratKeluar(existingSurat: surat),
    child: const Text('Edit')),
],
),
);
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      automaticallyImplyLeading: false,
      title: const Text('Arsip Surat Keluar',
        style: TextStyle(fontWeight: FontWeight.bold)),
      backgroundColor: Colors.amber,
      actions: [
        Padding(
          padding: const EdgeInsets.all(10.0),
          child: IconButton(
            icon: const Icon(Icons.search),
            onPressed: () {
              TextEditingController searchController =
                TextEditingController();
              showDialog(
                context: context,
                builder: (context) => AlertDialog(

```

```

title: const Text('Cari Surat Keluar'),
content: TextField(
  controller: searchController,
  decoration: const InputDecoration(
    labelText: 'Cari berdasarkan Tujuan atau Perihal',
  ),
  onChanged: (value) {
    setState(() {
      if (value.isEmpty) {
        _loadSuratKeluar();
      } else {
        _suratKeluar = _suratKeluar.where((surat) {
          final data = surat.data() as Map<String, dynamic>;
          final tujuan =
            data['tujuan']?.toLowerCase() ?? "";
          final perihal =
            data['perihal']?.toLowerCase() ?? "";
          return tujuan.contains(value.toLowerCase()) ||
            perihal.contains(value.toLowerCase());
        }).toList();
      }
    });
  },
),
actions: [
  TextButton(
    onPressed: () {
      setState(() {
        final value = searchController.text.trim();
        if (value.isEmpty) {
          _loadSuratKeluar();
        } else {
          _suratKeluar = _suratKeluar.where((surat) {
            final data =
              surat.data() as Map<String, dynamic>;
            final tujuan =
              data['tujuan']?.toLowerCase() ?? "";
            final perihal =
              data['perihal']?.toLowerCase() ?? "";
            return tujuan.contains(value.toLowerCase()) ||
              perihal.contains(value.toLowerCase());
          }).toList();
        }
      });
    },
  ),
],

```

```

        }
      });
      Navigator.pop(context);
    },
    child: const Text('Cari'),
  ),
],
),
);
},
),
),
],
),
body: ListView.builder(
  itemCount: _suratKeluar.length,
  itemBuilder: (context, index) {
    final surat = _suratKeluar[index];
    final data = surat.data() as Map<String, dynamic>;
    final tanggal = data['tanggal_surat'] is Timestamp
      ? (data['tanggal_surat'] as Timestamp)
        .toDate()
        .toString()
        .split(' ')[0]
      : data['tanggal_surat'].toString();
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: Container(
        decoration: BoxDecoration(
          color: Colors.orange[100],
          borderRadius: BorderRadius.circular(15),
        ),
        child: ListTile(
          title: Text(data['perihal'] ?? data['tujuan'] ?? ''),
          subtitle: Text('Tujuan: ${data['tujuan']} | Tanggal: $tanggal'),
          onTap: () => _showDetailDialog(surat),
          trailing: Row(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              IconButton(
                icon: const Icon(Icons.edit),
                onPressed: () =>

```

```

        _addOrEditSuratKeluar(existingSurat: surat)),
      IconButton(
        icon: const Icon(Icons.delete),
        onPressed: () => _deleteSuratKeluar(surat)),
    ],
  ),
),
);
},
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.amber,
  onPressed: () => _addOrEditSuratKeluar(),
  child: const Icon(Icons.add),
),
);
}
}

```

Kode Laporan Arsip Masuk

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:pdf/pdf.dart';
import 'package:pdf/widgets.dart' as pw;
import 'package:printing/printing.dart';

class ExportMasukPage extends StatefulWidget {
  const ExportMasukPage({Key? key}) : super(key: key);

  @override
  State<ExportMasukPage> createState() => _ExportMasukPageState();
}

class _ExportMasukPageState extends State<ExportMasukPage> {
  List<Map<String, dynamic>> _allData = [];
  String _selectedWaktu = 'Semua';

  final List<String> waktuList = [
    'Semua',
    'Mingguan',
    'Bulanan',
    'Tahunan',
  ];

  @override
  void initState() {
    super.initState();
    _loadData();
  }

  Future<void> _loadData() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user == null) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Silakan login terlebih dahulu')),
      );
      return;
    }
  }
```

```

final userDoc = await FirebaseFirestore.instance
    .collection('users')
    .doc(user.uid)
    .get();
final role = userDoc.data()?['role'] ?? 'user';

```

```

QuerySnapshot<Map<String, dynamic>> snapshot;

```

```

if (role == 'admin' || role == 'ketua') {
    snapshot =
        await FirebaseFirestore.instance.collectionGroup('surat_masuk').get();
} else {
    snapshot = await FirebaseFirestore.instance
        .collection('users')
        .doc(user.uid)
        .collection('surat_masuk')
        .get();
}

```

```

final List<Map<String, dynamic>> tempList = [];
for (var doc in snapshot.docs) {
    final item = doc.data();
    item['id'] = doc.id;
    tempList.add(item);
}

```

```

setState(() {
    _allData = tempList;
});
}

```

```

List<Map<String, dynamic>> _filterByWaktu(List<Map<String, dynamic>>
data) {
    final now = DateTime.now();
    print("Sekarang: $now");

```

```

    if (_selectedWaktu == 'Semua') return data;

```

```

    return data.where((item) {
        Timestamp? ts = item['tanggal_surat_masuk'] ?? item['tanggal_surat'];
        if (ts is! Timestamp) return false;
        DateTime tgl = ts.toDate();

```

```

print("Item tgl: $tgl"); // Tambahan debug

switch (_selectedWaktu) {
    case 'Mingguan':
        DateTime mingguAwal = now.subtract(Duration(days: now.weekday - 1));
        DateTime mingguAkhir = mingguAwal.add(const Duration(days: 6));
        return tgl.isAfter(mingguAwal.subtract(const Duration(days: 1))) &&
            tgl.isBefore(mingguAkhir.add(const Duration(days: 1)));
    case 'Bulanan':
        return tgl.year == now.year && tgl.month == now.month;
    case 'Tahunan':
        return tgl.year == now.year;
    default:
        return true;
}
}).toList();
}

Future<void> exportPDF(BuildContext context) async {
    try {
        if (_allData.isEmpty) {
            await _loadData();
        }

        final filteredData = _filterByWaktu(_allData);

        if (filteredData.isEmpty) {
            ScaffoldMessenger.of(context).showSnackBar(
                const SnackBar(content: Text('Tidak ada data sesuai filter waktu')),
            );
            return;
        }

        final pdf = pw.Document();
        final List<List<String>> tableData = [];
        int index = 1;

        for (final item in filteredData) {
            String tanggalSurat = "";
            if (item['tanggal_surat_masuk'] is Timestamp) {
                DateTime date = (item['tanggal_surat_masuk'] as Timestamp).toDate();

```

```

        tanggalSurat =
            '${date.day.toString().padLeft(2, '0')} ${_getNamaBulan(date.month)}
            ${date.year}';
    } else {
        tanggalSurat = item['tanggal_surat_masuk']?.toString() ?? "";
    }

    tableData.add([
        '$index',
        tanggalSurat,
        item['no_surat'] ?? "",
        item['perihal'] ?? "",
        item['pengirim'] ?? "",
        item['lampiran'] ?? "",
        item['status'] ?? "",
    ]);
    index++;
}

pdf.addPage(
    pw.Page(
        pageFormat: PdfPageFormat.a4,
        build: (context) => pw.Column(
            crossAxisAlignment: pw.CrossAxisAlignment.start,
            children: [
                pw.Text('Laporan Arsip Surat Masuk',
                    style: pw.TextStyle(fontSize: 20)),
                pw.SizedBox(height: 20),
                pw.Table.fromTextArray(
                    headers: [
                        'No',
                        'Tanggal Masuk',
                        'No Surat',
                        'Perihal',
                        'Pengirim',
                        'Lampiran',
                        'Status',
                    ],
                    data: tableData,
                ),
            ],
        ),
    ),
);

```

```

    ),
  );

  await Printing.layoutPdf(onLayout: (format) => pdf.save());
} catch (e) {
  print("Error saat export PDF: $e");
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Gagal export PDF: $e')),
  );
}
}

```

```

String _getNamaBulan(int month) {
  const bulanList = [
    'Januari',
    'Februari',
    'Maret',
    'April',
    'Mei',
    'Juni',
    'Juli',
    'Agustus',
    'September',
    'Oktober',
    'November',
    'Desember'
  ];
  return bulanList[month - 1];
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        'Laporan Arsip Masuk',
        style: TextStyle(fontWeight: FontWeight.bold),
      ),
      automaticallyImplyLeading: false,
      backgroundColor: Colors.amber,
      actions: const [
        Padding(

```

```

padding: EdgeInsets.only(right: 12),
child: Icon(Icons.picture_as_pdf),
),
],
),
body: Padding(
padding: const EdgeInsets.all(16.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
DropdownButtonFormField<String>(
value: _selectedWaktu,
decoration: const InputDecoration(
labelText: 'Filter Waktu',
border: OutlineInputBorder(),
),
items: waktuList
.map((waktu) =>
DropdownMenuItem(value: waktu, child: Text(waktu)))
.toList(),
onChanged: (value) async {
if (value != null) {
setState(() {
_selectedWaktu = value;
});
await _loadData();
}
},
),
const SizedBox(height: 20),
ElevatedButton.icon(
onPressed: () => exportPDF(context),
icon: const Icon(Icons.picture_as_pdf),
label: const Text('Download ke PDF'),
style: ElevatedButton.styleFrom(
backgroundColor: Colors.amber,
minimumSize: const Size.fromHeight(50),
),
),
],
),
),
),

```

Kode Laporan Arsip Keluar

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:pdf/pdf.dart';
import 'package:pdf/widgets.dart' as pw;
import 'package:printing/printing.dart';

class ExportKeluarPage extends StatefulWidget {
  const ExportKeluarPage({super.key});

  @override
  State<ExportKeluarPage> createState() => _ExportKeluarPageState();
}

class _ExportKeluarPageState extends State<ExportKeluarPage> {
  String _selectedWaktu = 'Semua';
  List<Map<String, dynamic>> _allData = [];

  final List<String> waktuList = [
    'Semua',
    'Mingguan',
    'Bulanan',
    'Tahunan',
  ];

  Future<void> _loadData() async {
    final user = FirebaseAuth.instance.currentUser;
    if (user == null) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Silakan login terlebih dahulu')),
      );
      return;
    }

    final userDoc = await FirebaseFirestore.instance
      .collection('users')
      .doc(user.uid)
      .get();

    final role = userDoc.data()?['role'] ?? 'user';
```

```
QuerySnapshot<Map<String, dynamic>> snapshot;
```

```
if (role == 'admin' || role == 'ketua') {  
    snapshot = await FirebaseFirestore.instance  
        .collectionGroup('surat_keluar')  
        .get();  
} else {  
    snapshot = await FirebaseFirestore.instance  
        .collection('users')  
        .doc(user.uid)  
        .collection('surat_keluar')  
        .get();  
}
```

```
final List<Map<String, dynamic>> tempList = [];  
for (var doc in snapshot.docs) {  
    final item = doc.data();  
    item['id'] = doc.id;  
    tempList.add(item);  
}
```

```
setState(() {  
    _allData = tempList;  
});  
}
```

```
List<Map<String, dynamic>> _filterByWaktu(List<Map<String, dynamic>>  
data) {
```

```
    final now = DateTime.now();
```

```
    if (_selectedWaktu == 'Semua') return data;
```

```
    return data.where((item) {  
        if (item['tanggal_surat'] is! Timestamp) return false;  
        DateTime tgl = (item['tanggal_surat'] as Timestamp).toDate();
```

```
        switch (_selectedWaktu) {  
            case 'Mingguan':  
                DateTime mingguAwal = now.subtract(Duration(days: now.weekday - 1));  
                DateTime mingguAkhir = mingguAwal.add(const Duration(days: 6));  
                return tgl.isAfter(mingguAwal.subtract(const Duration(days: 1))) &&
```

```

        tgl.isBefore(mingguAkhir.add(const Duration(days: 1)));
    case 'Bulanan':
        return tgl.year == now.year && tgl.month == now.month;
    case 'Tahunan':
        return tgl.year == now.year;
    }
    return true;
}).toList();
}

```

```

Future<void> exportPDF(BuildContext context) async {
    try {
        if (_allData.isEmpty) {
            await _loadData();
        }
    }
}

```

```

final filteredData = _filterByWaktu(_allData);

```

```

if (filteredData.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Tidak ada data sesuai filter waktu')),
    );
    return;
}

```

```

final pdf = pw.Document();
final List<List<String>> tableData = [];
int index = 1;

```

```

for (final item in filteredData) {
    String tanggalSurat = "";
    if (item['tanggal_surat'] is Timestamp) {
        DateTime date = (item['tanggal_surat'] as Timestamp).toDate();
        tanggalSurat =
            '${date.day.toString().padLeft(2, '0')} ${_getNamaBulan(date.month)}
            ${date.year}';
    } else {
        tanggalSurat = item['tanggal_surat']?.toString() ?? "";
    }
}

```

```

tableData.add([
    '$index',

```

```

        tanggalSurat,
        item['no_surat'] ?? ",
        item['perihal'] ?? ",
        item['tujuan'] ?? ",
        item['lampiran'] ?? ",
        item['status'] ?? ",
    ];
    index++;
}

pdf.addPage(
  pw.Page(
    pageFormat: PdfPageFormat.a4,
    build: (context) => pw.Column(
      crossAxisAlignment: pw.CrossAxisAlignment.start,
      children: [
        pw.Text('Laporan Arsip Surat Keluar',
          style: pw.TextStyle(fontSize: 20)),
        pw.SizedBox(height: 20),
        pw.Table.fromTextArray(
          headers: [
            'No',
            'Tanggal Surat',
            'No Surat',
            'Perihal',
            'Tujuan',
            'Lampiran',
            'Status',
          ],
          data: tableData,
        ),
      ],
    ),
  ),
);

await Printing.layoutPdf(onLayout: (format) => pdf.save());
} catch (e) {
  print("Error saat export PDF: $e");
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Gagal export PDF: $e')),
  );
};

```

```

    }
}

String _getNamaBulan(int month) {
  const bulanList = [
    'Januari',
    'Februari',
    'Maret',
    'April',
    'Mei',
    'Juni',
    'Juli',
    'Agustus',
    'September',
    'Oktober',
    'November',
    'Desember'
  ];
  return bulanList[month - 1];
}

@override
void initState() {
  super.initState();
  _loadData();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text(
        'Laporan Arsip Keluar',
        style: TextStyle(fontWeight: FontWeight.bold),
      ),
      automaticallyImplyLeading: false,
      backgroundColor: Colors.amber,
      actions: const [
        Padding(
          padding: EdgeInsets.only(right: 12),
          child: Icon(Icons.picture_as_pdf),
        ),
      ],
    ),
  ),
)

```

```

body: Padding(
  padding: const EdgeInsets.all(16),
  child: Column(
    children: [
      DropdownButtonFormField<String>(
        value: _selectedWaktu,
        decoration: const InputDecoration(
          labelText: 'Filter Waktu',
          border: OutlineInputBorder(),
        ),
        items: waktuList
          .map((waktu) =>
            DropdownMenuItem(value: waktu, child: Text(waktu)))
          .toList(),
        onChanged: (value) async {
          if (value != null) {
            setState(() {
              _selectedWaktu = value;
            });
            await _loadData(); // Perbarui data saat filter berubah
          }
        },
      ),
      const SizedBox(height: 20),
      ElevatedButton.icon(
        onPressed: () => exportPDF(context),
        icon: const Icon(Icons.picture_as_pdf),
        label: const Text('Download ke PDF'),
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.amber,
          minimumSize: const Size.fromHeight(50),
        ),
      ),
    ],
  ),
);
}

```

Kode Tentang IPPNU

```
import 'package:flutter/material.dart';

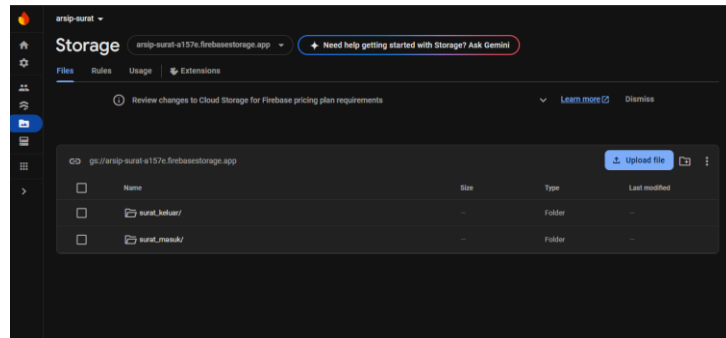
class TentangPage extends StatelessWidget {
  const TentangPage({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        automaticallyImplyLeading: false,
        title: const Text('Tentang IPPNU',
          style: TextStyle(fontWeight: FontWeight.bold)),
        backgroundColor: Colors.amber,
      ),
      body: SafeArea(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Konten Utama
            Expanded(
              child: SingleChildScrollView(
                child: Padding(
                  padding: const EdgeInsets.all(16),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      // Logo
                      Center(
                        child: Image.asset(
                          'assets/ippnu.png', // ganti sesuai lokasi file gambar
                          height: 200,
                        ),
                      ),
                      // Visi
                      Padding(
                        padding: const EdgeInsets.all(10.0),
                        child: const Text(
                          'Visi IPPNU',
                          style: TextStyle(
```

```

        fontSize: 18,
        fontWeight: FontWeight.bold,
    ),
),
),
const SizedBox(height: 8),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: const Text(
    'Terbentuknya kesempurnaan Pelajar Putri Indonesia yang '
    'bertakwa, berakhlakul karimah, berilmu, dan '
    'berwawasan kebangsaan.',
    style: TextStyle(fontSize: 16),
    textAlign: TextAlign.justify,
  ),
),
const SizedBox(height: 24),
// Misi
Padding(
  padding: const EdgeInsets.all(10.0),
  child: const Text(
    'Misi IPPNU',
    style: TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
    ),
  ),
),
const SizedBox(height: 8),
Padding(
  padding: const EdgeInsets.all(10.0),
  child: const Text(
    '1. Membangun kader NU yang berkualitas, berakhlakul karimah,
    ,
    'bersikap demokratis dalam kehidupan bermasyarakat, berbangsa
    dan bernegara.\n'
    '2. Mengembangkan wacana dan kualitas sumber daya kader
    menuju terciptanya kesetaraan gender.\n'
    '3. Membentuk kader yang dinamis, kreatif, dan inovatif.',
    style: TextStyle(fontSize: 16),
    textAlign: TextAlign.justify,
  ),
),

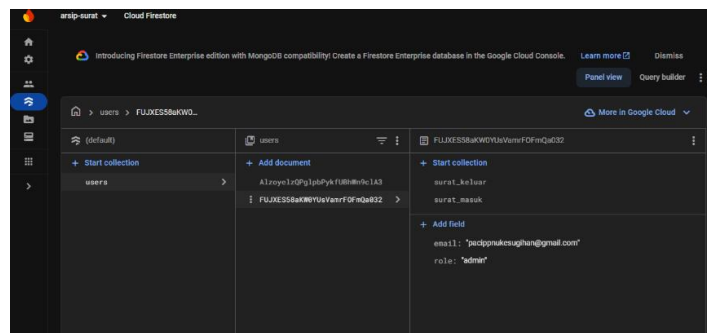
```

Lampiran III Dokumentasi



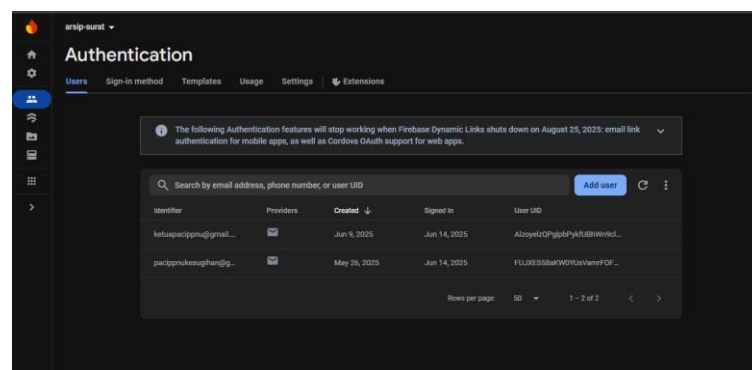
Gambar Tampilan Firebase Storage – Folder Penyimpanan File Surat

Gambar ini menunjukkan struktur penyimpanan file di Firebase Storage, dengan dua folder utama: surat_masuk/ dan surat_keluar/. File digital yang diunggah akan otomatis tersimpan sesuai jenis surat, sehingga lebih mudah diakses dan aman dari kehilangan arsip fisik.



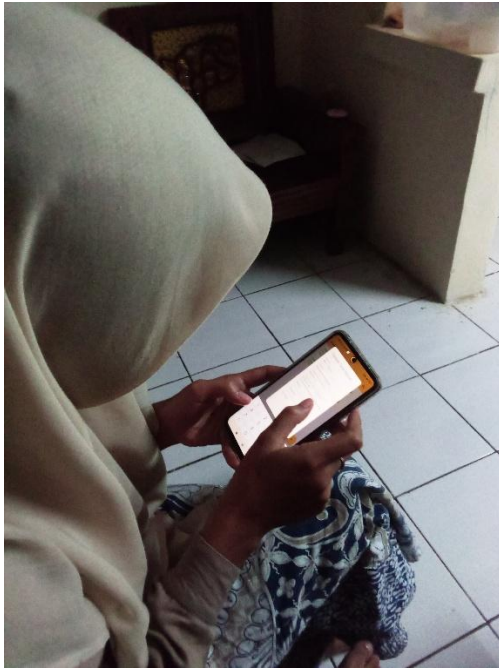
Struktur Data Pengguna pada Firebase Firestore

Gambar ini menunjukkan struktur database users di Cloud Firestore. Setiap pengguna memiliki dokumen berisi email dan role, serta subkoleksi surat_masuk dan surat_keluar untuk menyimpan arsip berdasarkan jenis surat dan identitas pengguna.



Gambar Tampilan Firebase Authentication – Data Pengguna Terdaftar

Firebase Authentication menampilkan dua akun pengguna untuk membedakan akses admin dan ketua berdasarkan email login.



Gambar Pengurus PAC IPPNU Kesugihan Saat Mencoba Aplikasi Arsip Surat