

LAMPIRAN

1. main.dart

```
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:flutter_dotenv/flutter_dotenv.dart';
import 'package:flutter_easyloading/flutter_easyloading.dart';
import 'package:stacked_services/stacked_services.dart';

import 'app/app.dialogs.dart';
import 'app/app.locator.dart';
import 'app/app.router.dart';
import 'app/themes/app_theme.dart';

Future main() async {
  WidgetsFlutterBinding.ensureInitialized();
  HttpOverrides.global = MyHttpOverrides();
  await dotenv.load(fileName: ".env");
  await setupAllLocator();
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Panti Asuhan Aisyiyah Abadi',
      theme: appTheme,
      debugShowCheckedModeBanner: false,
      navigatorKey: StackedService.navigatorKey,
      onGenerateRoute: StackedRouter().onGenerateRoute,
      builder: EasyLoading.init(),
    );
  }
}

Future<void> setupAllLocator() async {
  await setupLocator();
  setupDialogUi();
  // setupBottomsheetUi();
  // setupSnackBarUi();
}

class MyHttpOverrides extends HttpOverrides {
  @override
  HttpClient createHttpClient(SecurityContext? context) {
    return super.createHttpClient(context)
  }
}
```

```

    ..badCertificateCallback =
      (X509Certificate cert, String host, int port) => true;
  }
}

```

2. app.dart

```

import 'package:stacked_services/stacked_services.dart';
import 'package:stacked/stacked_annotations.dart';

import '../services/http_services.dart';
import '../services/my_easyloading.dart';
import '../services/other_function.dart';
import
'../ui/views/admin_index_tracking/add_siswa_dialog/add_siswa_dialog/add_siswa_dialog_view.
dart';
import
'../ui/views/admin_index_tracking/dana_sosial_khusus/add_donatur_dialog/add_donatur_dialo
g_view.dart';
import '../ui/views/admin_index_tracking/dana_sosial_khusus/dana_sosial_khusus_view.dart';
import '../ui/views/admin_index_tracking/admin_index_tracking_view.dart';
import '../ui/views/admin_index_tracking/dana_sosial_admin/dana_sosial_admin_view.dart';
import '../ui/views/admin_index_tracking/data_siswa/data_siswa_view.dart';
import
'../ui/views/admin_index_tracking/edit_siswa/edit_dialog_siswa/edit_dialog_siswa_view.dart';
import '../ui/views/admin_index_tracking/edit_siswa/edit_siswa_view.dart';
import '../ui/views/admin_index_tracking/filter_dialog/filter_dialog_view.dart';
import '../ui/views/admin_index_tracking/profil/profil_view.dart';
import '../ui/views/admin_index_tracking/sejarah/sejarah_view.dart';
import
'../ui/views/admin_index_tracking/struktur_organisasi/edit_struktur_organisasi_dialog/edit_stru
krur_organisasi_dialog_view.dart';
import '../ui/views/admin_index_tracking/struktur_organisasi/struktur_organisasi_view.dart';
import '../ui/views/admin_index_tracking/visi_misi/visi_misi_view.dart';
import '../ui/views/detail_dana_sosial/detail_dana_sosial_view.dart';
import '../ui/views/login_screen/login_screen_view.dart';
import
'../ui/views/pimpinan_index_tracking/pimpinan_index_tracking/pimpinan_index_tracking_view.
dart';
import '../ui/views/splash_screen/splash_screen_view.dart';
import '../ui/views/tambah_dana_sosial/tambah_dana_sosial_view.dart';
import '../ui/views/user_index_tracking/user_index_tracking_view.dart';

@StackedApp(
  routes: [
    MaterialRoute(page: SplashScreenView, initial: true),
    MaterialRoute(page: LoginScreenView),
    MaterialRoute(
      page: AdminIndexTrackingView,

```

```

children: [
    // MaterialRoute(page: AdminIndexView, initial: true),
    MaterialRoute(page: DanaSosialAdminView),
    MaterialRoute(page: DanaSosialKhususView),
    MaterialRoute(page: DataSiswaView),
    MaterialRoute(page: ProfilView),
    MaterialRoute(page: VisiMisiView),
    MaterialRoute(page: SejarahView),
    MaterialRoute(
        page: StrukturOrganisasiView,
    ),
],
),
MaterialRoute(page: TambahDanaSosialView),
MaterialRoute(page: EditSiswaView),
MaterialRoute(page: DetailDanaSosialView),
MaterialRoute(
    page: PimpinanIndexTrackingView,
    children: [
        // MaterialRoute(page: AdminIndexView, initial: true),
        MaterialRoute(page: DanaSosialAdminView),
        MaterialRoute(page: DanaSosialKhususView),
        MaterialRoute(page: DataSiswaView),
        MaterialRoute(page: ProfilView),
    ],
),
MaterialRoute(
    page: UserIndexTrackingView,
    children: [
        // MaterialRoute(page: AdminIndexView, initial: true),
        MaterialRoute(page: DanaSosialAdminView),
        MaterialRoute(page: DanaSosialKhususView),
        MaterialRoute(page: DataSiswaView),
        MaterialRoute(page: ProfilView),
        MaterialRoute(page: VisiMisiView),
        MaterialRoute(page: SejarahView),
        MaterialRoute(page: StrukturOrganisasiView),
    ],
),
],
dialogs: [
    StackedDialog(classType: AddSiswaDialogView),
    StackedDialog(classType: FilterDialogView),
    StackedDialog(classType: EditStrukturOrganisasiDialogView),
    StackedDialog(classType: EditDialogSiswaView),
    StackedDialog(classType: AddDonaturDialogView),
],
dependencies: [
    LazySingleton(classType: NavigationService),
    LazySingleton(classType: DialogService),

```

```

        LazySingleton(classType: SnackbarService),
        LazySingleton(classType: BottomSheetService),
        //

        LazySingleton(classType: MyEasyLoading),
        LazySingleton(classType: MyHttpServices),
        LazySingleton(classType: OtherFunction),
    ],
    logger: StackedLogger(),
  )
  class App {}

```

3. http_services.dart

```

import 'package:dio/dio.dart';
import 'package:flutter_dotenv/flutter_dotenv.dart';

import '../app/app.logger.dart';

class MyHttpServices {
  final log = getLogger('MyHttpServices');
  final _options = BaseOptions(
    baseUrl: dotenv.env['api_url']!,
    connectTimeout: const Duration(seconds: 120),
    receiveTimeout: const Duration(seconds: 120),
  );

  late Dio _dio;

  MyHttpServices() {
    _dio = Dio(_options);
  }

  Future<Response> get(String path) async {
    try {
      return await _dio.get(path);
    }
  }
}

```

```

    } on DioError catch (e) {
      log.e(e.message);
      log.e(e.response);
      rethrow;
    }
  }
}

```

```

Future<Response> postWithFormData(String path, FormData formData) async {
  try {
    return await _dio.post(path, data: formData);
  } on DioError catch (e) {
    log.e(e.message);
    log.e(e.response!.statusCode);
    log.e(e.response!.statusMessage);
    rethrow;
  }
}

```

```

// // delete
// Future<Response> delete(String path, FormData data) async {
//   try {
//     // log.i('path: $path');
//     return await _dio.delete(
//       path,
//       data: data,
//       // encoding: Encoding.getByName('utf-8'),
//       options: Options(
//         headers: {
//           'Content-Type': 'application/x-www-form-urlencoded',
//         },
//       ),
//     );
//   }
// }

```

```

// } on DioError catch (e) {
//   log.e(e.message);
//   log.e(e.response);
//   rethrow;
// }
//}
}

```

4. my_easyloading.dart

```

import 'package:flutter_easyloading/flutter_easyloading.dart';

class MyEasyLoading {
  showLoading() {
    EasyLoading.show(
      status: 'loading...',
      maskType: EasyLoadingMaskType.black,
      dismissOnTap: false,
    );
  }

  dismissLoading() {
    EasyLoading.dismiss();
  }

  customLoading(String message) {
    EasyLoading.show(
      status: message,
      maskType: EasyLoadingMaskType.black,
      dismissOnTap: false,
    );
  }

  showSuccess(String message) {
    EasyLoading.showSuccess(message);
  }

  showError(String message) {
    EasyLoading.showError(message);
  }

  showInfo(String message) {
    EasyLoading.showInfo(message);
  }
}

```

```

        showProgress(double progress, String status) {
            EasyLoading.showProgress(progress, status: status);
        }
    }
}

```

5. other_function.dart

```

import 'package:intl/intl.dart';

class OtherFunction {
    int umur(String tanggalLahir) {
        // change tanggalLahir to DateTime
        DateTime date = DateTime.parse(tanggalLahir);
        // get current date
        DateTime now = DateTime.now();
        // get difference in year
        int year = now.year - date.year;
        return year;
    }

    String commaFormat(int number) {
        final formatter = NumberFormat('#,###');
        return formatter.format(number);
    }

    String changeMonth(String month) {
        switch (month) {
            case 'Januari':
                return '01';
            case 'Februari':
                return '02';
            case 'Maret':
                return '03';
            case 'April':
                return '04';
            case 'Mei':
                return '05';
            case 'Juni':
                return '06';
            case 'Juli':
                return '07';
            case 'Agustus':
                return '08';
            case 'September':
                return '09';
            case 'Oktober':
                return '10';
            case 'November':
                return '11';
        }
    }
}

```



```

        case 'Desember':
            return '12';
        default:
            return "";
    }
}

```

```

String changeMonthYear(String s) {
    // get the last 2 digits
    String month = s.substring(s.length - 2);
    // get the first 4 digits
    String year = s.substring(0, 4);
    // return the month and year
    switch (month) {
        case '01':
            return 'Januari $year';
        case '02':
            return 'Februari $year';
        case '03':
            return 'Maret $year';
        case '04':
            return 'April $year';
        case '05':
            return 'Mei $year';
        case '06':
            return 'Juni $year';
        case '07':
            return 'Juli $year';
        case '08':
            return 'Agustus $year';
        case '09':
            return 'September $year';
        case '10':
            return 'Oktober $year';
        case '11':
            return 'November $year';
        case '12':
            return 'Desember $year';
        default:
            return "";
    }
}

```

```

String getDayOfWeek(String date) {
    DateTime dateTime = DateTime.parse(date);
    List<String> daysOfWeek = [
        'Senin',
        'Selasa',
        'Rabu',
        'Kamis',

```

```

    'Jumat',
    'Sabtu',
    'Minggu'
  ];
  return daysOfWeek[dateTime.weekday - 1];
}

String capitalizeEachWord(String s) {
  return s
    .split(' ')
    .map((word) => word[0].toUpperCase() + word.substring(1))
    .join(' ');
}
}

```

6. admin_index_tracking_view_model.dart

```

import 'package:flutter/material.dart';
import 'package:panti_asuhan/services/other_function.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:stacked/stacked.dart';
import 'package:stacked_services/stacked_services.dart';

import '../app/app.locator.dart';
import '../app/app.logger.dart';
import '../app/app.router.dart';

class AdminIndexTrackingViewModel extends IndexTrackingViewModel {
  final log = getLogger('AdminIndexTrackingViewModel');
  final _navigationService = locator<NavigationService>();
  final _dialogService = locator<DialogService>();
  final Future<SharedPreferences> _prefs = SharedPreferences.getInstance();
  final otherFunction = locator<OtherFunction>();

  final _bottomNavBarList = [
    {
      'name': 'List',
      'icon': Icons.people_alt_outlined,
      'header': 'List Anak Panti'
    },
    {'name': 'Dana', 'icon': Icons.money, 'header': 'Dana Sosial'},
    // {'name': 'Dana', 'icon': Icons.money, 'header': 'Dana Sosial Khusus'},
    {
      'name': 'Profil',
      'icon': Icons.person_4_outlined,
      'header': 'Profil Panti Asuhan'
    },
    {'name': 'V & M', 'icon': Icons.list_alt_rounded, 'header': 'Visi & Misi'},
    {'name': 'Sejarah', 'icon': Icons.list_outlined, 'header': 'Sejarah'},
    {

```

```

        'name': 'S O',
        'icon': Icons.people_alt_outlined,
        'header': 'Struktur Organisasi'
    },
];

```

```

List<Map<String, dynamic>> get bottomNavBarList => _bottomNavBarList;

```

```

final List<String> _views = [
    AdminIndexTrackingViewRoutes.dataSiswaView,
    AdminIndexTrackingViewRoutes.danaSosialAdminView,
    // AdminIndexTrackingViewRoutes.danaSosialKhususView,
    AdminIndexTrackingViewRoutes.profilView,
    AdminIndexTrackingViewRoutes.visiMisiView,
    AdminIndexTrackingViewRoutes.sejarahView,
    AdminIndexTrackingViewRoutes.strukturOrganisasiView
];

```

```

String header = 'Dana Sosial';

```

```

Future<void> init() async {
    setIndex(1);
    // await 2 seconds to make sure the view is loaded
    // await Future.delayed(const Duration(milliseconds: 500));
    // _navigationService.navigateTo(
    //   _views[1],
    //   id: 3,
    // );
}

```

```

void handleNavigation(int index) {
    log.d("handleNavigation: $index");
    log.d("currentIndex: $currentIndex");

```

```

    if (currentIndex == index) return;

```

```

    setIndex(index);
    header = _bottomNavBarList[index]['header'] as String;
    _navigationService.navigateTo(
        _views[index],
        id: 3,
    );
}

```

```

logout(BuildContext context) {
    _dialogService
        .showConfirmationDialog(
            title: 'Logout',
            description: 'Apakah anda yakin ingin logout?',
            cancelTitle: 'Ya',

```

```

        confirmationTitle: 'Tidak',
        // barrierDismissible: true,
      )
      .then((value) {
        if (!value!.confirmed) {
          log.d('logout');
          _prefs.then((SharedPreferences prefs) {
            prefs.setBool('isLogin', false);
            prefs.remove('role');
            _navigationService.clearStackAndShow(Routes.loginScreenView);
          });
        } else {
          // _navigationService.back();
          // close dialog
          Navigator.pop(context);
          // Navigator.of(StackedService.navigatorKey! as BuildContext).pop();
          // log.d('cancel logout');
        }
      });
    }
  }
}

```

7. admin_index_tracking_view.dart

```

import 'package:flutter/material.dart';
import 'package:stacked/stacked.dart';
import 'package:stacked_services/stacked_services.dart';
import 'package:stylish_bottom_bar/model/bar_items.dart';
import 'package:stylish_bottom_bar/stylish_bottom_bar.dart';

import '../..../app/app.router.dart';
import '../..../app/themes/app_colors.dart';
import '../..../app/themes/app_text.dart';
import '../admin_index_tracking_view_model.dart';

class AdminIndexTrackingView extends StatelessWidget {
  const AdminIndexTrackingView({super.key});

  @override
  Widget build(BuildContext context) {
    return ViewModelBuilder<AdminIndexTrackingViewModel>.reactive(
      viewModelBuilder: () => AdminIndexTrackingViewModel(),
      onViewModelReady: (AdminIndexTrackingViewModel model) async {
        await model.init();
      },
      builder: (
        BuildContext context,
        AdminIndexTrackingViewModel model,
        Widget? child,

```

```

){
return WillPopScope(
  onWillPop: () async {
    return false;
  },
  child: Scaffold(
    appBar: AppBar(
      title: Text(
        model.header,
        style: const TextStyle(
          color: Colors.white,
          fontSize: 20,
        ),
      ),
    ),
    backgroundColor: mainColor,
    elevation: 0,
    automaticallyImplyLeading: false,
    actions: [
      IconButton(
        onPressed: () {
          model.logout(context);
        },
        icon: const Icon(Icons.logout, color: Colors.white),
      ),
    ],
  ),
  body: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 15),
    child: ExtendedNavigator(
      navigatorKey: StackedService.nestedNavigationKey(3),
      router: AdminIndexTrackingViewRouter(),
      initialRoute: AdminIndexTrackingViewRoutes.danaSosialAdminView,
    ),
  ),
  bottomNavigationBar: StylishBottomBar(
    items: [
      for (var item in model.bottomNavBarList)
        BottomBarItem(
          icon: Icon(item['icon'],
            color: model.currentIndex ==
              model.bottomNavBarList.indexOf(item)
                ? sixthGrey
                : backgroundColor),
          title: Text(
            model.otherFunction.capitalizeEachWord(item['name']),
            style: regularTextStyle.copyWith(
              color: model.currentIndex ==
                model.bottomNavBarList.indexOf(item)
                  ? sixthGrey
                  : Colors.grey,
            ),
          ),
        ),
    ],
  ),
);
}

```

```

        overflow: TextOverflow.ellipsis,
      ),
    ),
    backgroundColor: model.currentIndex ==
      model.bottomNavBarList.indexOf(item)
      ? Colors.white
      : Colors.grey,
  ),
],
currentIndex: model.currentIndex,
hasNotch: true,
backgroundColor: mainColor,
onTap: (value) {
  model.handleNavigation(value);
},
option: BubbleBarOptions(
  barStyle: BubbleBarStyle.horizotnal,
  bubbleFillStyle: BubbleFillStyle.fill,
  opacity: 0.3),
),
),
);
},
);
}
}

```

8. dana_sosial_admin_view_modal.dart

```

import 'package:dio/dio.dart';
import 'package:flutter/material.dart';
import 'package:flutter_dotenv/flutter_dotenv.dart';
import 'package:flutter_file_downloader/flutter_file_downloader.dart';
import 'package:shared_preferences/shared_preferences.dart';

import '../..../app/app.dialogs.dart';
import '../..../app/app.locator.dart';
import '../..../app/app.logger.dart';
import '../..../app/app.router.dart';
import '../..../app/core/custom_base_view_model.dart';
import '../..../model/dana_sosial_model.dart';
import '../..../services/http_services.dart';
import '../..../services/my_easyloading.dart';
import '../..../services/other_function.dart';

class DanaSosialAdminViewModel extends CustomBaseViewModel {
  final log = getLogger('DanaSosialAdminViewModel');
  final _httpService = locator<MyHttpServices>();
  final easyLoading = locator<MyEasyLoading>();
  // final otherF

```

```

String url = dotenv.env['url']!;

int bulan = DateTime.now().month;

List<DanaSosialModel> danaSosialModelList = [];
List<Map<String, dynamic>> filteredByDateData = []; // newly added
List<Map<String, dynamic>> monthIncomeOutcome = []; // newly added
List<Map<String, dynamic>> yearIncomeOutcome = []; // newly added
int totalIncome = 0; // newly added
int totalOutcome = 0; // newly added

String? role;
bool? isLogin;

int jumlahDonasi = 0;
int jumlahPengeluaran = 0;

Future<void> init() async {
  await getData();
  await getJumlahDonasi();
  prefs.then((SharedPreferences prefs) {
    role = prefs.getString('role');
    isLogin = prefs.getBool('isLogin');
  });
  // log.i(bulan);
}

getJumlahDonasi() async {
  setBusy(true);
  easyLoading.showLoading();
  // get the month
  var bulan = DateTime.now().month;
  // log.i(bulan);
  // change bulan to string and add 0 if it is less than 10
  String bulanString = bulan.toString().length == 1 ? '0$bulan' : '$bulan';
  // log.i(bulanString);
  try {
    var response = await _httpClient.get('pemasukan?bulan=$bulanString');
    // log.i(response.data['jumlah']);
    // var theJumlahDonasi = response.data['jumlah'];
    jumlahDonasi = response.data['jumlah'];
    jumlahPengeluaran = response.data['jumlah_pengeluaran'];
  } catch (e) {
    log.e(e);
  } finally {
    setBusy(false);
    easyLoading.dismissLoading();
  }
}

```

```

getData() async {
  setBusy(true);
  easyLoading.showLoading();
  try {
    var response = await _httpService.get('dana_sosial');
    // log.i(response.data);
    danaSosialModelList = [];

    var datanya = response.data['data'];
    // log.i(datanya.length);
    if (datanya.length > 0) {
      for (var item in datanya) {
        danaSosialModelList.add(DanaSosialModel.fromJson(item));
      }
    }

    changeByDate(danaSosialModelList);
    getTahunan();
    notifyListeners();
    // log.i(danaSosialModelList.length);
  } catch (e) {
    log.e(e);
  } finally {
    setBusy(false);
    easyLoading.dismissLoading();
  }
}

getTahunan() async {
  try {
    var response = await _httpService.get('data_tahunan');

    var datanya = response.data['data'];
    log.i(datanya.length);
    for (int i = 0; i < datanya.length; i++) {
      log.i(datanya[i]);
      yearIncomeOutcome.add(datanya[i]);
    }
    notifyListeners();
  } catch (e) {
    log.e(e);
  }
}

//newly added
changeByDate(List<DanaSosialModel> data) {
  filteredByDateData = [];

  for (var item in data) {
    var monthKey =

```



```

        item.tanggal!.substring(0, 7); // Extracting the year and month
var dateKey = item.tanggal; // The full date

var monthData = filteredByDateData.firstWhere(
  (element) => element['month'] == monthKey,
  orElse: () => {
    'month': monthKey,
    'data': [],
  });

var dateData =
  monthData['data'].firstWhere((element) => element['date'] == dateKey,
    orElse: () => {
      'date': dateKey,
      'data_dana': [],
    });

dateData['data_dana'].add(item);

if (!monthData['data'].contains(dateData)) {
  monthData['data'].add(dateData);
}

if (!filteredByDateData.contains(monthData)) {
  filteredByDateData.add(monthData);
}
}
filteredByDateData.sort((a, b) => a['month'].compareTo(b['month']));
// log.i(filteredByDateData);

// group the data by month

monthIncomeOutcome = [];

for (var item in filteredByDateData) {
  var income = 0;
  var outcome = 0;

  for (var dateData in item['data']) {
    for (var danaSosialModel in dateData['data_dana']) {
      if (danaSosialModel.bentuk == 'Pemasukan' &&
        danaSosialModel.jenisDonasi == 'Uang') {
        income += int.parse(danaSosialModel.jumlah ?? '0');
      } else if (danaSosialModel.bentuk == 'Pengeluaran' &&
        danaSosialModel.jenisDonasi == 'Uang') {
        outcome += int.parse(danaSosialModel.jumlah ?? '0');
      }
    }
  }
}

```

```

        monthIncomeOutcome.add({
            'month': item['month'],
            'income': income,
            'outcome': outcome,
        });
    }

    // log.i(monthIncomeOutcome);
    totalIncome = 0;
    totalOutcome = 0;

    for (var item in monthIncomeOutcome) {
        totalIncome += int.parse(item['income'].toString());
        totalOutcome += int.parse(item['outcome'].toString());
    }

    log.i(totalIncome);
    log.i(totalOutcome);
}

goToTambahDanaSosial() {
    navigationService.navigateTo(Routes.tambahDanaSosialView);
}

getFilter(String sql) async {
    setBusy(true);
    easyLoading.showLoading();
    FormData formData = FormData.fromMap({
        'sql': sql,
    });
    try {
        var response = await _httpService.postWithFormData(
            'filter_dana',
            formData,
        );
        // log.i(response.data);
        danaSosialModelList = [];

        var datanya = response.data['data'];
        jumlahDonasi = response.data['jumlah_donasi'];
        jumlahPengeluaran = response.data['jumlah_pengeluaran'];
        // log.i(datanya.length);
        if (datanya.length > 0) {
            for (var item in datanya) {
                danaSosialModelList.add(DanaSosialModel.fromJson(item));
            }
        }

        setBusy(false);
        changeByDate(danaSosialModelList);
    }
}

```

```

        notifyListeners();
        // log.i(danaSosialModelList);
    } catch (e) {
        log.e(e);
        setBusy(false);
    } finally {
        easyLoading.dismissLoading();
    }
}

filterDialog(BuildContext context) async {
    // create a dialog
    final res = await dialogService.showCustomDialog(
        variant: DialogType.filterDialogView,
    );

    if (res!.confirmed) {
        String jenisDonasi = res.data['jenisDonasi'] == 'Semua'
            ? "
            : "jenis = '${res.data['jenisDonasi']}' and ";
        String bulan = res.data['bulan'] == 'Semua'
            ? "
            : "tanggal like '%-${OtherFunction().changeMonth(res.data['bulan'])}' and ";
        String tahun = res.data['tahun'] == 'Semua'
            ? "
            : "tanggal like \"%${res.data['tahun']}\" and ";
        String status = res.data['status'] == 'Semua'
            ? "
            : (res.data['status'] == 'Belum Dikonfirmasi')
            ? 'status = 0'
            : 'status = 1';

        if (jenisDonasi == 'Semua' &&
            bulan == 'Semua' &&
            tahun == 'Semua' &&
            status == 'Semua') {
            getData();
            return;
        }

        String sql =
            'Select * from tb_dana_sosial where $jenisDonasi$bulan$tahun$status';

        // check the last 3 character if it is 'or ' then remove it
        // if (sql.substring(sql.length - 3) == 'or ') {
        //     sql = sql.substring(0, sql.length - 3);
        // }
        if (sql.substring(sql.length - 4) == 'and ') {
            sql = sql.substring(0, sql.length - 4);
        }
    }
}

```

```

        // log.i(sql);

        getFilter(sql);
    }
}

goToEditDanaSosial(int id) async {
  navigationService.navigateTo(
    Routes.detailDanaSosialView,
    arguments: DetailDanaSosialViewArguments(
      id: id,
    ),
  );
}

deleteData(int parse) async {
  await dialogService
    .showDialog(
      title: 'Hapus Data',
      description: 'Apakah anda yakin ingin menghapus data ini?',
      buttonTitle: 'Hapus',
      cancelTitle: 'Batal',
      buttonTitleColor: Colors.red,
      cancelTitleColor: Colors.green,
    )
    .then(
      (value) async {
        if (value!.confirmed) {
          easyLoading.showLoading();
          setBusy(true);
          try {
            await _httpService.postWithFormData(
              'hapus_dana_sosial',
              FormData.fromMap({
                'id_dana_sosial': parse,
              }));
            // log.i(response.data);
            easyLoading.dismissLoading();
            easyLoading.showSuccess('Data berhasil dihapus');
            getData();
            getJumlahDonasi();
          } on DioError catch (e) {
            // easyLoading.dismissLoading();
            log.e(e);
            easyLoading.showError('Terjadi kesalahan');
          } finally {
            setBusy(false);
            easyLoading.dismissLoading();
          }
        }
      }
    )

```

```

    } else {
        // log.i('cancel');
        return;
    }
},
);
}

goToLaporanBulanan(data) async {
    // seeparate by "-", the first index is year, the second index is month
    String year = data.substring(0, data.indexOf('-'));
    String month = data.substring(data.indexOf('-') + 1);

    try {
        setBusy(true);
        easyLoading.customLoading("Mengunduh Laporan Bulanan...");
        await _httpService.get('laporan_bulanan?tahun=$year&bulan=$month');

        // if (response.data) {

        String urlPdf = '${url}assets/pdf/laporan_bulanan_$month,$year.pdf';
        log.i(urlPdf);
        FileDownloader.downloadFile(
            url: urlPdf,
            // name: "THE FILE NAME AFTER DOWNLOADING", //(optional)
            onProgress: (fileName, progress) {
                // change progress to 0-1
                double progressPercent = progress / 100;

                easyLoading.showProgress(
                    progressPercent,
                    "Downloading: $progress%",
                );
            },
            onDownloadCompleted: (String path) {
                easyLoading.dismissLoading();
                snackbarService.showSnackbar(
                    message: "Laporan Bulanan Berhasil Tersimpan di $path",
                    duration: const Duration(seconds: 3),
                );
            },
            onDownloadError: (String error) {
                // log.i('DOWNLOAD ERROR: $error');
                snackbarService.showSnackbar(
                    message: "Laporan Bulanan Gagal Tersimpan: $error",
                    duration: const Duration(seconds: 3),
                );
            }
        );
    } catch (e) {

```

```

        log.e(e);
    } finally {
        setBusy(false);
        easyLoading.dismissLoading();
    }
}

getLaporanHarian(String filteredByDateData, String dayOfWeek) async {
    String date = filteredByDateData;
    String day = dayOfWeek;

    try {
        setBusy(true);
        easyLoading.customLoading("Mengunduh Laporan Harian...");
        await _httpService.get('laporan_harian?tanggal=$date&hari=$day');

        String urlPdf = '${url}assets/pdf/laporan_harian_$day,$date.pdf';
        log.i(urlPdf);

        FileDownloader.downloadFile(
            url: urlPdf,
            // name: "THE FILE NAME AFTER DOWNLOADING", //(optional)
            onProgress: (fileName, progress) {
                // change progress to 0-1
                double progressPercent = progress / 100;

                easyLoading.showProgress(
                    progressPercent,
                    "Downloading: $progress%",
                );
            },
            onDownloadCompleted: (String path) {
                easyLoading.dismissLoading();
                snackbarService.showSnackBar(
                    message: "Laporan Harian Berhasil Tersimpan di $path",
                    duration: const Duration(seconds: 3),
                );
            },
            onDownloadError: (String error) {
                // log.i('DOWNLOAD ERROR: $error');
                snackbarService.showSnackBar(
                    message: "Laporan Harian Gagal Tersimpan: $error",
                    duration: const Duration(seconds: 3),
                );
            }
        );
    } catch (e) {
        log.e(e);
    } finally {
        setBusy(false);
        easyLoading.dismissLoading();
    }
}

```

```

    }
}

goToLaporanTahunan(String yearIncomeOutcome) async {
  // log.i(yearIncomeOutcome);

  try {
    setBusy(true);
    easyLoading.customLoading("Mengunduh Laporan Tahun...");
    await _httpService.get('laporan_tahunan?tahun=$yearIncomeOutcome');

    // if (response.data) {

    String urlPdf = '${url}assets/pdf/laporan_tahunan_$yearIncomeOutcome.pdf';
    log.i(urlPdf);
    FileDownloader.downloadFile(
      url: urlPdf,
      // name: "THE FILE NAME AFTER DOWNLOADING", //(optional)
      onProgress: (fileName, progress) {
        // change progress to 0-1
        double progressPercent = progress / 100;

        easyLoading.showProgress(
          progressPercent,
          "Downloading: $progress%",
        );
      },
      onDownloadCompleted: (String path) {
        easyLoading.dismissLoading();
        snackbarService.showSnackBar(
          message: "Laporan Tahunan Berhasil Tersimpan di $path",
          duration: const Duration(seconds: 3),
        );
      },
      onDownloadError: (String error) {
        // log.i('DOWNLOAD ERROR: $error');
        snackbarService.showSnackBar(
          message: "Laporan Tahunan Gagal Tersimpan: $error",
          duration: const Duration(seconds: 3),
        );
      },
    );
  } catch (e) {
    log.e(e);
  } finally {
    setBusy(false);
    easyLoading.dismissLoading();
  }
}
}

```



FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH PAREPARE

Alamat : Jl. Jend Ahmad Yani Km. 6 Telp. (0421) 22757 Parepare

KARTU MONITOR PEMBIMBINGAN SKRIPSI MAHASISWA

Nama : NEUR FADILLA PUTRI S
NIM : 217 280 113
JUDUL : pengembangan aplikasi dana sosial pada panti asuhan
PROGRAM STUDI : TEKNIK INFORMATIKA
PEMBIMBING I/II : (I) Muli Basti, ST., MT.

No.	TANGGAL BIMBINGAN	Komentar/Saran Perbaikan dari Pembimbing	Paraf (Tuliskan sebagai Pemb. I/II)
1.	13/09/2022	- penulisan kesantunan dan format - Dit Gas II masukkan Gulisan program Ya sudah	
	29/09/2022		

Catatan :

1. Jumlah bimbingan minimal masing-masing 3 kali mulai dari penunjukan pembimbing sampai dengan seminar proposal, Hasil & Skripsi
2. Kartu ini selalu dibawa/disertakan pada saat mahasiswa melakukan bimbingan dengan dosen pembimbing.

Ketua Program Studi
Teknik Informatika

Marlina, S.Kom., M.Kom
NBM. 1162 680



FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH PAREPARE

Alamat : Jl. Jend Ahmad Yani Km. 6 Telp. (0421) 22757 Parepare

KARTU MONITOR PEMBIMBINGAN SKRIPSI MAHASISWA

Nama : NUR FADILLA PUTRI S
NIM : 217 280 113
JUDUL : Aplikasi pengumpulan data pada panti jomang berbasis android
PROGRAM STUDI : TEKNIK INFORMATIKA
PEMBIMBING I/II : //

No.	TANGGAL BIMBINGAN	Komentar/Saran Perbaikan dari Pembimbing	Paraf (Tuliskan sebagai Pemb. I/II)
1	18/4-22	Sebaiknya dikembangkan dengan mengacu pada penelitian di ZIS yang telah	/
2	15/5-22	Pertajam metodologi yg & korelasikan dgn tujuan penelitian	/
3	19/7-22	Pertajam storitas admin & user 1 & rancangan interface. Ace ujes proposal	/

Catatan :

1. Jumlah bimbingan minimal masing-masing 3 kali mulai dari penunjukan pembimbing sampai dengan seminar proposal, Hasil & Skripsi
2. Kartu ini selalu dibawa/disertakan pada saat mahasiswa melakukan bimbingan dengan dosen pembimbing.

Ketua Program Studi
Teknik Informatika


Marlina, S.Kom., M.Kom.
NBM. 1162 680



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

SKRIPSI

Mahasiswa : NUR FADILLA PUTRI. S	Pembimbing I : MUH BASRI ST., MT.
NIM : 217280113	Pembimbing II : AHMAD SELAO, S.TP., M. Sc.
Judul Skripsi : Pengembangan Aplikasi Dana Sosial Pada Pantu Asuhan Berbasis Android	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Isikan data keari mulah anah anah - tambelin foto		Konsultasi 1 Tujuan penelitian & metode & instrumen	
Konsultasi 2		Konsultasi 2 Referensi & jurnal penelitian yg relevan & perbedaan dgn penelitian anda, susunlah & susun konsep & jelaskan sumbernya	
Konsultasi 3		Konsultasi 3 Metode harus mengacu literatur terbaru yg jelas & betul termonit di ran UML	
Konsultasi 4		Konsultasi 4 Hasil penelitian & bahas berdasarkan output pent (tengkapai pengelol pembahasan & hasil output	
Konsultasi 5		Konsultasi 5 Output awal perlu pembahasan lagi!	

Lanjut ke halaman sebelah...

Perhatian :

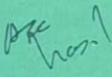
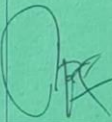
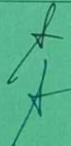
1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton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



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

SKRIPSI

Mahasiswa : NUR FADILLA PUTRI. S	Pembimbing I : MUH BASRI S.T.,M.T.
NIM : 217280113	Pembimbing II : AHMAD SELAO, S.TP.,M. Sc.
Judul Skripsi : Pengembangan Aplikasi Dana Sosial Pada Panti Asuhan Berbasis Android	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 		Konsultasi 1 Pelembagaan antara tugas peneliti & hasil peneliti	
Konsultasi 2		Konsultasi 2 SD	
Konsultasi 3		Konsultasi 3 Aproposurisasi antara hasil software/apli kasi & tugas peneliti secara jelas	
Konsultasi 4		Konsultasi 4 Hasil berupa gambar & skema dijelaskan agar mudah dimengerti	
Konsultasi 5		Konsultasi 5 Abstract & Sistematika	

Perhatian :

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



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

SKRIPSI

Mahasiswa : NUR FADILLA PUTRI. S	Pembimbing I : MUH BASRI ST., MT.
NIM : 217 280 113	Pembimbing II : AHMAD SELAO, S.TP., M.Sc.
Judul Skripsi : PENGEMBANGAN APLIKASI DANA SOSIAL PADA PANTI ASUHAN BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 <i>Aspek tentang dana sosial agar di cantumkan/ditambahkan</i>		Konsultasi 1	
Konsultasi 2 <i>Revisi, bagaimana tentang Manajemen Dana Sosial di panti boleh di kaitkan dengan konsep yg ada</i>		Konsultasi 2	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

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



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

SKRIPSI

Mahasiswa : NUR FADILLA PUTRI. S	Pembimbing I : MUH BASRI ST., MT.
NIM : 217 280 113	Pembimbing II : AHMAD SELAO, S.TP., M.Sc.
Judul Skripsi : PENGEMBANGAN APLIKASI DANA SOSIAL PADA PANTI ASUHAN BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 <i>Aspek tentang dana sosial agar di cantumkan/ditambahkan</i>		Konsultasi 1	
Konsultasi 2 <i>Revisi, bagaimana tentang Manajemen Dana Sosial di Panti boleh dikaitkan dengan konsep yg ada</i>		Konsultasi 2	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

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

