

Lampiran 2: Sk Penelitian Di Puskesmas Negeri Besar

 Bandar Lampung, 05 Juli 2021

Nomor : Penelitian.001/DMJ/DEKAN/BAAK/VII-21
Lampiran : -
Perihal : Permohonan Izin Penelitian

Kepada Yth,
Bapak Suardi Ahmad, S.Kep., MM
Di -
Puskesmas Rawat Inap Negeri Besar
Jl. Saraludin No.198, Tiuh Baru, Kec. Negeri Besar

Dengan hormat,

Sehubungan dengan peraturan Akademik Institut Bisnis dan Informatika (IBI) bahwa mahasiswa/i Strata Satu (S1) yang akan menyelesaikan studinya diwajibkan untuk memiliki pengalaman kerja dengan melaksanakan Penelitian dan membuat laporan yang waktunya disesuaikan dengan kalender Institut Bisnis dan Informatika (IBI) Darmajaya.

Untuk itu kami mohon kerja sama Bapak/Ibu agar kiranya dapat menerima mahasiswa/i untuk melakukan Penelitian, yang pelaksanaannya dimulai dari tanggal **10 Juli 2021 s.d 18 Juli 2021** (selama delapan hari).

Adapun mahasiswa/i tersebut adalah :

Nama	: Diola Sari Utami
NPM	: 1711050061
Jurusan	: S1 Sistem Informasi
Jenjang	: Strata Satu (S1)

Demikian permohonan ini dibuat, atas perhatian dan kerjasama yang baik kami ucapkan terimakasih.

Dekan
Fakultas Ilmu Komputer,

Agus Sunar, S.T., M.Engg
Telp. 06390203



Tembusan:
1. Jurusan S1 Sistem Informasi
2. Arsip

Jalan Z. A. Pagar Alam, No.93, Labuhan Ratu, Bandar Lampung, Lampung
www.darmajaya.ac.id
info@darmajaya.ac.id
0721-787214

Scanned by TapScanner

Lampiran 3: mewawancarai kepala puskesmas dan admin pendaftaran



Lampiran 4: listing program

```
import 'dart:math' as math;
import 'package:hexcolor/hexcolor.dart';
import 'package:puskes_ap/screens/fitness_app/fitness_app_theme.dart';
import 'package:puskes_ap/screens/fitness_app/models/tabIcon_data.dart';
// ignore: unused_import
import 'package:puskes_ap/main.dart';
import 'package:flutter/material.dart';
import 'package:puskes_ap/screens/daftar.dart';

// ignore: unused_import
import '../main.dart';
import '../models/tabIcon_data.dart';

class BottomBarView extends StatefulWidget {
  final int id;

  const BottomBarView(
    {Key? key,
    required this.id,
    this.tabIconsList,
    this.changeIndex,
    this.addClick})
    : super(key: key);

  final Function(int index)? changeIndex;
  final Function()? addClick;
  final List<TabIconData>? tabIconsList;
```

```
@override
// ignore: no_logic_in_create_state
_BottomBarViewState createState() => _BottomBarViewState(id);
}
```

```
class _BottomBarViewState extends State<BottomBarView>
  with TickerProviderStateMixin {
  AnimationController? animationController;
  final int id;
  _BottomBarViewState(this.id);
```

```
@override
void initState() {
  animationController = AnimationController(
    vsync: this,
    duration: const Duration(milliseconds: 1000),
  );
  animationController?.forward();
  super.initState();
}
```

```
@override
Widget build(BuildContext context) {
  return Stack(
    alignment: AlignmentDirectional.bottomCenter,
    children: <Widget>[
      AnimatedBuilder(
```

```

animation: animationController!,
builder: (BuildContext context, Widget? child) {
  return Transform(
    transform: Matrix4.translationValues(0.0, 0.0, 0.0),
    child: PhysicalShape(
      color: FitnessAppTheme.white,
      elevation: 16.0,
      clipper: TabClipper(
        radius: Tween<double>(begin: 0.0, end: 1.0)
          .animate(CurvedAnimation(
            parent: animationController!,
            curve: Curves.fastOutSlowIn))
          .value *
        38.0),
      child: Column(
        children: <Widget>[
          SizedBox(
            height: 62,
            child: Padding(
              padding:
                const EdgeInsets.only(left: 8, right: 8, top: 4),
              child: Row(
                children: <Widget>[
                  Expanded(
                    child: TabIcons(
                      tabIconData: widget.tabIconsList?[0],
                      removeAllSelect: () {

```

```

        setRemoveAllSelection(
            widget.tabIconsList?[0]);
        widget.changeIndex!(0);
    }),
),
Expanded(
    child: TabIcons(
        tabIconData: widget.tabIconsList?[1],
        removeAllSelect: () {
            setRemoveAllSelection(
                widget.tabIconsList?[1]);
            widget.changeIndex!(1);
        }
    ),
),
SizedBox(
    width: Tween<double>(begin: 0.0, end: 1.0)
        .animate(CurvedAnimation(
            parent: animationController!,
            curve: Curves.fastOutSlowIn))
        .value *
    64.0,
),
Expanded(
    child: TabIcons(
        tabIconData: widget.tabIconsList?[2],
        removeAllSelect: () {
            setRemoveAllSelection(

```



```
padding:
  EdgeInsets.only(bottom: MediaQuery.of(context).padding.bottom),
child: SizedBox(
  width: 38 * 2.0,
  height: 38 + 62.0,
  child: Container(
    alignment: Alignment.topCenter,
    color: Colors.transparent,
    child: SizedBox(
      width: 38 * 2.0,
      height: 38 * 2.0,
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ScaleTransition(
          alignment: Alignment.center,
          scale: Tween<double>(begin: 0.0, end: 1.0).animate(
            CurvedAnimation(
              parent: animationController!,
              curve: Curves.fastOutSlowIn)),
        child: Container(
          // alignment: Alignment.center,s
          decoration: BoxDecoration(
            color: FitnessAppTheme.nearlyDarkBlue,
            gradient: LinearGradient(
              colors: [
                FitnessAppTheme.nearlyDarkBlue,
                HexColor('#6A88E5'),
```

```
    ],
    begin: Alignment.topLeft,
    end: Alignment.bottomRight),
  shape: BoxShape.circle,
  boxShadow: <BoxShadow>[
    BoxShadow(
      color: FitnessAppTheme.nearlyDarkBlue
        .withOpacity(0.4),
      offset: const Offset(8.0, 16.0),
      blurRadius: 16.0),
  ],
),
child: Material(
  color: Colors.transparent,
  child: InkWell(
    splashColor: Colors.white.withOpacity(0.1),
    highlightColor: Colors.transparent,
    focusColor: Colors.transparent,
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => Daftar(id: id)));
    },
  child: const Icon(
    Icons.add,
    color: FitnessAppTheme.white,
```



```

class TabIcons extends StatefulWidget {
  const TabIcons({Key? key, this.tabIconData, this.removeAllSelect})
    : super(key: key);

  final TabIconData? tabIconData;
  final Function()? removeAllSelect;
  @override
  _TabIconsState createState() => _TabIconsState();
}

class _TabIconsState extends State<TabIcons> with TickerProviderStateMixin {
  @override
  void initState() {
    widget.tabIconData?.animationController = AnimationController(
      vsync: this,
      duration: const Duration(milliseconds: 400),
    )..addListener((AnimationStatus status) {
      if (status == AnimationStatus.completed) {
        if (!mounted) return;
        widget.removeAllSelect!();
        widget.tabIconData?.animationController?.reverse();
      }
    });
    super.initState();
  }
}

```

```

void setAnimation() {
  widget.tabIconData?.animationController?.forward();
}

@override
Widget build(BuildContext context) {
  return AspectRatio(
    aspectRatio: 1,
    child: Center(
      child: InkWell(
        splashColor: Colors.transparent,
        focusColor: Colors.transparent,
        highlightColor: Colors.transparent,
        hoverColor: Colors.transparent,
        onTap: () {
          if (!widget.tabIconData!.isSelected) {
            setAnimation();
          }
        },
        child: IgnorePointer(
          child: Stack(
            alignment: AlignmentDirectional.center,
            children: <Widget>[
              ScaleTransition(
                alignment: Alignment.center,
                scale: Tween<double>(begin: 0.88, end: 1.0).animate(
                  CurvedAnimation(

```

```
        parent: widget.tabIconData!.animationController!,
        curve: const Interval(0.1, 1.0,
            curve: Curves.fastOutSlowIn))),
    child: Image.asset(widget.tabIconData!.isSelected
        ? widget.tabIconData!.selectedImagePath
        : widget.tabIconData!.imagePath),
),
Positioned(
    top: 4,
    left: 6,
    right: 0,
    child: ScaleTransition(
        alignment: Alignment.center,
        scale: Tween<double>(begin: 0.0, end: 1.0).animate(
            CurvedAnimation(
                parent: widget.tabIconData!.animationController!,
                curve: const Interval(0.2, 1.0,
                    curve: Curves.fastOutSlowIn))),
    child: Container(
        width: 8,
        height: 8,
        decoration: const BoxDecoration(
            color: FitnessAppTheme.nearlyDarkBlue,
            shape: BoxShape.circle,
        ),
    ),
),
),
```



```

        alignment: Alignment.center,
        scale: Tween<double>(begin: 0.0, end: 1.0).animate(
            CurvedAnimation(
                parent: widget.tabIconData!.animationController!,
                curve: const Interval(0.5, 0.6,
                    curve: Curves.fastOutSlowIn))),
        child: Container(
            width: 6,
            height: 6,
            decoration: const BoxDecoration(
                color: FitnessAppTheme.nearlyDarkBlue,
                shape: BoxShape.circle,
            ),
        ),
    ),
),
],
),
),
),
),
),
);
}
}

```

```

class TabClipper extends CustomClipper<Path> {
  TabClipper({this.radius = 38.0});
}

```

```
final double radius;
```

```
@override
```

```
Path getClip(Size size) {
```

```
    final Path path = Path();
```

```
    final double v = radius * 2;
```

```
    path.lineTo(0, 0);
```

```
    path.arcTo(Rect.fromLTWH(0, 0, radius, radius), degreeToRadians(180),
```

```
        degreeToRadians(90), false);
```

```
    path.arcTo(
```

```
        Rect.fromLTWH(
```

```
            ((size.width / 2) - v / 2) - radius + v * 0.04, 0, radius, radius),
```

```
        degreeToRadians(270),
```

```
        degreeToRadians(70),
```

```
        false);
```

```
    path.arcTo(Rect.fromLTWH((size.width / 2) - v / 2, -v / 2, v, v),
```

```
        degreeToRadians(160), degreeToRadians(-140), false);
```

```
    path.arcTo(
```

```
        Rect.fromLTWH((size.width - ((size.width / 2) - v / 2)) - v * 0.04, 0,
```

```
            radius, radius),
```

```
        degreeToRadians(200),
```

```
        degreeToRadians(70),
```

```
        false);
```

```
path.arcTo(Rect.fromLTWH(size.width - radius, 0, radius, radius),
    degreeToRadians(270), degreeToRadians(90), false);
path.lineTo(size.width, 0);
path.lineTo(size.width, size.height);
path.lineTo(0, size.height);

path.close();
return path;
}

@Override
bool shouldReclip(TabClipper oldClipper) => true;

double degreeToRadians(double degree) {
    final double radian = (math.pi / 180) * degree;
    return radian;
}
}
```