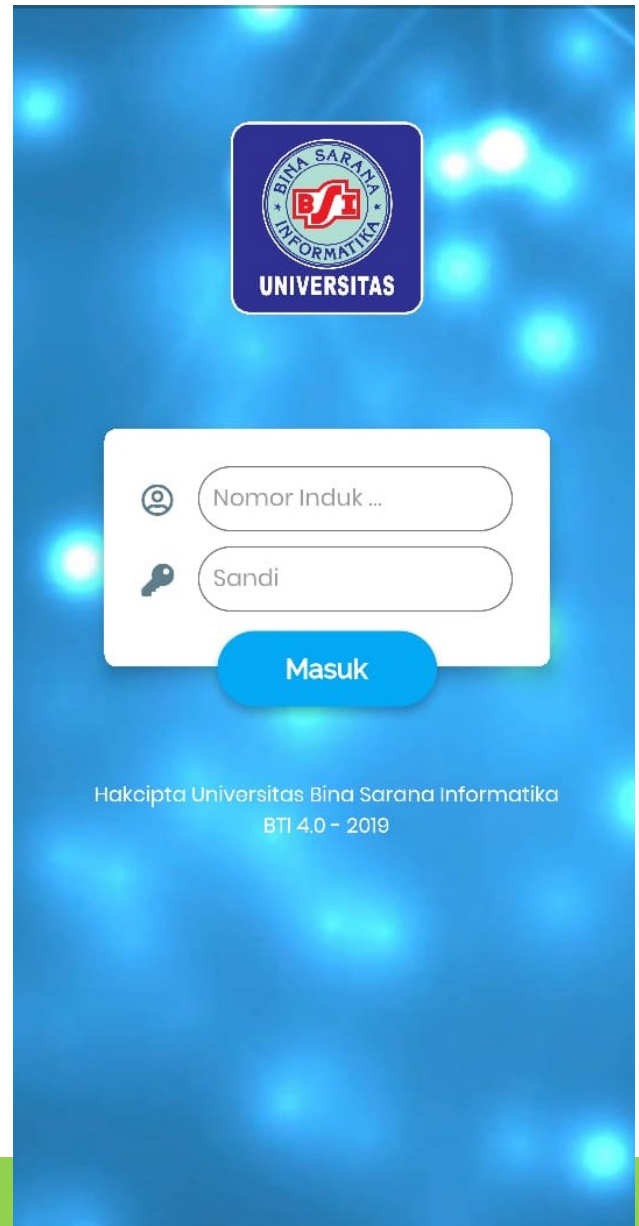




PANDUAN PENGGUNAAN APLIKASI MYUBSI STUDENTS BERBASIS MOBILE UNIVERSITAS BINA SARANA INFORMATIKA

Panduan Penggunaan

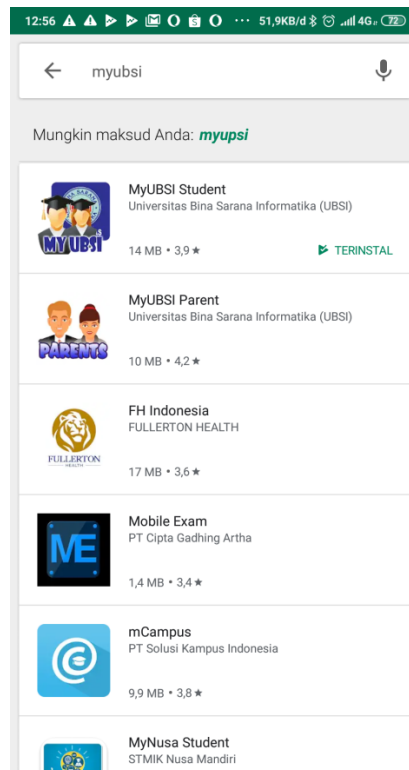
Pengantar : Memulai My UBSI Student

MyUBSI Student merupakan aplikasi yang berbasis android yang bermuat basis data untuk mahasiswa Universitas BSI yang bertujuan untuk mempermudah mahasiswa mendapatkan informasi perkuliahan didalam *handphone* mereka. Aplikasi MyUBSI Student ini dibangun dengan beberapa tujuan utama antara lain:

1. Memudahkan mahasiswa untuk dapat menerima informasi terkini di universitas BSI
2. Mempermudah mahasiswa untuk melihat berbagai kegiatan perkuliahan di Universitas BSI
3. Mempermudah mahasiswa dalam mengakses informasi nilai dan perkuliahan

Mengakses My UBSI Student

My UBSI Student atau MyUBSI Student UBSI adalah aplikasi yang berbasis android yang dapat diakses dengan menggunakan Handphone berbasis android serta terhubung dengan internet serta aplikasi ini bias di cari pada Play store dengan nama pencarian MyUBSI Student UBSI

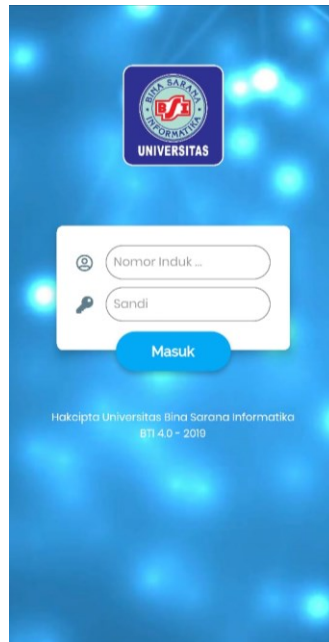


Gambar 1. Aplikasi MyUBSI Student di Google Playstore

Pada tahapan awal sebelum menggunakan MyUBSI Student, diperlukan mengdownload dan menginstall aplikasi MyUBSI Student melalui play store pada operating system Android. Pastikan instalasi dan aplikasi sudah terpasang di handphone pengguna.

Halaman Depan MyUBSI Student

Halaman depan ini digunakan untuk membatasi akses maupun memverifikasi pengguna atau mahasiswa Universitas Bina Sarana Informatika, dimana pada halaman depan MyUBSI Student pengguna wajib menginputkan Username dan Password.

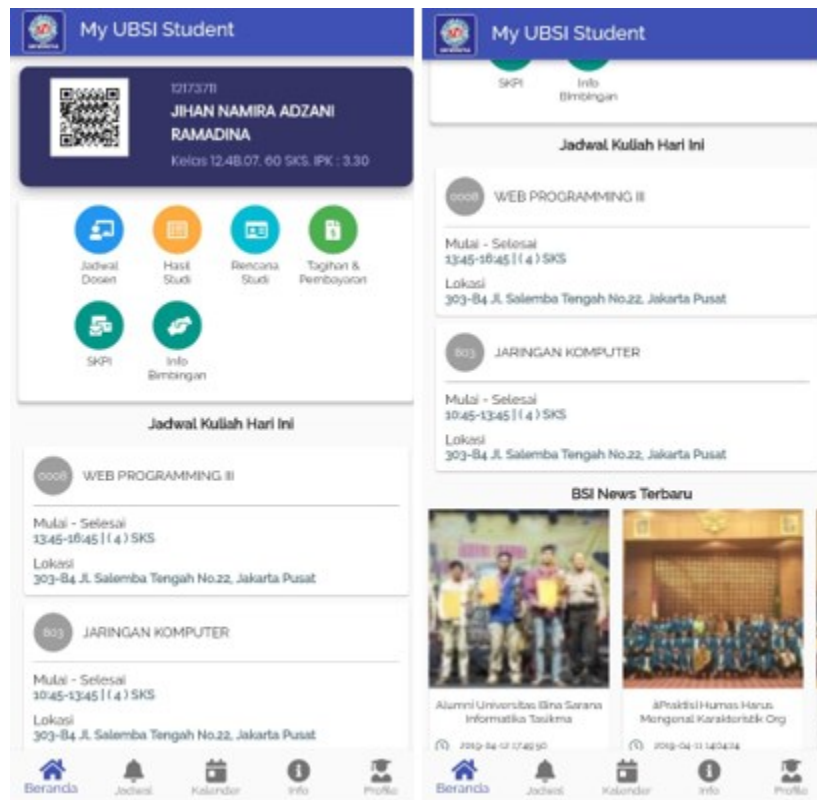


Gambar 2 Halaman Login MyUBSI Student

1. Pada bagian ini nomer induk di inputkan Nomer Induk Mahasiswa (NIM) yang telah diberikan dan pada kolom sandi diinputkan password NIM tersebut.
2. Pada tombol Masuk di klik jika Nomer Induk dan Password telah di inputkan dengan benar kemudian klik tombol **masuk**.

Halaman Menu Utama MyUBSI Student

Pada saat mahasiswa telah login pada aplikasi MyUBSI Student UBSI maka mahasiswa tersebut akan dirujuk ke halaman utama dari aplikasi tersebut. Berikut ini penjelsan tentang halaman utama MyUBSI Student:



Gambar 3. Halaman Utaman MyUBSI Student

Pada halaman utama Terdapat beberapa navigasi diantaranya profile mahasiswa yang berisikan NIM, Nama Mahasiswa, Kelas, SKS yang telah diambil dan IPK yang diperoleh sampai saat ini. Selain itu ada beberapa navigasi yang bisa digunakan antara lain Jadwal Dosen, Hasil Studi, Rencana Studi, Tagihan dan Pembayaran, SKPI dan info Bimbingan

Pada halaman utama juga terdapat informasi jadwal kuliah yang berlangsung pada hari ini atau ketika mahasiswa membuka aplikasi MyUBSI Student maka mahsiswa tersebut langsung mengetahui Jadwal kuliahhari ini akan ditayangkan pada halaman utama aplikasi ini.

Selain informasi jadwal kuliah hari ini. Pada halaman utama terdapat informasi kegiatan yang sedang berlangsung di Universitas Bina Sarana Informatika. Dengan nama NSI News

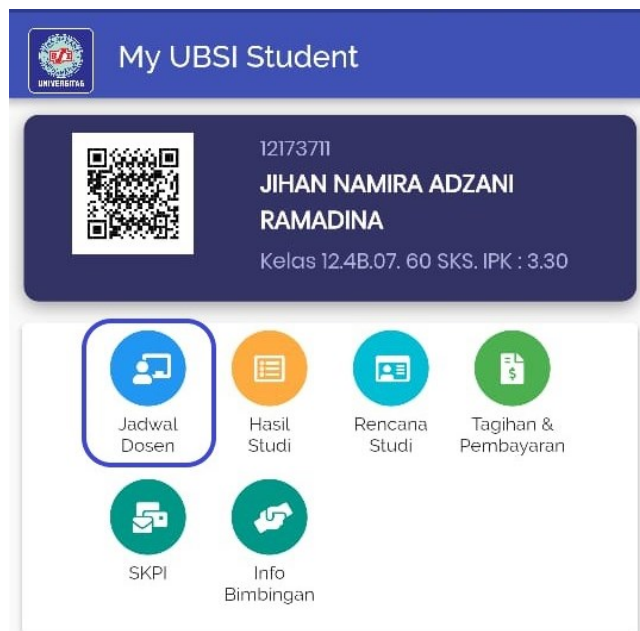
terbaru. Yang bertujuan agar mahasiswa dapat mengetahui informasi kegiatan yang ada dikampus.

Pada panel bawah terdapat navigasi ke halaman – halaman lainnya. Navigasi tersebut antara lain: Beranda, Jadwal (lihat Jadwal Perkuliahan berlangsung), Kalender (lihat Kalender Akademik), Info (informasi Akademik) dan Profile (lihat Profile Mahasiswa).

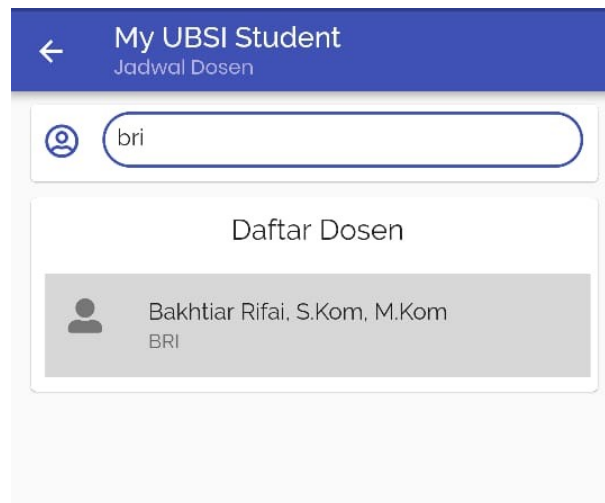
Menggunakan MyUBSI Student

1. Menu Jadwal Dosen

Setiap mahasiswa di Universitas Bina Sarana Informatika dapat menggunakan menu jadwal dosen untuk mencari nama dosen yang dituju dengan memasukan initial kode dosen atau Nama Dosen pada menu cari dan akan menampilkan nama dosen dan jadwal mengajar dosen



Gambar 4. Jadwal Dosen



Gambar 5. Fitur Cari Dosen

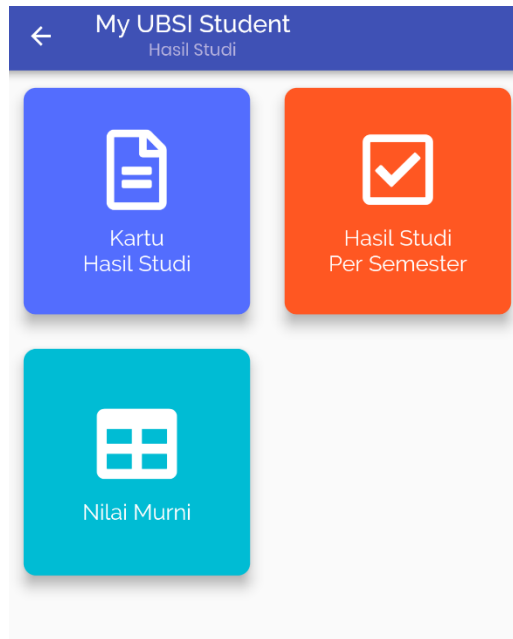
2. Hasil Studi

Mahasiswa dapat melihat Kartu Hasil Studi (KHS), Hasil Studi Per Semester dan Nilai Murni



Gambar 6. Hasil Studi

Ketika mahasiswa menekan button Hasil studi didalamnya akan menampilkan 3 menu antara lain menu Kartu Hasil Studi, Hasil Studi Per Semester, Nilai Murni



Gambar 6. Tampilan Hasil Studi

← My UBSI Student Hasil Studi Per Semester	20:09 9,5KB/4 3G 4G ← My UBSI Student Nilai Murni Per Semester
Semester 1	Semester 1
A 101 .- PENDIDIKAN PANCASILA 2 SKS B 104 .- BAHASA INGGRIS I 2 SKS A 153 .- PENGANTAR TEKNOLOGI INFORMASI DAN KOMUNIKASI 3 SKS B 207 .- LOGIKA & ALGORITMA 4 SKS A 712 .- ENTREPRENEURSHIP 3 SKS A 894 .- DASAR PEMROGRAMAN 4 SKS	57.0 101 .- PENDIDIKAN PANCASILA. (2 SKS) TGS 0.0 PRS 0.0 UTS 0.0 UAS 0.0 70.0 104 .- BAHASA INGGRIS I. (2 SKS) TGS 73.0 PRS 92.0 UTS 57.0 UAS 63.0 91.0 153 .- PENGANTAR TEKNOLOGI INFORMASI DAN KOMUNIKASI. (3 SKS) TGS 98.0 PRS 100.0 UTS 97.0 UAS 73.0 75.0 207 .- LOGIKA & ALGORITMA. (4 SKS) TGS 95.0 PRS 100.0 UTS 37.0 UAS 73.0 84.0 712 .- ENTREPRENEURSHIP. (3 SKS) TGS 90.0 PRS 86.0 UTS 57.0 UAS 100.0 82.0 894 .- DASAR PEMROGRAMAN . (4 SKS) TGS 0.0 PRS 0.0 UTS 0.0 UAS 0.0
6 Matakuliah. 18 SKS. IP = 3.67	6 Matakuliah. 18 SKS. IP = 3.67

Gambar 6. Tampilan Hasil Studi Per Semester & Nilai Murni Persemester

3. Rencana Studi

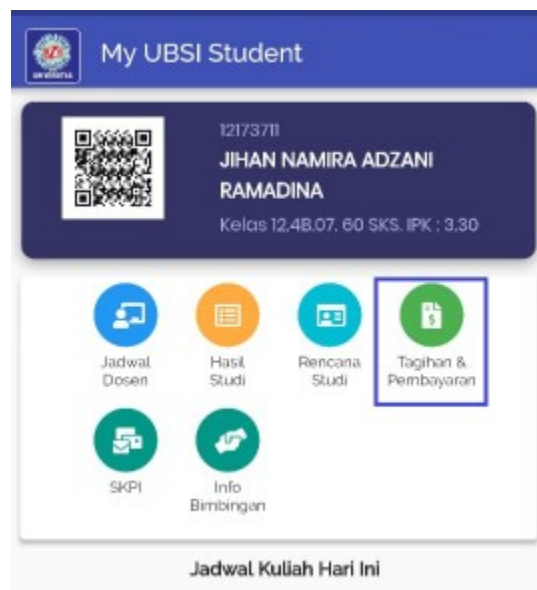
Mahasiswa juga dapat mendapatkan informasi Rencana Studi perkuliahan berlangsung dan semester sebelumnya



Gambar 7. Rencana Studi

4. Tagihan & Pembayaran

Mahasiswa dapat memperoleh status tagihan perkuliahan, history pembayaran yang telah dilakukan maupun pembayaran lainnya yang ditujukan untuk biaya kegiatan selama perkuliahan secara detail dan terperinci



My UBSI Student Tagihan dan Pembayaran			My UBSI Student Tagihan dan Pembayaran			My UBSI Student Tagihan dan Pembayaran		
Tagihan	Pembayaran Kuliah	Pembayaran Lainnya	Tagihan	Pembayaran Kuliah	Pembayaran Lainnya	Tagihan	Pembayaran Kuliah	Pembayaran Lainnya
Biaya Kuliah Semester 5 Kuliah : 1.980.000 Tagihan : 1.980.000			Agustus 2018 Biaya Kuliah Semester 4 Tanggal : Tue, 21-Aug-2018 11:43:30 12.2B.07 BAKU oleh BAKU (Untuk Beasiswa) SSP : 0 Kuliah : 1.980.000 Total : 1.980.000			Januari 2019 Biaya Her Semester 3 Biaya Rp. 50.000 Penyctor INDOMARET, JIHAN NAMIRA ADZANI RAMADINA Waktu 2019-01-31, 15:11:25 Tagihan : 50.000		
Biaya Kuliah Semester 6 Kuliah : 1.980.000 Tagihan : 1.980.000			Juli 2018 Biaya Kuliah Semester 3 Tanggal : Tue, 31-Jul-2018 15:19:54 12.2B.07 Zo5O1 oleh SULASMIATI SSP : 1.000.000 Kuliah : 1.980.000 Total : 2.980.000			Oktober 2018 null Biaya Rp. 50.000 Penyctor H2H, JIHAN NAMIRA ADZANI RAMADINA Waktu 2018-10-16, 14:37:06 Tagihan : 50.000		
			Januari 2018 Biaya Kuliah Semester 2 Tanggal : Tue, 30-Jan-2018 12:16:49 12.2B.07 434K1 oleh SULASMIATI SSP : 1.000.000			April 2018 Seminar Created a Good Database for Your Application (AMIK Jakarta) Biaya Rp. 50.000 Penyctor JIHAN NAMIRA ADZANI RAMADINA		
Total Tagihan IDR. 3.960.000			Total IDR. 9.920.000			Total Pembayaran IDR. 390.000		

Gambar 8. Tagihan & Pembayaran

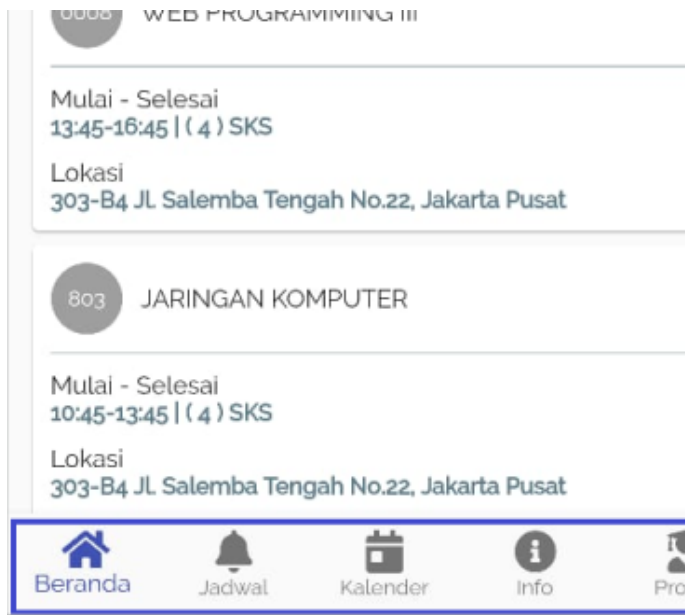
5. SKPI (Surat Keterangan Pendamping Ijazah)

SKPI adalah surat keterangan pendamping ijazah yang diperlukan oleh mahasiswa dan mahasiswa mengupload untuk sebagai kompetensi mahasiswa dan sebagai pendamping ijazah



6. Info Bimbingan

Info bimbingan akan menampilkan nama dosen pembimbing baik pembimbing 1 dan asisten pembimbing yang akan diperoleh jika mahasiswa telah berhak bimbingan dan akan menampilkan informasi outline, kelompok dan informasi bimbingan yang di infokan baik pembimbing utama ataupun asisten pembimbing.

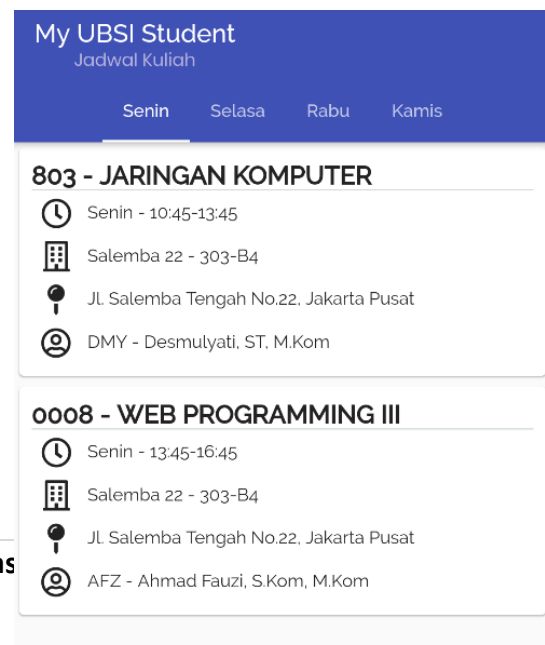


Selain menu icon diatas aja juga menu bar yang terletak dibagian bawah yang berisikan Beranda, Jadwal Akademik, Kalender Akademik, Info akademik dan Profile Mahasiswa.

1.

2. Jadwal Akademik

Jadwal Akademik ini berisikan jadwal kuliah mahasiswa yang bersangkutan sesuai dengan hari dan matakuliah yang diperoleh.



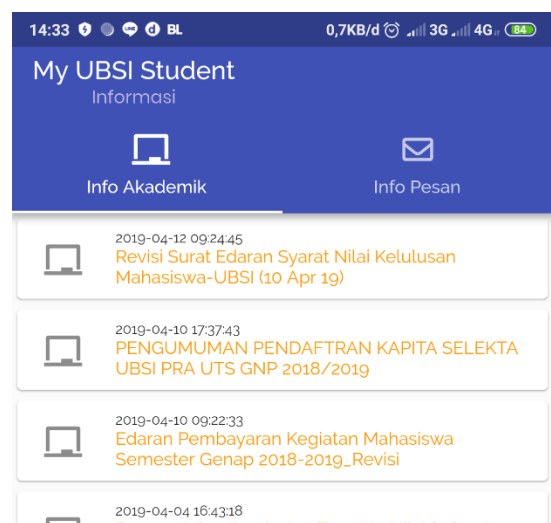
3. Kalender Akademik

Kalender akademik ini menampilkan informasi kegiatan akademik yang berlangsung selama perkuliahan, bias di lihat berdasarkan bulan.



4. Info Akademik

info Akademik menampilkan informasi kegiatan akademik yang diinfokan oleh Rektor, Wakil Rektor bidang akademik maupun kemahasiswaan, serta memberikan informasi terkini.



5. Profile Mahasiswa

profile mahasiswa menampilkan data mahasiswa universitas bina sarana informatika, kelas dna IPK, alamat, email dan data pribadi mahasiswa.

The screenshot displays the 'My UBSI Student Profile' page. At the top, the status bar shows the time 14:33, signal strength, and battery level at 64%. The page title is 'My UBSI Student Profile'. Below this, the user's name 'JIHAN NAMIRA ADZANI RAMADINA' and ID '12173711' are shown next to a placeholder profile picture. Academic details include 'Kelas 12.4B.07' and 'IPK 3.30'. A section for personal information lists 'Tempat Lahir' (Jakarta), 'Tanggal Lahir' (2000-02-06), 'Jenis Kelamin' (Perempuan), and 'Agama' (Islam). Address details include 'Alamat' (jln.budi mulia 4 pademangan barat jakarta utara), 'RT/RW' (014/014), 'Kota' (Kota Jakarta Utara), and 'Kode Pos' (14420). Contact information shows 'Telepon' as '-' and 'E-Mail' as 'jhnnmr06@gmail.com'. The 'Kampus' is listed as 'Salemba 22'. At the bottom, a navigation bar contains icons for 'Beranda', 'Jadwal', 'Kalender', 'Info', and 'Profile'.

JIHAN NAMIRA ADZANI RAMADINA 12173711	
Kelas 12.4B.07	IPK 3.30
Tempat Lahir	Jakarta
Tanggal Lahir	2000-02-06
Jenis Kelamin	Perempuan
Agama	Islam
Alamat	jln.budi mulia 4 pademangan barat jakarta utara
RT/RW	014/014
Kota	Kota Jakarta Utara
Kode Pos	14420
Telepon	-
E-Mail	jhnnmr06@gmail.com
Kampus	Salemba 22

Navigation Bar:

- Beranda
- Jadwal
- Kalender
- Info
- Profile

Source Code

```
import 'package:flutter/material.dart';
import 'utils/assets.dart';
import './page/login_page.dart';
import './page/main_page.dart';
import './page/splash_page.dart';
import './page/webview_page.dart';
import './page/pagemenu/cari_jadwal_dosen.dart';
import './page/pagemenu/hasil_study.dart';
import './page/pagemenu/nilaimurni.dart';
import './page/pagemenu/study_per_semester.dart';
import './page/pagemenu/khs.dart';
import './page/pagemenu/rencana_study.dart';
import './page/pagemenu/tagihan.dart';
import './page/pagemenu/kalender.dart';
import './page/walkthrough/walkthrough_main.dart';
import './page/subjadwal/materi_page.dart';
import 'package:background_fetch/background_fetch.dart';
import 'package:flutter/services.dart';
import './helper/alarmmanager.dart';

const int APP_MODE = 1; //1 : UBSI, 2:STMIK
//to compile flutter build apk --release --target-platform android-arm64
//build
void main() {
  runApp(MyApp());

  setup();
}

Future setup() async {
  print('backgroundfetch configure');
  AlarmManager(durasi: Duration(minutes: 1), callback: wakeup, funcName:
'wakeup' ).initialize();

  /* int i = await BackgroundFetch.configure(BackgroundFetchConfig(
    minimumFetchInterval: 1,
    stopOnTerminate: false,
    enableHeadless: true
  ), wakeup);
  print('background result $i'); */
```

```

}

Future wakeup() async {
  print('wakeup ${DateTime.now()}');
  try {
    // BackgroundFetch.finish(BackgroundFetch.FETCH_RESULT_NEW_DATA);
  } on PlatformException catch(eo) {} catch(e) { print('$e'); }
}

class MyApp extends StatelessWidget {
  Route<dynamic> _routes(RouteSettings settings) {
    switch(settings.name) {
      case MainPage.tag: return MyMaterialPageRoute(builder: (_) => new
MainPage(), settings: settings);
      case SplashScreenPage.tag: return MyMaterialPageRoute(builder:
(_) => new SplashScreenPage(), settings: settings);
      case WalkthroughMainPage.tag: return MyMaterialPageRoute(builder:
(_) => new WalkthroughMainPage(), settings: settings);
      case LoginPage.tag: return MyMaterialPageRoute(builder: (_) => new
LoginPage(), settings: settings);
      case WebView.tag: return MyMaterialPageRoute(builder: (_) => new
WebView(), settings: settings);
      case CariJadwalDosen.tag: return MyMaterialPageRoute(builder:
(_) => new CariJadwalDosen(), settings: settings);
      case HasilStudy.tag: return MyMaterialPageRoute(builder: (_) => new
HasilStudy(), settings: settings);
      case KHS.tag: return MyMaterialPageRoute(builder: (_) => new KHS(),
settings: settings);
      case HasilStudySemester.tag: return MyMaterialPageRoute(builder:
(_) => new HasilStudySemester(), settings: settings);
      case NilaiMurni.tag: return MyMaterialPageRoute(builder: (_) => new
NilaiMurni(), settings: settings);
      case RencanaStudy.tag: return MyMaterialPageRoute(builder: (_) => new
RencanaStudy(), settings: settings);
      case Tagihan.tag: return MyMaterialPageRoute(builder: (_) => new
Tagihan(), settings: settings);
      case KalenderAkademik.tag: return MyMaterialPageRoute(builder:
(_) => new KalenderAkademik(), settings: settings);
      case MateriPage.tag: return MyMaterialPageRoute(builder: (_) => new
MateriPage(), settings: settings);
    }
    return null;
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: Res.APP_TITLE,
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        // primarySwatch: Colors.indigo,
        primarySwatch: APP_MODE == 1 ? Colors.indigo : Colors.brown,
        fontFamily: 'Raleway',
      ),
    ),
  }
}

```

```

        home: new SplashScreenPage(),
        onGenerateRoute: _routes ,
    );
}
}

class MyMaterialPageRoute<T> extends MaterialPageRoute<T> {

  MyMaterialPageRoute({WidgetBuilder builder, RouteSettings settings})
    : super(builder: builder, settings: settings);

  @override
  Widget buildTransitions(BuildContext context, Animation<double> animation,
    Animation<double> secondaryAnimation, Widget child) {
    if (settings.isInitialRoute) return child;
    // return new RotationTransition(turns: secondaryAnimation, child: child);
    // return new FadeTransition(opacity: animation, child: child);
    return new SlideTransition(
      position: new Tween<Offset>(
        begin: const Offset(1.0, 0.0),
        end: Offset.zero,
      ).animate(animation),
      child: FadeTransition(opacity: animation, child: child),
    );
  }
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/utills/assets.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class AkademikPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      body: AkademikPageBody(),
    );
  }
}

class AkademikPageBody extends StatefulWidget {
  @override
  _AkademikPageBodyState createState() => _AkademikPageBodyState();
}

class _AkademikPageBodyState extends State<AkademikPageBody> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("${Res.APP_TITLE}"),
        leading: Scaffold(
          backgroundColor: Colors.indigo,
          body: Padding(
            padding: EdgeInsets.fromLTRB(13.0, 7.0, 0.0, 0.0),

```

```

        child: Image(
          image: AssetImage("assets/img/logo.png"),
          width: 43.0,
          height: 43.0,
        ),
      ),
    ),
  ),
  body: GridView.count(
    crossAxisCount: 2,
    children: <Widget>[
      CardBoardIcon(
        icon: FontAwesomeIcons.envelopeOpen,
        color: Colors.blue,
        title: "Surat\nPengantar Riset",
      ),
      CardBoardIcon(
        icon: FontAwesomeIcons.envelope,
        color: Color(0xFF965f12),
        title: "Surat\nKeterangan",
      ),
      CardBoardIcon(
        icon: Icons.touch_app,
        color: Color(0xFF885F7F),
        title: "Pengajuan\nCut",
      ),
      CardBoardIcon(
        icon: Icons.payment,
        color: Color(0xFF29700f),
        title: "Pembayaran\n",
      ),
      CardBoardIcon(
        icon: FontAwesomeIcons.bookOpen,
        color: Color(0xFF20375b),
        title: "Tugas Akhir /\n Skripsi",
      ),
      CardBoardIcon(
        icon: FontAwesomeIcons.userGraduate,
        color: Color(0xFF58307a),
        title: "Wisuda",
      ),
    ],
  ),
);
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/assets.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/utills/account_card.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import './pagemenu/cari_jadwal_dosen.dart';
import './pagemenu/hasil_study.dart';
import './pagemenu/rencana_study.dart';
import './pagemenu/tagihan.dart';

```

```

import './pagemenu/kalender.dart';
import '../helper/http_api.dart';
import '../helper/urladdress.dart';
import '../model/pengaturan.dart';
import './home/news_view.dart';
import './home/jdwkuliaah_today_view.dart';
import './skpi/skpi_page.dart';
import './bimbingan/bimbingan_page.dart';
import './bimbingan/infobimbingan_panel.dart';
import 'dart:convert';

class HomePage extends StatefulWidget {
  final Widget child;

  HomePage({Key key, this.child}) : super(key: key);

  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends StateExt<HomePage> {

  BuildContext context;
  Size deviceSize;
  State state;
  List listNews;

  Widget menuCard() => Padding(
    padding: const EdgeInsets.symmetric(horizontal: 5.0),
    child: Card(
      elevation: 3.0,
      child: Padding(
        padding: const EdgeInsets.all(5.0),
        child: Center(
          child: _itemsMenuAlt() ,
        ),
      ),
    ),
  );

  _HomePageState(){
    listNews = new List();
  }

  void initState() {
    super.initState();
  }

  Widget _itemsMenuAlt(){
    List<Widget> itemsMenu = <Widget>[
      LabelIcon(icon: FontAwesomeIcons.chalkboardTeacher,
        label: 'Jadwal\nDosen',
        onPressed: () {
Navigator.of(context).pushNamed(CariJadwalDosen.tag);
        },

```

```

        circleColor: Colors.blue ,),

LabelIcon(icon: FontAwesomeIcons.listAlt,
  label: 'Hasil\nStudi',
  onPressed: (){
    Navigator.of(context).pushNamed(HasilStudy.tag);
  },
  circleColor: Colors.orangeAccent ,),

LabelIcon(icon: FontAwesomeIcons.idCard,
  label: 'Rencana\nStudi',
  circleColor: Colors.cyan,
  onPressed: (){
    Navigator.of(context).pushNamed(RencanaStudy.tag);
  },
),

/*      LabelIcon(icon: FontAwesomeIcons.clipboardCheck,
  label: 'Daftar\nKegiatan',
  circleColor: Colors.redAccent,
  onPressed: (){
    info(context, 'Daftar Kegiatan', 'Akan tersedia
selanjutnya');
  },
),*/

LabelIcon(icon: FontAwesomeIcons.fileInvoiceDollar,
  label: 'Tagihan &\nPembayaran',
  circleColor: Colors.green,
  onPressed: (){
    Navigator.of(context).pushNamed(Tagihan.tag);
  },
),

/*      LabelIcon(icon: FontAwesomeIcons.calendarAlt,
  label: 'Kalender',
  circleColor: Colors.brown,
  onPressed: (){

Navigator.of(context).pushNamed(KalenderAkademik.tag);
  },
),*/

LabelIcon(icon: FontAwesomeIcons.mailBulk,
  label: 'SKPI',
  circleColor: Colors.teal,
  onPressed: (){
    Navigator.of(context).push(new MaterialPageRoute(
      builder: (bc)=>SKPIPage()
    ));
  },
),

LabelIcon(icon: FontAwesomeIcons.handsHelping,
  label: 'Info\nBimbingan',
  circleColor: Colors.teal , onPressed: (){

```

```

        Navigator.of(context).push( new MaterialPageRoute(
            builder: (bc)=>BimbinganPage()
        ));

    },)

];

return Wrap(
    children: itemsMenu
);
}

Widget allCards(BuildContext context) => SingleChildScrollView(
    child: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        children:<Widget>[
            AccountCard(),
            menuCard(),
            SizedBox(height: 10.0),

            InfoBimbinganPanel(),

            Padding(
                padding: EdgeInsets.only(left: 10.0),
                child:Text('Jadwal Kuliah Hari Ini', style:
TextStyle(fontWeight: FontWeight.bold, fontSize: 15.0)),
            ),

            JadwalKuliahTodayView(),

            Padding(
                padding: EdgeInsets.only(left: 10.0),
                child:Text('BSI News Terbaru', style:
TextStyle(fontWeight: FontWeight.bold, fontSize: 15.0)),
            ),

            SizedBox(height: 270.0, child: NewsView()),

        ])
    );

Widget appBar(){
    return AppBar(
        title: new Text(Res.APP_TITLE,),
        leading: Padding(
            padding: EdgeInsets.fromLTRB(13.0, 7.0, 0.0, 0.0),
            child: Assets.logo
        ),
    );
}

@override

```

```

Widget build(BuildContext context) {
  this.context = context;
  deviceSize = MediaQuery.of(context).size;

  return Scaffold(
    appBar: appBar(),
    body: allCards(context)
  );
}

}

import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubs/helper/urladdress.dart';
import 'package:m_student_ubs/model/infoakademik.dart';
import 'package:m_student_ubs/utills/mywidget.dart';
import 'package:m_student_ubs/utills/assets.dart';
import 'package:m_student_ubs/page/webview_page.dart';
import 'package:m_student_ubs/load_online/sync_db.dart';

class InfoPageBody extends StatefulWidget {
  @override
  _InfoPageBodyState createState() => _InfoPageBodyState();
}

class _InfoPageBodyState extends State<InfoPageBody>
  with SingleTickerProviderStateMixin {

  TabController tabController;
  final CListView infoAkad = new CListView();
  final CListView infoPesan = new CListView();

  _InfoPageBodyState() {

    infoAkad.onRefresh = () {
      infoAkad.isLoading = true;
      return refreshInfoAkademik();
    };

    infoAkad.onLoadMore = () {
      _pageIA++;
      refreshInfoAkademik();
    };

    infoAkad.itemBuilder = (BuildContext c, int index){
      return item(index);
    };
  }

  @override
  void initState() {
    tabController = new TabController(vsync: this, length: 2);
    super.initState();
    infoAkad.isLoading = true;
  }
}

```

```

        refreshInfoAkademik();
    }

    @override
    void dispose() {
        tabController.dispose();
        super.dispose();
    }

    @override
    Widget build(BuildContext context) {

        return new Scaffold(
            appBar: new AppBar(
                title: TitleActionbar(subTitle: 'Informasi'),
                bottom: new TabBar(
                    controller: tabController,
                    tabs: <Widget>[
                        new Tab(icon: new Icon(FontAwesomeIcons.chalkboard), text: "Info
Akademik",),
                        new Tab(icon: new Icon(FontAwesomeIcons.envelope),text: "Info
Pesan",),
                    ],
                ),
            ),
            body: new TabBarView(
                controller: tabController,
                children: <Widget>[
                    infoAkad.build(context),
                    infoPesan.build(context),
                ],
            ),
        );
    }

    int _pageIA = 1;
    Future refreshInfoAkademik() async{
        print('refresh');
        setState(() { });
        int _limit = 50;
        int off = (_pageIA - 1) * _limit;

        List list = await (new InfoakademikModel()).read(limit: _limit, offset:
off);
        if( list != null){ infoAkad.addItem(list); }else{ list = new List();}

        if(list.length == 0){ await loadInfoAkademikServer(); }

        list = await (new InfoakademikModel()).read(limit: _limit, offset:
off);
        if( list != null){ infoAkad.addItem(list); }

        setState(() { infoAkad.isLoading = false; });
        infoAkad.isLoadMore = false;
    }
}

```

```

Future<void> loadInfoAkademikServer() async{
  int add = await Sync.infoAkademik(_pageIA);

  if(add == 0 && infoAkad.isLoadMore){
    _pageIA--;
  }
}

Widget item(int index){
  InfoakademikModel iam =
  InfoakademikModel().fromMap(infoAkad.data[index]);

  return Card( key: Key(index.toString()),
    child: InkWell(
      onTap: (){
        InfoakademikModel().update('id=?', [ iam.id ],
data: { InfoakademikModel.COL_HAS_READ : 1 } );
        // setState(() {
infoAkad.data[index]['has_read'] = 1; });

        Navigator.of(context).push(new
MaterialPageRoute(builder: (_) {
          return WebView(settings: {
            'title': iam.judul,
            'url': UrlAddr.URL_GOOGLEVIEWER +
Uri.decodeFull('${iam.file}') + '&embedded=true'
          });
        }));
      },
      child: new Container(
padding: EdgeInsets.all(5.0),
child: new Column(
  children: <Widget>[
    new ListTile(
      leading: InkWell(
        child: Icon( iam.is_favorite == 1 ?
FontAwesomeIcons.star : FontAwesomeIcons.chalkboard),
      ),
      title: new Text("${iam.tgl}", style:
        TextStyle(fontSize: 11.0),),
      subtitle: new Text("${iam.judul}",
        style: TextStyle(fontSize:
14.0,
        color: iam.has_read
== 1 ? Colors.black : Colors.orange))
    ),
  ],
),
),
);
}

}

import 'package:flutter/material.dart';

```

```

import 'package:flutter/cupertino.dart';
import 'package:m_student_ubs/helper/http_api.dart';
import 'package:m_student_ubs/utills/mywidget.dart';
import 'package:m_student_ubs/load_online/sync_db.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubs/model/jadwal_kuliah.dart';
import 'package:m_student_ubs/model/user.dart';
import 'package:m_student_ubs/utills/font_style.dart';
import './subjadwal/materi_page.dart';

class JadwalPage extends StatefulWidget {
  @override
  _JadwalPageState createState() {
    return new _JadwalPageState();
  }
}

class _JadwalPageState extends State<JadwalPage> with
TickerProviderStateMixin{
  List<CListView> listLV = List();
  List<Map> listHari = List();
  TabController _tabController;
  UserModel user;
  bool isLoading = false;

  void initState() {
    super.initState();
    _tabController = new TabController(vsync: this, length: 0);

    UserModel().getUser().then((u) async{
      user = u;
      await initListLV();

      refreshAll();
    });
  }

  Future<void> initListLV() async{
    final jk = await JadwalkuliahModel().read(columns: ['hari_t',
'nohari'], where:'kd_lokal=?',
                                whereArgs: [user.kd_lokal],
                                groupBy:
'nohari', orderBy: 'nohari');

    for(int i=0;i<jk.length;i++){
      final m = jk[i];

      listHari.add({'nohari':'${m['nohari']}', 'hari':m['hari_t'] });
      listLV.add( CListView(
        itemBuilder: (b,index)=>_item(i, index),
        onRefresh: (){
          return _refresh( i,1 );
        }
      ) );
    }
  }

  Future<void> refreshAll() async{

```

```

        if(listHari.length == 0){
            await Sync.jadwalKuliah(user.kd_lokal);
            await initListLV();
        }

        String nowhari = '${DateTime.now().weekday+1}';
        int selindex = 0;

        isLoading = true;
        listLV.forEach((f){ f.isLoading = true; });
        setState((){});

        for(int i=0; i<listHari.length; i++){
            await _refresh(i, 0, 0);
            if(nowhari ==listHari[i]['nohari']){
                selindex = i;
            }
        }

        listLV.forEach((f){ f.isLoading = false;});

        _tabController = new TabController(vsync: this, length: listLV.length,
        initialIndex: selindex);
        isLoading = false;
        setState((){});
    }

    Future<void> _loadDB(int iLV, String nohari)async{
        listLV[iLV].data = await JadwalkuliahModel().read(where:'kd_lokal=? AND
        nohari=?',
            whereArgs: [user.kd_lokal, nohari],
            orderBy: 'nohari ASC, jam_t ASC' );
    }

    Future _refresh(int iLV, [int mode=0, int update=1])async{
        isLoading = true;
        listLV[iLV].isLoading = true; if(update==1)setState((){});

        await _loadDB(iLV, listHari[iLV]['nohari' ]);
        if(mode == 1 || listLV[iLV].data.length == 0){
            await Sync.jadwalKuliah(user.kd_lokal);
            await _loadDB(iLV, listHari[iLV]['nohari' ]);
        }

        isLoading = true;
        listLV[iLV].isLoading = false; if(update==1) setState((){});
        print('_Refresh      $iLV      ${listLV[iLV].data.length}      =
        ${listLV[iLV].isLoading}');
    }

    Widget _item(int iLV, int index){
        final j =JadwalkuliahModel().fromMap(listLV[iLV].data[index]);
        return Card(
            child: InkWell(
                onTap: (){
                    Navigator.of(context).push(new MaterialPageRoute(
                        builder: (_)=>MateriPage(jadwal:j)

```

```

    ));
  },
  child: Container(
    padding: EdgeInsets.all(10.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: <Widget>[
        Container(
          decoration: BoxDecoration(border:
Border(bottom: BorderSide(width: 1.0, color: Colors.blueGrey.shade100))),
          child: Text('${j.kd_mtk} - ${j.nm_mtk}', style:
TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold) ),
        ),
        IconAndLabel(icon: FontAwesomeIcons.clock,
child: Text('${j.hari_t} - ${j.jam_t}') ),
        IconAndLabel(icon: FontAwesomeIcons.building,
child: Text('${j.nm_kampus} - ${j.no_ruang} ')),
        IconAndLabel(icon: FontAwesomeIcons.mapPin,
child: Text('${j.alm_kampus}')),
        IconAndLabel(icon: FontAwesomeIcons.userCircle,
child: Text('${j.kd_dosen} - ${j.dosen1} ')),
      ],
    ),
  ),
);
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: new AppBar(
      title: TitleActionBar( subTitle: 'Jadwal Kuliah' ),
      bottom: listLV.length==0 ? null : TabBar(
        controller: _tabController,
        isScrollable: true,
        tabs: listHari.map((f){
          return Tab(child:Text('${f['hari']}') ) ;
        }).toList()
      ),
    ),
    body: isLoading ? Center(child:CupertinoActivityIndicator()) :
listLV.length == 0 ? emptyPage('Maaf, saat ini jadwal anda
tidak tersedia.') :
    TabBarView(
      controller: _tabController,
      children: listLV.map((f){
        return f.build(context);
      }).toList()
    ),
  );
}
}

import 'package:flutter/material.dart';

```

```

import 'package:flutter/cupertino.dart';
import 'dart:convert';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../utils/assets.dart';
import '../utils/logo.dart';
import '../utils/gradien.dart' as Gradien;
import 'package:flutter/services.dart';
import '../helper/http_api.dart';
import '../helper/urladdress.dart';
import '../helper/pref.dart';
import '../page/walkthrough/walkthrough_main.dart';
import 'package:m_student_ubsj/model/user.dart';
import 'package:m_student_ubsj/utils/mywidget.dart';

class LoginPage extends StatefulWidget {
  LoginPage({Key key}) : super(key: key);

  static const String tag = 'login-page';

  @override
  _LoginPageState createState() => new _LoginPageState();
}

class _LoginPageState extends State<LoginPage>
  with SingleTickerProviderStateMixin {

  final GlobalKey<ScaffoldState> _scaffoldKey = new
  GlobalKey<ScaffoldState>();
  bool isLoading = false;
  final FocusNode nim = FocusNode();
  final FocusNode pass = FocusNode();

  TextEditingController nimController;
  TextEditingController passController;

  _LoginPageState() {
    nimController = new TextEditingController(text: ''); //21150079
    passController = new TextEditingController(text: ''); //1996-06-15
    //nimController.text = '12164854';
    //passController.text = '1999-03-28';
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      key: _scaffoldKey,
      body: new Container(
        decoration: new BoxDecoration(
          image: backgroundImage,
        ),
        child: new ListView(
          padding: const EdgeInsets.all(0.0),
          children: <Widget>[
            new Column(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[

```

```

        Padding(
          padding: EdgeInsets.only(top: 100.0),
        ),
        new Logo(logo: logoImage),
        Padding(
          padding: EdgeInsets.only(top: 50.0),
          child: _loginCard(context),
        )
      ],
    ),
  ],
),
),
);
}

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

  SystemChrome.setPreferredOrientations([
    DeviceOrientation.portraitUp,
    DeviceOrientation.portraitDown,
  ]);
}

void showInSnackBar(String value, {Color backgroundColor=Colors.blue, Color
textColor=Colors.white}) {
  FocusScope.of(context).requestFocus(new FocusNode());
  _scaffoldKey.currentState?.removeCurrentSnackBar();
  _scaffoldKey.currentState.showSnackBar(new SnackBar(
    content: new Text(
      value,
      textAlign: TextAlign.center,
      style: TextStyle(
        color: textColor,
        fontSize: 16.0,
        fontFamily: "WorkSansSemiBold"),
    ),
    backgroundColor: backgroundColor,
    duration: Duration(seconds: 2),
  ));
}

Widget _loginCard(BuildContext context) {
  return isLoading ? Container(margin: EdgeInsets.only(top: 60.0),
  child: CupertinoActivityIndicator(radius: 20.0,)) : Container(
    padding: EdgeInsets.only(top: 23.0),
    child: Column(
      children: <Widget>[
        Stack(
          alignment: Alignment.topCenter,
          overflow: Overflow.visible,
          children: <Widget>[
            Card(
              elevation: 10.0,
              color: Colors.white,

```

```

    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(8.0),
    ),
    child: Container(
      width: 300.0,
      height: 160.0,
      child: Column(
        children: <Widget>[
          Padding(
            padding: EdgeInsets.only(
              top: 25.0, bottom: 5.0, left: 25.0, right: 25.0),
            child: TextField(
              focusNode: nim,
              controller: nimController,
              keyboardType: TextInputType.number,

              style: StyleLogin.textLogin,
              decoration:
StyleLogin.inputDecoration(hintText: 'Nomor Induk Mahasiswa')
            ),
          ),

          Padding(
            padding: EdgeInsets.only(
              top: 5.0, bottom: 15.0, left: 25.0, right: 25.0),
            child: TextField(
              focusNode: pass,
              controller: passController,
              obscureText: true,
              style: StyleLogin.textLogin,
              decoration:
'Sandi', icon: FontAwesomeIcons.key)
            ),
          ),
        ],
      ),
    ),
  ),
  Container(
    margin: EdgeInsets.only(top: 140.0),
    decoration: BoxDecoration( boxShadow: <BoxShadow>[

    ]),
    child: RaisedButton(
      child: Container(
        padding: EdgeInsets.fromLTRB(30.0, 15.0, 30.0,
15.0),

        child:Text('Masuk', style: StyleLogin.buttonLogin)
      ),
      color: Colors.lightBlue,
      elevation: 5.0,
      shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(50.0)),
      onPressed: onLoginClick
    )
  )

```

```

        ),
        Container(
          margin: EdgeInsets.only(top: 240),
          child: Wrap(
            direction: Axis.vertical,
            crossAxisAlignment: WrapCrossAlignment.center,
            children: <Widget>[
              Text(Res.HAK_CIPTA, style:StyleLogin.textHakCipta),
              Text('BTI          4.0          -          2019',
style:StyleLogin.textHakCipta),
            ],
          ),
        ),
      ],
    ),
  ],
),
],
),
],
),
);
}

void onLoginClick(){
  setState(){isLoading = true; }};
  //Navigator.of(context).pushNamed(WalkthroughMainPage.tag);
  HTTP.o().sendPOST(UrlAddr.URL_LOGIN, {
    'nim': nimController.text,
    'sandi': passController.text
  }).then((response){
    print('response $response');
    dynamic js = json.decode(response);

    if(js != null){
      int res = int.parse( js['result'].toString(), radix: 10 );
      if(res > 0){
        simpanLog(js).then((b){

          setState(){isLoading = false; }};
          if(b){

Navigator.of(context).pushReplacementNamed(WalkthroughMainPage.tag);
          }else{
            showInSnackBar('Terjadi kesalahan');
          }
        });
      }else{

        setState(){isLoading = false; }};
        showInSnackBar('Kesalahan ${js['msg']}', backGroundColor:
Colors.red, textColor: Colors.yellow);
      }

    }

  }).catchError((onError){
    setState(){isLoading = false; }};

```

```

        print('terjadi error $onError');
        alert( context, 'Kesalahan Jaringan', 'Maaf terjadi kesalahan
jaringan');
    });
}

Future<bool> simpanLog(dynamic data) async{
    Pref.o().setString('nim', data['nim']);
    Pref.o().setString('token', data['token']);
    int res = 0;
    String response = await HTTP.o().sendPOST(UrlAddr.URL_PROFILE, {});
    dynamic js = json.decode(response);
    print('isi response $response');

    if(js != null){
        if( js['result'] == 0 || js['data'] == null ){
            Pref.o().clear();
            showInSnackBar('Akun anda tidak ditemukan');
        }else{
            UserModel um = new UserModel().fromMap(js['data']);
            res = await um.create();
        }
    }
    print('hasil res = $res');
    return res > 0;
}

}

class StyleLogin{
    static const buttonLogin = TextStyle(color: Colors.white, fontSize: 18.0,
fontWeight: FontWeight.bold, );
    static const textLogin = TextStyle( fontFamily: "WorkSansSemiBold",
fontSize: 16.0, color: Colors.black);
    static const textHakCipta = TextStyle( fontFamily: "Poppins", fontSize:
14.0, color: Colors.white);
    static const hinttextLogin = TextStyle( fontFamily: "Poppins", fontSize:
16.0, color: Colors.grey);

    static InputDecoration inputDecoration({String hintText='', IconData
icon=FontAwesomeIcons.userCircle}){
        return InputDecoration(
            contentPadding: EdgeInsets.all(10.0),
            border: OutlineInputBorder(borderRadius:
BorderRadius.circular(43.0)),
            icon: Icon( icon, color: Colors.blueGrey, size: 22.0 ),
            hintText: hintText,
            hintStyle: StyleLogin.hinttextLogin
        );
    }
}

import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../helper/http_api.dart';
//import '../helper/urladdress.dart';
import './home_page.dart';

```

```

import './login_page.dart';
//import './akademik_page.dart';
import 'package:m_student_ubs/helper/pref.dart';
import 'package:m_student_ubs/helper/db.dart';
import 'package:m_student_ubs/utills/mywidget.dart';
import './pagemenu/kalender.dart';
import './jadwal_page.dart';
import './info_page.dart';
import './profile_page.dart';

class MainPage extends StatefulWidget {
  static const String tag = 'main-page';

  @override
  _MainPageState createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  int _currentIndex = 0;
  var _controller = PageController(
    initialPage: 0,
  );

  @override
  void initState() {
    super.initState();
    // StateExts.cekTokenValid(context);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: PageView(
        controller: _controller,
        children: <Widget>[
          HomePage(),
          JadwalPage(),
          KalenderAkademik(),
          // AkademikPage(),
          InfoPageBody(),
          ProfilePage(),
        ],
        physics: NeverScrollableScrollPhysics(),
      ),
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _currentIndex,
        onTap: (index) {
          print('tap $index');
          _currentIndex = index;
          _controller.animateToPage(index, duration: Duration(milliseconds:
100), curve: Curves.bounceIn);
          setState(() {});
        },
        type: BottomNavigationBarType.fixed,
        items: [
          BottomNavigationBarItem(

```

```

        icon: Icon(
          FontAwesomeIcons.home,
        ),
        title: Text(
          'Beranda',
        )),
      BottomNavigationBarItem(
        icon: Icon(
          FontAwesomeIcons.solidBell,
        ),
        title: Text(
          'Jadwal',
        )),
      /* BottomNavigationBarItem(
        icon: Icon(
          FontAwesomeIcons.graduationCap,
        ),
        title: Text(
          'Akademik',
        )),*/
      BottomNavigationBarItem(
        icon: Icon(
          FontAwesomeIcons.calendarDay,
        ),
        title: Text(
          'Kalender',
        )),
      BottomNavigationBarItem(
        icon: Icon(
          FontAwesomeIcons.infoCircle,
        ),
        title: Text(
          'Info',
        )),
      BottomNavigationBarItem(
        icon: Icon(
          FontAwesomeIcons.userGraduate,
        ),
        title: Text(
          'Profile',
        )),
    ],
  ),
);
}
}

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:m_student_ubsj/model/user.dart';
import 'package:m_student_ubsj/utills/mywidget.dart';
import 'package:m_student_ubsj/utills/assets.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class ProfilePage extends StatefulWidget {
  final Widget child;

```

```

    ProfilePage({Key key, this.child}) : super(key: key);

    _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends StateExt<ProfilePage> {
    UserModel _user = new UserModel();
    bool isLoading = false;

    @override
    void initState() {
        // TODO: implement initState
        super.initState();
        new UserModel().getUser().then((u) {
            _user = u;
            setState(() {

            });
        });
    }

    Widget _fotoDanNama() {
        return Row(
            children: <Widget>[
                ClipRRect(
                    borderRadius: BorderRadius.circular(1000.0),
                    child: FadeInImage.assetNetwork(placeholder:
'assets/img/no_foto.png', image: '', width: 50.0,)),
                SizedBox(width: 10.0,),
                Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: <Widget>[
                        Text('${_user.nm_mhs}', style: TextStyle(fontWeight:
FontWeight.bold)),
                        Text('${_user.nim}'),
                    ],
                ),
            ],
        );
    }

    Widget _rowCenter(Widget child1, Widget child2){
        return Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: <Widget>[
                Expanded(
                    flex: 1,
                    child: Center(child:child1),
                ),
                Expanded(
                    flex: 1,
                    child: Center(child:child2),
                ),
            ],
        );
    }
}

```

```

Widget _cardbiodata(){
  return Card(
    child: Container(
      padding: EdgeInsets.all(20.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: <Widget>[
          _fotoDanNama(),
          SizedBox(height: 10.0,),
          _rowCenter(Text('Kelas'), Text('IPK')),
          _rowCenter(Text('${_user.kd_lokal}', style:RC.style),
Text('${_user.ipk.toStringAsFixed(2)}', style: RC.style)),
        ],
      ),
    ),
  );
}

Widget _cardDataPribadi(){
  return Container(
    padding: EdgeInsets.only(top:8.0),
    child: Card(
      child:Container(
        padding: EdgeInsets.all(10.0),
        child: Column(
          children: <Widget>[
            RowDataIconLabel(icon:      FontAwesomeIcons.home,
wLeft: Text('Tempat Lahir'),
textAlign: TextAlign.right,)),
            RowDataIconLabel(icon:      FontAwesomeIcons.calendar,
wLeft: Text('Tanggal Lahir'),
textAlign: TextAlign.right,)),
            RowDataIconLabel(icon:      FontAwesomeIcons.genderless,
wLeft: Text('Jenis Kelamin'),
textAlign: TextAlign.right,)),
            RowDataIconLabel(icon:
FontAwesomeIcons.addressCard, wLeft: Text('Agama'),
wRight:      Text('${_user.agama()}'),
textAlign: TextAlign.right,)),
          ],
        ),
      ),
    );
}

Widget _cardAlamat(){

```

```

return Container(
  padding: EdgeInsets.only(top:8.0),
  child: Card(
    child:Container(
      padding: EdgeInsets.all(10.0),
      child: Column(
        children: <Widget>[

          RowDataIconLabel(icon:  FontAwesomeIcons.streetView,
wLeft: Text('Alamat'),
                                wRight:      Text('${_user.alm}',
textAlign: TextAlign.right,)),

          RowDataIconLabel(icon:  FontAwesomeIcons.userShield,
wLeft: Text('RT/RW'),
                                wRight:      Text('${_user.rt_rw}',
textAlign: TextAlign.right,)),

          RowDataIconLabel(icon:      FontAwesomeIcons.city,
wLeft: Text('Kota'),
                                wRight:      Text('${_user.kota}',
textAlign: TextAlign.right,)),

          RowDataIconLabel(icon:      FontAwesomeIcons.passport,
wLeft: Text('Kode Pos'),
                                wRight:      Text('${_user.kd_pos}',
textAlign: TextAlign.right,)),

        ],
      ),
    ),
  );
}

Widget _cardKontak(){
  return Container(
    padding: EdgeInsets.only(top:8.0),
    child: Card(
      child:Container(
        padding: EdgeInsets.all(10.0),
        child: Column(
          children: <Widget>[

            RowDataIconLabel(icon:      FontAwesomeIcons.phone,
wLeft: Text('Telepon'),
                                wRight:      Text('${_user.telp}',
textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:      FontAwesomeIcons.envelope,
wLeft: Text('E-Mail'),
                                wRight:      Text('${_user.emailaddress}',
Text('${_user.emailaddress}', textAlign: TextAlign.right,)),

          ],

```

```

        ),
    ),
);
}

Widget _cardAkademik(){
  return Container(
    padding: EdgeInsets.only(top:8.0),
    child: Card(
      child:Container(
        padding: EdgeInsets.all(10.0),
        child: Column(
          children: <Widget>[

            RowDataIconLabel(icon:  FontAwesomeIcons.university,
wLeft: Text('Kampus'),
wRight:  Text('${_user.nm_kampus}',
textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:  FontAwesomeIcons.flag,
wLeft: Text('Jurusan'),
wRight:  Text('${_user.nm_jrs}',
textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:  FontAwesomeIcons.bookmark,
wLeft: Text('Kelas'),
wRight:  Text('${_user.kd_lokal}',
textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:
FontAwesomeIcons.calendarCheck, wLeft: Text('Tahun Masuk'),
wRight:  Text('${_user.th_masuk}',
textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:  FontAwesomeIcons.clock,
wLeft: Text('Waktu'),
wRight:
Text('${_user.waktuKuliah()}', textAlign: TextAlign.right,)),

            RowDataIconLabel(icon:  FontAwesomeIcons.userCheck,
wLeft: Text('Kondisi'),
wRight:
Text('${_user.nm_kondisi}', textAlign: TextAlign.right,)),

          ],
        ),
      ),
    );
}

@override
Widget build(BuildContext context) {
  return Scaffold(

```

```

        backgroundColor: Colors.blueGrey[50],
        appBar: AppBar(
          title: TitleActionBar(subTitle: 'Profile'),
        ),
        body: isLoading ? Center(child: CupertinoActivityIndicator() ) :
SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      _cardbiodata(),
      _cardDataPribadi(),
      _cardAlamat(),
      _cardKontak(),
      _cardAkademik(),
      Padding(
        padding: EdgeInsets.all(10.0),
        child: RaisedButton(
          color: Colors.red,
          child: Text('Keluar', style:
TextStyle(color: Colors.white),),
          onPressed: () {
            confirm(context, 'Keluar', 'Apakah anda yakin
keluar dari ${Res.APP_TITLE}?', (s) {
              if(s) {
                setState(() { isLoading = true; });
                StateExt.logoutserver(context).then((e) {
                  setState(() { isLoading = false; });
                });
              }
            });
          },
        ),
      ],
    ),
  ),
);
}

}

class RC{
  static const style = TextStyle(fontWeight: FontWeight.bold, fontSize:
14.0);
}

import 'dart:async';
import 'package:flutter/material.dart';
import './login_page.dart';
import './main_page.dart';
import 'package:m_student_ubs/helper/pref.dart';
import 'package:m_student_ubs/main.dart';
import 'package:m_student_ubs/model/user.dart';

class SplashScreenPage extends StatefulWidget {

```

```

static const String tag = 'splash-page';

@override
_SplashScreenState createState() => new _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreenPage>
    with SingleTickerProviderStateMixin {

    var _visible = true;

    AnimationController animationController;
    Animation<double> animation;

    startTime() async {
        var _duration = new Duration(seconds: 2);
        return new Timer(_duration, navigationPage);
    }

    void navigationPage() {
        pindahPage();
    }

    Future pindahPage() async {
        bool state = false;
        int logininvalid = 0;
        try {
            final nim = await Pref.o().prefGetString('nim');
            final token = await Pref.o().prefGetString('token');
            logininvalid = await Pref.o().prefGetInt('LOGININVALID');

            final um = await UserModel().read(where: 'nim=?', whereArgs: [nim]);
            if(um.length > 0) state = true;
        } catch (e) {
            print('error $e');
        }

        print('kondisi state $state');

        if(state && logininvalid != 1 ) {
            Navigator.of(context).pushReplacementNamed(MainPage.tag);
        } else {
            Navigator.of(context).pushReplacementNamed(LoginPage.tag);
        }
    }

    @override
    void initState() {
        super.initState();
        animationController = new AnimationController(
            vsync: this,
            duration: new Duration(seconds: 1),
        );

        animation =
            new CurvedAnimation(parent: animationController, curve:
Curves.easeOut);

```

```

        animation.addListener(() => this.setState(() {}));
        animationController.forward();

        setState(() {
            _visible = !_visible;
        });
        startTime();
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: APP_MODE ==1 ? Colors.indigo : Colors.brown,
            body: Stack(
                fit: StackFit.expand,
                children: <Widget>[
                    new Column(
                        mainAxisAlignment: MainAxisAlignment.center,
                        children: <Widget>[
                            new Image.asset(
                                'assets/img/logo_apk.png',
                                width: animation.value * 125,
                                height: animation.value * 125,
                            ),
                        ],
                    ),
                ],
            ),
        );
    }
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/assets.dart';
import 'package:flutter_webview_plugin/flutter_webview_plugin.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';

class WebView extends StatefulWidget {
    static const tag = 'webview-page';
    dynamic settings;

    WebView({this.settings});
    _WebViewState createState() => _WebViewState(settings: this.settings);
}

class _WebViewState extends State<WebView> {
    dynamic settings;

    _WebViewState({this.settings}){
        final fwp = new FlutterWebviewPlugin();
        fwp.onStateChanged.listen((ws){
            print(' webview change state $ws ');
        }, onDone: (){
            print(' open url done ');
        });
    }
}

```

```

    }

    @override
    Widget build(BuildContext context) {
      String url = Uri.encodeFull("${this.settings['url']}");

      print('webview ope $url');
      return WebviewScaffold(
        url: url,
        appBar: AppBar(
          title: TitleActionbar(subTitle: '${this.settings['title']}'),
        ), enableAppScheme: true,
        withLocalStorage: true,
        withJavascript: true,
        withZoom: true,

        initialChild: Container(child: Center(child:
CupertinoActivityIndicator(),),),),
      );
    }
  }

}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/bimbingan.dart';
import 'package:m_student_ubsu/model/user.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'package:url_launcher/url_launcher.dart';
import 'package:flutter_html/flutter_html.dart';
import 'package:m_student_ubsu/page/pagemenu/jadwal_dosen.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class BimbinganPage extends StatefulWidget {
  @override
  _BimbinganPageState createState() => _BimbinganPageState();
}

class _BimbinganPageState extends StateExt<BimbinganPage> {
  bool _isLoading = true;
  List _data;
  UserModel _userModel;
  BimbinganModel _bimbing = new BimbinganModel();

  _BimbinganPageState() {
    _data = new List();
  }

  void initState() {
    super.initState();
    UserModel().getUser().then((e) {
      _userModel = e;
      refresh();
    });
  }
}

```

```

    }

    Future refresh([int mode=0])async{
      updateDisplay((){ _isLoading = true; });

      try{
        _data = await BimbinganModel().read(where: 'nim=?', whereArgs:
[_userModel.nim], orderBy: 'periode DESC', limit: 1, offset: 0);

        if(_data.length == 0 || mode==1){
          await Sync.getPembimbing();
          _data = await BimbinganModel().read(where: 'nim=?', whereArgs:
[_userModel.nim], orderBy: 'periode DESC', limit: 1, offset: 0);
        }

        if(_data.length > 0) {
          _bimbing = BimbinganModel().fromMap(_data[0]);
          try{
            if(_bimbing.link.substring(0,4).toLowerCase() != 'http'){
              _bimbing.link = 'http://${_bimbing.link}';
            }
          }catch(e){
            _bimbing.link = '';
          }
        }
      }catch(e){

    }
    updateDisplay((){ _isLoading = false; });

  }

  Widget _cardPembimbing(){
    return Card(
      margin: EdgeInsets.all(10.0),
      elevation: 4.0,
      child: Container(
        padding: EdgeInsets.all(15.0),
        child: Column(
          children: <Widget>[
            InkWell(
              child:
                RowDataIconLabel(icon:
FontAwesomeIcons.userCircle,          wLeft:          Text('Pembimbing'),          wRight:
Text('${_bimbing.pembimbing}          -          ${_bimbing.nm_dosen}',          textAlign:
TextAlign.right,)),),
              onTap: (){
                final dosen = {'kd_dosen':_bimbing.pembimbing,
'nama':_bimbing.nm_dosen, 'gelar':''};
                Navigator.push(context, new MaterialPageRoute(
                  builder: (b)=>JadwalDosen(dosen: dosen,)
                ));
              },
            ),
            InkWell(
              child:
                RowDataIconLabel(icon:
FontAwesomeIcons.solidUserCircle,      wLeft:          Text('Asisten'),          wRight:

```

```

Text('${_bimbing.asisten} - ${_bimbing.nm_ass}', textAlign:
TextAlign.right,)),
        onTap: (){
            final dosen = {'kd_dosen':_bimbing.asisten,
'nama':_bimbing.nm_ass, 'gelar':''};
            Navigator.push(context, new MaterialPageRoute(
                builder: (b)=>JadwalDosen(dosen: dosen,)
            ));
        }
    ),
],
),
);
}

Widget _cardMetaTA(){
    return Card(
        margin: EdgeInsets.symmetric(horizontal: 10.0),
        elevation: 4.0,
        child: Container(
            padding: EdgeInsets.all(15.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                    Text('Outline', style: TextStyle(fontWeight:
FontWeight.bold)),
                    SizedBox(height: 5.0),
                    Text('${_bimbing.nm_jns_ta}'),
                    SizedBox(height: 10.0),
                    Text('Kelompok', style: TextStyle(fontWeight:
FontWeight.bold)),
                    SizedBox(height: 5.0),
                    (_bimbing.nama_kelompok == null ? Text('Tidak ada') :
_bimbing.nama_kelompok.length > 3 ? Text('${_bimbing.nama_kelompok}') :
Text('-') )
                ],
            ),
        ),
    );
}

Widget _cardInfoTA(){
    return Card(
        margin: EdgeInsets.symmetric(horizontal: 10.0),
        child: Container(
            padding: EdgeInsets.all(15.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: <Widget>[
                    Text('Informasi', style: TextStyle(fontWeight:
FontWeight.bold)),
                    Container(
                        child: _bimbing.info == 'null' ? Text("Belum ada
informasi.") : Text('${_bimbing.info}'),

```

```

        ),
        SizedBox(height: 20.0,),
        Html(data: '<b>Website:</b><br/><a
href="\${_bimbing.link}">\${_bimbing.link}</a>', onLinkTap: (s){
            _launchURL(s);
        },)
    ],
),
);
}

Future _launchURL(String url)async{
    if(await canLaunch(url)){
        await launch(url);
    }else{
        alert(context, 'Url Gagal', 'Url $url Gagal dibuka');
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: TitleActionbar(subTitle: 'Informasi Bimbingan',),
        ),
        body: _isLoading ? Center(child:CupertinoActivityIndicator()) :
            _data.length == 0 ? emptyPage('Belum ada informasi') :
            RefreshIndicator(
                color: Colors.red,
                onRefresh: (){
                    return refresh(1);
                },
                child: ListView(
                    shrinkWrap: true,
                    children: <Widget>[
                        _cardPembimbing(),
                        _cardMetaTA(),
                        _cardInfoTA()
                    ],
                ),
            ),
    );
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsiload_online/sync_db.dart';
import 'package:m_student_ubsimodel/bimbingan.dart';
import 'package:m_student_ubsimodel/user.dart';
import 'package:m_student_ubsiutils/mywidget.dart';
import 'package:flutter_html/flutter_html.dart';
import 'package:url_launcher/url_launcher.dart';

class InfoBimbinganPanel extends StatefulWidget {
    @override

```

```

    _InfoBimbinganPanelState createState() => _InfoBimbinganPanelState();
}

class _InfoBimbinganPanelState extends StateExt<InfoBimbinganPanel> {
    UserModel _user;
    BimbinganModel bimbing;
    List data;
    bool _loading = false;

    _InfoBimbinganPanelState() {
        data = new List();
        bimbing = new BimbinganModel();
    }

    @override
    void initState() {
        // TODO: implement initState
        super.initState();
        UserModel().getUser().then((u) {
            _user = u;
            _refresh();
        });
    }

    Future _refresh() async {
        updateDisplay(() { _loading = true; });
        try {
            int diff = 3600 * 1000 * 16;
            data = await BimbinganModel().read(where: 'nim=? AND synctime-? < ?',
                whereArgs: [_user.nim,
                    DateTime.now().millisecondsSinceEpoch, diff], orderBy: 'periode DESC', limit:
1, offset: 0);

            if (data.length == 0) {
                await Sync.getPembimbing();

                data = await BimbinganModel().read(where: 'nim=? AND synctime-? <
?',
                    whereArgs: [_user.nim, DateTime.now().millisecond,
diff], orderBy: 'periode DESC', limit: 1, offset: 0);
            }

            if (data.length > 0) {
                bimbing = new BimbinganModel().fromMap(data[0]);
                String s = bimbing.link.trim().toLowerCase().substring(0, 4);
                if (s != 'http') {
                    bimbing.link = 'http://${bimbing.link}';
                }
            }

        } catch (e) {}
        updateDisplay(() { _loading = false; });
        return ;
    }

    @override

```

```

Widget build(BuildContext context) {

  return _loading ? Center(child:CupertinoActivityIndicator()) :
    data.length == 0 || bimbing.info == 'null' ? Container(padding:
EdgeInsets.all(0),) :
    Card(
      elevation: 7.0,
      margin: EdgeInsets.fromLTRB(10.0, 0.0, 10.0, 20.0),
      child: Container(
        padding: EdgeInsets.all(10.0),
        child: Html(data: '<center><b>Informasi
Bimbingan</b></center><br/><p>${bimbing.info}</p><a
href="${bimbing.link}">${bimbing.link}</a>',
          onLinkTap: (s){
            openWeb(s);
          },
        ),
      ),
    );
}

Future openWeb(String url)async{
  if(await canLaunch(url)){
    launch(url);
  }else{
    alert(context, 'Gagal Buka Url', 'Url $url gagal dibuka');
  }
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/jadwal_kuliah.dart';
import 'package:m_student_ubsu/model/user.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'package:intl/intl.dart';

class JadwalKuliahTodayView extends StatefulWidget {
  @override
  _JadwalKuliahTodayViewState createState() => _JadwalKuliahTodayViewState();
}

class _JadwalKuliahTodayViewState extends StateExtends<JadwalKuliahTodayView> {
  List listdata;

  void initState() {
    super.initState();
    listdata = new List();
    loaddata();
  }

  Future loaddata()async{
    int nohari = DateTime.now().weekday+1;
    print('nohari $nohari');
    UserModel u = await UserModel().getUser();

```

```

String jamnow = DateFormat('HH:mm:ss').format(DateTime.now());

print('loaddata jamnow $jamnow');
List lst = await JadwalkuliahModel().read( where: 'nohari=? AND
(jam_selesai >=? )', whereArgs: [nohari, jamnow]);
if(lst.length == 0){
    await Sync.jadwalKuliah(u.kd_lokal);
    lst = await JadwalkuliahModel().read( where: 'nohari=? AND
(jam_selesai >=? )', whereArgs: [nohari, jamnow]);
}
listdata = lst;
updateDisplay();
return;
}

Widget pointLabel(String strBullet, String strTitle){
    return Row(
        children: <Widget>[
            Expanded(
                flex: 1,
                child: CircleAvatar(
                    backgroundColor: Colors.grey,
                    child: Text('$strBullet', style: TextStyle(fontSize: 12.0,
color: Colors.white)),
                ),
            ),
            SizedBox(width: 10.0,),
            Expanded(child: Text('$strTitle'), flex: 8)
        ],
    );
}

static const TextStyle styl1 = const TextStyle(fontSize: 13.0, fontWeight:
FontWeight.bold, color: Colors.blueGrey);

Widget item(dynamic dt){
    JadwalkuliahModel jm =JadwalkuliahModel().fromMap(dt);
    return Card(
        child: Container(
            padding: EdgeInsets.all(10.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: <Widget>[
                    pointLabel(jm.kd_mtk, '${jm.nm_mtk}'),
                    Padding(child: Container(
                        decoration: BoxDecoration(border:
Border(bottom: BorderSide(color: Colors.blueGrey, width: 0.5)) )
                    ),
                    padding: EdgeInsets.symmetric(vertical: 10.0)),
                    Text('Mulai - Selesai'),
                    Text('${jm.jam_t} | ( ${jm.sks} ) SKS', style: styl1),
                    Padding(padding: EdgeInsets.only(top: 10.0),
child: Text('Lokasi')),
                    Text('${jm.no_ruang} ${jm.alm_kampus}', style: styl1)
                ],
            ),
        ),
    );
}

```

```

        ),
    ),
);
}

@override
Widget build(BuildContext context) {
    return Padding(
        padding: EdgeInsets.all(10.0),
        child: listdata.length == 0 ? Text('Tidak ada jadwal kuliah
selanjutnya.') : ListView(
            shrinkWrap: true,
            physics: ClampingScrollPhysics(),
            children: listdata.map((f) {
                return item(f);
            }).toList()
        )
    );
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/helper/http_api.dart';
import 'package:m_student_ubsu/helper/urladdress.dart';
import 'package:m_student_ubsu/model/pengaturan.dart';
import 'package:m_student_ubsu/page/webview_page.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'dart:convert';

class NewsView extends StatefulWidget {
    final Widget child;

    NewsView({Key key, this.child}) : super(key: key);

    _NewsViewState createState() => _NewsViewState();
}

class _NewsViewState extends StateExtends<NewsView> {
    List listNews;

    _NewsViewState() {
        listNews = new List();
    }

    void initState() {
        super.initState();
        loadNews();
    }

    Future loadNews() async {
        final r = await HTTP.o().sendPOST(UrlAddr.URL_NEWS, {});
        try {
            final artikels = json.decode(r);
            if(artikels['data'] != null) {

```

```

        final art = artikels['data'];
        listNews = art;
        if (art.length > 0) {
            PengaturanModel p = new PengaturanModel();
            p.kunci = 'NEWS';
            p.nilai = json.encode(art);
            await p.replace('kunci=?', [p.kunci]);
        }
    }
} catch (e) {
    print('error loadnews $e');
}

if (listNews.length == 0) {
    final setting = await PengaturanModel().read(where: 'kunci=?',
whereArgs: ['NEWS']);
    if (setting.length > 0) {
        listNews = json.decode(setting[0]['nilai']);
    }
}
updateDisplay();
}

Widget item(dynamic f) {
    final max = f['judul'].toString().length > 50 ? 50 :
f['judul'].toString().length;
    return Container(
        padding: EdgeInsets.all(3.0),
        width: 200.0,
        child: Card(
            child: Column(
                children: <Widget>[
                    Expanded(
                        flex: 3,
                        child:
FadeInImage.assetNetwork(placeholder: 'assets/img/no_foto.png',
image: '${f['thumb']}', fit: BoxFit.cover, ),
                    ),
                    SizedBox(height: 10.0, ),
                    Expanded(
                        flex: 1,
                        child: Container(
                            padding: EdgeInsets.fromLTRB(5.0,
0.0, 5.0, 0.0),
                            width: 195.0,
                            child: Column(
                                children: <Widget>[
                                    Expanded(
                                        flex: 2,

child: Text('${f['judul'].toString().substring(0, max)}',
TextAlign.center, style: TextStyle(fontSize: 12.0), ),
                                    ),
                                    Expanded(
                                        flex: 1,

child: Padding(child: IconLabel(Text('${f['tgl']}', style: TextStyle(fontSize:

```

```

10.0)), icon: FontAwesomeIcons.clock), padding:
EdgeInsets.only(bottom:5.0))
    )
  ]
)
)
),
],
)
));
}

@override
Widget build(BuildContext context) {
  return ListView(
    physics: ClampingScrollPhysics(),
    shrinkWrap: true,
    scrollDirection: Axis.horizontal,
    children: listNews.map((f) {
      return InkWell(child:item(f), onTap: (){
        Navigator.of(context).push(new MaterialPageRoute(
          builder: (bc)=>WebView(
            settings:{
              'title':'BSI News',
'url':'http://news.bsi.ac.id/index.php/artikel/baca/${f['id']}'
            )
          ));
        },));
    }).toList(),
  );
}
}

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

class InfoPesan extends StatelessWidget {

  List<dynamic> data = new List();
  Map mapID = new Map();

  void addItem(List items){
    for(int i=0; i<items.length;i++){
      final item = items[i];
      final idx = mapID[item['id']];
      if(idx == null){
        mapID[item['id']] = data.length;
        data.add(item);
      }else{
        data[idx] = item;
      }
    }
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      body: new Container(

```

```

        child: new ListView.builder(
          itemCount: data.length,
          itemBuilder: (context, index) {
            return new Card(
              child: new Container(
                padding: EdgeInsets.all(5.0),
                child: new Column(
                  children: <Widget>[
                    new ListTile(
                      leading: new Icon(FontAwesomeIcons.envelope),
                      title: new Text("${data[index]}"),
                      subtitle: new Text("Keterangan Detail"),
                    )
                  ],
                ),
            ),
          ),
        );
      },
    ),
  );
}
}

```

```

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/assets.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/utills/font_style.dart';
import 'package:m_student_ubsu/helper/urladdress.dart';
import 'package:m_student_ubsu/helper/http_api.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/page/pagemenu/jadwal_dosen.dart';
import 'dart:convert';

```

```

class CariJadwalDosen extends StatefulWidget {
  static const tag = 'jadwal-dosen';
  _CariJadwalDosenState createState() => _CariJadwalDosenState();
}

```

```

class _CariJadwalDosenState extends State<CariJadwalDosen> {
  List<dynamic> data;
  Map<String, int> map;

  int _page = 1;
  String _keyword = '';
  bool _isLoading = false;

  _CariJadwalDosenState(){
    data = new List();
    map = new Map();
  }
  @override
  void initState() {
    // TODO: implement initState
    data.clear();
  }
}

```

```

        map.clear();
        super.initState();
    }

    @override
    void dispose() {
        // TODO: implement dispose
        super.dispose();
    }

    void addItem(List item){
        for(Map m in item){
            int idx = map[ m['kd_dosen'] ];
            if(idx == null){
                map[m['kd_dosen']] = data.length;
                data.add(m);
            }else{
                data[idx] = m;
            }
        }
    }

    Widget _appBar(){
        return AppBar(
            title: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                    Text(Res.APP_TITLE),
                    Text('Jadwal Dosen', style: Style.commonTextStyle,)
                ],),
        );
    }

    Widget _cardInputKodeDosen(){
        return Container(
            margin: EdgeInsets.fromLTRB(10.0, 5.0, 10.0, 2.0),
            child: Card(
                child: Container(
                    padding: EdgeInsets.fromLTRB(10.0, 5.0, 10.0, 10.0),
                    child: TextField(
                        onSubmitted: (keyword){
                            _page = 1;
                            _keyword = keyword;
                            data = new List();
                            _loadDosen(keyword);
                        },
                        decoration: InputDecoration(
                            hintText: 'Nama / Kode
Dosen',
                            contentPadding:
EdgeInsets.all(10.0),
                            icon:
Icon(FontAwesomeIcons.userCircle),
                            border: OutlineInputBorder(
                                borderRadius:
BorderRadius.circular(52.0)

```

```

        ),
      ),
    ),
  );
}

Widget _listviewDosen(){
  if(!_isLoading){
    return CupertinoActivityIndicator();
  }

  return data.length == 0 ? Text('Belum ada') :
    ListView.builder(
      shrinkWrap: true,
      physics: ScrollPhysics(),
      itemCount: data.length,
      itemBuilder: (bc,index){
        var d = data[index];

        return InkWell(
          onTap: (){
            Navigator.of(context).push(new MaterialPageRoute(
              builder: (c)=>JadwalDosen(dosen: d)
            ));
          },
          child: ListTile(
            leading: Icon(FontAwesomeIcons.userAlt),
            title: Text('${d['nama']}, ${d['gelar']}'),
            subtitle: Text('${d['kd_dosen']}') ,
          ),
        );
      },
    );
}

Widget _cardDaftarDosen(){
  return Container(
    margin: EdgeInsets.fromLTRB(10.0, 1.0, 10.0, 2.0),
    width: double.infinity,
    child: Card(
      child: Container(
        padding: EdgeInsets.fromLTRB(10.0, 10.0, 5.0, 10.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: <Widget>[
            Container(
              width: double.infinity,
              margin: EdgeInsets.fromLTRB(0.0, 0.0, 0.0,
20.0),
              child: Center(child:Text('Daftar Dosen', style:
TextStyle(fontSize: 20.0, fontFamily: 'Raleway' )))
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

        ], _listviewDosen()
      ),
    ),
  );
}

Widget _body(){
  return SingleChildScrollView(
    child: Column(
      children: <Widget>[
        _cardInputKodeDosen(),
        _cardDaftarDosen(),
      ],
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: _appBar(),
    body: _body(),
  );
}

Future<void> _loadDosen(String keyword)async{
  _isLoading = true; setState((){});

  final response = await HTTP.o().sendPOST(UrlAddr.URL_LIST_DOSEN, {
    'keyword' : keyword,
    'page' : _page
  });
  var js = json.decode(response);
  if(js['data'] != null){
    map.clear();
    data.clear();
    addItem(js['data']);
  }

  _isLoading = false;  setState((){});
}
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import './khs.dart';
import './study_per_semester.dart';
import './nilaimurni.dart';

class HasilStudy extends StatefulWidget {

```

```

    static const tag = 'hasil-study';
    _HasilStudyState createState() => _HasilStudyState();
}

class _HasilStudyState extends State<HasilStudy> {

  Widget _body(){
    return GridView.count(
      crossAxisCount: 2,
      children: <Widget>[
        CardBoardIcon(
          color:Colors.indigoAccent,
          icon:FontAwesomeIcons.fileAlt,
          title: 'Kartu\nHasil Studi',
          onTap: (){
            Navigator.of(context).pushNamed(KHS.tag);
          },
        ),

        CardBoardIcon(
          color:Colors.deepOrange,
          icon:FontAwesomeIcons.checkSquare,
          title: 'Hasil Studi\nPer Semester',
          onTap: (){
            Navigator.of(context).pushNamed(HasilStudySemester.tag);
          },
        ),

        CardBoardIcon(
          color:Colors.cyan,
          icon:FontAwesomeIcons.table,
          title: 'Nilai Murni',
          onTap: (){
            Navigator.of(context).pushNamed(NilaiMurni.tag);
          },
        )
      ],
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: TitleActionBar(subTitle: 'Hasil Studi',)
      ),
      body: _body()
    );
  }
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubs/helper/http_api.dart';
import 'package:m_student_ubs/helper/urladdress.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubs/utills/mywidget.dart';
import 'package:m_student_ubs/utills/font_style.dart';

```

```

import 'dart:convert';

class JadwalDosen extends StatefulWidget {
  dynamic dosen;
  JadwalDosen({this.dosen});

  _JadwalDosenState createState() => _JadwalDosenState(dosen: dosen);
}

class _JadwalDosenState extends State<JadwalDosen> with
SingleTickerProviderStateMixin {
  dynamic dosen;
  List<dynamic> data;
  Map<String, List> hariJdw;
  bool isLoading = false;
  TabController _tabController;

  _JadwalDosenState({this.dosen});

  void initState() {
    super.initState();
    _tabController = new TabController(vsync: this, length: 0);
    data = new List();
    hariJdw = new Map();

    _loadJadwal('${dosen['kd_dosen']}');
  }

  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
  }

  Future _loadJadwal(String kdDosen) async{
    setState(() { isLoading = true; });
    final response = await HTTP.o().sendPOST(UrlAddr.URL_JADWAL_DOSEN, {
      'kd_dosen' : kdDosen
    });

    int initsel = 0;
    final js = json.decode(response);
    if(js != null){
      data = js['data'];
      Map hidx = new Map();
      data.forEach((j){
        if(hariJdw['${j['hari_t']}'] == null){
          hariJdw['${j['hari_t']}'] = new List();
          hariJdw['${j['hari_t']}'].add( j );
          hidx['${j['hari_t']}'] = '${j['nohari']}';
        }
      });

      String idxtoday = '${DateTime.now().weekday+1}';
      data = new List();
      hariJdw.forEach((k,v){

```

```

        data.add(k);
    });

    for(int i=0;i<data.length;i++){
        String nohari = hidx[data[i]];
        if(nohari == idxtoday)initssel = i;
    }

    _tabController =new TabController(vsync: this, length: data.length,
initialIndex: initssel);
    }

    setState(() { isLoading = false; });
}

Widget _listViewJadwal(){
    if(!isLoading){
        return Center(child: CupertinoActivityIndicator() );
    }else if(data.length == 0){
        return Center(child: Text('Konten Belum Tersedia') );
    }

    return TabBarView(
        controller: _tabController,
        children: data.map((h){

            final data = hariJdw[h];
            final f = CListView(

                itemBuilder: (b,index){
                    return itemList(data[index]);
                },onRefresh: (){
                    return _loadJadwal('${dosen['kd_dosen']}');
                }
            );
            f.data = data;
            return f.build(context);
        }).toList()
    );
}

Widget itemList(dynamic jdw){
    return Card(
        margin: EdgeInsets.fromLTRB(4.0, 2.0, 4.0, 2.0),
        child: Container(
            padding: EdgeInsets.all(10.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                    Text('${jdw['kd_mtk']} - ${jdw['mtk']}', style:
TextStyle(fontSize: 14.0, color: Colors.blueGrey, fontWeight:
FontWeight.bold),),
                    SizedBox(height: 18.0,),
                    IconLabel(Text('${jdw['kd_dosen']} /
${jdw['kd_dosen2']}', style: TextStyle(fontSize: 16.0), ),
                        icon: FontAwesomeIcons.userCircle,),
                ],
            ),
        ),
    );
}

```

```

                SizedBox(height: 9.0,),
                IconLabel(Text('${jdwl['hari_t']}. ${jdwl['jam_t']} -
({jdwl['sks']} SKS)', style: TextStyle(fontSize: 16.0), ),
                    icon: FontAwesomeIcons.clock,)),

                SizedBox(height: 9.0,),
                IconLabel(Text('${jdwl['kd_lokal']}', style:
TextStyle(fontSize: 16.0), ),
                    icon: FontAwesomeIcons.flag,)),

                SizedBox(height: 9.0,),
                IconLabel(Text('${jdwl['kampus']} -
${jdwl['no_ruang']}', style: TextStyle(fontSize: 16.0), ),
                    icon: FontAwesomeIcons.solidBuilding,)),

            ],
        ),
    );
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: TitleActionBar(subTitle: '${dosen['nama']}',
${dosen['gelar']}',),
            bottom: TabBar(
                controller: _tabController,
                isScrollable: false,
                tabs: data.map((h){
                    return Tab(text: '$h',);
                }).toList()
            ),
        ),
        body: _listViewJadwal()
    );
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/kalender.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/helper/http_api.dart';
import 'package:intl/intl.dart';

class KalenderAkademik extends StatefulWidget {
    static const tag = 'kalender-akademik';
    _KalenderAkademikState createState() => _KalenderAkademikState();
}

```

```

class _KalenderAkademikState extends State<KalenderAkademik> with
TickerProviderStateMixin{
  List<CListView> lv;
  dynamic data;
  TabController _tabController;
  List<String> _periode;
  bool isLoading = false;
  final GlobalKey _keyTab = new GlobalKey();

  void initState() {
    super.initState();
    _periode = new List();
    _tabController = new TabController(vsync: this, length: _periode.length);
    lv = List();
    _refresh();
  }

  Future _loadDB() async {
    DateTime dt = DateTime.now();
    this.data = await KalenderModel().read( where: 'strftime(?, tgl_awal) >=
?', whereArgs: ['%Y', dt.year]);
  }

  Future _refresh([int mode = 0]) async {
    isLoading = true;
    setState(() {});

    await _loadDB();
    if (data.length == 0 || mode == 1) {
      await Sync.kalenderAkademik();
      await _loadDB();
    }

    String periode = '';
    List tmp = new List();
    int initIndex = 0;

    for(Map m in data){
      final k = KalenderModel().fromMap(m);
      final tgl = Convert.bulanTahun( k.tgl_awal );

      if(periode != tgl ){
        periode = tgl;
        _periode.add(periode);

        if(_periode.length > 1){
          addListView(tmp);
          tmp = new List();

          final current =Convert.bulanTahun( DateFormat('yyyy-MM-
dd').format(DateTime.now()) );
          if(periode == current){
            initIndex =_periode.length-1;
          }
        }
      }
      tmp.add(m);
    }
  }
}

```

```

    }

    if(tmp.length > 0){
        addListView(tmp);
    }

    _tabController = new TabController(vsync: this, length: _periode.length,
initialIndex: initIndex);

    isLoading = false;
    setState(() {});
}

void addListView(final dynamic dt){
    int len = lv.length;
    final l = new CListView(
        itemBuilder: (b,index){
            return _item(len, index);
        }, onRefresh: (){
            _refresh(1);
        });

    l.data = dt;
    lv.add(l);
}

Widget _item(int idxLV, int index) {
    final dt = lv[idxLV].data[index];
    final k = KalenderModel().fromMap(dt);

    if(k.tgl_akhir == '---'){
        return Container(
            padding: EdgeInsets.fromLTRB(20.0,10.0,20.0,10.0),
            child: Text(
                k.tgl_awal,
                style: TextStyle(fontWeight: FontWeight.bold,
fontSize: 21.0)
            )
        );
    }

    DateTime now = DateTime.now();
    DateTime tgl = DateFormat('yyyy-MM-dd').parse(k.tgl_awal);
    Color wrn = now.compareTo(tgl) >= 0 ? Colors.grey : Colors.indigoAccent;

    return Card(
        child: ListTile(
            leading: LabelTopDown(top: Convert.dateOnly(k.tgl_awal), down:
Convert.sortMonth(k.tgl_awal), backColor: wrn, ),
            subtitle: Text('${Convert.tglRange(k.tgl_awal, k.tgl_akhir)}',
                style: TextStyle(fontWeight: FontWeight.bold)),
            title: Text('${k.kegiatan}')),
        );
    }

    @override
    Widget build(BuildContext context) {
        return Container(

```

```

        child: Scaffold(
          appBar: AppBar(
            title: TitleActionbar(subTitle: 'Kalender Akademik'),
            bottom: lv.length == 0 ? null : TabBar(
              isScrollable: true,
              controller: _tabController,
              tabs: _periode.map((f){
                return Tab(child: Text('$f'),);
              }).toList()
            ),
          ),
          body: isLoading ? Center(child:CupertinoActivityIndicator()) :
            lv.length == 0 ? emptyPage('Belum ada Kalender Tersedia') :
            TabBarView(
              key: _keyTab,
              controller: _tabController,
              children: lv.map((f){
                return f.build(context);
              }).toList()
            )
        ),
      );
    }
  }

import 'package:flutter/material.dart';
import 'package:m_student_ubsj/model/user.dart';
import 'package:m_student_ubsj/model/hasilstudy.dart';
import 'package:m_student_ubsj/utills/mywidget.dart';
import 'package:m_student_ubsj/helper/http_api.dart';
import 'package:m_student_ubsj/helper/urladdress.dart';
import 'package:m_student_ubsj/model/hasilstudy.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'dart:convert';
import 'package:m_student_ubsj/main.dart';
import 'package:m_student_ubsj/utills/mywidget.dart';

class KHS extends StatefulWidget {
  static const tag = 'hasil-study-khs';
  _KHSState createState() => _KHSState();
}

class _KHSState extends State<KHS> {
  bool _isExist = false;
  int sks = 0;
  int mutu = 0;
  double ipk = 0.0;
  Map _nilai;

  Map maps = new Map();
  UserModel _user;
  CListView cListView;

  _KHSState(){
    _nilai = {'A':0, 'B':0, 'C':0, 'D':0, 'E':0};
  }
}

```

```

void initState() {
  super.initState();

  cListView = new CListView(
    itemBuilder: (b,index){
      if(index == 0){
        return _resumeNilai();
      }else{
        return _itemList(index - 1);
      }
    }, onRefresh: (){
      _refresh(1);
    },
  );

  UserModel().getUser().then((u) {
    _user = u;
    _refresh();
  });

  _isExist = true;
}

Widget _resumeNilai(){
  Map<String,Color> mc = new Map();
  mc = {'A' : Colors.cyan, 'B' : Colors.blueGrey, 'C' : Colors.amber,
    'D' : Colors.orange, 'E' : Colors.red };

  return CardTwoColumn(
    w2: 3,
    childCol1: <Widget>[
      RowDataIconLabel(wLeft: Text('Nilai A : '), wRight:
Text('${_nilai['A']}'),),
      RowDataIconLabel(wLeft: Text('Nilai B : '), wRight:
Text('${_nilai['B']}'),),
      RowDataIconLabel(wLeft: Text('Nilai C : '), wRight:
Text('${_nilai['C']}') ),),
    ],
    childCol2: <Widget>[
      RowDataIconLabel(wLeft: Text('Nilai D : '), wRight:
Text('${_nilai['D']}') ),),
      RowDataIconLabel(wLeft: Text('Gagal : '), wRight:
Text('${_nilai['E']}') ),),
    ],
  );
}

@override
void dispose() {
  _isExist = false;
  // TODO: implement dispose
  super.dispose();
}

@override
void setState(fn) {
  // TODO: implement setState

```

```

        if(!_isExist) super.setState(fn);
    }

    Future<void> loadDB()async{
        final ls = await HasilstudyModel().read(where: "nim=? AND
grade_akhir<>'\"", whereArgs: [_user.nim]);

        for(Map m in ls){
            String key = m['nim'] + '|' + m['kd_mtk'];

            _nilai[ m['grade_akhir'].toString().toUpperCase() ]++;

            int idx = maps[key];
            if(idx == null){
                maps[key] = cListView.data.length;
                cListView.data.add(m);
            }else{
                maps[key] = idx;
            }
        }
    }

    Future<void> loadServer()async{
        String response = await HTTP.o().sendPOST(UrlAddr.URL_KHS, {});
        final js = json.decode(response);

        if(js['data']!=null){
            cListView.data = js['data'];

            for(Map m in cListView.data){
                final khs = HasilstudyModel().fromMap(m);
                await khs.replace( 'nim=? AND kd_mtk=?', [khs.nim, khs.kd_mtk]);
            }
        }
    }

    Future<void> _refresh([int mode = 0]) async{
        if(!cListView.isLoading){
            setState((){cListView.isLoading = true;});
            await loadDB();

            if(mode == 1 || cListView.data.length == 0){
                await loadServer();
                await loadDB();
            }

            getResumeStudy();
            cListView.data.length +=1;
            setState((){cListView.isLoading = false;});
        }
    }

    void getResumeStudy(){
        sks = 0; mutu = 0;
        for(Map m in cListView.data){
            sks+= int.parse('0${m['sks']}');
            mutu+= int.parse('0${m['mutu']}');
        }
    }

```

```

    }
    ipk = mutu / sks;
  }

Widget _infoResume(){
  return Container(
    color: APP_MODE == 1 ? Colors.blueAccent : Colors.brown,
    height: 50.0,
    padding: EdgeInsets.all(9.0),
    child: Row(
      children: <Widget>[
        Expanded( flex: 3,
          child: Container( child: Center(child:Text('SKS
$sk $sks ', style: TextStyle(color: Colors.white) ) ) )
        ),
        Expanded( flex: 3,
          child: Container( child: Center(child:Text('IPK
${ipk.toStringAsFixed(2)} ', style: TextStyle(color: Colors.white) ) ) )
        )
      ],
    ),
  );
}

Widget _itemList(int index){
  final h = cListView.data[index];
  Map<String,Color> mc = new Map();
  mc = {'A' : Colors.cyan, 'B' : Colors.blueGrey, 'C' : Colors.amber,
    'D' : Colors.orange, 'E' : Colors.red };
  var c = mc[h['grade_akhir']];
  c = c == null ? Colors.white : c;
  return ListTile(
    leading: CircleBadge(text: h['grade_akhir'], color: c,),
    title: Text('${h['kd_mtk']} .- ${h['mtk']}'),
    subtitle: Text('${h['sks']} SKS '),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: TitleActionbar(subTitle: 'Kartu Hasil Study',)
    ),
    body: Column(
      children: <Widget>[
        Expanded(flex: 1, child:_infoResume()),
        Expanded(flex: 13, child: cListView.build(context) )
      ],
    ),
  );
}

import 'package:flutter/material.dart';

```

```

import 'package:m_student_ubsi/helper/http_api.dart';
import 'package:m_student_ubsi/helper/urladdress.dart';
import 'package:m_student_ubsi/utills/mywidget.dart';
import 'package:m_student_ubsi/model/hasilstudy.dart';
import 'package:m_student_ubsi/model/user.dart';
import 'dart:convert';

class NilaiMurni extends StatefulWidget {
  static const tag = 'hasil-study-nilai-murni';
  _NilaiMurniState createState() => _NilaiMurniState();
}

class _NilaiMurniState extends State<NilaiMurni> {
  bool _isExist = false;
  int _csmt = 1;
  int sks = 0;
  double ip = 0.0;
  Map maps = new Map();
  UserModel _user;
  List _listSemester = [1,2,3,4,5,6,7];
  CListView cListView;

  _NilaiMurniState(){
    cListView = new CListView(
      itemBuilder: (b,index){
        return _itemList(index);
      }, onRefresh: (){
        _refreshData(1);
      }
    );
  }

  void initState() {
    _isExist = true;
    super.initState();

    UserModel().getUser().then((e) {
      this._user = e;
      _getListSemester();
      _refreshData();
    });
  }

  @override
  void dispose() {
    // TODO: implement dispose
    _isExist = false;
    super.dispose();
  }

  @override
  void setState(fn) {
    // TODO: implement setState
    if(_isExist) super.setState(fn);
  }
}

```

```

    }

    Future<void> _getListSemester() async {
        final ls = await HasilstudyModel().read(columns: ['smt'], where: 'nim=?',
        whereArgs: [_user.nim],
                                groupBy: 'smt'
                                );

        if(ls.length > 0){
            _listSemester.clear();
            ls.forEach((f){
                _listSemester.add(int.parse(f['smt']));
            });
        }
        setState(() {});
    }

    Widget _cardPilihSemester(){
        return Expanded(
            flex: 1,
            child: Card(
                child: Container(
                    width: double.infinity,
                    child: Center(child: DropdownButton(
                        items: _listSemester.map((val) {
                            return
                                DropdownMenuItem(
                                    value: val,
                                    child:
                                        SizedBox(width: 200.0, child: Text('Semester $val')),
                                ),
                        }).toList(),
                        value: _csmt,
                        onChanged: (val) { _csmt = val;
                            _refreshData(); setState(() {}); },
                    ),
                ),
            ),
        );
    }

    Future<void> loadDB() async {
        final ls = await HasilstudyModel().read(where: 'nim=? AND smt=?',
        whereArgs: [_user.nim, _csmt]);
        cListView.data = ls;
    }

    Future<void> loadServer() async {
        String response = await HTTP.o().sendPOST(UrlAddr.URL_KHS, {});
        final js = json.decode(response);

        if(js['data'] != null){
            for(Map m in js['data']){
                final khs = HasilstudyModel().fromMap(m);
            }
        }
    }

```

```

        await khs.replace('nim=? AND kd_mtk=?', [khs.nim, khs.kd_mtk]);
    }
}

Future<void> _refreshData([int mode=0])async{
  setState((){ cListView.isLoading = true; });
  await loadDB();
  if(mode == 1 || cListView.data.length == 0){
    await loadServer();
    await loadDB();
  }

  sks =0; ip = 0.0;
  int mutu = 0;

  for(int i=0;i<cListView.data.length;i++){
    final d = cListView.data[i];
    sks += d['sks'];
    mutu += d['mutu'];
  }
  ip = mutu/sks;

  setState((){ cListView.isLoading = false; });
}

Widget _listViewHasil(){
  return Expanded(
    flex: 10,
    child: cListView.build(context)
  );
}

Widget _itemList(int index){
  final h = HasilstudyModel().fromMap( cListView.data[index] );
  Map<String,Color> mc = new Map();
  mc = {'A' : Colors.cyan, 'B' : Colors.blueGrey, 'C' : Colors.amber,
    'D' : Colors.orange, 'E' : Colors.red };
  var c = mc[h.grade_akhir];
  c = c == null ? Colors.white : c;
  return ListTile(
    leading: Container(
      width: 60.0,
      height: 40.0,
      child: Center(child:Text('${h.total_nilai}',
        style: TextStyle(fontSize: 14.0,
          fontWeight:
FontWeight.bold,
          color: Colors.white
        ),
      ),
    ),
    decoration: BoxDecoration(color: c)
  ),
  title: Text('${h.kd_mtk} .- ${h.mtk}. (${h.sks} SKS) '),

```

```

        subtitle: Text('TGS ${h.tgs} | PRS ${h.presensi} | UTS ${h.uts} |
UAS ${h.uas}'),
      );
    }

    Widget _panelResume(){
      return Expanded(
        flex:1,
        child: Container(
          color: Colors.blueGrey,
          child: Center(
            child:Text('${cListView.data.length} Matakuliah. $sks SKS. IP
= ${ip.toStringAsFixed(2)}',
              style:    TextStyle(color:    Colors.white,    fontWeight:
FontWeight.bold)
            )
          )
        ),
      );
    }

    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          title: TitleActionBar(subTitle: 'Nilai Murni Per Semester',),
        ),body: Column(
          children: <Widget>[
            _cardPilihSemester(),
            _listViewHasil(),
            _panelResume()
          ],
        ),
      );
    }
  }

import 'package:flutter/material.dart';
import 'package:m_student_ubs/utlis/mywidget.dart';
import 'package:m_student_ubs/helper/http_api.dart';
import 'package:m_student_ubs/helper/urladdress.dart';
import 'package:m_student_ubs/model/hasilstudy.dart';
import 'package:m_student_ubs/model/user.dart';
import 'dart:convert';

class RencanaStudy extends StatefulWidget {
  static const tag = 'rencana-study';
  _RencanaStudyState createState() => _RencanaStudyState();
}

class _RencanaStudyState extends State<RencanaStudy> {
  CListView listView;
  UserModel _user = new UserModel();

  //List<String> semester = List.generate(7, (v){ final s = v+1; return '$s';
});
  //String cSemester = '1';

```

```

bool isExist = false;
int _sks = 0;

_RencanaStudyState(){
    listView = new CListView(
        itemBuilder: (b,index){
            return items(index);
        },
        onRefresh: (){
            refresh(1);
        }
    );
}

void initState() {
    super.initState();
    isExist = true;
    UserModel().getUser().then((e) {
        _user = e;
        refresh();
    });
}

Future<void> refresh([int mode=0]) async{
    if(!listView.isLoading){
        setState((){ listView.isLoading = true; });

        await loadDB();
        if(listView.data.length == 0 || mode == 1){
            await loadServer();
            await loadDB();
            // await _loadSmtDB();
        }

        setState((){ listView.isLoading = false; });
    }
}

Future<void> loadDB()async{
    listView.data = await HasilstudyModel().read(where: "nim=? AND
grade_akhir=''", whereArgs: [_user.nim]);
    _sks = 0;
    listView.data.forEach((m) {
        _sks += int.parse('0${m['sks']}');
    });
}

Future<void> loadServer()async{
    final r = await HTTP.o().sendPOST(UrlAddr.URL_KHS, {});
    final js = json.decode(r);
    if(js!= null){
        for(Map m in js['data']){
            final md =HasilstudyModel().fromMap(m);
            md.replace( 'nim=? AND kd_mtk=?', [md.nim, md.kd_mtk]);
        }
    }
}

```

```

Future<void> _loadSmtDB() async{
  /*
    _user = await UserModel().getUser();
    final hsm = await HasilstudyModel().read(columns: ['smt'], where:
'nim=?', whereArgs: [_user.nim], groupBy: 'smt');
    if(hsm.length > 0){
      semester.clear();
      hsm.forEach((f){
        semester.add('${f['smt']}');
      });
      setState((){});
    }
  */
}

@override
void dispose() {
  // TODO: implement dispose
  isExist = false;
  super.dispose();
}

@override
void setState(fn) {
  // TODO: implement setState
  if(isExist) super.setState(fn);
}

/*Widget _cardSelectSemester(){
  return Card(
    child: Container(
      padding: EdgeInsets.all(10.0),
      width: double.infinity,
      child: Center(
        child: DropdownButton(
          items: semester.map((f){
            return DropdownMenuItem(
              value: '$f',
              child: SizedBox(width: 200.0, child:
Text('Semester $f'),)
            );
          }).toList() ,
          value: cSemester,
          onChanged: (s){
            cSemester = s;
            refresh();
            setState((){});
          },
        ),
      ),
    ),
  ),
)

```

```

    );
  }*/

Widget items(int index){
  final d = HasilstudyModel().fromMap( listView.data[index] );
  return Container(
    color: index % 2 == 0 ? Colors.grey.shade100 : Colors.white,
    child: ListTile(
      title: Text('${d.kd_mtk} - ${d.mtk}'),
      subtitle: Text('${d.sks} SKS'),
    )
  );
}

Widget _resumeSKS(){
  return Container(
    child: Center(
      child: Text('Total ${listView.data.length} Matakuliah | $_sks
SKS'),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: TitleActionBar(subTitle: 'Rencana Study',),
    ),
    body: Column(
      children: <Widget>[
        // Expanded(child:_cardSelectSemester(), flex:1),
        Expanded(child:listView.build(context), flex:9),
        Expanded(child: _resumeSKS())
      ],
    )
  );
}

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

class SKIPage extends StatefulWidget {
  final Widget child;

  SKIPage({Key key, this.child}) : super(key: key);

  _SKIPageState createState() => _SKIPageState();
}

class _SKIPageState extends State<SKIPage> {

  @override
  Widget build(BuildContext context) {
    return Container(

```

```

        child: widget.child,
      );
    }
  }

import 'package:flutter/material.dart';
import 'package:m_student_ubsj/helper/http_api.dart';
import 'package:m_student_ubsj/helper/urladdress.dart';
import 'package:m_student_ubsj/utills/mywidget.dart';
import 'package:m_student_ubsj/model/hasilstudy.dart';
import 'package:m_student_ubsj/model/user.dart';
import 'dart:convert';

class HasilStudySemester extends StatefulWidget {
  static const tag = 'hasil-study-semester';
  _HasilStudySemesterState createState() => _HasilStudySemesterState();
}

class _HasilStudySemesterState extends State<HasilStudySemester> {
  bool _isExist = false;
  int _csmt = 1;
  int sks = 0;
  double ip = 0.0;
  Map maps = new Map();
  UserModel _user;
  List _listSemester = [1,2,3,4,5,6,7];
  CListView cListView;

  _HasilStudySemesterState(){
    cListView = new CListView(
      itemBuilder: (b,index){
        return _itemList(index);
      }, onRefresh: (){
        _refreshData(1);
      }
    );
  }

  void initState() {
    _isExist = true;
    super.initState();

    UserModel().getUser().then((e) {
      this._user = e;
      _getListSemester();
      _refreshData();
    });
  }

  @override
  void dispose() {
    // TODO: implement dispose
    _isExist = false;
    super.dispose();
  }
}

```

```

    }

    @override
    void setState(fn) {
      // TODO: implement setState
      if(!_isExist) super.setState(fn);
    }

    Future<void> _getListSemester() async{
      final ls = await HasilstudyModel().read(columns: ['smt'], where: 'nim=?',
whereArgs: [_user.nim],
                                           groupBy: 'smt'
                                           );

      if(ls.length > 0){
        _listSemester.clear();
        ls.forEach((f){
          _listSemester.add(int.parse(f['smt']));
        });
      }
      setState((){});
    }

    Widget _cardPilihSemester(){
      return Expanded(
        flex: 1,
        child: Card(
          child: Container(
            width: double.infinity,
            child: Center(child: DropdownButton(
              items: _listSemester.map((val){
                return
                DropdownMenuItem(
                  value: val,
                  child:
                  SizedBox(width:200.0, child: Text('Semester $val'),),
                ),
              ).toList(),
              value: _csmt,
              onChanged: (val){_csmt = val;
                _refreshData(); setState((){}); },
            ),
          ),
        ),
      );
    }

    Future<void> loadDB() async{
      final ls = await HasilstudyModel().read(where: 'nim=? AND smt=?',
whereArgs: [_user.nim, _csmt]);
      cListView.data = ls;
    }

    Future<void> loadServer() async{

```

```

String response = await HTTP.o().sendPOST(UrlAddr.URL_KHS, {});
final js = json.decode(response);

if(js['data']!=null){
  for(Map m in js['data']){
    final khs = HasilstudyModel().fromMap(m);
    await khs.replace( 'nim=? AND kd_mtk=?', [khs.nim, khs.kd_mtk]);
  }
}

Future<void> _refreshData([int mode=0])async{
  setState((){ cListView.isLoading = true; });
  await loadDB();
  if(mode == 1 || cListView.data.length == 0){
    await loadServer();
    await loadDB();
  }

  sks =0; ip = 0.0;
  int mutu = 0;

  for(int i=0;i<cListView.data.length;i++){
    final d = cListView.data[i];
    sks += d['sks'];
    mutu += d['mutu'];
  }
  ip = mutu/sks;

  setState((){ cListView.isLoading = false; });
}

Widget _listViewHasil(){
  return Expanded(
    flex: 10,
    child: cListView.build(context)
  );
}

Widget _itemList(int index){
  final h = cListView.data[index];
  Map<String,Color> mc = new Map();
  mc = {'A' : Colors.cyan, 'B' : Colors.blueGrey, 'C' : Colors.amber,
    'D' : Colors.orange, 'E' : Colors.red };
  var c = mc[h['grade_akhir']];
  c = c == null ? Colors.white : c;
  return ListTile(
    leading: Container(
      width: 40.0,
      height: 40.0,
      child: Center(child:Text('${h['grade_akhir']}',
        style: TextStyle(fontSize: 20.0,
          fontWeight:
FontWeight.bold,
          color: Colors.white
        ),
      ),
    ),
  );
}

```

```

        ),
        decoration: BoxDecoration(shape: BoxShape.circle, color: c)
    ),

    title: Text('${h['kd_mtk']} .- ${h['mtk']}'),
    subtitle: Text('${h['sks']} SKS '),
  );
}

Widget _panelResume(){
  return Expanded(
    flex:1,
    child: Container(
      color: Colors.blueGrey,
      child: Center(
        child:Text('${cListView.data.length} Matakuliah. $sks SKS. IP
= ${ip.toStringAsFixed(2)}',
          style: TextStyle(color: Colors.white, fontWeight:
FontWeight.bold)
        )
      ),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: TitleActionBar(subTitle: 'Hasil Studi Per Semester',),
    ),body: Column(
      children: <Widget>[
        _cardPilihSemester(),
        _listViewHasil(),
        _panelResume()
      ],
    ),
  );
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'package:m_student_ubsu/model/suratriset.dart';
import 'package:m_student_ubsu/model/user.dart';

class SuratRisetPage extends StatefulWidget {

  SuratRisetPage() : super();

  _SuratRisetPageState createState() => _SuratRisetPageState();
}

class _SuratRisetPageState extends State<SuratRisetPage> {
  List data;
  UserModel _user;

```

```

bool _loading = false;

_SuratRisetPageState(){
  data = new List();
}

Future _refresh()async{
  setState(){ _loading=true; });
  await Sync.listSuratRiset();
  _user = await UserModel().getUser();
  data = await SuratrisetModel().read(where: 'nim=?', whereArgs:
[_user.nim]);

  setState(){ _loading=false; });
}

Widget item(dynamic f){
  final s = SuratrisetModel().fromMap(f);
  return Card(
    child: Container(
      padding: EdgeInsets.all(10.0),
      child: Text('${s.no}'),
    ),
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: TitleActionbar(subTitle: 'Surat Pengantar Riset'),
    ),
    body: ListView(
      children: data.map((f){
        return item(f);
      }).toList()
    ),
  );
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/user.dart';
import './sub_tagihan/tagihan_pageview.dart';
import './sub_tagihan/bayarkuliah_pageview.dart';
import './sub_tagihan/bayarlainnya_pageview.dart';

class Tagihan extends StatefulWidget {
  static const tag = 'tagihan';
  _TagihanState createState() => _TagihanState();
}

class _TagihanState extends StateExtends<Tagihan> {
  TagihanPageView _tagihanPageView;
  BayarKuliahPageView _bayarKuliahPageView;
  BayarLainnyaPageView _bayarLainnyaPageView;

```

```

void initState() {
  super.initState();
  UserModel().getUser().then((u) {
    _tagihanPageView = new TagihanPageView(this, user: u);
    _bayarKuliahPageView = new BayarKuliahPageView(this, user: u);
    _bayarLainnyaPageView = new BayarLainnyaPageView(this, user:u);
  });
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    body: DefaultTabController(
      length: 3,
      child: NestedScrollView(
        headerSliverBuilder: (BuildContext ctx, bool
innerIsScroll){
          return <Widget>[
            SliverAppBar(
              floating: true,
              pinned: true,
              title: TitleActionBar(subTitle: 'Tagihan dan
Pembayaran',),,
              bottom: TabBar(
                indicatorColor: Colors.redAccent,
                indicatorWeight: 3.0,
                isScrollable: true,
                tabs: <Widget>[
                  Tab( text: 'Tagihan',),,
                  Tab( text: 'Pembayaran Kuliah',),,
                  Tab( text: 'Pembayaran Lainnya',),,
                ],
              ),
            ),
          ],
        ),
      ),
    );
  /* return Scaffold(
    appBar: AppBar(
      title: TitleActionBar(subTitle: 'Tagihan dan
Pembayaran'),,
    ),
    body: DefaultTabController(
      length: 3,

```

```

        child: Column(
          children: <Widget>[
            Container(
              child: Material(
                color: Colors.indigo,
                child: TabBar(
                  indicatorColor: Colors.redAccent,
                  indicatorWeight: 3.0,
                  isScrollable: true,
                  tabs: <Widget>[
                    Tab( text: 'Tagihan',),
                    Tab(      text:      'Pembayaran
Kuliah',),
                    Tab(      text:      'Pembayaran
Lainnya',),
                  ],
                ),
              ),
            ),
            Expanded(
              child: TabBarView(
                children: <Widget>[
                  Tab(child:
                    _tagihanPageView.build(context) ),
                  Tab(child:
                    _bayarKuliahPageView.build(context) ),
                  Tab(child:
                    _bayarLainnyaPageView.build(context) ),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  */

}
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/payment.dart';
import 'package:m_student_ubsu/model/user.dart';
import 'package:intl/intl.dart';
import 'package:m_student_ubsu/helper/http_api.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'dart:convert';

class BayarKuliahPageView{
  UserModel user;
  StateExts state;
  CListView lv;
  int total = 0;

```

```

BayarKuliahPageView(this.state, {this.user}){
    lv = new CListView(
        itemBuilder: (b,index)=>_item(index),
        onRefresh: ()=>_refresh(1)
    );

    _refresh();
}

Future<void> _loadDB()async{
    lv.data = await PaymentModel().read(where: 'nim=? AND (tgl_byr <> ?
OR tgl_byr IS NOT NULL ) AND SUBSTR(kd_jnsbyr,1,1)=?',
                                whereArgs: [user.nim,
'', 'K'], orderBy: 'tgl_byr DESC');
}

Future<void> _refresh([int mode=0])async{
    lv.isLoading = true; state.updateDisplay();

    await _loadDB();

    if(lv.data.length == 0 || mode==1 ){
        print('load invoice tagihan');
        await Sync.invoice();
        await _loadDB();
    }

    total = 0;
    String lastTgl = '';
    List tmp = new List();
    lv.data.forEach((m){
        final p = new PaymentModel().fromMap(m);
        total += p.total + p.nominal;

        final tgl =Convert.bulanTahun(p.tgl_byr);
        if(tgl !=lastTgl){
            lastTgl = tgl;
            tmp.add( {'jam':'---', 'tgl_byr':tgl} );
        }
        tmp.add(m);
    });
    lv.data = tmp;
    lv.isLoading = false; state.updateDisplay();
}

Widget _item(int index){
    final p = PaymentModel().fromMap(lv.data[index]);
    if(p.jam == '---'){
        return Container(
            padding: EdgeInsets.all(5.0),
            child:Text('${p.tgl_byr}', style: TextStyle(fontWeight:
FontWeight.bold))
        );
    }
}

```

```

final formatter = new NumberFormat('#,###');
return Card(
  child: Container(
    padding: EdgeInsets.all(10.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        IconLabel(Text('${p.nm_jnsbyr}', style:
TextStyle(fontWeight: FontWeight.bold), ), icon:FontAwesomeIcons.tag),
        SizedBox(height: 10.0,),
        IconLabel(Text('Tanggal : ${Convert.tanggal(p.tgl_byr)
} | ${p.jam}'), icon:FontAwesomeIcons.calendar),
        SizedBox(height: 10.0,),
        IconLabel(Text('${p.kd_lokal}'),
icon:FontAwesomeIcons.flag),
        SizedBox(height: 10.0,),
        IconLabel(Text('${p.atm}      oleh      ${p.penyetor}'),
icon:FontAwesomeIcons.userAlt),
        SizedBox(height: 10.0,),
        IconLabel(Text('SSP      :      ${formatter.format(p.ssp)}'),
icon:FontAwesomeIcons.moneyBillAlt),
        SizedBox(height: 10.0,),
        IconLabel(Text('Kuliah      :
${formatter.format(p.kuliah)}'), icon:FontAwesomeIcons.moneyBillWave),
        SizedBox(height: 10.0,),
        IconLabel(Text('Total : ${formatter.format(p.total)}',
style:      TextStyle(fontWeight:      FontWeight.bold),      textAlign:
TextAlign.right,)),
      ],
    ),
  ),
);
}

```

```

Widget build(BuildContext context){
  final formatter = NumberFormat('#,###');
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: <Widget>[
      Expanded(
        flex: 11,
        child: lv.build(context),
      ),
      Expanded(
        flex:1,
        child: Container(
          color: Colors.indigo,
          width: double.infinity,
          padding: EdgeInsets.fromLTRB(10.0, 15.0, 10.0, 20.0),
          child:
            Center(child:Text('Total      IDR.
${formatter.format(total)}',      style:      TextStyle(color:      Colors.white,
fontWeight: FontWeight.bold),))
          ),
      ],
    ),
  ),
);
}

```

```

    );
  }
}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/payment.dart';
import 'package:m_student_ubsu/model/user.dart';
import 'package:intl/intl.dart';
import 'package:m_student_ubsu/helper/http_api.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';

class BayarLainnyaPageView{
  UserModel user;
  StateExts state;
  CListView lv;
  int total = 0;

  BayarLainnyaPageView(this.state, {this.user}){
    lv = new CListView(
      itemBuilder: (b,index)=>_item(index),
      onRefresh: ()=>_refresh(1)
    );

    _refresh();
  }

  Future<void> _loadDB()async{
    lv.data = await PaymentModel().read(where: 'nim=? AND
SUBSTR(kd_jnsbyr,1,1)<>? AND (tgl_byr <> ? OR tgl_byr IS NOT NULL )',
                                whereArgs: [user.nim,
'K', ''], orderBy: 'tgl_byr DESC');
  }

  Future<void> _refresh([int mode=0])async{
    lv.isLoading = true; state.updateDisplay();

    await _loadDB();

    if(lv.data.length == 0 || mode==1 ){
      await Sync.otherPayment();
      await _loadDB();
    }

    total = 0;
    String lastTgl = '';
    List tmp = new List();
    lv.data.forEach( (m) {
      final p = new PaymentModel().fromMap(m);
      total += p.total + p.nominal;

      final tgl =Convert.bulanTahun(p.tgl_byr);
      if(tgl !=lastTgl){

```

```

        lastTgl = tgl;
        tmp.add( {'jam':'---', 'tgl_byr':tgl} );
    }
    tmp.add(m);
});
lv.data = tmp;
lv.isLoading = false; state.updateDisplay();
}

Widget _item(int index){
    final p = PaymentModel().fromMap(lv.data[index]);
    if(p.jam == '---'){
        return Container(
            padding: EdgeInsets.all(5.0),
            child:Text('${p.tgl_byr}', style: TextStyle(fontWeight:
FontWeight.bold))
        );
    }

    final formatter = new NumberFormat('#,###');
    List<Widget> listitems = List();

    listitems.add( Text('${p.nm_jnsbyr}', style: TextStyle(fontWeight:
FontWeight.bold),) );
    listitems.add(SizedBox(height: 10.0,));

    if(p.nominal!=0){
        listitems.add(
RowDataIconLabel(icon:FontAwesomeIcons.fileInvoiceDollar,
wLeft:Text('Biaya'), wRight: Text('Rp.  ${formatter.format(p.nominal)}',
textAlign: TextAlign.right,) ) );
    }

    listitems.add( RowDataIconLabel(icon:FontAwesomeIcons.userCircle,
wLeft:Text('Penyeter'), wRight: Text('${p.atm}, ${p.penyeter}', textAlign:
TextAlign.right,) ) );
    listitems.add( RowDataIconLabel(icon:FontAwesomeIcons.clock,
wLeft:Text('Waktu'), wRight: Text('${p.tgl_byr}, ${p.jam}', textAlign:
TextAlign.right,) ) );
    listitems.add( IconLabel(Text('Tagihan : ${formatter.format(p.total +
p.nominal)}', style: TextStyle(fontWeight: FontWeight.bold), textAlign:
TextAlign.right,)) );

    return Card(
        child: Container(
            padding: EdgeInsets.all(10.0),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: listitems
            ),
        ),
    );
}

Widget build(BuildContext context){

```

```

final formatter = NumberFormat('#,###');
return Column(
  mainAxisAlignment: MainAxisAlignment.center,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: <Widget>[
    Expanded(
      flex: 11,
      child: lv.build(context),
    ),
    Expanded(
      flex:1,
      child: Container(
        color: Colors.indigo,
        width: double.infinity,
        padding: EdgeInsets.fromLTRB(10.0, 15.0, 10.0, 20.0),
        child: Center(child:Text('Total Pembayaran IDR.
${formatter.format(total)}', style: TextStyle(color: Colors.white,
fontWeight: FontWeight.bold),))
      ),
    ],
  );
}

}

import 'package:flutter/material.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:m_student_ubsu/model/payment.dart';
import 'package:m_student_ubsu/model/user.dart';
import 'package:intl/intl.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsu/load_online/sync_db.dart';
import 'dart:convert';

class TagihanPageView{
  UserModel user;
  StateExts state;
  CListView listTagihan;
  int totalTagihan = 0;

  TagihanPageView(this.state, {this.user}){
    listTagihan = new CListView(
      itemBuilder: (b,index)=>itemTagihan(index),
      onRefresh: ()=>_refreshTagihan(1)
    );

    _refreshTagihan();
  }

  Future<void> _loadTagihanDB()async{
    listTagihan.data = await PaymentModel().read(where: 'nim=? AND
(tgl_byr = ? OR tgl_byr IS NULL )',
                                                    whereArgs:
[user.nim, ''], orderBy: 'tgl_byr DESC');
  }
}

```

```

Future<void> _refreshTagihan([int mode=0])async{
    listTagihan.isLoading = true; state.updateDisplay();

    await _loadTagihanDB();

    if(listTagihan.data.length == 0 || mode==1 ){
        print('load invoice tagihan');
        await Sync.invoice();
        await _loadTagihanDB();
    }

    totalTagihan = 0;
    listTagihan.data.forEach((m){
        final p = new PaymentModel().fromMap(m);
        totalTagihan += p.total + p.nominal;
    });

    listTagihan.isLoading = false; state.updateDisplay();
}

Widget itemTagihan(int index){
    final p = PaymentModel().fromMap(listTagihan.data[index]);
    final formatter = new NumberFormat('#,###');
    List<Widget> listitems = List();

    listitems.add(
        IconLabel(Text('${p.nm_jnsbyr}', style:
TextStyle(fontWeight: FontWeight.bold), ), icon:FontAwesomeIcons.tag) );
    listitems.add( SizedBox(height: 10.0) );
    if(p.ssp != 0){
        listitems.add(
            IconLabel(Text('SSP
${formatter.format(p.ssp)}'), icon: FontAwesomeIcons.fileInvoice, ) );
        listitems.add( SizedBox(height: 10.0) );
    }

    if(p.kuliah!=0){
        listitems.add(
            IconLabel(Text('Kuliah
${formatter.format(p.kuliah)}'), icon: FontAwesomeIcons.fileInvoiceDollar, )
        );
        listitems.add( SizedBox(height: 10.0) );
    }

    if(p.nominal!=0){
        listitems.add(
            IconLabel(Text('Biaya
${formatter.format(p.nominal)}'), icon: FontAwesomeIcons.fileInvoiceDollar, )
        );
        listitems.add( SizedBox(height: 10.0) );
    }

    listitems.add( IconLabel(Text('Tagihan : ${formatter.format(p.total +
p.nominal)}', style: TextStyle(fontWeight: FontWeight.bold), textAlign:
TextAlign.right,)) );

    return Card(

```

```

        child: Container(
          padding: EdgeInsets.all(10.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: listitems
          ),
        ),
      ),
    );
  }

Widget _tagihanContent(BuildContext context){
  final formatter = NumberFormat('#,###');
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: <Widget>[
      Expanded(
        flex: 11,
        child: listTagihan.build(context),
      ),
      Expanded(
        flex:1,
        child: Container(
          color: Colors.indigo,
          width: double.infinity,
          padding: EdgeInsets.fromLTRB(10.0, 15.0, 10.0, 20.0),
          child: Center(child:Text('Total Tagihan IDR.
${formatter.format(totalTagihan)}', style: TextStyle(color: Colors.white,
fontWeight: FontWeight.bold),))
        ),
      ),
    ],
  );
}

@override
Widget build(BuildContext context) {
  return _tagihanContent(context);
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsj/model/skpi.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:m_student_ubsj/utills/mywidget.dart';
import 'package:m_student_ubsj/helper/http_api.dart';
import 'package:m_student_ubsj/helper/urladdress.dart';
import 'package:m_student_ubsj/load_online/sync_db.dart';
import 'package:file_picker/file_picker.dart';
import 'dart:io';
import 'dart:convert';
import 'package:path/path.dart';
import 'package:m_student_ubsj/page/webview_page.dart';

typedef OnFinishSave(bool state);

```

```

class SKPIForm extends StatefulWidget {
  final SkpiModel skpi;
  final OnFinishSave onFinishSave;

  SKPIForm({this.skpi, this.onFinishSave});

  @override
  _SKPIFormState createState() => _SKPIFormState(skpi:this.skpi,
onFinishSave: this.onFinishSave);
}

class _SKPIFormState extends State<SKPIForm> {
  final OnFinishSave onFinishSave;
  SkpiModel skpi;
  String tmpBukti1 = '', tmpBukti2 = '';
  File fbuktiPenghargaan, fbuktiMagang;
  TextEditingController _nmOrganisasi, _jabatan, _nmKegiatan, _capaian,
_nmInstansi, _posisi;
  bool _isloading = false;

  _SKPIFormState({this.skpi, this.onFinishSave});

  void initState() {
    super.initState();
    _nmOrganisasi = new TextEditingController(text: skpi.nm_organisasi);
    _jabatan = new TextEditingController(text: skpi.jabatan);
    _nmKegiatan = new TextEditingController(text: skpi.nm_kegiatan);
    _capaian = new TextEditingController(text: skpi.capaian);
    _nmInstansi = new TextEditingController(text: skpi.nm_instansi);
    _posisi = new TextEditingController(text: skpi.posisi);
  }

  Widget _cardOrganisasi(){
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        Padding(padding: EdgeInsets.only(left: 10.0, top:10.0),
child:Text('Pengalaman Organisasi', style: sb1)),
        Card(
          margin: EdgeInsets.all(10.0),
          elevation: 7.0,
          child: Container(
            padding: EdgeInsets.all(20.0),
            child: Column(
              children: <Widget>[
                TextField( controller: _nmOrganisasi,
                  decoration: InputDecoration( labelText:
'Nama Organisasi', labelStyle: sb1, icon: Icon(FontAwesomeIcons.tags, size:
15.0,) )
                ),
                TextField( controller: _jabatan,
                  decoration: InputDecoration( labelText:
'Jabatan', labelStyle: sb1, icon: Icon(FontAwesomeIcons.chartLine, size:
15.0) )
                ),

```

```

        DropDownIcon(label: 'Tahun Pengalaman',
value: skpi.thn_1,
        onChanged: (e){
            setState(() { skpi.thn_1 = e; });
        },
        iconData: FontAwesomeIcons.calendarAlt,
        listitems: List.generate(8, (idx){ return
DateTime.now().year-idx; })),
    ),
    ],
    ),
    ),
    ],
    );
}

List<Widget> listInputPenghargaan(){
    List ls = <Widget>[
        TextField( controller: _nmKegiatan,
            decoration: InputDecoration( labelText: 'Nama Kegiatan',
labelStyle: sb1, icon: Icon(FontAwesomeIcons.clipboardList, size: 15.0) )
        ),
        TextField( controller: _capaian,
            decoration: InputDecoration( labelText: 'Capaian
Prestasi', labelStyle: sb1, icon: Icon(FontAwesomeIcons.trophy, size: 15.0) )
        ),
        DropDownIcon(label: 'Tahun', value: skpi.thn_2,
            onChanged: (e){
                setState(() { skpi.thn_2 = e; });
            },
            iconData: FontAwesomeIcons.calendarAlt,
            listitems: List.generate(8, (idx){ return
DateTime.now().year-idx; })),
        SizedBox(height: 10.0, ),
        btnBukti('Bukti Penghargaan', 1, btnBuktiPenghargaan_onclick)
    ];

    if(skpi.bukti1 != ''){
        ls.add(RaisedButton(
            onPressed: (){
                confirm(this.context, 'Hapus Bukti Penghargaan',
                    'Bukti Penghargaan ${skpi.bukti1} akan dihapus,
mau dilanjutkan?', (s){
                    if(s){
                        hapusBukti(1);
                    }
                } );
            },

```

```

                shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20.0)),
                color: Colors.red, textColor: Colors.white,
                child: Text('Hapus File Bukti'),
            ));
        }

        return ls;
    }

    Widget _cardPenghargaan(){
        return Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: <Widget>[
                Padding(padding: EdgeInsets.only(left: 10.0, top:10.0),
child:Text('Penghargaan / Prestasi', style: sb1)),
                Card(
                    margin: EdgeInsets.all(10.0),
                    elevation: 7.0,
                    child: Container(
                        padding: EdgeInsets.all(20.0),
                        child: Column(
                            children: listInputPenghargaan()
                        ),
                    ),
                ),
            ],
        );
    }

    Widget _cardMagang(){
        return Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: <Widget>[
                Padding(padding: EdgeInsets.only(left: 10.0, top:10.0),
child:Text('Pengalaman Magang', style: sb1)),
                Card(
                    margin: EdgeInsets.all(10.0),
                    elevation: 7.0,
                    child: Container(
                        padding: EdgeInsets.all(20.0),
                        child: Column(
                            children: listInputMagang()
                        ),
                    ),
                ),
            ],
        );
    }

    List<Widget> listInputMagang(){
        List lst = <Widget>[
            TextField( controller: _nmInstansi,
                decoration: InputDecoration( labelText: 'Nama
Instansi', labelStyle: sb1, icon: Icon(FontAwesomeIcons.building, size: 15.0)
            )
        ],
    }

```

```

        TextField( controller: _posisi,
                    decoration: InputDecoration( labelText: 'Posisi',
labelStyle: sb1, icon: Icon(FontAwesomeIcons.addressCard, size: 15.0) )
                ),

        DropDownIcon(label: 'Lama', value: skpi.lamanya,
onChanged: (e){
                    setState(() { skpi.lamanya = e; });
                },
                iconData: FontAwesomeIcons.calendarAlt,
                dropItems: List.generate(6, (idx){
                    int bln = idx+1;
                    return DropdownMenuItem(
                        value: '${bln}',
                        child: Text('$bln bulan'),
                    );
                }
            ),

        DropDownIcon(label: 'Tahun', value: skpi.thn_3,
onChanged: (e){
                    setState(() { skpi.thn_2 = e; });
                },
                iconData: FontAwesomeIcons.calendarAlt,
                listItems: List.generate(8, (idx){
                    return
DateTime.now().year-idx;
                }
            ),

        SizedBox(height: 10.0,),

        btnBukti('Bukti Magang', 2, btnBuktiMagang_onclick)

    ];
    if(skpi.bukti2 != ''){
        lst.add(RaisedButton(
            onPressed: (){
                confirm(this.context, 'Hapus Bukti Magang',
                    'Bukti Magang ${skpi.bukti2} akan dihapus, mau
dilanjutkan?', (s){
                    if(s){
                        hapusBukti(2);
                    }
                }
            );
        },
        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20.0)),
        color: Colors.red, textColor: Colors.white,
        child: Text('Hapus File Bukti'),
    ));
    }
    return lst;
}

Future<bool> hapusBukti(int idx) async{
    _isLoading = true;
    bool ret = false;

```

```

        setState(() {});

        final senddata = { 'file' : idx == 1? skpi.bukti1 : skpi.bukti2,
'idx':idx, 'periode': skpi.periode};
        final response = await HTTP.o().sendPOST(UrlAddr.URL_SKPI_DELETE,
senddata);

        print('isi $senddata response $response');
        try{
            final js = json.decode(response);
            if(js['result'] == 1){
                if(idx==1){
                    ret = true;
                    skpi.bukti1 = '';
                    fbuktiPenghargaan = null;
                }else{
                    skpi.bukti2 = '';
                    fbuktiMagang = null;
                }
            }
        }catch(e){}
        _isLoading = false;
        setState(() {});
        return ret;
    }

    Widget btnBukti(String title, int idx, VoidCallback onTap){
        String deskripsi = idx == 1 ? '${skpi.bukti1}' : '${skpi.bukti2}';
        if(deskripsi == ''){
            deskripsi = (idx == 1 ? (tmpBukti1 == '' ? '' : '${tmpBukti1}') :
(tmpBukti2 == '' ? '' : '${tmpBukti2}')) );
            if(deskripsi != '') deskripsi += ' (siap di unggah)';
        }
        return Column(
            children: <Widget>[
                Text('$deskripsi'),
                RaisedButton(
                    onPressed: onTap,
                    child: Text('$title', style: TextStyle(color:
Colors.white)),),
                color: Colors.teal,
                shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20.0)),)
            ],
        );
    }

    Future<bool> simpanSKPI()async{

        bool ret = false;
        try{
            _isLoading = true;
            setState(() {});

            List fupload = new List();

```

```

        if(fbuktiMagang != null){
            fupload.add({'b64': base64Encode(await fbuktiMagang.readAsBytes()),
'idx':2, 'ext':extension(fbuktiMagang.path) });
            fbuktiMagang = null;
            skpi.bukti2 = '';
        }

        if(fbuktiPenghargaan != null){
            fupload.add({'b64': base64Encode(await
fbuktiPenghargaan.readAsBytes()),
'idx':1,
'ext':extension(fbuktiPenghargaan.path) });
            fbuktiPenghargaan = null;
            skpi.bukti2 = '';
        }

        final datasend = {
            'periode' : skpi.periode,
            'nm_organisasi' : _nmOrganisasi.text,
            'jabatan' : _jabatan.text,
            'thn_1' :skpi.thn_1,
            'nm_kegiatan' : _nmKegiatan.text,
            'capaian' : _capaian.text,
            'thn_2': skpi.thn_2,
            'nm_instansi' : _nmInstansi.text,
            'posisi' : _posisi.text,
            'lamanya': skpi.lamanya,
            'thn_3' : skpi.thn_3,
            'files' : fupload
        };

        final response = await HTTP.o().sendPOST(UrlAddr.URL_SKPI_SAVE,
datasend);
        print('response simpan response $response');
        final f = json.decode(response);
        if(f['state'] > 0){
            ret = true;
            await Sync.getSKPI();
        }

    }catch(e){
        print('error simpan $e');
    }

    tmpBukti1 = '';
    tmpBukti2 = '';

    if(onFinishSave != null){
        onFinishSave(ret);
    }

    await Sync.getSKPI();
    final ls = await SkpiModel().read(where: 'nim=? AND periode=?',
whereArgs: [skpi.nim, skpi.periode]);
    if(ls.length > 0){
        skpi = SkpiModel().fromMap(ls[0]);
    }
}

```

```

        setState(() { _isLoading = false; });

        return ret;
    }

    void btnBuktiPenghargaan_onclick() {
        if(skpi.bukti1 == '') {
            FilePicker.getFile().then((f) {
                fbuktiPenghargaan = f;
                if(f==null){
                    tmpBukti1 = '';
                } else if(f.lengthSync()/1024 < 3072) {
                    tmpBukti1 = basename(f.path);
                } else {
                    alert(this.context, 'Ukuran file', 'Ukuran File tidak boleh
lebih dari 3MB');
                }
                setState(() {});
            });
        } else {
            Navigator.push(this.context, new MaterialPageRoute(
                builder: (bc)=>WebView(
                    settings: {'url':UrlAddr.PATH_SKPI_BUKTI + '1-
${skpi.bukti1}',
                                'title':'Bukti Magang' })
                ));
        }
    }

    void btnBuktiMagang_onclick() {
        if(skpi.bukti2 == '') {
            FilePicker.getFile().then((f) {
                fbuktiMagang = f;

                if(f == null){ tmpBukti2 = ''; }
                else if(f.lengthSync()/1024 < 3072 ) { tmpBukti2 =
basename(f.path); }
                else {
                    alert(this.context, 'Ukuran file', 'Ukuran File tidak boleh
lebih dari 3MB');
                }
                setState(() {});
            });
        } else {
            Navigator.push(this.context, new MaterialPageRoute(
                builder: (bc)=>WebView(
                    settings: {'url':UrlAddr.PATH_SKPI_BUKTI + '2-
${skpi.bukti2}',
                                'title':'Bukti Magang' })
                ));
        }
    }

    @override
    Widget build(BuildContext context) {

```

```

        return _isLoading ? Center(child: CupertinoActivityIndicator(radius:
20.0,)) : Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
        _cardOrganisasi(),
        _cardPenghargaan(),
        _cardMagang(),
        Container(
            padding: EdgeInsets.all(10.0),
            child: RaisedButton(
                onPressed: () {
                    simpanSKPI().then((e) {
                        if (e) {
                            info(context, 'Simpan SKPI', 'SKPI
periode ${skpi.periode} Berhasil disimpan');
                        } else {
                            alert(context, 'Simpan SKPI', 'SKPI
periode ${skpi.periode} Gagal disimpan');
                        }
                    });
                },
                color: Colors.red,
                child: Text('Simpan',
style: TextStyle(color: Colors.white)),
                elevation: 3.0,
                shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(5.0)),
            )
        ),
    ],
);
}

final TextStyle sb1 = TextStyle(fontWeight: FontWeight.bold);
}

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:m_student_ubs/utills/mywidget.dart';
import 'package:m_student_ubs/model/skpi.dart';
import 'package:m_student_ubs/model/user.dart';
import 'package:m_student_ubs/load_online/sync_db.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import './skpi_form.dart';

class SKPIPage extends StatefulWidget {
    @override
    _SKPIPageState createState() => _SKPIPageState();
}

class _SKPIPageState extends State<SKPIPage> with TickerProviderStateMixin {
    Map<String, List> maps;
    UserModel _user;
    List _listTabs;
    TabController tabController;
    bool _loading = true;

```

```

_SKPIPageState() {
    maps = new Map();
    _listTabs = new List();
    tabController = new TabController(vsync: this, length: maps.length);
}

void initState() {
    super.initState();

    UserModel().getUser().then((u) {
        _user = u;
        refresh().then((onValue) {
            setState(() {});
        });
    });
}

Future refresh([int mode = 0]) async {
    setState(() { _loading = true; });
    List l = await SkpiModel().read(
        where: 'nim=?', whereArgs: [_user.nim],
        orderBy: 'nim ASC, periode ASC');
    print('refresh isi $l');

    if (l.length == 0 || mode == 1) {
        await Sync.getSKPI();
        l = await SkpiModel().read(
            where: 'nim=?', whereArgs: [_user.nim],
            orderBy: 'nim ASC, periode ASC');
    }

    _listTabs = new List();
    l.forEach((f) {
        final skpi = SkpiModel().fromMap(f);
        if (maps[skpi.periode] == null) {
            maps[skpi.periode] = new List();
        }
        maps[skpi.periode].add(skpi);
        _listTabs.add('$ ${skpi.periode}');
    });

    tabController = new TabController(vsync: this, length: maps.length);

    setState(() { _loading = false; });
    return maps;
}

Widget tabbarview() {
    return TabBarView(
        controller: tabController,
        children: _listTabs.map((f) {
            final dt = maps[f];
            final items = dt.map((itm) {
                return SKPIForm(skpi: itm);
            }).toList();
        }));
}

```

```

        return ListView(
          shrinkWrap: true,
          physics: AlwaysScrollableScrollPhysics(),
          children: items,
        );
      }).toList()
    );
  }

Widget content(){
  final c = maps.length == 0 ?
    Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Icon(FontAwesomeIcons.exclamationTriangle, size:
90.0, color: Colors.amber,)),
          SizedBox(height: 30.0,)),
          Text('Anda belum memiliki hak mengisi SKPI')
        ],
      ),
    ) : tabbarview();
  return _loading ? Center(child:CupertinoActivityIndicator()) : c;
}

Widget tabbar(){
  return maps.length > 0 ?
    TabBar(
      isScrollable: true,
      controller: tabController,
      tabs: _listTabs.map((f){
        return Tab(child: Text('Periode $f'),);
      }).toList()
    )
    : null;
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: TitleActionBar(subTitle: 'SKPI',),
      bottom: tabbar()
    ),
    body: content(),
  );
}
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsj/model/jadwal_kuliah.dart';
import 'package:m_student_ubsj/utlis/mywidget.dart';
import 'dart:io';
import 'package:path_provider/path_provider.dart';
import 'package:path/path.dart' as path;

```

```

import 'package:m_student_ubsj/helper/zipunzip.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:dio/dio.dart';
import 'package:percent_indicator/circular_percent_indicator.dart';
import './pdf_viewer_page.dart';

class ListMateriPage extends StatefulWidget {
  final Widget child;
  final JadwalKuliahModel model;
  final dynamic itemmateri;

  ListMateriPage({Key key, this.child, this.model, this.itemmateri}):
    super(key: key);

  _ListMateriPageState createState() => _ListMateriPageState(model: model,
    itemMateri: itemmateri);
}

class _ListMateriPageState extends State<ListMateriPage> {
  final JadwalKuliahModel model;
  final dynamic itemMateri;
  List data;
  String pathFiles = '';
  bool isLoading = false;
  bool isDownload = false;
  bool isModeDelete = false;
  double percentDownload = 0;
  int sizeDownload = 0, sizeProgress = 0;

  _ListMateriPageState({this.model, this.itemMateri}){
    data = new List();
  }

  void initState() {
    super.initState();
    enumFiles();
    isModeDelete = false;
  }

  Future enumFiles()async{
    isLoading = true;
    data = new List();
    setState(() {});

    final appDir = await getApplicationDocumentsDirectory();
    final dir = Directory(path.join(appDir.path, itemMateri['title'],
model.kd_mtk));
    pathFiles = dir.path;
    pathFiles = path.join(pathFiles, path.basename(itemMateri['link']));

    final exist = await dir.exists();
    if(!exist){ await dir.create(recursive: true); }

    List flist = dir.listSync();
    for(final ls in flist){
      final isfile = await File(ls.path).exists();
      print('file ${ls.path}');
    }
  }
}

```

```

        if(isfile){
            final ext = ls.path.substring(ls.path.length-3
).toLowerCase();
            if(ext == 'zip'){
                print('fileszip : ${ls.path} di ${dir.path}');
                final result = await UnZip(ls.path, dir.path).doUnzip();
                if(!result){
                    alert(context, 'Ekstrak ZIP', 'Gagal ekstrak file');
                }else{
                }
            }
        }
    }

    flist = dir.listSync();
    for(final ls in flist){
        File file = File(ls.path);

        if(file.existsSync()){
            final ext = ls.path.substring(ls.path.length-3
).toLowerCase();
            if(ext != 'zip'){
                data.add({'title':path.basename(ls.path), 'path':ls.path,
'size':file.lengthSync()});
            }
        }
    }

    data.sort((d1,d2){
        return d1['title'].toString().compareTo(d2['title'].toString());
    });

    isLoading = false;
    setState(() {});
}

Widget dataEmpties(){
    double ukuran = (itemMateri['size']/1024);
    return Center(
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
                Icon(FontAwesomeIcons.exclamationTriangle, size:
50.0, color: Colors.amberAccent,),
                SizedBox(height: 20.0,),
                Text('Berkas materi belum tersedia di memory hp'),
                SizedBox(height: 10.0,),
                OutlineButton(
                    color: Colors.transparent,
                    shape: RoundedRectangleBorder(borderRadius:
BorderRadius.all(Radius.circular(20.0))),
                    child: Text('Tap untuk mengunduh
${itemMateri['title']}'),
                    onPressed: (){
                        downloadFile();

```

```

        },
        SizedBox(height: 10.0,),
        Text('Ukuran ${ukuran.toStringAsFixed(2)} KB'),
      ],
    ),
  );
}

CancelToken _cancelToken;
Future downloadFile() async {
  _cancelToken = new CancelToken();
  setState(() { isDownload = true; });

  percentDownload = 0;
  try {
    print('download ${itemMateri['link']} ke $pathFiles');
    Response r = await Dio().download(itemMateri['link'], pathFiles,
cancelToken: _cancelToken, onReceiveProgress: (int received, int total) {
      setState(() {
        sizeDownload = total;
        sizeProgress = received;
        percentDownload = sizeProgress/sizeDownload ;
print('percent $percentDownload');

        if(percentDownload >= 1.0) {
          enumFiles();
        }
      });
    } );
    print(r);
  } catch (e) {
    print(e);
  }

  setState(() { isDownload = false; });
}

Widget downloadPage() {
  return Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        CircularPercentIndicator(
          radius: 70.0,
          animation: true,
          animateFromLastPercent: true,
          percent: percentDownload,
          progressColor: Colors.indigo,
          center: Icon(FontAwesomeIcons.download,
color:Colors.grey),
        ),
        Text('Mengunduh $sizeProgress / $sizeDownload '),
        SizedBox(height:10.0),
        OutlineButton(

```

```

                shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20.0)),
                child: Text('Batal'),
                onPressed: () {
                    _cancelToken.cancel('cancelled');
                    isDownload = false;
                    setState(() {});
                },
            ),
        ],
    ),
);
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: TitleActionbar(subTitle: '${itemMateri['title']}',),
            actions: btnActions(),
        ),
        body: isDownload ? downloadPage() : contentList()
    );
}

List<Widget> btnActions(){
    List<Widget> ret = new List();
    if(isModeDelete){
        int sel = 0;
        List<File> filelist = new List();
        data.forEach((f){
            sel += f['sel'] == 1 ? 1 : 0;
            if(f['sel'] == 1){
                filelist.add(File(f['path']));
            }
        });
        if(sel > 0){
            ret.add(IconButton(icon: Icon(FontAwesomeIcons.trash), iconSize: 17,
onPressed: (){
                confirm(context, 'Konfirmasi', '$sel Berkas akan dihapus, apakah
anda yakin?', (r){

                    Navigator.pop(context);
                    if(r){
                        filelist.forEach((f){
                            f.deleteSync();
                        });
                        enumFiles();
                    }
                    isModeDelete = false;
                    setState(() {});
                });
            }));
        }
    }
}

```

```

    return ret;
}

Widget itemFile(dynamic dt){
    double size = dt['size'] / 1024;

    return Card(
        child: InkWell(
            onTap: (){
                if(isModeDelete){
                    dt['sel'] = dt['sel'] != 1 ? 1 : 0;
                    setState((){});
                }else{
                    Navigator.of(context).push(new MaterialPageRoute(
                        builder: (bc)=>PdfMateriViewerPage(pdfFile:
dt['path'])
                    ));
                }
            },
            onLongPress: (){
                isModeDelete = !isModeDelete;
                setState((){});
            },
            child: Container(
                padding: EdgeInsets.all(20.0),
                child: Row(
                    children: <Widget>[
                        isModeDelete ? (dt['sel'] == 1 ?
Icon(FontAwesomeIcons.checkCircle) : Icon(FontAwesomeIcons.circle) ) :
Icon(FontAwesomeIcons.filePdf, color: Colors.redAccent),
                        SizedBox(width: 10.0),
                        Wrap(
                            direction: Axis.vertical,
                            children: <Widget>[
                                Text('${dt['title']}', style:
TextStyle(fontSize: 16.0, fontWeight: FontWeight.bold),),
                                Text('${ size.toStringAsFixed(2) } KB'),
                            ],
                        ),
                    ],
                ),
            ),
        ),
    );
}

Widget contentList(){
    return isLoading ? Center(child: CupertinoActivityIndicator()) :
        data.length == 0 ? dataEmpties() :
        ListView(
            children: data.map((f){
                return itemFile(f);
            }).toList()
        );
}
}

```

```

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubs/Utils/mywidget.dart';
import 'package:m_student_ubs/model/jadwal_kuliah.dart';
import 'package:m_student_ubs/helper/http_api.dart';
import 'package:m_student_ubs/helper/urladdress.dart';
import 'dart:convert';
import 'package:intl/intl.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import './listmateri_page.dart';

class MateriPage extends StatefulWidget {
  final Widget child;
  static const tag = 'materi-page';
  final JadwalkuliahModel jadwal;

  MateriPage({Key key, this.child, this.jadwal}) : super(key: key);

  _MateriPageState createState() => _MateriPageState(jadwal: jadwal);
}

class _MateriPageState extends State<MateriPage> {
  final JadwalkuliahModel jadwal;
  List data;
  bool isLoading = false;

  _MateriPageState({this.jadwal}){
    data = new List();
  }

  void initState() {
    super.initState();
    loadItem();
  }

  Future loadItem()async{
    isLoading = true;
    setState(() {});

    try{
      final ls = await JadwalkuliahModel().read(where: 'kd_mtk=?',
whereArgs: [jadwal.kd_mtk]);
      final jdw = JadwalkuliahModel().fromMap(ls[0]);

      this.data = json.decode(jdw.files);
    }catch(e){
      this.data = new List();
    }

    if(this.data.length == 0){
      final response = await HTTP.o().sendPOST(UrlAddr.URL_MATERI, {'kd':
jadwal.kd_mtk});
      print('response $response');
      try{
        final js = json.decode(response);
        this.data = js['data']['list'];
      }
    }
  }
}

```

```

        jadwal.files = json.encode(this.data);
        jadwal.update('kd_mtk=?', [jadwal.kd_mtk] );
    }catch(e){
        this.data = new List();
    }
}

isLoading = false;
setState((){});
}

Widget item(dynamic dt){
    final f = NumberFormat('#,###.##');
    return Card(
        child: InkWell(
            onTap: (){
                Navigator.of(context).push(new MaterialPageRoute(
                    builder: (_)=>ListMateriPage(itemmateri: dt, model:
jadwal,)
                ));
            },
            child: Container(
                padding: EdgeInsets.fromLTRB(10.0, 15.0, 10.0, 15.0),
                child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: <Widget>[
                        Expanded(
                            flex:2,
                            child:
Container(child:Icon(FontAwesomeIcons.archive, color: Colors.amber),
                                padding: EdgeInsets.only(right:
10.0)),
                            ),
                        Expanded(
                            flex: 12,
                            child: Column(
                                crossAxisAlignment:
CrossAxisAlignment.stretch,
                                children: <Widget>[
                                    Text('${dt['title']}', style:
TextStyle(fontSize: 18.0, fontWeight: FontWeight.bold),),
                                    Text('Ukuran Paket :
${f.format(dt['size']/1024)} KB', textAlign: TextAlign.left,)
                                ])
                            ),
                        Expanded(
                            flex:1,
                            child: Icon(FontAwesomeIcons.chevronRight, size:
10.0, color: Colors.grey,),
                        )
                    ],
                ),
            ),
        );
}

```

```

Widget tips(){
  return Card(
    child: Container(
      padding: EdgeInsets.all(10.0),
      child: Text('Pilih paket berkas materi yang ingin dibuka',
style: TextStyle(color: Colors.orange),),
      color: Colors.amber[50],
    )
  );
}
@override
Widget build(BuildContext context) {
  List<Widget> list = new List();

  list.add(tips());

  data.forEach((f) {
    list.add( item(f) );
  });

  return Scaffold(
    appBar: AppBar(
      title: TitleActionbar(subTitle: '${jadwal.nm_mtk}')
    ),body: isLoading ? Center(child: CupertinoActivityIndicator()) :
      ListView(
        children: list
      ),
  );
}
}

import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:m_student_ubsu/utills/mywidget.dart';
import 'package:path/path.dart';
import 'package:flutter_full_pdf_viewer/flutter_full_pdf_viewer.dart';
import 'package:flutter_full_pdf_viewer/full_pdf_viewer_plugin.dart';
import 'package:flutter_full_pdf_viewer/full_pdf_viewer_scaffold.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:share_extend/share_extend.dart';

class PdfMateriViewerPage extends StatefulWidget {
  final Widget child;
  final String pdffile;

  PdfMateriViewerPage({Key key, this.child, this.pdffile}) : super(key: key);

  _PdfMateriViewerPageState createState() =>
  _PdfMateriViewerPageState(pdffile: pdffile);
}

class _PdfMateriViewerPageState extends State<PdfMateriViewerPage> {
  String pdffile = '';

  _PdfMateriViewerPageState({this.pdffile});
}

```

```

@override
Widget build(BuildContext context) {
  return PDFViewerScaffold(
    appBar: AppBar(
      title: TitleActionBar(subTitle: basename(pdffile), title:
Text('Baca Materi')),
      actions: <Widget>[
        IconButton(
          icon: Icon(FaIcons.shareAlt, color:
Colors.white70),
          iconSize: 17.0,
          onPressed: (){
            ShareExtend.share(pdffile, 'file');
          },
        ),
      ],
    ),
    path: pdffile,
  );
}

```

```

import 'dart:async';
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';
import 'package:m_student_ubsj/model/user.dart';
import 'package:m_student_ubsj/model/infoakademik.dart';
import 'package:m_student_ubsj/model/hasilstudy.dart';
import 'package:m_student_ubsj/model/payment.dart';
import 'package:m_student_ubsj/model/jadwal_kuliah.dart';
import 'package:m_student_ubsj/model/kalender.dart';
import 'package:m_student_ubsj/model/pengaturan.dart';
import 'package:m_student_ubsj/model/suratriset.dart';
import 'package:m_student_ubsj/model/skpi.dart';
import 'package:m_student_ubsj/model/bimbingan.dart';

```

```

class DB{
  static final DB _instance = new DB.o();
  static Database _db;
  static const String DBNAME = "db_students_ubsj.db";
  static const int DBVERSION = 2;

```

```

  DB.o();

```

```

  factory DB()=>_instance;

```

```

  Future<Database> get db async{
    if(_db != null){
      return _db;
    }else{
      _db = await initDB();
    }

```

```

    return _db;
  }

```

```

  Future deleteDB() async{

```

```

        String dbPath = await getDatabasesPath();
        String path = join(dbPath, DBNAME);

        await close();
        await deleteDatabase(path);
        await initDB();
    }

    Future<Database> initDB() async {
        print("initDB dijalankan");
        String dbPath = await getDatabasesPath();
        String path = join(dbPath, DBNAME);
        // await deleteDatabase(path);

        Database ret = await openDatabase(path, version: DBVERSION, onCreate:
        _onCreate, onUpgrade: _onUpgrade);

        return ret;
    }

    void _onUpgrade(Database db, int oldVer, int newVer) async {
        if (oldVer == 1 && newVer == 2) {
            try {
                db.execute( SkpiModel.defineTable() );
                db.execute( BimbinganModel.defineTable() );
                print('update table sukses');
            } catch (e) {
                print('error upgrade table $e');
            }
        }
    }

    void _onCreate(Database db, int newVersion) async {
        try {
            db.execute( UserModel.defineTable() );
            db.execute( InfoakademikModel.defineTable() );
            db.execute( HasilstudyModel.defineTable() );
            db.execute( PaymentModel.defineTable() );
            db.execute( KalenderModel.defineTable() );
            db.execute( JadwalkuliahModel.defineTable() );
            db.execute( PengaturanModel.defineTable() );
            db.execute( SuratrissetModel.defineTable() );
            db.execute( SkpiModel.defineTable() );
            db.execute( BimbinganModel.defineTable() );

            print('create table success');
        } catch (e) {
            print("Kesalahan onCreate table : " + e.toString());
        }
    }

    Future close() async {
        try { await _db.close(); } catch (e) {}
    }
}

```

}