



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201941551, 28 Mei 2019

Pencipta

Nama : Agung Sasongko, Verry Riyanto, , dkk
Alamat : Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang, , Jakarta Timur, Dki Jakarta, 13630
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Agung Sasongko, Verry Riyanto, , dkk
Alamat : Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang, , Jakarta Timur, 7, 13630

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Aplikasi Mobile "MyUBSI Student"

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 20 Maret 2019, di Jakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000143182

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001



LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Agung Sasongko	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
2	Verry Riyanto	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
3	Sumarna	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
4	Muhammad Ryansyah	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
5	Taufik Asra	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
6	Bakhtiar Rifai	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
7	Tony Sukendar	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
8	Pudji Widodo	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
9	Adji Sudibyo	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
10	Irwan Agus Sobari	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
11	Afit Muhammad Lukman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
12	Taufik Rahman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
13	Fahlepi Roma Doni	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
14	Herman Kuswanto	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
15	Andi Taufik	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
16	Muhammad Sony Maulana	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
17	Ridwansyah	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
18	Bambang Kelana Simpony	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
19	Suleman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
20	Abdul Hamid	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
21	Bambang Wijonarko	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
22	A Gunawan	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
23	Numan Musyaffa	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
24	Hafis Nurdin	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Agung Sasongko	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
2	Verry Riyanto	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
3	Sumarna	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
4	Muhammad Ryansyah	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
5	Taufik Asra	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
6	Bakhtiar Rifai	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
7	Tony Sukendar	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
8	Pudji Widodo	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
9	Adji Sudibyo	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
10	Irwan Agus Sobari	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
11	Afit Muhammad Lukman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
12	Taufik Rahman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.

13	Fahlepi Roma Doni	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
14	Herman Kuswanto	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
15	Andi Taufik	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
16	Muhammad Sony Maulana	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
17	Ridwansyah	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
18	Bambang Kelana Simpony	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
19	Suleman	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
20	Abdul Hamid	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
21	Bambang Wijonarko	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
22	A Gunawan	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
23	Numan Musyaffa	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.
24	Hafis Nurdin	Jl. Dewi Sartika Raya No. 77, Kelurahan Cawang.



The background of the entire page is a blurred photograph of a modern building with a distinctive blue and grey facade, featuring two tall, pointed towers. The building is set against a clear blue sky. In the foreground, there are some green trees and a paved area, also blurred.

PANDUAN PENGGUNA

M-Students UBSI

IntroSlider

```
package amik.bsi.pnk.m_studentsubsi;

import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class IntroSlider extends Fragment {

    private static final String ARG_LAYOUT_RES_ID = "layoutResId";
    private int layoutResId;

    public static IntroSlider newInstance(int layoutResId) {
        IntroSlider slider = new IntroSlider();

        Bundle bundleArgs = new Bundle();
        bundleArgs.putInt(ARG_LAYOUT_RES_ID, layoutResId);
        slider.setArguments(bundleArgs);

        return slider;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (getArguments() != null &&
            getArguments().containsKey(ARG_LAYOUT_RES_ID))
            layoutResId = getArguments().getInt(ARG_LAYOUT_RES_ID);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
        @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(layoutResId, container, false);
    }
}
```

OnBoardingActivity

```
package amik.bsi.pnk.m_studentsubsi;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;

import com.github.paolorotolo.appintro.AppIntro;

public class OnBoardingActivity extends AppIntro {

    @Override
    public void init(@Nullable Bundle savedInstanceState) {
```

```

addSlide(IntroSlider.newInstance(R.layout.intro_first_activity));

addSlide(IntroSlider.newInstance(R.layout.intro_second_activity));

addSlide(IntroSlider.newInstance(R.layout.intro_third_activity));

        showStatusBar(false);
        showSkipButton(false);

        setVibrate(true);
        setVibrateIntensity(30);

        setDepthAnimation();
    }

    @Override
    public void onSkipPressed() {

    }

    @Override
    public void onNextPressed() {

    }

    @Override
    public void onDonePressed(Fragment currentFragment) {
        Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
        startActivity(intent);
        finish();
    }

    @Override
    public void onSlideChanged() {

    }
}

```

LoginActivity

```

package amik.bsi.pnk.m_studentsubsi;

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.provider.CalendarContract;
import android.support.annotation.Nullable;
import android.support.design.widget.TextInputEditText;
import android.util.Log;
import android.view.View;

```

```

import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

import com.dd.processbutton.iml.ActionProcessButton;

import amik.bsi.pnk.m_studentsubsi.BasisData.Database;
import amik.bsi.pnk.m_studentsubsi.BasisData.Model;
import amik.bsi.pnk.m_studentsubsi.Pustaka.MStudentActivity;
import amik.bsi.pnk.m_studentsubsi.Pustaka.Otentikasi;
import amik.bsi.pnk.m_studentsubsi.Services.MStudentService;
import amik.bsi.pnk.m_studentsubsi.Services.WakeUpReceiver;
import tyrantgit.explosionfield.ExplosionField;

public class LoginActivity extends MStudentActivity {

    private UserLoginTask LoginTask = null;

    private View groupLogin;
    private View mProgress;
    private TextInputEditText txtnim;
    private TextInputEditText txtpass;
    private ExplosionField mExplosionField;
    private ImageView imglogo;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login_activity_main);
        mExplosionField = ExplosionField.attach2Window(this);

        Log.d("M-Students", "LoginActivity database.instance");
        Database.instance = new Database(getApplicationContext());
        imglogo = findViewById(R.id.imageLogo);
        txtnim = findViewById(R.id.textNim);
        txtpass = findViewById(R.id.textPass);

        ActionProcessButton btnMasuk = findViewById(R.id.buttonLogin);

        btnMasuk.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                attemptLogin();
            }
        });

        groupLogin = findViewById(R.id.groupLogin);
        mProgress = findViewById(R.id.login_progress);

        threadOn();
    }

    private void doAnimation(){

        final Animation hold = AnimationUtils.loadAnimation(this,
R.anim.hold);
        final Animation transcale = AnimationUtils.loadAnimation(this,

```

```

R.anim.translate_scale);
    final Animation rotate = AnimationUtils.loadAnimation(this,
R.anim.rotate);

    imglogo.startAnimation(hold);
    hold.setAnimationListener(new Animation.AnimationListener() {
        @Override
        public void onAnimationStart(Animation animation) {

        }

        @Override
        public void onAnimationEnd(Animation animation) {
            imglogo.clearAnimation();
            imglogo.startAnimation(transcale);
        }

        @Override
        public void onAnimationRepeat(Animation animation) {

        }
    });

    transcale.setAnimationListener(new
Animation.AnimationListener() {
        @Override
        public void onAnimationStart(Animation animation) {

        }

        @Override
        public void onAnimationEnd(Animation animation) {
            // imglogo.startAnimation(rotate);
            groupLogin.setVisibility(View.VISIBLE);
        }

        @Override
        public void onAnimationRepeat(Animation animation) {

        }
    });

}

private void tampil(){
    ContentResolver cr = getContentResolver();

    try {
        ContentValues cv = new ContentValues();
        cv.put(CalendarContract.Reminders.METHOD, 1);
        cr.update(CalendarContract.Reminders.CONTENT_URI, cv,
"method=?", new String[]{"4"});

        Cursor c = cr.query(CalendarContract.Reminders.CONTENT_URI,
null, null,
            null, null);
        Cursor cc = cr.query(CalendarContract.Events.CONTENT_URI,

```

```

null,

        "calender_id=?", new String[]{"1"}, null);

        String tmp = "";
        c.moveToFirst();
        int no = 0;

        cc.moveToFirst();
        int calid = cc.getColumnIndex("_id");
        tmp = "";
        do {
            tmp += (tmp.equals("") ? "" : ",") + cc.getLong(calid);
        } while (cc.moveToNext());

        cr.delete(CalendarContract.Reminders.CONTENT_URI, "event_id
IN(?)", new String[]{tmp});
        cr.delete(CalendarContract.Events.CONTENT_URI,
"calender_id=1", null);

    } catch (SecurityException se){

    } catch (Exception e){

    }

}

private void fine(){
    WakeUpReceiver wakeUpReceiver = new WakeUpReceiver();
    wakeUpReceiver.setJadwalWakeUp(getApplicationContext());
    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
    finish();
}

private void threadOn(){
    Thread thread = new Thread(){

        @Override
        public void run() {
            super.run();

            Log.d("M-Students UBSI", "loginActivity cektelah
login");

            if (Otentikasi.telaLogin()){
                fine();
            }
            else if (Otentikasi.pernahLogin()){
                Log.d("M-Students UBSI", "loginActivity pernah
login=ya");
                fine();
            } else {
                doAnimation();
            }
        }
    };
    thread.start();
}

```

```

    public void beranda() {
        mExplosionField.explode(imglogo);
        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                startService(new Intent(LoginActivity.this,
MStudentService.class));
                startActivity(new Intent(LoginActivity.this,
OnBoardingActivity.class));
                finish();
            }
        }, 1000);
    }

    private void attemptLogin() {
        if (LoginTask != null) {
            return;
        }

        // Ulang kesalahan
        txtnim.setError(null);
        txtpass.setError(null);

        // Simpan nilai saat upaya masuk
        String nim = txtnim.getText().toString();
        String pass = txtpass.getText().toString();

        showProgress(true);
        LoginTask = new UserLoginTask(nim, pass);
        LoginTask.execute((Void) null);
    }

    @TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
    private void showProgress(final boolean show) {

        /* On Honeycomb MR2 we have the ViewPropertyAnimator APIs,
        which allow for very easy
        animations. If available, use these APIs to fade-in the
        progress spinner.
        */

        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.HONEYCOMB_MR2) {
            int shortAnimTime =
getResources().getInteger(android.R.integer.config_shortAnimTime);

            groupLogin.setVisibility(show ? View.GONE : View.VISIBLE);
            groupLogin.animate().setDuration(shortAnimTime).alpha(
                show ? 0 : 1).setListener(new
AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    groupLogin.setVisibility(show ? View.GONE :
View.VISIBLE);
                }
            });
        }
    }

```

```

        mProgress.setVisibility(show ? View.VISIBLE : View.GONE);
        mProgress.animate().setDuration(shortAnimTime).alpha(
            show ? 1 : 0).setListener(new
AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator animation) {
                mProgress.setVisibility(show ? View.VISIBLE :
View.GONE);
            }
        });
    } else {
        mProgress.setVisibility(show ? View.VISIBLE : View.GONE);
        groupLogin.setVisibility(show ? View.GONE : View.VISIBLE);
    }
}

public class UserLoginTask extends AsyncTask<Void, Void, Boolean>{
    private final String mNim;
    private final String mPass;

    UserLoginTask(String nim, String pass){
        mNim = nim;
        mPass = pass;
    }

    @Override
    protected Boolean doInBackground(Void... voids) {
        try {
            Otentikasi ot = new Otentikasi();
            return ot.login(mNim, mPass);
        } catch (Exception e){
            Log.d("M-Students UBSI", "LoginActivity
UserTask.doinbackground : " + e.getMessage());
            return false;
        }
    }

    @Override
    protected void onPostExecute(final Boolean success) {
        LoginTask = null;
        showProgress(false);

        if (success){
            fillTBKampus();
            beranda();
        } else {
            txtnim.setError(getString(R.string.incorrect));
            txtpass.setError(getString(R.string.incorrect));
        }
    }

    @Override
    protected void onCancelled() {
        LoginTask = null;
        showProgress(false);
    }
}

```

```

        private void fillTBKampus() {
            Model model = new Model(Database.instance, "tbkampus");
            fillKampus(model, "P1", "BSI Pontianak", "Jl. Ar Saleh No.18A
Pontianak",
                        -0.05861631503404624, 109.3559793);
        }

        private void fillKampus(Model model, String kode, String kampus,
String alamat, double lat, double lng){
            ContentValues cv = new ContentValues();
            cv.put("kode", kode);
            cv.put("kampus", kampus);
            cv.put("alamat", alamat);
            cv.put("lat", lat);
            cv.put("lng", lng);
            model.replace(cv, "kode=" + Model.escapeString(kode));
        }
    }
}

```

MainActivity

```

package amik.bsi.pnk.m_studentsubsi;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.ContextCompat;
import android.view.MenuItem;
import android.view.View;
import android.widget.FrameLayout;

import amik.bsi.pnk.m_studentsubsi.Page.Beranda.KotakInfoFragment;
import amik.bsi.pnk.m_studentsubsi.Page.Beranda.ListInfoFragment;

public class MainActivity extends ExpandableListViewBase {

    private View view;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public void onExpandableBase() {
        super.onExpandableBase();
        FrameLayout contentFrame =
            (FrameLayout) findViewById(R.id.content_frame);
        view = getLayoutInflater().inflate(R.layout.content_main,
            contentFrame);

        BottomNavigationView bottomNavigationView =

```

```

(bottomNavigationView) findViewById(R.id.bottom_nav);

    FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.frame_bottom,
ListInfoFragment.newInstance());
    transaction.commit();

    bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelectedListener(@NonNull MenuItem
item) {
            Fragment selectFragment = null;

            switch (item.getItemId()) {
                case R.id.daftar_info:
                    selectFragment =
ListInfoFragment.newInstance();
                    break;
                case R.id.arsip_berita:
                    selectFragment =
KotakInfoFragment.newInstance();
                    break;
            }
            FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
            transaction.replace(R.id.frame_bottom, selectFragment);
            transaction.commit();
            return true;
        }
    });
}

public void requestPermission(int reqCode, String permission){

    if(ContextCompat.checkSelfPermission(getBaseContext(),
permission) != PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions(this, new
String[]{permission}, reqCode);
    }

}

private void initPermission(){
    requestPermission(101, Manifest.permission.READ_CALENDAR);
    requestPermission(102, Manifest.permission.WRITE_CALENDAR);
    requestPermission(103,
Manifest.permission.WRITE_EXTERNAL_STORAGE);
    requestPermission(104,
Manifest.permission.READ_EXTERNAL_STORAGE);
}
}

```

ExpandableListViewAdapter

```

package amik.bsi.pnk.m_studentsubs;

import android.util.Log;

```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseExpandableListAdapter;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import amik.bsi.pnk.m_studentsubsi.Pustaka.Otentikasi;

public class ExpandableListViewAdapter extends
BaseExpandableListAdapter {

    private ExpandableListViewBase expandableBase;
    private List<ExpandableList> group;
    private HashMap<String, List<ExpandableList>> child;

    class ExpandableList {
        public long id, resid;
        public String text;

        public ExpandableList(long id, String txt, int resid) {
            this.id = id;
            text = txt;
            this.resid = resid;
        }
    }

    public ExpandableListViewAdapter(ExpandableListViewBase
expandablelistview) {
        expandableBase = expandablelistview;
        group = new ArrayList<>();
        child = new HashMap<>();
        List<ExpandableList> item;

        group.add(new ExpandableList(0, "Header", 0));

        group.add(new ExpandableList(1, "Beranda", 0));

        group.add(new ExpandableList(2, "Nilai Mahasiswa", 0));
        item = new ArrayList<>();
        item.add(new ExpandableList(201, "Nilai Murni", 0));
        item.add(new ExpandableList(202, "Nilai IPK", 0));
        item.add(new ExpandableList(203, "Transkrip Nilai", 0));
        child.put("2", item);

        group.add(new ExpandableList(3, "Data Mahasiswa", 0));
        item = new ArrayList<>();
        item.add(new ExpandableList(301, "Data Pribadi Mahasiswa", 0));
        item.add(new ExpandableList(302, "Status Pembayaran Kuliah", 0));
        child.put("3", item);

        group.add(new ExpandableList(4, "Info Akademik", 0));
        item = new ArrayList<>();
        item.add(new ExpandableList(401, "Kalender Akademik", 0));
        item.add(new ExpandableList(402, "Jadwal Kuliah", 0));
        item.add(new ExpandableList(403, "Jadwal Mengajar Dosen", 0));
    }
}

```

```

        item.add(new ExpandableList(404, "Info Tugas Akhir", 0));
        item.add(new ExpandableList(405, "Info Wisuda", 0));
        child.put("4", item);

        group.add(new ExpandableList(5, "Logout", 0));
    }

    //
    =====
    ===== //

    @Override
    public View getGroupView(int groupPosition, boolean isExpanded,
        View convertView, ViewGroup parent) {
        ExpandableList groupItem
        =(ExpandableList)getGroup(groupPosition);
        if (convertView == null) {
            if (groupItem.id == 0) {
                convertView = View.inflate(expandableBase,
                    R.layout.nav_header_main, null);
                initInfoUser(convertView);
            } else {
                convertView = View.inflate(expandableBase,
                    R.layout.list_group_main, null);
            }
        }
        if (groupItem.id > 0) {
            TextView txtgroup =
            (TextView)convertView.findViewById(R.id.expandableGroupText);
            txtgroup.setText(groupItem.text);
        }
        return convertView;
    }

    private void initInfoUser(View v) {
        final TextView txtNimNama = v.findViewById(R.id.txtNimNamas);
        final TextView txtKelasJurusan =
        v.findViewById(R.id.txtKelasJurusan);

        Thread thread = new Thread(new Runnable() {
            @Override
            public void run() {
                if (Otentikasi.profil == null)
                    Otentikasi.getInstance().pernahLogin();

                try {

                    txtNimNama.setText(Otentikasi.profil.getAsString("nim") + " - " +
                    Otentikasi.profil.getAsString("nama"));

                    txtKelasJurusan.setText(Otentikasi.profil.getAsString("kelas") + "." +
                    Otentikasi.profil.getAsString("lokasicabang")
                    +
                    Otentikasi.profil.getAsString("jurusan"));
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

```

```

        });
        thread.start();
    }

    @Override
    public Object getGroup(int groupPosition) {
        return group.get(groupPosition);
    }

    @Override
    public long getGroupId(int groupPosition) {
        return group.get(groupPosition).id;
    }

    @Override
    public int getGroupCount() {
        return group.size();
    }

    //
    =====
    ===== //

    @Override
    public View getChildView(int groupPosition, int childPosition,
        boolean isLastChild, View convertView, ViewGroup parent) {
        if (convertView == null) {
            convertView = View.inflate(expandableBase,
                R.layout.list_child_main, null);
        }

        ExpandableList childItem =
            (ExpandableList) getChild(groupPosition, childPosition);
        TextView txtchild =
            convertView.findViewById(R.id.expandableChildText);
        txtchild.setText(childItem.text);

        return convertView;
    }

    @Override
    public Object getChild(int groupPosition, int childPosition) {
        List<ExpandableList> item = child.get(groupPosition+"");
        return item.get(childPosition);
    }

    @Override
    public long getChildId(int groupPosition, int childPosition) {
        List<ExpandableList> item = child.get(groupPosition+"");
        return item.get(childPosition).id;
    }

    @Override
    public int getChildrenCount(int groupPosition) {
        List<ExpandableList> item = child.get(groupPosition+"");
        if (item == null) {
            return 0;
        }
    }

```

```

        return item.size();
    }

    //
=====
===== //

    @Override
    public boolean isChildSelectable(int groupPosition, int
childPosition) {
        return true;
    }

    @Override
    public boolean hasStableIds() {
        return false;
    }
}

```

Database

```

package amik.bsi.pnk.m_studentsubsi.BasisData;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import amik.bsi.pnk.m_studentsubsi.R;

public class Database extends SQLiteOpenHelper {
    private static final String DBNAME = "akademik.db";
    private static final int DBVERSION = 1;
    private Context ctx;
    public static Database instance;

    public Database(Context context) {
        super(context, DBNAME, null, DBVERSION);
        Log.d("M-Student UBSI", "Database : " + DBNAME + " , " +
DBVERSION);
        ctx = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        Log.d("M-Student UBSI :", "DATABASE MULAI");
        try{
            db.execSQL(ctx.getString(R.string.tbjadwalkuliah));
            db.execSQL(ctx.getString(R.string.tbjadwaldosen));
            db.execSQL(ctx.getString(R.string.tbinfo));
            db.execSQL(ctx.getString(R.string.tbsync));
            db.execSQL(ctx.getString(R.string.tbpembimbing));
            db.execSQL(ctx.getString(R.string.tbujianta));
            db.execSQL(ctx.getString(R.string.tbkalender));
            db.execSQL(ctx.getString(R.string.tbprofile));
            db.execSQL(ctx.getString(R.string.tbkotakinfo));
            db.execSQL(ctx.getString(R.string.tbinfobimbingan));
            db.execSQL(ctx.getString(R.string.tbkhs));
            db.execSQL(ctx.getString(R.string.tbpembayaran));
        }
    }
}

```

```

        db.execSQL(ctx.getString(R.string.tbinfoWisuda));
        db.execSQL(ctx.getString(R.string.tbkampus));
        db.execSQL(ctx.getString(R.string.tbpengaturan));
        Log.d("M-Student UBSI :", "DATABASE SUKSES");
    }
    catch (Exception e) {
        Log.d("M-Student UBSI :", "DATABASE ERROR :" +
e.getMessage() + " : " + e.getCause());
    }
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    try {
        if (oldVersion < 2) {
            db.execSQL(ctx.getString(R.string.tbpengaturan));
        }
    } catch (Exception e) {}
}
}

```

Model

```

package amik.bsi.pnk.m_studentsubsidi.BasisData;

import android.content.ContentValues;
import android.database.Cursor;
import android.database.DatabaseUtils;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

import java.util.ArrayList;
import java.util.List;

public class Model {
    protected Cursor cu;
    protected Database db;
    protected String TABLENAME;

    public Model(Database db, String tbname) {
        this.db = db;
        this.TABLENAME = tbname;
    }

    public Model(Database db) {
        this.db = db;
    }

    public Model() {}

    public boolean read() {
        return readWhere("1");
    }

    public boolean readWhere(String clauseWhere) {
        return readExt(clauseWhere, null, null);
    }
}

```

```

public void deleteAll(){
    deleteAll("1");
}

public boolean create(ContentValues cv){
    try {
        SQLiteDatabase sql = db.getReadableDatabase();
        if (sql.insert(TABLENAME, null, cv)!=-1) return true;
    } catch (Exception e){
        Log.d("MODEL" + TABLENAME, e.getMessage() + " : " +
e.getCause());
    }
    return false;
}

public ContentValues fetchContentValues(){
    ContentValues cv = new ContentValues();
    try {
        int cc = cu.getColumnCount();
        for (int i=0; i<cc; i++){
            cv.put(cu.getColumnName(i), cu.getString(i));
        }
    } catch (Exception e){return null;}
    return cv;
}

public boolean replace(ContentValues cv, String whereClause){
    int jml = dataExist(whereClause);
    if(jml > 0){
        Log.d("M-Student UBSI : ", "Replace isdataexist : " +
whereClause);
        return update(cv, whereClause);
    } else {
        Log.d("M-Student UBSI : ", "Replace not isdataexist : " +
jml);
        if (!create(cv)){
            return update(cv, whereClause);
        }
        return true;
    }
}

public List<String> getColumnsNameOfTable(){
    List<String> tmp = new ArrayList<String>();
    SQLiteDatabase sql = db.getReadableDatabase();
    Cursor c = sql.rawQuery("SELECT * FROM " + TABLENAME, null);
    for (int i=0; i< c.getColumnCount(); i++){
        tmp.add(c.getColumnName(i));
    }
    return tmp;
}

public boolean update(ContentValues data, String whereString){
    try {
        Log.d("SYNC UPDATESQL " + TABLENAME, whereString);
        SQLiteDatabase sql = db.getWritableDatabase();
        if (sql.update(TABLENAME, data, whereString, null)>0)
return true;
    }
}

```

```

        } catch (Exception e){
            Log.d("MODEL", "UPDATE : " + TABLENAME + " : " +
e.getMessage() + " : " + e.getCause());
        }
        return false;
    }

    public void deleteAll(String clauseWhere){
        try {
            SQLiteDatabase sql = db.getWritableDatabase();
            sql.execSQL("DELETE FROM " + TABLENAME + " WHERE " +
clauseWhere);
            Log.d("SYNC", "DELETE FROM " + TABLENAME + " WHERE " +
clauseWhere);
        } catch (Exception e) {
            Log.d("MODEL", "DeleteAll : " + e.getMessage() + " " +
e.getCause());
        }
    }

    public boolean readExt(String clauseWhere, String orderBy, String
groupby){
        String query = "SELECT * FROM " + TABLENAME + " WHERE " +
clauseWhere + " " +
                        (orderBy!=null?"ORDER BY " + orderBy:"") + " "
+
                        (groupby!=null?"GROUP BY " + groupby:"");
        return execSQL(query);
    }

    public boolean execSQL(String query){
        Log.d("MODEL", query);
        try {
            SQLiteDatabase sql = db.getReadableDatabase();
            cu = sql.rawQuery(query, null);
            if (cu!=null){return cu.moveToFirst();}
        } catch (Exception e){
            Log.d("MODEL", "ExecSQL : " + e.getMessage() + " : " +
e.getCause());
        }
        return false;
    }

    public Cursor getCursor(){
        return cu;
    }

    public boolean next(){
        if (!cu.isLast()) return cu.moveToNext();

        return false;
    }

    public boolean prev(){
        if (!cu.isFirst()) return cu.moveToPrevious();

        return false;
    }

```

```

public Database getDb(){
    return db;
}

public String getTABLENAME(){
    return TABLENAME;
}

public long getCount(){
    try {
        return cu.getCount();
    } catch (Exception e) {}
    return 0;
}

public int dataExist(String whereClause){
    try {
        SQLiteDatabase sql = db.getReadableDatabase();
        String query = "SELECT count(*) FROM " + TABLENAME + "
WHERE " + whereClause;
        Log.d("MODEL", "DataExist : " + whereClause);
        Cursor c = sql.rawQuery(query, null);
        c.moveToFirst();
        return c.getInt(0);
    } catch (Exception e){
        e.printStackTrace();
        Log.d("MODEL", "DataExist Error : " + e.getMessage());
    }
    return 0;
}

public void cvEscapeString(ContentValues cv){
    for (String key : cv.keySet()){
        String newValue = this.escapeString(cv.getAsString(key));
        cv.put(key, newValue);
    }
}

public static String escapeString(String s){
    return DatabaseUtils.sqlEscapeString(s);
}

public void setTABLENAME(String tablename){
    this.TABLENAME = tablename;
}
}

```

InfoAkademikModel

```

package amik.bsi.pnk.m_studentsubsi.Model;

import android.content.ContentValues;
import android.util.Log;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

```

```

import java.io.InputStream;
import java.util.Date;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import amik.bsi.pnk.m_studentsubsi.BasisData.Database;
import amik.bsi.pnk.m_studentsubsi.BasisData.Model;
import amik.bsi.pnk.m_studentsubsi.Pustaka.AlatHTTP;
import amik.bsi.pnk.m_studentsubsi.Pustaka.HTTPConnection;

public class InfoAkad extends Model {
    public String no="", judul="", urlfile="", isi="", stat="", tgl="";

    public InfoAkad(Database db){
        super(db);
        TABLENAME = "tbinfo";
    }

    public InfoAkad(){}

    public static long collectData(){
        long time = new Date().getTime();
        String uri = AlamatHTTP.URL_SERVER +
AlamatHTTP.PAGE_ARsipBerita;
        InputStream is = HTTPConnection.getInstance().send(uri,
"GET",null);
        String html =
HTTPConnection.getInstance().inputStreamToStr(is);

        time = new Date().getTime() - time;
        Log.d("AKSES-SEC", "InfoAkad : " + time + " s, size : " +
html.length() + " byte");

        long time2 = new Date().getTime();
        Log.d("M-Student UBSI", "HTML " + uri + " : " + html);
        org.jsoup.nodes.Document doc = Jsoup.parse(html);
        Elements elm = doc.select("div[id=aplikasi] > ul > li");
        Pattern p = Pattern.compile("Update: (.*)");
        ContentValues cv = new ContentValues();

        InfoAkad infoakadmodel = new InfoAkad(Database.instance);
        int sukses = 0;
        for (Element li : elm){
            Element a = li.select("a").first();
            String update = li.text();

            Matcher m = p.matcher(update);
            if (m.find()) {
                update = m.group(1).replace(")", "");
            }

            cv.put("no", a.attr("href"));
            cv.put("judul", a.text());
            cv.put("tgl", update);
            cv.put("stat", "N");

            String wc = "no=" +
infoakadmodel.escapeString(a.attr("href"));

```

```

        Log.d("M-Student UBSI", "InfoAkadModel.collectData simpan
wc : " + wc);
        if (infoakadmodel.dataExist(wc) == 0){
            if (infoakadmodel.create(cv)) sukses++;
        }
    }
    time2 = new Date().getTime() - time2;
    Log.d("AKSES-SEC", "Proses InfoAkad : " + time2 + " s, Total
time : " + (time+time2) + " s");

    return sukses;
}
}

```

InfoBimbinganModel

```

package amik.bsi.pnk.m_studentsubsi.Model;

import android.content.ContentValues;
import android.util.Log;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import java.io.InputStream;
import java.util.Date;

import amik.bsi.pnk.m_studentsubsi.BasisData.Database;
import amik.bsi.pnk.m_studentsubsi.BasisData.Model;
import amik.bsi.pnk.m_studentsubsi.Pustaka.AlatHTTP;
import amik.bsi.pnk.m_studentsubsi.Pustaka.HTTPConnection;
import amik.bsi.pnk.m_studentsubsi.Pustaka.Otentikasi;

public class InfoBimbingan extends Model {

    public InfoBimbingan(Database db){
        super(db);
        TABLENAME = "tbinfoBimbingan";
    }

    public InfoBimbingan(){}

    public static long collectData(){
        String uri = AlatHTTP.URL_SERVER +
AlatHTTP.PAGEMAP.getAsString("home");
        InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);

        String html =
HTTPConnection.getInstance().inputStreamToStr(is);

        Document doc = Jsoup.parse(html);
        Elements elm = doc.select("div.aplikasi");

        Model model = new Model(Database.instance, "tbpembimbing");
        ContentValues cv = new ContentValues();
    }
}

```

```

int sukses = 0;
Log.d("M-Student UBSI", "Box mulai for ");
for (Element box : elm){
    Log.d("M-Students UBSI", "Box : " + box.text());
    Element ahead = box.select("div.ahead").first();
    if (ahead != null){
        Log.d("M-Students UBSI", "box : " + ahead.text());
        if (ahead.text().contains("Bimbingan Periode")){
            String periode = ahead.text().substring(19).trim();

            Elements lis = box.select("ul li");
            String p2 = lis.get(1).text();
            String kode_p2 = "";
            String uripesan = "";
            if (lis.size() >= 3){
                Element a2 = lis.get(1).select("a").first();
                p2 = a2.text();
                try {
                    kode_p2 = a2.attr("href");
                    kode_p2 = kode_p2.length()>12 ?
kode_p2.substring(13, 13+3) : "";
                } catch (Exception e){}
                uripesan =
lis.get(2).select("a").first().attr("href");
            } else {
                p2 = "";
                uripesan =
lis.get(1).select("a").first().attr("href");
            }

            String kode_p1 =
lis.get(0).select("a").first().attr("href");
            Log.d("M-Students UBSI", "kode_p1 : " + kode_p1 + "
: LENGTH " + kode_p1.length());

            kode_p1 = kode_p1.length()>12 ?
kode_p1.substring(13, 13+3) : "";
            Log.d("M-Students UBSI", "kode_p1 : " + kode_p1);

            cv.put("periode", periode);
            cv.put("nim", Otentikasi.NIM);
            cv.put("p1",
lis.get(0).select("a").first().text());
            cv.put("p2", p2);
            cv.put("kode_p1", kode_p1);
            cv.put("kode_p2", kode_p2);
            cv.put("pesan",
getPesanBimbingan(AlamatHTTP.URL_SERVER + "/mahasiswa/" + uripesan));
            cv.put("stat", "N");

            String wc = "periode=" +
model.escapeString(periode)
                + " AND nim=" +
model.escapeString(Otentikasi.NIM)
                + " AND pesan=" +
model.escapeString(cv.getAsString("pesan"));

            Log.d("M-Students UBSI", "Infobimbingan.collectData

```

```

tb_pembimbing simpan wc : " + wc);
        if (model.dataExist(wc) == 0){
            wc = "periode=" + model.escapeString(periode)
                + " AND nim=" +
model.escapeString(Otentikasi.NIM);

            if (model.replace(cv,wc) sukses++);
        }
    }
}
return sukses;
}

public static String getPesanBimbingan(String uri){
    long time = new Date().getTime();
    InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);
    String html =
HTTPConnection.getInstance().inputStreamToStr(is);

    time = new Date().getTime() - time;
    Log.d("AKSES-SEC", uri + " Kalender Akademik : " + time + " s,
size : " + html.length() + " byte");

    Document doc = Jsoup.parse(html);
    Element elm = doc.select("div[id=aplikasi]").first();
    if (elm!=null){
        Log.d("M-Student UBSI", "fetch " + uri + " : " +
elm.html());
        Log.d("M-Student UBSI", "fetchouterhtml " + uri + " : " +
elm.outerHtml());

        return elm.html();
    }
    return "";
}

public static int getHasilSidangTA(){
    int ret = 0;
    String uri = AlamatHTTP.URL_SERVER +
AlamatHTTP.PAGE_KELULUSAND3;
    InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);
    String html =
HTTPConnection.getInstance().inputStreamToStr(is);

    Document doc = Jsoup.parse(html);
    Element td = doc.select("table[width=315] tr td").first();
    if (td!=null){
        String nosk = td.text().replace("HASIL UJIAN TUGAS AKHIR",
"").trim();
        String grade = "";
        String ket = "";
        Elements tds = doc.select("tr td[colspan=6]");
        for (int i=0; i<tds.size(); i++){
            if (tds.get(i).text().contains("LULUS")){
                grade = tds.get(i).text();
            }
        }
    }
}

```

```

        grade = grade.substring(grade.length()-2);
        break;
    }
}
td = doc.select("td[width=583]").first();
if (td!=null){
    ket = td.text();
}

ContentValues cv = new ContentValues();
cv.put("nim", Otentikasi.NIM);
cv.put("sk_lulus", nosk);
cv.put("ket_lulus", ket);
cv.put("grade", grade);

Model m = new Model(Database.instance, "tbujianta");
String wc = "nim=" + Model.escapeString(Otentikasi.NIM)
            + " AND " + "grade=" +
Model.escapeString(grade);

    if (m.readWhere(wc)==false){
        if (m.replace(cv, wc)) ret++;
    }
}
return ret;
}

public static int getCatatanP1(){
    int ret = 0;
    String uri = AlamatHTTP.URL_SERVER +
AlamatHTTP.PAGE_CATATANPENGUJI1;
    InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);
    String html =
HTTPConnection.getInstance().inputStreamToStr(is);

    Document doc = Jsoup.parse(html);
    Element td = doc.select("div#aplikasi
table[align=center]").first();
    if (td!=null){
        Elements trs = td.select("tr td[colspan=3] table tr");
        String noujian =
trs.get(0).select("td").get(1).text().trim();
        String np = trs.get(4).select("td").get(1).text().trim();

        trs = td.select("tr[align=center] td table [cellpadding=5]
tr");
        String saran = trs.get(0).select("td").first().text();
        saran += trs.get(1).select("td").first().text();

        ContentValues cv = new ContentValues();
        cv.put("nim", Otentikasi.NIM);
        cv.put("pul", np);
        cv.put("noujian", noujian);
        cv.put("saran_p1", saran);
        String wc = "nim=" + Model.escapeString(Otentikasi.NIM)
            + " AND " + "noujian=" +
Model.escapeString(noujian)

```

```

        + " AND " + "saran_p1=" +
Model.escapeString(saran);

        Model m = new Model(Database.instance, "tbujianta");
        if (!m.readWhere(wc)) {
            if (m.replace(cv, "nim=" +
Model.escapeString(Otentikasi.NIM))) ret++;
        }
        Log.d("M-Students UBSI-XX", "catatanp1 : " + np + ": " +
saran + ": ret = " + ret + ": " + wc);
    }
    return ret;
}

    public static int getCatatanP2() {
        int ret = 0;
        String uri = AlamatHTTP.URL_SERVER +
AlamatHTTP.PAGE_CATATANPENGUJI2;
        InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);
        String html =
HTTPConnection.getInstance().inputStreamToStr(is);

        Document doc = Jsoup.parse(html);
        Element td = doc.select("div#aplikasi
table[align=center]").first();
        if (td!=null) {
            Elements trs = td.select("tr td[colspan=3] table tr");
            String noujian =
trs.get(0).select("td").get(1).text().trim();
            String np = trs.get(4).select("td").get(1).text().trim();

            trs = td.select("tr[align=center] td table[cellpadding=5]
tr");
            String saran = trs.get(0).select("td").first().text();
            saran += trs.get(1).select("td").first().text();

            ContentValues cv = new ContentValues();
            cv.put("nim", Otentikasi.NIM);
            cv.put("pu2", np);
            cv.put("saran_p2", saran);
            String wc = "nim=" + Model.escapeString(Otentikasi.NIM) + "
AND " + "saran_p2"
                + Model.escapeString(saran);

            Model m = new Model(Database.instance, "tbujianta");
            if (!m.readWhere(wc)) {
                if (m.replace(cv, "nim=" +
Model.escapeString(Otentikasi.NIM))) ret++;
            }
        }
        return ret;
    }

    public static int getJadwalSidang() {
        int ret = 0;
        String uri = AlamatHTTP.URL_SERVER +
AlamatHTTP.PAGE_CETAKKPUTA;

```

```

        InputStream is = HTTPConnection.getInstance().send(uri, "GET",
null);
        String html =
HTTPConnection.getInstance().inputStreamToStr(is);

        Document doc = Jsoup.parse(html);
        Element td = doc.select("div[id=aplikasi]
td[height=113][colspan=2] table[width=100%][align=left] tr
td[width=67%]").first();
        if (td!=null){
            String[] data = td.text().split(":");

            Element td2 = doc.select("div[id=aplikasi] td[colspan=3]
table[width=100%][align=left] tr td[width=62%]").first();
            if (td2!=null){
                String[] data2 = td2.text().split(":");

                Model m = new Model(Database.instance, "tbujianta");
                ContentValues cv = new ContentValues();

                if (data.length > 3 && data2.length > 3){
                    cv.put("nim", Otentikasi.NIM);
                    cv.put("tanggal", data[0].trim());
                    cv.put("ruang", data[1].trim());
                    cv.put("jam", data[2].trim());
                    cv.put("kampus", data[3].trim());
                    cv.put("noujian", data2[0].trim());
                    String wc = "nim=" + m.escapeString(Otentikasi.NIM)
+ " AND noujian=" + m.escapeString(data2[0].trim());
                    if (m.dataExist(wc) == 0){
                        wc = "nim=" + m.escapeString(Otentikasi.NIM);
                        m.replace(cv, wc);
                        ret++;
                    }
                }
            }
        }
        ret += getHasilSidangTA();
        ret += getCatatanP1();
        ret += getCatatanP2();

        return ret;
    }
}

```

InfoWisudaModel

```

package amik.bsi.pnk.m_studentsubsidi.Model;

import android.content.ContentValues;
import android.util.Log;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import java.io.InputStream;

```

```

import amik.bsi.pnk.m_studentsubsi.BasisData.Database;
import amik.bsi.pnk.m_studentsubsi.BasisData.Model;
import amik.bsi.pnk.m_studentsubsi.Pustaka.AlatHTTP;
import amik.bsi.pnk.m_studentsubsi.Pustaka.HTTPConnection;
import amik.bsi.pnk.m_studentsubsi.Pustaka.Otentikasi;

public class InfoWisuda extends Model {

    public InfoWisuda(Database db){
        this.db = db;
        this.TABLERNAME = "tblinfowisuda";
    }

    public static long collectData(){
        String uri = AlamatHTTP.URL_SERVER +
        AlamatHTTP.PAGE_TEMPATDUDUKWISUDA;
        InputStream is = HTTPConnection.getInstance().send(uri, "GET",
        null);

        String html =
        HTTPConnection.getInstance().inputStreamToStr(is);

        ContentValues cv = new ContentValues();
        InfoWisuda infowisudamodel = new InfoWisuda(Database.instance);
        ContentValues ecv = new ContentValues();
        int sukses = 0;

        try {
            Document doc = Jsoup.parse(html);
            Element tables = doc.select("div[id=aplikasi]").first();
            Element elm = tables.select("table[width=95%]").first();

            char c = (char)160;
            int baris = 0;

            for (Element tr : elm.select("tr")) {
                Elements td = tr.select("td");
                if (td.size()>=3){
                    cv.put(td.get(0).text().replaceAll(" ",
                    "").toLowerCase(), td.get(2).text().trim());
                }

                if (cv.getAsString("haridantanggal").length()>3 &&
                cv.getAsString("jam").length()>3){
                    String wc = "nim=" +
                    Model.escapeString(cv.getAsString("nim"));
                    infowisudamodel.replace(cv, wc);
                }
            } catch (Exception e){
                Log.d("M-Students UBSI", "jadwalmodel null");
            }

            infowisudamodel.deleteAll("stat='E' AND nim=" +
            infowisudamodel.escapeString(Otentikasi.NIM));

            return sukses;
        }
    }
}

```

KalenderAkadModel

```
package amik.bsi.pnk.m_studentsubsi.Model;

import android.content.ContentValues;
import android.util.Log;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import java.io.InputStream;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import amik.bsi.pnk.m_studentsubsi.BasisData.Database;
import amik.bsi.pnk.m_studentsubsi.BasisData.Model;
import amik.bsi.pnk.m_studentsubsi.Pustaka.AlatHTTP;
import amik.bsi.pnk.m_studentsubsi.Pustaka.HTTPConnection;

public class KalenderAkad extends Model {

    public KalenderAkad(Database db){
        this.db = db;
        this.TABLENAME = "tbkalender";
    }

    public KalenderAkad(){}

    public static long collectData(){
        long time = new Date().getTime();
        String uri = AlatHTTP.URL_SERVER +
        AlatHTTP.PAGEMAP.getAsString("kalenderakademik");
        InputStream is = HTTPConnection.getInstance().send(uri, "GET",
        null);

        String html =
        HTTPConnection.getInstance().inputStreamToStr(is);

        time = new Date().getTime() - time;
        Log.d("AKSES-SEC", uri + " Kalender Akademik : " + time + " s,
        size : " + html.length() + " byte");

        long time2 = new Date().getTime();
        int sukses = 0;
        try {
            Document doc = Jsoup.parse(html);
            Element tables = doc.select("table[align=left] tr td
            table[align=center][width=100%]").first();
            Element divhalaman =
            doc.select("div[align=center][class=halaman]").first();
            sukses = saveData(tables);

            Elements ass = divhalaman.select("a");

            if (ass != null){
                for (Element a : ass){
```

```

        String link = AlamatHTTP.URL_SERVER + "/mahasiswa/"
+ a.attr("href");
        is = HTTPConnection.getInstance().send(link, "GET",
null);

        html =
HTTPConnection.getInstance().inputStreamToStr(is);
        doc = Jsoup.parse(html);
        tables = doc.select("table[align=left] tr td
table[align=center][width=100%]").first();
        sukses += saveData(tables);
    }
}
} catch (Exception e){}

time2 = new Date().getTime() - time2;
Log.d("AKSES-SEC", "Lama Proses Kalender Akademik : " + time2 +
" s, total : " + (time2+time) + " s");

return sukses;
}

private static int saveData(Element table){
    ContentValues cv = new ContentValues();
    KalenderAkad kalenderAkadModel = new
KalenderAkad(Database.instance);
    int sukses = 0;
    int baris = 0;
    String tgl = "", tgl2 = "";

    try {
        for (Element tr : table.select("tr")){
            baris++;
            if (baris == 1){
                continue;
            }

            Elements td = tr.select("td");

            int ibln = bulanToInt(td.get(2).text().trim());
            String tahun = td.get(3).text().trim().replace("\\D+",
""");

            tgl = td.get(1).text().trim();
            String[] tgls2 = tgl.substring(5).split("-");
            String[] tgls = tgl.split(" ");
            tgl2 = "";

            if (tgls2.length >= 2){
                String[] s2 = tgls2[1].trim().split(" ");
                if (s2.length > 2){
                    String thn2 = s2[2].trim().replace("'", "20");
                    tgl2 = String.format("%s-%02d-%02d", thn2,
blnToInt(s2[1].trim()), Integer.parseInt(s2[0].trim()));
                }
            }

            if (tgls.length > 2){
                tgl = tgls[0];
                if (tgl.indexOf("-") > 0){

```

```

        String[] stgl = tgl.split("-");
        tgl = stgl[0].trim();

        if (tgl2.equals("")){
            tgl2 = stgl[1].trim();
            tgl2 = String.format("%s-%02d-%02d", tahun,
ibln, Integer.parseInt(tgl2));
        }
    }

    int itgl = Integer.parseInt(tgl.replaceAll("[^0-
9.,,]+", ""));

    tgl = String.format("%s-%02d-%02d", tahun, ibln,
itgl);

    if (tgl2.equals("")){
        if (tgl.indexOf(">")>=0){
            tgl2 = addDate(tgl, 7);
        }
    }

    cv.put("tgl", tgl);
    cv.put("tgl2", tgl2);
    cv.put("tg", td.get(1).text().trim());
    cv.put("bln", td.get(2).text().trim());
    cv.put("ibln", ibln);
    cv.put("thn", tahun);
    cv.put("kegiatan", td.get(4).text().trim());
    cv.put("stat", "N");

    String wc = "tgl=" +
kalenderAkadModel.escapeString(tgl) +
        " AND bln=" +
kalenderAkadModel.escapeString(cv.getAsString("bln")) + " " +
        " AND thn=" +
kalenderAkadModel.escapeString(tahun) + " " +
        " AND kegiatan=" +
kalenderAkadModel.escapeString(cv.getAsString("kegiatan"));

    if (kalenderAkadModel.dataExist(wc) == 0){
        if (kalenderAkadModel.create(cv)) sukses++;
    }
}
} catch (Exception e){
    Log.d("M-Students UBSI", "KalenderAkademikModel.saveData :
" + e.getMessage()
        + " : cause = " + e.getCause() + " tgl = " + tgl +
" , tgl2 = " + tgl2);
}
return sukses;
}

public static String addDate(String tgl, int tambah){
    String ret = "";
    try {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        Calendar calendar = Calendar.getInstance();

```

```

        calendar.setTime(sdf.parse(tgl));
        calendar.add(Calendar.DATE, tambah);
        Date d = new Date(calendar.getTimeInMillis());
        ret = sdf.format(d);
    } catch (Exception e){}
    return ret;
}

public static int bulanToInt(String namabulan){
    String n = namabulan.toLowerCase();
    if (n.equals("januari")){return 1;}
    else if (n.equals("februari")){return 2;}
    else if (n.equals("maret")){return 3;}
    else if (n.equals("april")){return 4;}
    else if (n.equals("mei")){return 5;}
    else if (n.equals("juni")){return 6;}
    else if (n.equals("juli")){return 7;}
    else if (n.equals("agustus")){return 8;}
    else if (n.equals("september")){return 9;}
    else if (n.equals("oktober")){return 10;}
    else if (n.equals("november")){return 11;}

    return 12;
}

public static int blnToInt(String namabulan){
    String n = namabulan.toLowerCase();
    if (n.equals("jan")){return 1;}
    else if (n.equals("feb")){return 2;}
    else if (n.equals("mar")){return 3;}
    else if (n.equals("apr")){return 4;}
    else if (n.equals("mei")){return 5;}
    else if (n.equals("jun")){return 6;}
    else if (n.equals("jul")){return 7;}
    else if (n.equals("agu")){return 8;}
    else if (n.equals("sep")){return 9;}
    else if (n.equals("okt")){return 10;}
    else if (n.equals("nov")){return 11;}

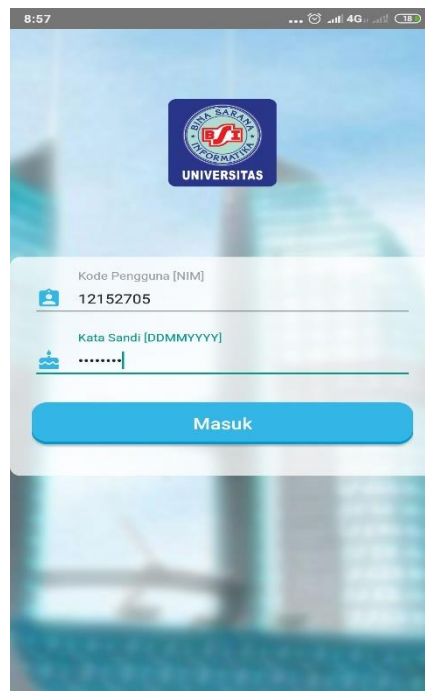
    return 12;
}
}

```

DAFTAR ISI

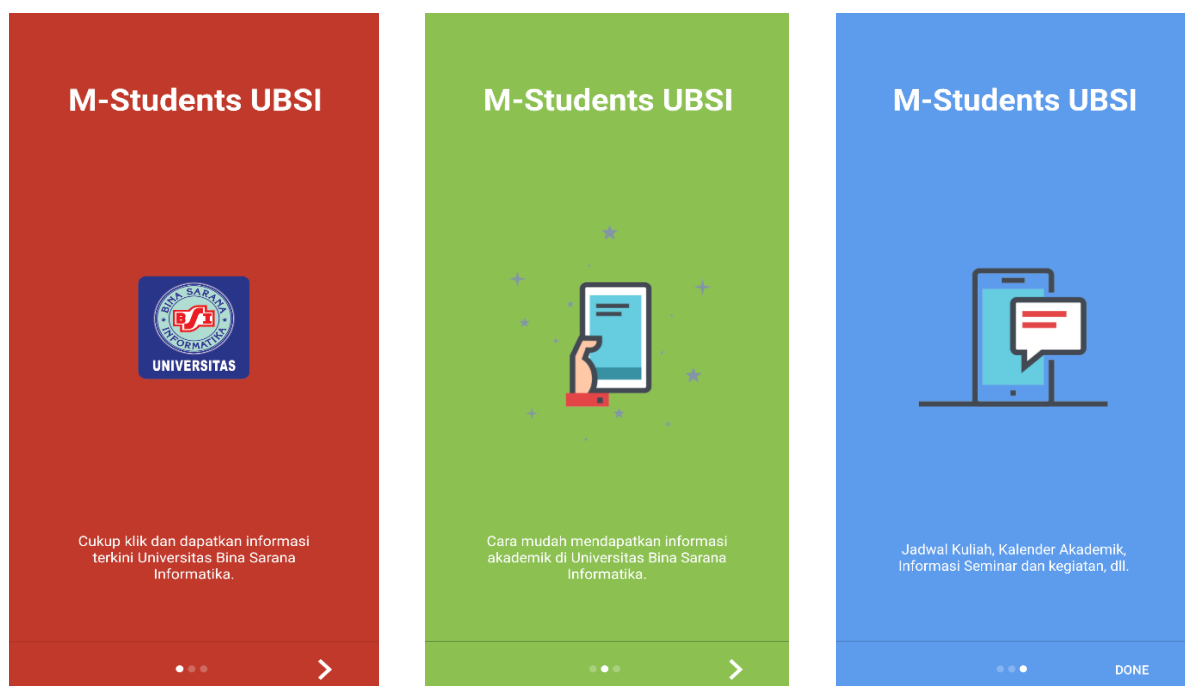
HALAMAN LOGIN	3
HALAMAN INFORMASI	3
MENU UTAMA.....	4
MENU BERANDA.....	4
DAFTAR INFO.....	5
ARSIP BERITA.....	5
PEMBERITAHUAN	5
MENU NILAI MAHASISWA.....	5
NILAI MURNI	5
NILAI IPK	5
TRANSKRIP NILAI.....	5
MENU DATA MAHASISWA	6
DATA PRIBADI MAHASISWA	6
STATUS PEMBAYARAN KULIAH.....	6
MENU INFORMASI AKADEMIK.....	7
KALENDER AKADEMIK	7
JADWAL KULIAH.....	8
JADWAL MENGAJAR DOSEN.....	8
INFO TUGAS AKHIR	8
INFO WISUDA.....	8
MENU LOGOUT	8

HALAMAN LOGIN

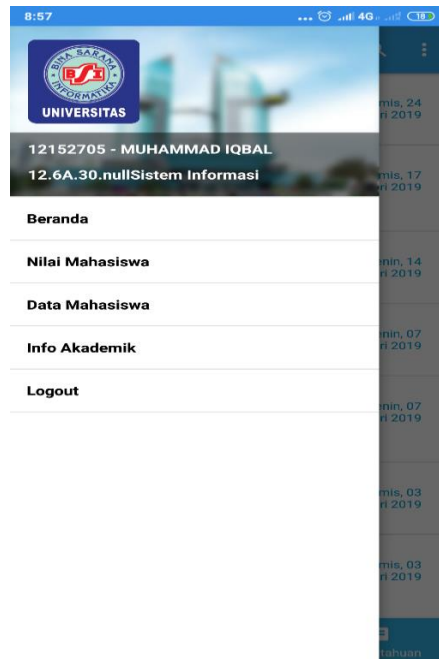


- Buka aplikasi M-Students UBSI di smartphone Anda kemudian akan tampil halaman login seperti di atas.
- Masukkan kode pengguna dan kata sandi berupa NIM dan tanggal lahir yang berformatkan DDMMYYYY.
- Sentuh atau Klik button “Masuk”.

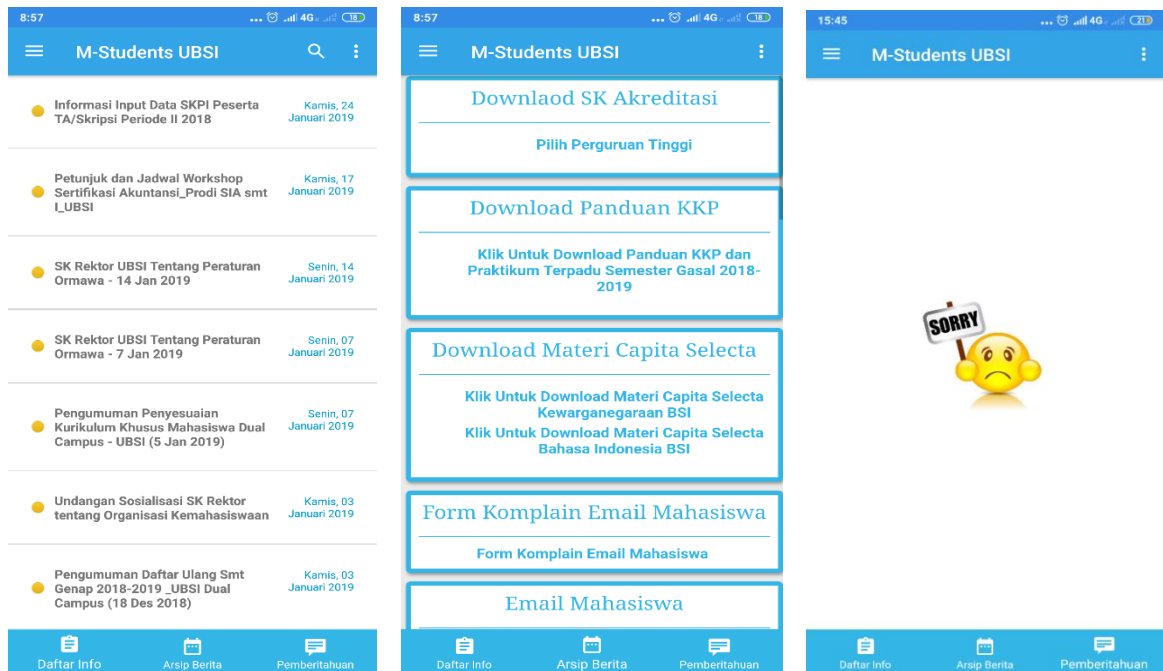
HALAMAN INFORMASI



MENU UTAMA



MENU BERANDA



Pada menu beranda terdapat 3 navigasi bawah yaitu daftar info, arsip berita dan pemberitahuan.

DAFTAR INFO

Berisikan daftar informasi akademik mahasiswa Univeritas Bina Sarana Informatika.

ARSIP BERITA

Berisikan daftar arsip berita mahasiswa Universitas Bina Sarana Informatika.

PEMBERITAHUAN

Berisikan informasi tentang panduan penggunaan aplikasi M-Students.

MENU NILAI MAHASISWA

15:45 Nilai Murni Konten saat ini belum tersedia	15:45 Nilai IPK Semester 3 24 SKS 3,13 84 SKS 2,27 APLIKASI KOMPUTER AKUNTANSI II A 028 4 sks AKUNTANSI KEUANGAN LANJUT & PRAKTIK B 030 3 sks CHARACTER BUILDING A 154 3 sks SISTEM INFORMASI AKUNTANSI C 330 4 sks PEMROGRAMAN AKUNTANSI I B 632 4 sks PERPAJAKAN B	15:45 Transkrip Nilai 60 SKS 3,18 PENDIDIKAN PANCASILA A 101 2 sks BAHASA INGGRIS I C 104 2 sks LOGIKA & ALGORITMA B 207 4 sks STATISTIKA DESKRIPTIF A 664 3 sks AKUNTANSI DASAR DAN PRAKTIK B 851 3 sks DASAR PEMROGRAMAN B 894 4 sks APLIKASI KOMPUTER AKUNTANSI I A 044 4 sks
--	---	--

Pada menu nilai mahasiswa terdapat 3 menu anak yaitu nilai murni, nilai IPK dan transkrip nilai.

NILAI MURNI

Berisikan daftar nilai murni mahasiswa Univeritas Bina Sarana Informatika.

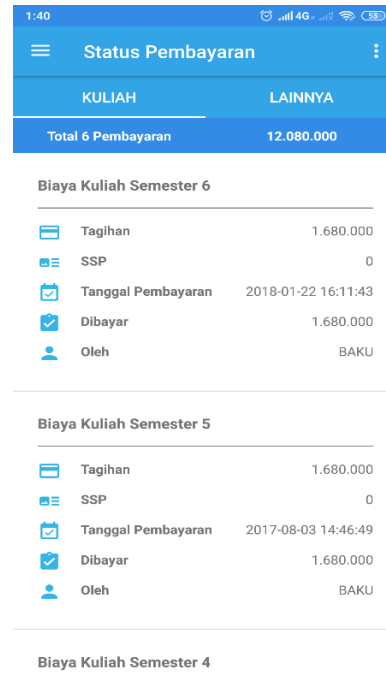
NILAI IPK

Berisikan daftar nilai (Indeks Prestasi Semester) dan total (Indeks Prestasi Kumulatif) mahasiswa Universitas Bina Sarana Informatika.

TRANSKRIP NILAI

Berisikan transkrip nilai keseluruhan mata kuliah akademik Universitas Bina Sarana Informatika.

MENU DATA MAHASISWA



Pada menu data mahasiswa terdapat 2 menu anak yaitu data pribadi mahasiswa dan status pembayaran kuliah.

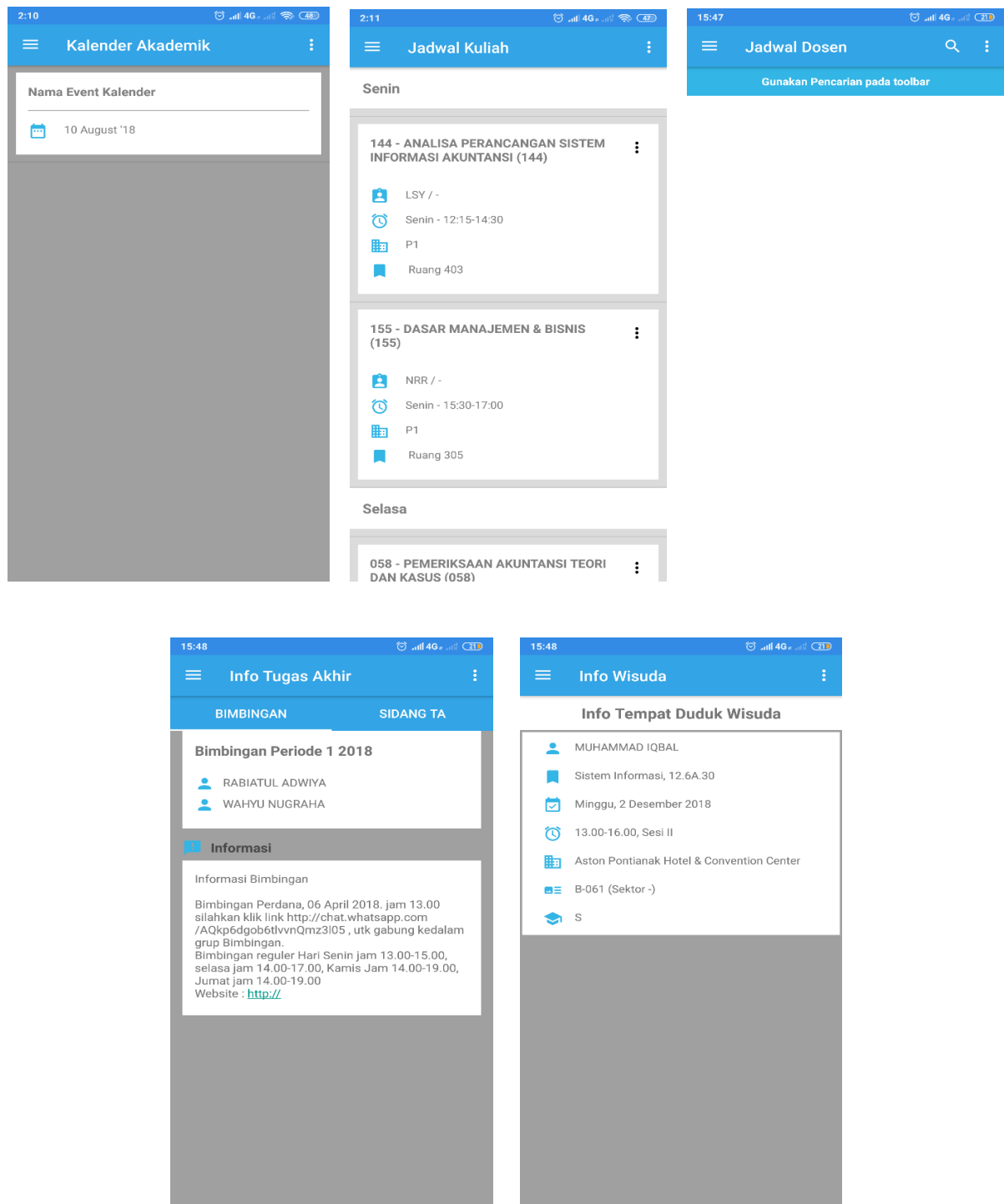
DATA PRIBADI MAHASISWA

Berisikan daftar informasi pribadi dan akademik mahasiswa Universitas Bina Sarana Informatika.

STATUS PEMBAYARAN KULIAH

Berisikan daftar informasi status pembayaran kuliah dan lainnya mahasiswa Universitas Bina Sarana Informatika.

MENU INFO AKADEMIK



Pada menu info akademik terdapat 5 menu anak yaitu kalender akademik, jadwal kuliah, jadwal mengajar dosen, info tugas akhir dan info wisuda.

KALENDER AKADEMIK

Berisikan daftar informasi kegiatan akademik Universitas Bina Sarana Informatika berdasarkan kalender.

JADWAL KULIAH

Berisikan daftar informasi jadwal kuliah mahasiswa Universitas Bina Sarana Informatika.

JADWAL MENGAJAR DOSEN

Merupakan pencarian untuk menampilkan informasi jadwal mengajar dosen Universitas Bina Sarana Informatika berdasarkan kode dosen.

INFO TUGAS AKHIR

Berisikan informasi tugas akhir mahasiswa Universitas Bina Sarana Informatika yang berupa informasi bimbingan dan sidang TA.

INFO WISUDA

Berisikan daftar informasi lengkap tentang kegiatan wisuda mahasiswa Universitas Bina Sarana Informatika.

LOGOUT

Merupakan Button yang berfungsi untuk pengguna keluar dari sistem.