LAMPIRAN

1. Lampiran *coding Main Activity*

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private DrawerLayout drawerLayout;
    private ProgressDialog progressDialog;

    public void showProgress(final String arg0) {
        dismissProgress();
        progressDialog = new ProgressDialog(MainActivity.this);
        progressDialog.setCancelable(false);
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage(arg0);
        progressDialog.show();
    }


    public void dismissProgress() {
        if (progressDialog != null) {
            if (progressDialog.isShowing())
                progressDialog.dismiss();
            progressDialog = null;
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
        findViewById(R.id.nav_btn).setOnClickListener(this);
        findViewById(R.id.beranda).setOnClickListener(this);
        findViewById(R.id.acaraku).setOnClickListener(this);
        findViewById(R.id.tentang).setOnClickListener(this);
        findViewById(R.id.headbar).setOnClickListener(this);
        findViewById(R.id.action).setOnClickListener(this);
        new CountDownTimer(2000, 500) {
            @Override
            public void onTick(long millisUntilFinished) {

            }

            @Override
            public void onFinish() {
                startApp();
            }
        }.start();
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.i("User", "?> " + App.instance().getUser().getNim());
        if (appStart && App.instance().getUser().getNim() == 0) {
            doLogout();
        } else if (App.instance().getUser().getNim() != 0) {
            requestPermissionsStorage();
            App.instance().initAlarm();
            ((TextView)
findViewById(R.id.nimku)).setText(App.instance().getUser().getNim() +
"");
            ((TextView)
findViewById(R.id.namaku)).setText(App.instance().getUser().getNama());
            ((TextView) findViewById(R.id.action)).setText("Logout");
        }
    }

    public static final int PERMISSIONS_REQUEST_READ_EXTERNAL_STORAGE =
23422;

    public void requestPermissionsStorage() {
        if (Build.VERSION.SDK_INT >= 23) {
            List<String> perms = new ArrayList<>();
            if
(checkSelfPermission(android.Manifest.permission.WRITE_EXTERNAL_STORAGE
) != PackageManager.PERMISSION_GRANTED)

perms.add(android.Manifest.permission.WRITE_EXTERNAL_STORAGE);
            if
(checkSelfPermission(android.Manifest.permission.READ_EXTERNAL_STORAGE)
!= PackageManager.PERMISSION_GRANTED)

perms.add(android.Manifest.permission.READ_EXTERNAL_STORAGE);
            if (!perms.isEmpty()) {
                requestPermissions(perms.toArray(new
String[perms.size()]), PERMISSIONS_REQUEST_READ_EXTERNAL_STORAGE);
                return;
            }
        }
        App.instance().openStorage();
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        boolean f = false;
        switch (requestCode) {

            case PERMISSIONS_REQUEST_READ_EXTERNAL_STORAGE: {
                for (int i = 0; i < permissions.length; i++) {
                    if (grantResults[i] ==
PackageManager.PERMISSION_GRANTED) {
                        Log.d("Permissions", "Permission Granted: " +
permissions[i]);
                        f = true;
                    } else if (grantResults[i] ==
PackageManager.PERMISSION_DENIED) {
                        f = false;
                        Log.d("Permissions", "Permission Denied: " +
permissions[i]);
                    }
                }
                if (f) {
                    App.instance().openStorage();
                } else {
                    Toast.makeText(getApplicationContext(), "Harus
memberi permisi ke Penyimpanan agar aplikasi berjalan !",
Toast.LENGTH_LONG).show();
```

```
                    return;
                }
            }
            break;
            default: {
                super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
            }
        }
    }

    private boolean appStart;

    public void startApp() {
        if (App.instance().getUser().getNim() == 0) {
            doLogout();
        } else {
            findViewById(R.id.splash).setVisibility(View.GONE);
            drawerLayout.setVisibility(View.VISIBLE);
            actionBeranda();
        }
        appStart = true;
    }

    public void triggerDrawer() {
        if (drawerLayout.isDrawerOpen(Gravity.LEFT))
            drawerLayout.closeDrawer(Gravity.LEFT);
        else drawerLayout.openDrawer(Gravity.LEFT);
    }

    private void closeDrawer() {
        if (drawerLayout.isDrawerOpen(Gravity.LEFT))
            drawerLayout.closeDrawer(Gravity.LEFT);
        findViewById(R.id.beranda).setBackgroundColor(screen == 1 ?
getResources().getColor(R.color.black_semi_transparent) :
Color.TRANSPARENT);
        findViewById(R.id.acaraku).setBackgroundColor(screen == 2 ?
getResources().getColor(R.color.black_semi_transparent) :
Color.TRANSPARENT);
        findViewById(R.id.tentang).setBackgroundColor(screen == 3 ?
getResources().getColor(R.color.black_semi_transparent) :
Color.TRANSPARENT);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.nav_btn:
                triggerDrawer();
                break;
            case R.id.beranda:
                actionBeranda();
                break;
            case R.id.acaraku:
                actionAcaraku();
                break;
            case R.id.tentang:
                actionTentang();
                break;
            case R.id.headbar:
                startActivity(new Intent(this, UserActivity.class));
                break;
            case R.id.action:
                new AlertDialog.Builder(this).setTitle("Logout")
                        .setMessage("Anda yakin ingin logout pengguna 
?")
                        .setPositiveButton("Ya", new 
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, 
int which) {
                                triggerDrawer();
                                doLogout();
                            }
                        }).setNegativeButton("Tidak", null).show();
        }
    }

    private void doLogout() {
        showProgress("Loading ...");

        if (App.instance().getUser().getNim() != 0) {
            App.instance().save();
            App.instance().loadData();
            Restful.update(App.instance().getUser(), new 
Callback<Restful.UserCallback>() {
                @Override
                public void onResponse(Call<Restful.UserCallback> call, 
Response<Restful.UserCallback> response) {
                    dismissProgress();
                    if (response.body() != null) {
                        App.instance().logout();
                        beranda.initView();
                        Intent i = new Intent(MainActivity.this, 
UserActivity.class);
                        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                        startActivity(i);
                        finish();
                    }
                }

                @Override
                public void onFailure(Call<Restful.UserCallback> call, 
Throwable t) {
                    dismissProgress();
                    Toast.makeText(MainActivity.this, 
t.getLocalizedMessage(), Toast.LENGTH_LONG).show();
                }
            });
        } else {
            App.instance().logout();
            Intent i = new Intent(MainActivity.this, 
UserActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(i);
            finish();
        }
    }

    @Override
    public void onBackPressed() {
        if (screen == 1) super.onBackPressed();
        else actionBeranda();
    }

    private int screen;
    private final Beranda beranda = new Beranda();

    public void actionBeranda() {
        ((TextView)
findViewById(R.id.tv_ab)).setText(getResources().getString(R.string.app
_name));

getSupportFragmentManager().beginTransaction().replace(R.id.container,
beranda).commit();
        screen = 1;
        closeDrawer();
    }

    private final Acaraku acaraku = new Acaraku();

    public void actionAcaraku() {
        ((TextView) findViewById(R.id.tv_ab)).setText("Acara Saya");

getSupportFragmentManager().beginTransaction().replace(R.id.container,
acaraku).commit();
        screen = 2;
        closeDrawer();
    }

    public void actionTentang() {
        ((TextView) findViewById(R.id.tv_ab)).setText("Tentang
Aplikasi");
        screen = 3;

getSupportFragmentManager().beginTransaction().replace(R.id.container,
new Tentang()).commit();
        closeDrawer();
    }
}
```

2. Lampiran coding Jadwal Acara *Activity*

```
public class
JadwalAcaraActivity extends AppCompatActivity {

    private ProgressDialog progressDialog;

    public void showProgress(final String arg0) {
        dismissProgress();
        progressDialog = new
ProgressDialog(JadwalAcaraActivity.this);
        progressDialog.setCancelable(false);

progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage(arg0);
        progressDialog.show();
    }


    public void dismissProgress() {
        if (progressDialog != null) {
            if (progressDialog.isShowing())
                progressDialog.dismiss();
            progressDialog = null;
        }
    }

    public static final String TAG =
JadwalAcaraActivity.class.getSimpleName();

    private boolean before(Calendar param) {
        Calendar now = Calendar.getInstance();
        if (now.get(Calendar.DAY_OF_YEAR) <
param.get(Calendar.DAY_OF_YEAR))
            return true;
        if (now.get(Calendar.DAY_OF_YEAR) ==
param.get(Calendar.DAY_OF_YEAR) &&
                now.get(Calendar.HOUR_OF_DAY) <
param.get(Calendar.HOUR_OF_DAY))
            return true;
        if (now.get(Calendar.DAY_OF_YEAR) ==
param.get(Calendar.DAY_OF_YEAR) &&
                now.get(Calendar.HOUR_OF_DAY) ==
param.get(Calendar.HOUR_OF_DAY) &&
                now.get(Calendar.MINUTE) <
param.get(Calendar.MINUTE))
            return true;
        return false;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_jadwal_acara);
        findViewById(R.id.nav_btn).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });
        final JadwalAcara acara = App.instance().getJadwal();
        if (acara != null) {
            if (acara.getKategori() != null &&
!acara.getKategori().isEmpty())
                ((TextView)
findViewById(R.id.tv_ab)).setText(acara.getKategori());
            ((TextView)
findViewById(R.id.waktu)).setText(TextUtil.dayCalendarToString(acara
.getCalendar())
                    + ", " + acara.getTime());
            ((TextView)
findViewById(R.id.nama)).setText(acara.getNama());
            ((TextView)
findViewById(R.id.deskripsi)).setText(acara.getDeskripsi());
            ((TextView)
findViewById(R.id.lokasi)).setText(acara.getLokasi());
            ((TextView)
findViewById(R.id.pendaftaran)).setText(acara.getPendaftaran());
            ((TextView)
findViewById(R.id.fasilitas)).setText(acara.getFasilitas());
            final ImageView img = (ImageView)
findViewById(R.id.image);
            if (App.instance().getBitmap(acara.getGambar()) == null)
                ImageLoader.getInstance().loadImage(Restful.DATA +
acara.getGambar(), new ImageLoadingListener() {
                    @Override
                    public void onLoadingStarted(String imageUri,
View view) {

                    }

                    @Override
                    public void onLoadingFailed(String imageUri,
View view, FailReason failReason) {

                    }

                    @Override
                    public void onLoadingComplete(String imageUri,
View view, Bitmap loadedImage) {
                        App.instance().putBitmap(acara.getGambar(),
loadedImage);

img.setImageBitmap(App.instance().getBitmap(acara.getGambar()));
                    }

                    @Override
                    public void onLoadingCancelled(String imageUri,
View view) {

                    }
                });
            else
img.setImageBitmap(App.instance().getBitmap(acara.getGambar()));

(findViewById(R.id.go)).setVisibility(before(acara.getCalendar()) ?
View.VISIBLE : View.GONE);
            boolean acaraku =
App.instance().getFollows().containsKey(acara.getId());
            if (before(acara.getCalendar())) {
                findViewById(R.id.fb).setVisibility(View.GONE);
                findViewById(R.id.go).setVisibility(View.VISIBLE);
                if (acaraku) {
                    ((TextView)
findViewById(R.id.go)).setText("Hapus dari Acara Saya");
                } else {
                    ((TextView)
findViewById(R.id.go)).setText("Ikuti Acara");
                }
                findViewById(R.id.go).setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        boolean acaraku =
App.instance().getFollows().containsKey(acara.getId());
                        Log.i("acaraku", "? " + acaraku);
                        if (acaraku) {
                            Toast.makeText(JadwalAcaraActivity.this,
"Berhasil dihapus !", Toast.LENGTH_LONG).show();
                            Long val =
App.instance().getFollows().remove(acara.getId());
                            Log.i("delete", "? " + val);
                            App.instance().save();
                            App.instance().initAlarm();
                            Log.i("getFollows", "? " + new
GsonBuilder().create().toJson(App.instance().getFollows()));
                            ((TextView)
findViewById(R.id.go)).setText("Ikuti Acara");
                        } else {
                            ((TextView)
findViewById(R.id.go)).setText("Hapus dari Acara Saya");
                            Toast.makeText(JadwalAcaraActivity.this,
"Berhasil disimpan, Anda akan mendapatkan notifikasi jelang acara
dilaksanakan."
                                    , Toast.LENGTH_LONG).show();

App.instance().getFollows().put(acara.getId(),
acara.getCalendar().getTimeInMillis());
                            App.instance().save();
                            App.instance().initAlarm();
                        }
                    }
                });
            } else {
                boolean peserta = false;
                for (Peserta f :
App.instance().getUser().getPesertas()) {
                    if (f.getId_acara() == acara.getId()) {
                        ((EditText)
findViewById(R.id.feedback)).setText(f.getKomentar());
                        peserta = true;
                        break;
                    }
                }
                findViewById(R.id.go).setVisibility(View.GONE);
                if (acaraku) {
                    ((TextView)
findViewById(R.id.save)).setText("Beri Komentar");


findViewById(R.id.fb).setVisibility(View.VISIBLE);
                    findViewById(R.id.save).setOnClickListener(new
View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            String feedback = ((EditText)
findViewById(R.id.feedback)).getText().toString();

                            for (Peserta f :
App.instance().getUser().getPesertas()) {
                                if (f.getId_acara() ==
acara.getId()) {
                                    f.setKomentar(feedback);
                                    pushKomentar(f);
                                    return;
                                }
                            }
                            Peserta f = new Peserta();

f.setId_user(App.instance().getUser().getNim());
                            f.setId_acara(acara.getId());
                            f.setKomentar(feedback);
                            f.setWaktu(System.currentTimeMillis());
                            pushKomentar(f);
                        }
                    });
                } else {
                    findViewById(R.id.fb).setVisibility(View.GONE);
                }
            }

        }
    }
    private void pushKomentar(Peserta peserta) {
        final Callback<Restful.PersertaCallback> c = new 
Callback<Restful.PersertaCallback>() {
            @Override
            public void onResponse(Call<Restful.PersertaCallback> 
call, Response<Restful.PersertaCallback> response) {
                dismissProgress();
                if (response.body() != null) {
                    Toast.makeText(JadwalAcaraActivity.this, 
"Komentar berhasil disimpan, Terima kasih :)", 
Toast.LENGTH_LONG).show();
                    for (Peserta f : 
App.instance().getUser().getPesertas()) {
                        if (f.getId() == 
response.body().data.getId()) {
                            return;
                        }
                    }
                    
App.instance().getUser().getPesertas().add(response.body().data);
                    App.instance().save();
                }
            }

            @Override
            public void onFailure(Call<Restful.PersertaCallback> 
call, Throwable t) {
                dismissProgress();
                Toast.makeText(JadwalAcaraActivity.this, 
t.getLocalizedMessage(), Toast.LENGTH_LONG).show();
            }
        };
        showProgress("Loading ...");
        Restful.createPeserta(peserta, c);
    }

    private void pushPeserta(Peserta peserta) {
        final Callback<Restful.PersertaCallback> c = new 
Callback<Restful.PersertaCallback>() {
            @Override
            public void onResponse(Call<Restful.PersertaCallback> 
call, Response<Restful.PersertaCallback> response) {
                dismissProgress();
                if (response.body() != null) {
                    Toast.makeText(JadwalAcaraActivity.this, 
"Komentar berhasil disimpan, Terima kasih :)", 
Toast.LENGTH_LONG).show();
                    for (Peserta f : 
App.instance().getUser().getPesertas()) {
                        if (f.getId() == 
response.body().data.getId()) {
                            return;
                        }
                    }
                    
App.instance().getUser().getPesertas().add(response.body().data);
                    App.instance().save();
                }
            }

            @Override
            public void onFailure(Call<Restful.PersertaCallback> 
call, Throwable t) {
                dismissProgress();
                Toast.makeText(JadwalAcaraActivity.this,
t.getLocalizedMessage(), Toast.LENGTH_LONG).show();
            }
        };
        showProgress("Loading ...");
        Restful.createPeserta(peserta, c);
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        finish();
    }
}
```

3. Lampiran *coding* Notifikasi *Activity*

```
public class NotificationActivity extends AppCompatActivity
implements View.OnClickListener{
    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.hadir) {
            Peserta f = new Peserta();
            f.setId_user(App.instance().getUser().getNim());
            f.setId_acara(acara.getId());
            f.setWaktu(System.currentTimeMillis());
            pushPeserta(f);
        } else
            onBackPressed();
    }

    private ProgressDialog progressDialog;

    public void showProgress(final String arg0) {
        dismissProgress();
        progressDialog = new
ProgressDialog(NotificationActivity.this);
        progressDialog.setCancelable(false);

progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage(arg0);
        progressDialog.show();
    }


    public void dismissProgress() {
        if (progressDialog != null) {
            if (progressDialog.isShowing())
                progressDialog.dismiss();
            progressDialog = null;
        }
    }
    private void pushPeserta(Peserta peserta) {
        final Callback<Restful.PersertaCallback> c = new
Callback<Restful.PersertaCallback>() {
            @Override
            public void onResponse(Call<Restful.PersertaCallback>
call, Response<Restful.PersertaCallback> response) {
```

```
                dismissProgress();
                if (response.body() != null) {
                    Toast.makeText(NotificationActivity.this,
"Terima kasih Anda menghadiri acara " + acara.getNama()
                            + ", Silahkan mengisi komentar di detail
acara."
                            , Toast.LENGTH_LONG).show();
                    for (Peserta f :
App.instance().getUser().getPesertas()) {
                        if (f.getId() ==
response.body().data.getId()) {
                            return;
                        }
                    }

App.instance().getUser().getPesertas().add(response.body().data);
                    App.instance().save();
                    onBackPressed();
                }
            }

            @Override
            public void onFailure(Call<Restful.PersertaCallback>
call, Throwable t) {
                dismissProgress();
                Toast.makeText(NotificationActivity.this,
t.getLocalizedMessage(), Toast.LENGTH_LONG).show();
            }
        };
        showProgress("Loading ...");
        Restful.createPeserta(peserta, c);
    }

    public static final String TAG =
NotificationActivity.class.getSimpleName();
    private JadwalAcara acara;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notification);
        findViewById(R.id.hadir).setOnClickListener(this);
        findViewById(R.id.tidak).setOnClickListener(this);
        final int id = getIntent().getIntExtra(TAG, 0);
        if (id != 0) {
            App.instance().loadData();
            for (final JadwalAcara acara :
App.instance().getJadwalAcara()) {
                if (acara.getId() == id) {
                    NotificationActivity.this.acara = acara;
                    ((TextView)
findViewById(R.id.waktu)).setText(TextUtil.dayCalendarToString(acara
.getCalendar())
                            + ", " + acara.getTime());
                    ((TextView)
findViewById(R.id.nama)).setText(acara.getNama());
                    ((TextView)
findViewById(R.id.lokasi)).setText(acara.getLokasi());
                    final ImageView img = (ImageView)
findViewById(R.id.image);
                    if (App.instance().getBitmap(acara.getGambar())
== null)
ImageLoader.getInstance().loadImage(Restful.DATA +
acara.getGambar(), new ImageLoadingListener() {
                            @Override
                            public void onLoadingStarted(String
imageUri, View view) {

                            }

                            @Override
                            public void onLoadingFailed(String
imageUri, View view, FailReason failReason) {

                            }

                            @Override
                            public void onLoadingComplete(String
imageUri, View view, Bitmap loadedImage) {

App.instance().putBitmap(acara.getGambar(), loadedImage);

img.setImageBitmap(App.instance().getBitmap(acara.getGambar()));
                            }

                            @Override
                            public void onLoadingCancelled(String
imageUri, View view) {

                            }
                        });
                    else
img.setImageBitmap(App.instance().getBitmap(acara.getGambar()));
                    break;
                }
            }
        }
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent(this, MainActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(i);
        finish();
    }
}
```

4. Lampiran *coding User Activity*

```
public class UserActivity extends AppCompatActivity {

    public final static String TAG =
UserActivity.class.getSimpleName();
    private ProgressDialog progressDialog;

    public void showProgress(final String arg0) {

        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                dismissProgress();
                progressDialog = new
ProgressDialog(UserActivity.this);
                progressDialog.setCancelable(false);

progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                progressDialog.setMessage(arg0);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        progressDialog.show();
                    }
                });
            }
        });
    }


    public void dismissProgress() {

        if (progressDialog != null) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    if (progressDialog.isShowing())
                        progressDialog.dismiss();
                    progressDialog = null;
                }
            });
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user);
        if (App.instance().getUser().getNim() == 0)
            tampilLogin();
        else {
            register.setUser(App.instance().getUser());
            tampilRegister();
        }
    }

    @Override
    public void onBackPressed() {
        switch (screen) {
            case 1:
                super.onBackPressed();
                finish();
                break;
            case 2:
                if (App.instance().getUser().getNim() == 0)
                    tampilLogin();
                else finish();
                break;
        }
    }

    private final Login login = new Login();
    private final Register register = new Register();
    private int screen;
    public void tampilLogin() {
        
getSupportFragmentManager().beginTransaction().replace(R.id.containe
r, login).commit();
        screen = 1;
    }

    public void tampilRegister() {
        
getSupportFragmentManager().beginTransaction().replace(R.id.containe
r, register).commit();
        screen = 2;
    }
}
```