
	<p>Tersedia online di: http://ejournal-balitbang.kkp.go.id/index.php/JP e-mail: jurnalpari@gmail.com JURNAL PARI Volume 5 Nomor 2 Desember 2019 p-ISSN: 2502-0730 e-ISSN : 2549-0133</p>	
---	--	---

PERANCANGAN SISTEM INFORMASI PREDIKSI PASANG SURUT BERBASIS ANDROID DI PUSAT RISET KELAUTAN DESIGNING ANDROID-BASED TIDAL PREDICTION INFORMATION SYSTEMS AT THE MARINE RESEARCH CENTER

Dani Saepuloh

Pusat Riset Kelautan, Badan Riset dan Sumber Daya Manusia Kelautan dan Perikanan, KKP
Jl. Pasir Putih 1 Ancol Timur, Jakarta Telp : (021) 64711583, Fax : (021) 64711654
Diterima tanggal : 2 Agustus 2019 Diterima setelah perbaikan : 25 September 2019
disetujui terbit : 15 November 2019

ABSTRAK

Pasang surut air laut merupakan parameter yang sangat penting bagi navigasi di pelabuhan, Pusat Riset Kelautan, *Marine & Coastal Data Laboratory* (MCDL) membuat prediksi pasang surut di 39 pelabuhan seluruh Indonesia. Penelitian ini bertujuan menghasilkan sistem informasi prediksi pasang surut berbasis android dengan menggunakan bahasa pemrograman Java, Google Maps Api sebagai basemapsnya dan data prediksi pasang surut yang dihasilkan oleh MCDL. Informasi yang ditampilkan dalam bentuk grafik dan data tabel untuk 14 hari kedepan (2 minggu), dengan resolusi temporal per jam dengan pemuktahiran (*update*) informasi per 14 hari. Dalam aplikasi android prediksi pasang surut (pasut) terdiri dari 4 menu yaitu Pelabuhan, Glosarium, About, Disclaimer. Menu Pelabuhan menampilkan informasi nama pelabuhan perikanan. Menu Glosarium berisikan informasi daftar istilah pasang surut. Menu About berisikan informasi pemilik aplikasi yang bertujuan untuk pengguna dapat berinteraksi dengan pemilik aplikasi. Menu Disclaimer (penyanggahan) berisikan informasi bahwa informasi perkiraan pasut ini digunakan sebagai rujukan (referensi) ketinggian muka/paras air laut di lokasi Pelabuhan Perikanan untuk kepentingan operasi keluar dan masuknya kapal. Informasi tentang kedalaman batimetri dan kondisi dasar perairan lokasi tidak disediakan pada aplikasi ini.

KATA KUNCI : Prediksi Pasang Surut; Pasut; Android; Pusriskel; MCDL.

ABSTRACT

Tide is a very important parameter for navigation in ports, the Center for Marine Research, Marine & Coastal Data Laboratory (MCDL) makes tide predictions in 39 ports throughout Indonesia. This study aims to produce an Android-based tidal prediction information system using the Java programming language, Google Maps Api as its basemap and tidal prediction data generated by MCDL. Information displayed in graphical form and data tables for the next 14 days (2 weeks), with hourly temporal resolution with updated information per 14 days. In the tide prediction android application (tides) consists of 4 menus namely Port, Glossary, About, Disclaimer. The Port menu displays information on the name of the fishing port. The Glossary menu contains tidal term glossary information. The About menu contains application owner information that aims for users to interact with the application chooser. The Disclaimer Menu contains information that this tidal estimated information is used as a reference (reference) for sea level / sea level at the Fishery Port location for the purpose of operating in and out of ships. Information about the bathymetry depth and the bottom waters of the location is not provided in this application.

KEYWORDS: Tidal Prediction; Tidall; Android; Pusriskel; MCDL.

Korespondensi penulis:

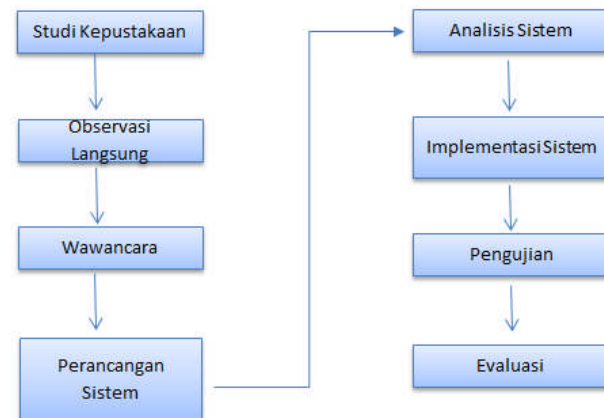
Jl. Pasir Putih 1 Ancol Timur, Jakarta Telp : (021) 64711583, Fax : (021) 64711654
email : d4n1.saepuloh@gmail.com

PENDAHULUAN

Pasang surut air laut merupakan parameter yang sangat penting bagi navigasi di pelabuhan, Pusat Riset Kelautan, *Marine & Coastal Data Laboratory* (MCDL) membuat prediksi pasang surut di 39 pelabuhan seluruh Indonesia. Penelitian ini bertujuan menghasilkan sistem informasi prediksi pasang surut berbasis android dengan menggunakan bahasa pemrograman Java, Google Maps Api sebagai basemapsnya dan data prediksi pasang surut yang dihasilkan oleh MCDL. Informasi yang ditampilkan dalam bentuk grafik dan data tabel untuk 14 hari kedepan (2 minggu), dengan resolusi temporal per jam dengan pemuktahiran (*update*) informasi per 14 hari. Dalam aplikasi android prediksi pasang surut (pasut) terdiri dari 4 menu yaitu Pelabuhan, Glosarium, About, Disclaimer. Menu Pelabuhan menampilkan informasi nama pelabuhan perikanan. Menu Glosarium berisikan informasi daftar istilah pasang surut. Menu About berisikan informasi pemilik aplikasi yang bertujuan untuk pengguna dapat berinteraksi dengan pemilih aplikasi. Menu Disclaimer (penyanggahan) berisikan informasi bahwa informasi perkiraan pasut ini digunakan sebagai rujukan (referensi) ketinggian muka/paras air laut di lokasi Pelabuhan Perikanan untuk kepentingan operasi keluar dan masuknya kapal. Informasi tentang kedalaman batimetri dan kondisi dasar perairan lokasi tersebut tersebut tidak disediakan pada aplikasi ini.

Info prakiraan pasut ini adalah hasil peramalan menggunakan 11 Komponen Harmonik, yakni 4 Komponen Semidiurnal (M2, S2, N2, K2), 4 Komponen Diurnal (K1, O1, P1, Q1), dan 3 Komponen turunan dari Semidiurnal (MF, MM, M4). Seluruh komponen tersebut adalah turunan dari data satelit altimetry Topex/Poseidon (Egbert & Erofeeva, 2002). Peramalan dilakukan terhadap koordinat-koordinat Pelabuhan Perikanan yang dikelola oleh Direktorat Jenderal Perikanan Tangkap, Kementerian Kelautan dan Perikanan Republik Indonesia (DGCF, 2009). Sedangkan verifikasi hasil peramalan terhadap data pengukuran/observasi telah dilakukan dengan hasil cukup memuaskan (Ramadhan, 2011; Pranowo & Wirasantosa, 2011).

METODOLOGI



Gambar 1. Tahapan langkah dalam penelitian ini

HASIL DAN PEMBAHASAN

A. Langkah Pembangunan Sistem

Gambaran Umum Sistem

Sistem informasi prediksi pasang surut dibangun dengan berbasis Android dengan menggunakan bahasa pemrograman Java dan dipadukan dengan database MySQL. Jenis pemrograman Java yang digunakan pada aplikasi ini adalah pemrograman berorientasi objek. Peta menggunakan Google Maps Api untuk mengetahui lokasi pelabuhan berada. Halaman pada Menu Utama meliputi beberapa bagian antara lain header, menu, content serta beberapa menu variasi tambahan guna mempercantik aplikasi. Header berisikan logo dan keterangan tentang nama aplikasi, menu berisikan 4 pilihan menu yaitu Pelabuhan, Glosarium, About, Disclaimer.

B. Implementasi Perangkat Keras dan Perangkat Lunak

1. Perangkat Keras (Hardware)

Perangkat keras yang digunakan untuk pembangunan aplikasi ini adalah sebuah komputer dengan spesifikasi:

- Processor Intel(R) Xeon(R) CPU E5620 @2.4 GHz 2.4 GHz
- Hardisk 1 TB 5400-rpm
- RAM 12 GB
- VGA Intel Graphics Media Accelerator (GMA) HD

2. Perangkat Lunak (Software)

Piranti lunak yang digunakan pada pembangunan aplikasi ini:

- Microsoft Windows 7 Professional Service Pack 1, 32 bit sebagai platform sistem operasi dalam pengembangan aplikasi
- XAMPP Web Server Version 1.7.3 sebagai webserver
- MySQL Database Version 5.1.41 sebagai DBMS
- phpMyAdmin Version 3.2.4 sebagai Database Manager.
- Android Studio coding (menuliskan source code Java for Android)
- Aplikasi Adobe Potoshop CS6 untuk mendesain tampilan aplikasi

C. Hasil Implementasi Antarmuka

1. *Splash Screen* (Halaman Pembuka)

Halaman yang akan ditampilkan pertama kali ketika user membuka aplikasi sistem informasi prediksi pasang surut. Halaman *Splash Screen* ini berisi elemen kontrol grafis yang terdiri dari jendela yang berisi gambar, logo, dan versi perangkat lunak saat ini. Layar splash biasanya muncul saat program sedang diluncurkan.

Kode Pada Halaman *Splash Screen* splashscreen.java

Kode program ini berfungsi untuk menarik (*load*) informasi dari server dan akan otomatis masuk kehalaman menu utama (*home*).



Gambar 2. Halaman *Splash Screen*

```

package com.p3sdlp.pasut;

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

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.p3sdlp.pasut.pustaka.DBHelper;
import com.p3sdlp.pasut.pustaka.ServiceHandler;

import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.View;
import android.widget.LinearLayout;
import android.app.ActivityManager;
import android.app.ActivityManager. RunningService
Info;
import android.app.AlertDialog;
import android.app.NotificationManager;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;

public class Splash extends ActionBarActivity {

    JSONArray pelabuhan = null;
    ArrayList<HashMap<String, String>>
daftarpelabuhan;
    HashMap<String, Integer> nNotip = new
HashMap<String, Integer>();
    Intent notip = new
Intent(Notification.class.getName());
    Boolean anydata = false;
    LinearLayout loading;

    private static final String TAG_PELABUHAN
= "pelabuhan";
    private static final String TAG_ID = "id";
    private static final String TAG_TITLE = "title";
    private static final String TAG_LAT =
"metakey";
    private static final String TAG_LNG =
"metadesc";

    private static final String TAG_DB = "station";
    private static final String
TAG_UNREAD_NOTIF = "unreadnotification";

    @Override
    protected void onCreate(Bundle
savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_splash);

        loading
        (LinearLayout)findViewById(R.id.LinearLayout3);
loading.setVisibility(View.INVISIBLE);

        daftarpelabuhan = new
ArrayList<HashMap<String, String>>();

        new GetData().execute();
    }

    private boolean
isNotifServiceRunning(Class<?> serviceClass) {
        ActivityManager manager =
        (ActivityManager)
        getSystemService(Context.ACTIVITY_SERVICE);
        for (RunningServiceInfo service :
        manager.getRunningServices(Integer.MAX_VALUE)) {
            if
            (serviceClass.getName().equals(service.service.getClassName()))
            {
                return true;
            }
        }
        return false;
    }

    public void prepareDB(Context c){
        DBHelper db = new DBHelper(c);

        if(db.cekAda(TAG_DB)==0){
            db.addNotif(TAG_DB, 0, 0);
        }

        if(!isNotifServiceRunning(Notification.class)){
            startService(notip);
        }

        if(db.cekAda(TAG_UNREAD_NOTIF)==0){
            db.addNotif(TAG_UNREAD_NOTIF, 0, 0);
        }else{
            db.updateNotif(TAG_UNREAD_NOTIF, 0);
        }

        for(int i = 0; i <
daftarpelabuhan.size(); i++){
            if(db.cekAdaByID(Integer.valueOf(daftarpelabuhan.get(i).get(TAG_ID)))
==0){
                db.addNotif(daftarpelabuhan.get(i).get(TAG_TITLE), 0,
Integer.valueOf(daftarpelabuhan.get(i).get(TAG_ID)));
            }else{
                db.updateNotifValue(daftarpelabuhan.get(i).get(TAG_TITLE),
Integer.valueOf(daftarpelabuhan.get(i).get(TAG_ID)));
            }
        }

        NotificationManager notif =
(NotificationManager)c.getSystemService(Context.NOTIFICATION_
SERVICE);
        notif.cancel(100);
    }

    private class GetData extends
AsyncTask<Void, Void, Void> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();

            loading.setVisibility(View.VISIBLE);
        }

        @Override
        protected Void
doInBackground(Void... arg0) {
            ServiceHandler sh = new
ServiceHandler();

            String url = "http://
p3sdlp.litbang.kkp.go.id/includes/app-
daftarPelabuhan.php?f=list";
            String jsonStr =
sh.makeServiceCall(url, ServiceHandler.GET);

            if (jsonStr != null) {
                try {

                    JSONObject jsonObj = new JSONObject(jsonStr);

                    pelabuhan
                    = jsonObj.getJSONArray(TAG_PELABUHAN);

                    for (int i =
0; i < pelabuhan.length(); i++) {
                        JSONObject c = pelabuhan.getJSONObject(i);
                        String id = c.getString(TAG_ID);
                        String title = c.getString(TAG_TITLE);
                    }
                }
            }
        }
    }
}

```




Gambar 3. Halaman Menu Utama (home)

MainActivity.java

kode file ini berfungsi menampilkan sebaran pelabuhan dalam peta

```
package com.p3sdlp.pasut;
```

```
import java.util.ArrayList;
import java.util.HashMap;
```

```
import com.google.android.gms.maps.
CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnInfo
WindowClickListener;
import com.google.android.gms.maps.GoogleMap.On
MarkerClickListener;
import com.google.android.gms.maps.OnMap
ReadyCallback;
import com.google.android.gms.maps.SupportMap
Fragment;
import com.google.android.gms.maps.model.Bitmap
Descriptor Factory;
import com.google.android.gms.maps.model.Camera
Position;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.p3sdlp.pasut.pustaka.DBHelper;
```

```
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
```

```
import android.support.v7.app.ActionBarActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.ImageButton;
```

```
@SuppressWarnings("NewApi")
public class MainActivity extends ActionBarActivity
implements
    OnMapReadyCallback,
    OnMarkerClickListener {
```

```
    private Toolbar mToolbar;
    private SupportMapFragment mapFragment;
    private MarkerOptions option;
```

```
    private static final String TAG_ID = "id";
    private static final String TAG_TITLE = "title";
    private static final String TAG_LAT =
"metakey";
    private static final String TAG_LNG =
"metadesc";
```

```
    private static final String TAG_DB = "station";
    private static final String TAG_PPS = "PPS";
    private static final String TAG_PPN = "PPN";
    private static final String TAG_PPI = "PPI";
    private static final String TAG_CPP = "CPP";
    private static final String TAG_PPP = "PPP";
```

```
    final Handler handler = new Handler();
    Runnable refresh;
```

```
    private ArrayList<HashMap<String, String>>
station;
    ImageButton InkPelabuhan;
    Boolean adaYangBaru;
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        Intent intent = getIntent();
        station =
(ArrayList<HashMap<String, String>>) intent
.getSerializableExtra("station");
```

```
        refresh = new Runnable() {
    @Override
    public void run() {
        cekPembaruan();
        handler.postDelayed(this, 2000);
    }
}
```

```

    }
};

    handler.postDelayed(refresh, 2000);

        Integer    iconnya    =
R.drawable.ic_launcher;

        InKPelabuhan = (ImageButton)
findViewById(R.id.btnPelabuhan);

        mToolbar    =    (Toolbar)
findViewById(R.id.toolbar_actionbar);
        setSupportActionBar(mToolbar);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

getSupportActionBar().setIcon(iconnya);
        getSupportActionBar().setTitle("
Prakiraan Pasut WPP RI");

        mapFragment    =
( S u p p o r t M a p F r a g m e n t )
getSupportFragmentManager()

.findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    private void clearUpdate(String target){
        DBHelper db = new DBHelper(this);
        if(db.cekNotif(target)==1){
            db.updateNotif(target, 0);
        }
    }

    public void cekPembaruan() {
        DBHelper db = new DBHelper(this);
        adaYangBaru = false;
        for(int x=0; x<station.size(); x++){
            i                f
            (db.cekNotifByID(Integer.valueOf(station.get(x).get(TAG_ID)))
            == 1) {
                adaYangBaru =
true;
            }
        }

        if (db.cekNotif(TAG_DB) == 1) {
            adaYangBaru = true;
        }

        if(adaYangBaru){

InkPelabuhan.setImageResource(R.drawable.ic_place
main_new);

        }else if(!adaYangBaru){

InkPelabuhan.setImageResource(R.drawable.ic_place
main);
        }

        public void kePelabuhan(View v) {
            DBHelper db = new DBHelper(this);
            if(db.cekNotif(TAG_DB)==1){
                db.updateNotif(TAG_DB, 0);
            }
            Intent listpelabuhan = new
Intent(getApplicationContext(),
DaftarPelabuhan.class);
            listpelabuhan.putExtra("station",
station);
            startActivity(listpelabuhan);
        }

        public void keGlosarium(View v) {
            Intent glosari = new
Intent(getApplicationContext(), Glosarium.class);
            startActivity(glosari);
        }

        public void keAbout(View v) {
            Intent about = new
Intent(getApplicationContext(), SinglePage.class);
            about.putExtra("page", "about");
            startActivity(about);
        }

        public void keDisclaimer(View v) {
            Intent disclaimer = new
Intent(getApplicationContext(),
SinglePage.class);
            disclaimer.putExtra("page",
"disclaimer");
            startActivity(disclaimer);
        }

        public double fix2digit(String value) {
            double    valDouble    =
Double.parseDouble(value) * 100;
            int vallnt = (int) valDouble;
            valDouble = vallnt / 100.0;
            return valDouble;
        }

        public boolean onMarkerClick(Marker marker)
{
            Intent listtahun = new
Intent(getApplicationContext(), DaftarTahun.class);

```

```

        listtahun.putExtra("pelabuhan",
marker.getTitle().trim().replaceAll("\\s+", ""));
        startActivity(listtahun);

        return true;
    }

    @Override
    public void onBackPressed() {
        handler.removeCallbacks(refresh);
        super.onBackPressed();
    }

    @Override
    public void onMapReady(GoogleMap map) {
        for (int obj = 0; obj < station.size();
obj++) {

            option = new
MarkerOptions();

            option.position(new
LatLng(fix2digit(station.get(obj).get(TAG_LAT)),fix2digit(station.
get(obj).get(TAG_LNG))))

            .title(station.get(obj).get(TAG_TITLE))
                .snippet("Tap here.");

            if(station.get(obj).get(TAG_TITLE).substring(0,
3).equalsIgnoreCase(TAG_PPS)){

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_BLUE));
            } else
            if(station.get(obj).get(TAG_TITLE).substring(0,
3).equalsIgnoreCase(TAG_PPN)){

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_RED));
            } else
            if(station.get(obj).get(TAG_TITLE).substring(0,
3).equalsIgnoreCase(TAG_PPI)){

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_GREEN));
            } else
            if(station.get(obj).get(TAG_TITLE).substring(0,
3).equalsIgnoreCase(TAG_CPP)){

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_YELLOW));
            } else
            if(station.get(obj).get(TAG_TITLE).substring(0,
3).equalsIgnoreCase(TAG_PPP)){

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_VIOLET));
            }else{

                option.icon(BitmapDescriptorFactory.defaultMarker(Bitmap
DescriptorFactory.HUE_MAGENTA));
            }

            map.addMarker(option);

            .title(station.get(obj).get(TAG_TITLE)).snippet("Tap
here.");

        }

        map.setOnInfoWindowClickListener(new
OnInfoWindowClickListener() {
            @Override
            public void
onInfoWindowClick(Marker marker) {
                // TODO Auto-
generated method stub

                clearUpdate(marker.getTitle());

                Intent listtahun =
new Intent(getApplicationContext(),
DaftarTahun.class);

                listtahun.putExtra("pelabuhan",
marker.getTitle().trim().replaceAll("\\s+", ""));

                listtahun.putExtra("from", "main");
                startActivity(listtahun);
            }
        });

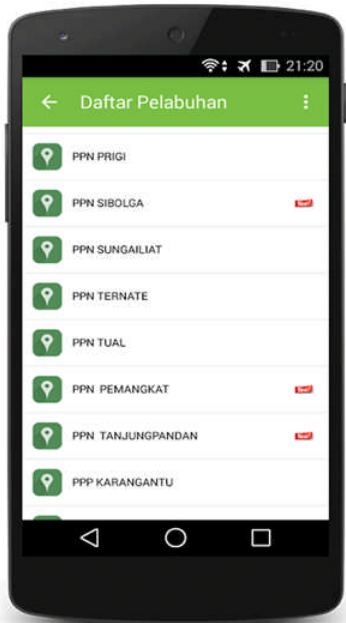
        CameraPosition cameraPosition =
new CameraPosition.Builder()
                .target(new
LatLng(-2.5, 118)).zoom(3).build();

        map.moveCamera(CameraUpdateFactory.newCamera
Position(cameraPosition));
    }
}

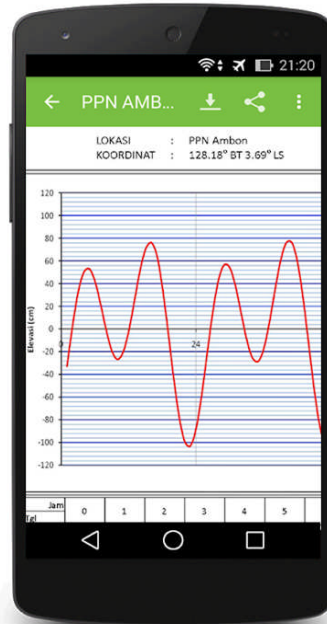
```

3. Halaman Pelabuhan

Menampilkan menu Pelabuhan yang berisi informasi nama pelabuhan perikanan. Tampilan bisa dilihat pada gambar 3 di bawah ini.



Gambar 3. Halaman Pelabuhan



Gambar 4. Halaman Prediksi Pasang Surut.

pelabuhan.java

file ini berfungsi untuk meload data pelabuhan.

package com.p3sdlp.pasut;

```
public class Pelabuhan {
    public String title;
    public Boolean baru;

    public Pelabuhan(String title, Boolean baru) {
        this.title = title;
        this.baru = baru;
    }
}
```

4. Halaman Prediksi Pasang Surut

Halaman ini menampilkan informasi yang ditampilkan dalam bentuk grafik dan data tabel untuk 14 hari kedepan (2 minggu), dengan resolusi temporal per jam dengan pemuktahiran (*update*) informasi per 14 hari pridiksi pasang surut bisa dilihat pada gambar 4.

5. Halaman Glosarium

Menampilkan menu Glosarium berisikan informasi daftar istilah pasang surut. Tampilan bisa di lihat pada gambar 5 di bawah ini:



Gambar 5. Halaman Glosarium

6. Halaman About

Menampilkan menu About berisikan informasi pemilik aplikasi yang bertujuan untuk pengguna dapat berinteraksi dengan pemilik aplikasi. Tampilan bisa di lihat pada gambar 6 di bawah ini.



Gambar 6. Halaman About

7. Halaman Disclaimer

Menampilkan menu disclaimer berisikan informasi bahwa informasi pasut dihasilkan oleh MCDL, segala penyalahgunaan penggunaan informasi yang ada dalam aplikasi ditanggung sendiri. Tampilan bisa dilihat pada gambar 7 di bawah ini:



Gambar 7. Halaman Disclaimer

D. Pengujian Sistem

Setelah tahap implementasi hasil rancangan sistem dilakukan, maka tahap selanjutnya adalah melakukan pengujian terhadap sistem. Pengujian terhadap sistem dilakukan dengan uji fungsionalitas menggunakan metode Black-Box Test.

Black-Box Test

Black-Box Test yaitu pengujian yang berfokus pada persyaratan fungsional suatu piranti lunak dengan membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional system (Pressman, 2010). Tabel di bawah ini merupakan hasil uji coba yang telah dilakukan terhadap sistem, berupa deskripsi fungsi-fungsi dari menu pada aplikasi serta menunjukkan kesesuaian fungsi yang dihasilkan ketika dilakukan pengetesan. Dari hasil uji coba tersebut dapat dilihat bahwa semua fungsi yang diharapkan telah dapat tercapai, ditandai dengan keterangan 'Berhasil' pada kolom 'Hasil Aktual'.

E. Evaluasi Sistem

Dalam pembangunan sebuah sistem, terkadang tidak semua hal yang telah direncanakan, dianalisis dan dirancang dapat diimplementasikan dengan baik. Berikut akan dijelaskan beberapa hal baik yang berhasil di implementasikan pada sistem (kelebihan sistem) maupun yang kurang berhasil diimplementasikan pada sistem (kekurangan sistem).

1. Kelebihan Sistem

- Semua masyarakat dapat menggunakan Sistem Informasi Prediksi Pasang Surut Berbasis Android, secara gratis.
- Tampilan aplikasi yang dibuat semudah mungkin.

2. Kekurangan Sistem

- Sistem aplikasi ini belum adanya menu admin (*backend*) data ditampilkan langsung dari website <http://pusriskel.litbang.kkp.go.id/>
- Belum kompatibel untuk Oparating System Applae (IOS).

Tabel 4.1 Black-Box Test

No	Nama Fungsi	Deskripsi	Hasil yang Diharapkan	Hasil Aktual
1.	<i>Splash Screen</i>	Halaman yang akan ditampilkan pertama kali ketika user membuka aplikasi sistem informasi prediksi pasang surut	Menampilkan halaman <i>Splash Screen</i>	Berhasil
2.	Menu Utama	Menampilkan menu utama (<i>home</i>), yang memberikan informasi lokasi pelabuhan yang ditampilkan melalui peta interaktif dan terdapat 4 menu dibawahnya yaitu Pelabuhan, Glosarium, About, Disclaimer.	Menampilkan menu utama, Informasi lokasi pelabuhan	Berhasil
3.	Pelabuhan	Menampilkan menu Pelabuhan yang berisi informasi nama pelabuhan perikanan	Menampilkan data Pelabuhan	Berhasil
4	Prediksi Pasang Surut	Menampilkan informasi yang ditampilkan dalam bentuk grafik dan data tabel untuk 14 hari kedepan (2 minggu), dengan resolusi temporal per jam dengan pemuktahiran (<i>update</i>) informasi per 14 hari pridiksi pasang surut	Menamfilkan informasi Prediksi Pasang Surut dalam bentuk grafik	Berhasil

SIMPULAN

Berdasarkan hasil dari analisa, perancangan, dan implementasi, maka didapatkan beberapa kesimpulan sebagai berikut :

1. Hasil penelitian telah membangun Sistem Informasi Prediksi Pasang Surut Berbasis Android.
2. Informasi prediksi pasang surut di tampilkan dalam bentuk grafik.
3. Informasi prediksi pasang surut untuk 14 hari kedepan (2 minggu), dengan resolusi temporal per jam dengan pemuktahiran (*update*) informasi per 14 hari.

DAFTAR PUSTAKA

<https://developer.android.com/guide> di akses pada tanggal 24 Oktober 2019

DJCF, 2009. Indonesian Fishing Ports 2009. *Guideline Book*. Direktorat General of Capture

Fisheries, Ministry of Marine Affairs & Fisheries of The Republic of Indonesia & Japan International Cooperation Agency.pp.113.

Egbert, G.D. & S.Y. Erofeeva, 2002.Efficient Inverse Modeling of Barotropic Ocean Tides. *Am. Met. Soc.*, (19): 183-204.

Pranowo, W.S. & S. Wirasantosa. 2011. Tidal regims of Arafura & Timor Seas. *Journal of Marine Research in Indonesia* 36(1): 21-28.

Pressman, Roger S. (2010). *Software Engineering: a practioner’s approach*. McGraw-Hill, New York.

Ramdhan, M., 2011. Komparasi Hasil Pengamatan Pasang Surut di Perairan Pulau Pramuka dan Kabupaten Pati dengan Prediksi Pasang Surut Tide Model Driver. *J. Segara*, Vol.7, No.1, p.1-10.