

DAFTAR PUSTAKA

- [1] G. Irianto, “Kebijakan dan pengelolaan air dalam pengembangan lahan rawa lebak,” pp. 9–20, 2005.
- [2] B. P. P. L. Rawa, *Profil BALITTRA*. Banjarbaru: Balai Penelitian Pertanian Lahan Rawa, 2011.
- [3] M. Alwi and C. Tapakrisnanto, “Potensi dan Karakteristik Lahan Rawa Lebak,” *Karakteristik dan Pengelolaan Lahan Rawa*, pp. 117–150, 2006.
- [4] M. Noor, K. Anwar, and B. Kartiwa, “Sistem Polder untuk Pengembangan Pertanian Berkelanjutan di Lahan Rawa Lebak,” pp. 22–38, 2017, [Online]. Available:
[http://repository.pertanian.go.id/bitstream/handle/123456789/6631/08.sistem polder untuk pengembangan pertanian berkelanjutan di lahan rawa lebak.pdf?sequence=1](http://repository.pertanian.go.id/bitstream/handle/123456789/6631/08.sistem%20polder%20untuk%20pengembangan%20pertanian%20berkelanjutan%20di%20lahan%20rawa%20lebak.pdf?sequence=1)
- [5] J. H. Purba, “KEBUTUHAN DAN CARA PEMBERIAN AIR IRIGASI UNTUK TANAMAN PADI SAWAH (*Oryza sativa* L .) (IRRIGATION WATER REQUIREMENTS AND APPLICATION METHODS FOR RICE PLANT (*Oryza sativa* L .)),” *J. Sains dan Teknol.*, vol. 10, no. 3, pp. 145–155, 2011.
- [6] N. A. Fuadi, M. Y. J. Purwanto, and S. D. Tarigan, “Kajian Kebutuhan Air dan Produktivitas Air Padi Sawah dengan Sistem Pemberian Air Secara SRI dan Konvensional Menggunakan Irigasi Pipa,” *J. Irig.*, vol. 11, no. 1, p. 23, 2016, doi: 10.31028/ji.v11.i1.23-32.
- [7] P. Kusumaningsih, U. Dhyana, and P. Bali, “Conference Paper · April 2021,” no. April, pp. 15–16, 2021.
- [8] R. R. Rachmawati, “Smart Farming 4.0 Untuk Mewujudkan Pertanian Indonesia Maju, Mandiri, Dan Modern,” *Forum Penelit. Agro Ekon.*, vol. 38, no. 2, p. 137, 2021, doi: 10.21082/fae.v38n2.2020.137-154.
- [9] V. Mayasari and A. Hairani, “Preliminary Prediction of Rice Planting for Planting Season 2019, 2020, and 2021, Based Water Balance in Freshwater Swampland,” *BIO Web Conf.*, vol. 20, no. 2020, p. 01004, 2020, doi: 10.1051/bioconf/20202001004.
- [10] T. Manik, R. Rosadi, and A. Karyanto, “Evaluasi Metode Penman-Monteith Dalam Menduga Laju Evapotranspirasi Standar (ET₀) di Dataran Rendah Propinsi Lampung, Indonesia,” *J. Keteknikan Pertan.*, vol. 26, no. 2, p. 21612, 2012.
- [11] I. Ali, “Big Data: Apa dan Pengaruhnya pada Perpustakaan? (What is Big Data and its Influence to Library),” *Media Pustak.*, vol. 22, no. 4, pp. 19–23, 2015, [Online]. Available:

<https://ejournal.perpusnas.go.id/mp/article/view/218>

- [12] A. I. Anton Yudhana, Sunardi, “Aplikasi Android Untuk Monitoring Kualitas Lahan Pertanian,” *Pros. SNST ke-9 Tahun*, pp. 43–47, 2018.
- [13] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, “Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin,” *9th Ind. Res. Work. National Semin.*, no. Juli, pp. 319–324, 2018.
- [14] M. Rais, “Penerapan Konsep Object Oriented Programming Untuk Aplikasi Pembuat Surat,” *PROtek J. Ilm. Tek. Elektro*, vol. 6, no. 2, pp. 96–101, 2019, doi: 10.33387/protk.v6i2.1242.
- [15] E. A. W. Sanad, “Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire,” *J. Penelit. Enj.*, vol. 22, no. 1, pp. 20–26, 2019, doi: 10.25042/jpe.052018.04.
- [16] Aprianto Budiman, M. Ficky Duskarnaen, and Hamidillah Ajie, “Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta,” *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 4, no. 2, pp. 32–36, 2020, doi: 10.21009/pinter.4.2.6.
- [17] Y. Astuti, B. Novianti, T. Hidayat, and D. Maulina, “Penerapan Metode Single Moving Average Untuk Peramalan Penjualan Mainan Anak,” *Semin. Nas. Sist. Inf. dan Tek. Inform. Sensitif*, vol. 4, no. July, p. 255, 2019.
- [18] ITU-T, “G.1010: End-user multimedia QoS categories,” *Int. Telecommun. Union*, vol. 1010, 2001, [Online]. Available: http://scholar.google.com.au/scholar?hl=en&q=ITU-T+Recommendation+G.1010&btnG=&as_sdt=1,5&as_sdtp=#7

LAMPIRAN

1. Aplikasi Admin

- Menu Utama

```
public class MainActivity extends AppCompatActivity {

    private Button tbDelete, tbprediksi, btWetDryNormal,
    mBtnLiatPrediksi, mBtnWaterReq;
    private Button mBtnGraphicWaterReq, mBtnMovingAvg;

    @Override
    public void onBackPressed() {
        new AlertDialog.Builder(this)
            .setMessage("Apakah anda ingin keluar dari menu
pengolahan data?")
            .setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {finish();
                }
            })
            .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    dialog.cancel();
                }
            })
            .show();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tbDelete = findViewById(R.id.bt_delete);
        tbprediksi = findViewById(R.id.bt_tambahprediksi);
        btWetDryNormal = findViewById(R.id.bt_wet_dry_normal);
        mBtnLiatPrediksi = findViewById(R.id.bt_liat_prediksi);
        mBtnWaterReq = findViewById(R.id.bt_input_water_req);
        mBtnGraphicWaterReq =
findViewById(R.id.bt_graphic_water_req);
        mBtnMovingAvg = findViewById(R.id.bt_moving_avg);

        tbDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(v.getContext(),
DeleteActivity.class);
                startActivity(intent);
            }
        });
    }
};
```

```

        tbprediksi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(v.getContext(),
PrediksiActivity.class);
                startActivity(intent);
            }
        });

        btWetDryNormal.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(v.getContext(),
WaterAvailabilityActivity.class);
                startActivity(intent);
            }
        });

        mBtnLiatPrediksi.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(),
GraphPrediksiActivity.class);
                startActivity(intent);
            }
        });

        mBtnWaterReq.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(),
WaterRequirementActivity.class);
                startActivity(intent);
            }
        });

        mBtnGraphicWaterReq.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(),
WaterReqGraphActivity.class);
                startActivity(intent);
            }
        });

        mBtnMovingAvg.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(),
MulaiActivity.class);
                startActivity(intent);
            }
        });

```

```

    }
}

```

- Akurasi Prediksi

```

public class MulaiActivity extends AppCompatActivity {

    static int deleteElement(double[] angka, int L, int R, int N)
    {
        int i, j = 0;
        for (i = 0; i < N; i++) {
            if (i <= L || i >= R) {
                angka[j] = angka[i];
                j++;
            }
        }
        return j;
    }

    private EditText input;
    private Button proses;
    private TextView tvmad, tvmse, tvmape, tvakurasi;
    private Spinner mSpinnerYear;

    private FirebaseDatabase database;
    private DatabaseReference reference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mulai);

        input = findViewById(R.id.input);
        proses = findViewById(R.id.proses);
        tvmad = findViewById(R.id.tvmad);
        tvmse = findViewById(R.id.tvmse);
        tvmape = findViewById(R.id.tvmape);
        tvakurasi = findViewById(R.id.tvakurasi);
        mSpinnerYear = findViewById(R.id.spinner_year_moving_avg);

        database = FirebaseDatabase.getInstance();
        reference = database.getReference("Data Net Laju");

        proses.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //int periode;
                float hasil1, hasil2, hasil3, hasil4, hperamalan,
                had, hse, hape;
                /*String hasilperamalan =
                tvhperamalan.getText().toString();*/
                int periode =
                Integer.valueOf(input.getText().toString().trim()) - 1;
            }
        });
    }
}

```

```

        String yearSpinner = (String)
mSpinnerYear.getSelectedItem();
        //String periodeSpinner = (String)
mSpinnerPeriode.getSelectedItem();

        int year = Integer.valueOf(yearSpinner);
        int startYear = (year - 1) - periode;

        reference.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
snapshot) {

                List<String> tahuns = new ArrayList<>();
                Map<String, List<Float>> nets = new
HashMap<>();

                for (int i = startYear; i <= year - 1;
i++) {
                    tahuns.add(String.valueOf(i));
                }

                List<Float> netLajus = new ArrayList<>();
                for (DataSnapshot item :
snapshot.getChildren()) {

                    if (item.hasChild("1")) {
                        Object o =
item.child("1").getValue();
                        String netLajuStr =
String.valueOf(o);
                        Float netLaju =
Float.valueOf(netLajuStr);
                        netLajus.add(netLaju);

                        continue;
                    }
                }

                //peramalan
                int pos = year - 1988;
                int posAkhir = (year - 1) - 1988;
                int posAwal = (year - 1 - periode) - 1988;

                float sumRamal = 0f;
                for (int i = posAwal; i <= posAkhir; i++)
                {
                    sumRamal += netLajus.get(i);
                }

                sumRamal /= (periode + 1);

                List<Float>avgNoZero = new ArrayList<>();

```

```

//error prediction
List<Float> movingAvg = new ArrayList<>();
for (int i = 0; i < netLajus.size(); i++)
{
    if ((i - periode) < 0) {
        movingAvg.add(0f);
        continue;
    }

    int awal = i - periode;
    float sumX = 0f;

    for (int j = awal; j <= i; j++) {
        sumX += netLajus.get(j);
    }

    sumX /= (periode + 1);
    movingAvg.add(sumX);
    avgNoZero.add(netLajus.get(i));
}

float yearmoving = movingAvg.get(pos-1);

//MAD
List<Float> mad = new ArrayList<>();
for(int i =0; i < (pos-1); i++){

    if(movingAvg.get(i) == 0f){
        continue;
    }

    float netLaju = netLajus.get(i+1);
    float avg = movingAvg.get(i);

    mad.add(Math.abs(netLaju-avg));
}

float sumMad = 0f;
for(Float item : mad) {
    sumMad += item;
}

sumMad /= mad.size();

tvmad.setText(String.valueOf(sumMad));

//MSE
float mse = 0f;
for(Float item : mad) {
    mse += (item * item);
}

mse /= mad.size();
tvmse.setText(String.valueOf(mse));

//MAPE

```

```

        float mape = 0f;
        for(int i = 0; i < mad.size(); i++){

            float m = mad.get(i);
            float avg = avgNoZero.get(i+1);

            float val = (m/avg);

            mape += Math.abs(val);

        }

        mape /= mad.size();

        tvmape.setText(String.valueOf(mape));

        float akurasi = 100f - mape;

tvakurasi.setText(String.valueOf(akurasi));

    }

    @Override
    public void onCancelled(@NonNull DatabaseError
error) {

    }

});
}
}
}
}
}
}

```

- Prediksi Net Laju

```

public class PrediksiActivity extends AppCompatActivity implements
View.OnClickListener {

    private Button mBtnPrediksi;
    private Spinner mSpinnerYear, mSpinnerPeriode;

    private FirebaseDatabase database;
    private DatabaseReference reference;

    private Retrofit retrofit;
    private FcmService fcmService;

    private static final String TAG = "prediksi";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_prediksi);
    }
}

```



```

        retrofit = ApiClient.getInstance();
        fcmService = retrofit.create(FcmService.class);

        mSpinnerYear = findViewById(R.id.spinner);
        mSpinnerPeriode = findViewById(R.id.spinner2);

        mBtnPrediksi = findViewById(R.id.button_prediksi);
        mBtnPrediksi.setOnClickListener(this);

        database = FirebaseDatabase.getInstance();
        reference = database.getReference("Data Net Laju");

    }

    @Override
    public void onClick(View view) {

        String yearSpinner = (String)
mSpinnerYear.getSelectedItem();
        String periodeSpinner = (String)
mSpinnerPeriode.getSelectedItem();

        int periode = Integer.valueOf(periodeSpinner) - 1;
        int year = Integer.valueOf(yearSpinner) ;
        int startYear = (year - 1) - periode;

        reference.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
snapshot) {

                List<String> tahuns = new ArrayList<>();
                Map<String, List<Float>> nets = new HashMap<>();

                for (int i = startYear; i <= year-1; i++) {
                    tahuns.add(String.valueOf(i));
                }

                for (String tahun : tahuns) {
                    for (DataSnapshot item :
snapshot.child(tahun).getChildren()) {

                        String hari = item.getKey();
                        Object o = item.getValue();
                        String netLajuStr = String.valueOf(o);
                        Float netLaju = Float.valueOf(netLajuStr);

                        if (nets.get(hari) == null) {
                            List<Float> floats = new
ArrayList<>();

                            floats.add(netLaju);
                            nets.put(hari, floats);
                        } else {
                            nets.get(hari).add(netLaju);
                        }
                    }
                }
            }
        })
    }
}

```

```

    }

    List<NetLaju> netLajus = new ArrayList<>();
    Map<String, Float> netLajus2 = new HashMap<>();

    for (Map.Entry<String, List<Float>> entry :
nets.entrySet()) {

        List<Float> floats = entry.getValue();
        float sum = 0f;

        for (Float netLaju : floats) {
            sum = sum + netLaju;
        }

        float avg = sum / floats.size();

        NetLaju netLaju = new
NetLaju(Integer.valueOf(entry.getKey().toString()), avg);
        netLajus.add(netLaju);

        netLajus2.put(entry.getKey(), avg);
    }

    DatabaseReference ref =
database.getReference("Data Net Laju");
    ref.child(yearSpinner).setValue(netLajus2);

    Notification notification = new Notification("Data
prediksi tahun " + year + " sudah di update",
        "Petani");
    MessageFcm messageFcm = new
MessageFcm("/topics/petani", notification);

    fcmService.sendMessage(messageFcm).enqueue(new
Callback<MessageFcmResponse>() {
        @Override
        public void
onResponse(Call<MessageFcmResponse> call,
Response<MessageFcmResponse> response) {
            Log.d(TAG, "onResponse: " +
response.code());
        }

        @Override
        public void onFailure(Call<MessageFcmResponse>
call, Throwable t) {
            Log.e("ERRORX", "onFailure: " +
t.getMessage());
        }
    });

    @Override
    public void onCancelled(@NonNull DatabaseError error)
{
        Log.d("ERROX", "onCancelled: " +

```

```

error.getMessage() );
        }
    });
}
}

```

- **Grafik Prediksi**

```

public class GraphPrediksiActivity extends AppCompatActivity
implements View.OnClickListener {

    private Spinner mSpinner;
    private Button mBtnShowPrediksi;
    private GraphView mGraphView;

    private FirebaseDatabase database;
    private DatabaseReference reference;

    private LineGraphSeries series;
    private RecyclerView mRecyclerView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_graph_prediksi);

        database = FirebaseDatabase.getInstance();

        mSpinner = findViewById(R.id.spinner_graph_prediksi_year);
        mBtnShowPrediksi =
findViewById(R.id.button_prediksi_result);
        mGraphView = findViewById(R.id.graph_prediksi);
        mRecyclerView = findViewById(R.id.rv_graph_prediksi);

        mBtnShowPrediksi.setOnClickListener(this);

        series = new LineGraphSeries();
        mGraphView.addSeries(series);

        mGraphView.getViewport().setScalable(true);
        mGraphView.getViewport().setScrollable(true);

        mGraphView.getViewport().setXAxisBoundsManual(true);
        mGraphView.getViewport().setMinX(1d);
        mGraphView.getViewport().setMaxX(365d);

        GridLabelRenderer gridLabelRenderer =
mGraphView.getGridLabelRenderer();
        gridLabelRenderer.setHorizontalAxisTitle("Hari");
        gridLabelRenderer.setVerticalAxisTitle("Net Laju
(mm/hari)");

```

```

        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(this);
        mRecyclerView.setLayoutManager(mLayoutManager);

    }

    @Override
    public void onClick(View view) {

        String year = (String) mSpinner.getSelectedItem();
        reference = database.getReference("Data Net
Laju").child(year);

        reference.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
snapshot) {
                DataPoint[] dp = new DataPoint[(int)
snapshot.getChildrenCount()];
                int index = 0;

                List<NetLaju> netLajus = new ArrayList<>();

                for (DataSnapshot myDataSnapshot :
snapshot.getChildren()) {
                    String hari = myDataSnapshot.getKey();

                    Object o = myDataSnapshot.getValue();
                    String netLajuStr = String.valueOf(o);
                    Float laju = Float.valueOf(netLajuStr);

                    NetLaju netLaju = new
NetLaju(Integer.valueOf(hari), laju);
                    netLajus.add(netLaju);

                    dp[index] = new DataPoint(netLaju.getHari(),
netLaju.getNetLaju());
                    index++;
                }
                series.resetData(dp);

                GraphPrediksiAdapter adapter = new
GraphPrediksiAdapter(netLajus);
                mRecyclerView.setAdapter(adapter);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error)
{

            }
        });
    }
}

```

- Pendugaan Net Laju