LAMPIRAN

DetailActivity.Java

```
package com.dicoding.firs_aid;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.google.android.material.bottomnavigation.BottomNavigationView;

public class DetailActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

setContentView(R.layout.activity_detail);

      // Ambil data dari Intent
      Intent intent = getIntent();
      String videoUrl = intent.getStringExtra("video");
      String name = intent.getStringExtra("name");
      String detail = intent.getStringExtra("detail");
      String imageUrl = intent.getStringExtra("image");

      // Inisialisasi tampilan
      WebView webView = findViewById(R.id.webViewDetail);
      ImageView imageView = findViewById(R.id.imageViewDetail);
      TextView textViewName = findViewById(R.id.textViewDetailName);
      TextView textViewDesc = findViewById(R.id.textViewDetailDesc);
      BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);

      // Set data ke tampilan
      textViewName.setText(name);
      textViewDesc.setText(detail);

Glide.with(this).load(imageUrl).into(imageView);

      // Konfigurasi WebView untuk memutar video hanya jika videoUrl tidak kosong
      if (videoUrl != null && !videoUrl.isEmpty()) {
          WebSettings webSettings = webView.getSettings();

webSettings.setJavaScriptEnabled(true);

webSettings.setDomStorageEnabled(true);

          webView.setWebChromeClient(new WebChromeClient());
          webView.setWebViewClient(new WebViewClient());

          // Buat embed video hanya jika URL benar
          String videoHtml = "<html><body style='margin:0;padding:0;'><iframe width='100%' height='100%' "
                  + "src='" + videoUrl + "' frameborder='0' allowfullscreen></iframe></body></html>";


webView.loadDataWithBaseURL(null, videoHtml, "text/html", "UTF-8", null);

webView.setVisibility(WebView.VISIBLE); // Tampilkan WebView jika ada video
      } else {

webView.setVisibility(WebView.GONE); // Sembunyikan WebView jika tidak ada video
      }
```

```
        // Bottom Navigation
        bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                Intent newIntent = null;
                int id = item.getItemId();

                if (id == R.id.nav_favorite) {
                    newIntent = new Intent(DetailActivity.this, FavoriteActivity.class);
                } else if (id == R.id.nav_home) {
                    newIntent = new Intent(DetailActivity.this, MainActivity.class);
                } else if (id == R.id.nav_profile) {
                    newIntent = new Intent(DetailActivity.this, ProfileActivity.class);
                }

                if (newIntent != null) {
                    startActivity(newIntent);
                    finish(); // Tutup aktivitas ini agar tidak menumpuk
                    return true;
                }
                return false;
            }
        });
    }
}
```

DetailP3Kactivity.Java

```
package com.dicoding.firs_aid;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.google.android.material.bottomnavigation.BottomNavigationView;

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

import java.io.IOException;

import okhttp3.Call;
```

```
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class DetailP3kActivity extends AppCompatActivity {

    public static final String ITEM_EXTRA = "item_extra";
    private P3k p3k;
    private static final String TAG = "DetailP3kActivity";

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

        setContentView(R.layout.activity_detail_p3k);

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
        bottomNavigationView.setOnNavigationItemSelectedListener(item -> {
            int id = item.getItemId();
            if (id == R.id.nav_favorite) {
                startActivity(new Intent(DetailP3kActivity.this, FavoriteActivity.class));
                return true;
            } else if (id == R.id.nav_home) {
                startActivity(new Intent(DetailP3kActivity.this, MainActivity.class));
                return true;
            } else if (id == R.id.nav_profile) {
                startActivity(new Intent(DetailP3kActivity.this, ProfileActivity.class));
                return true;
            }
            return false;
        });

        ImageView imgP3k = findViewById(R.id.imgP3k);
        TextView tvP3kName = findViewById(R.id.tvP3kName);
        TextView tvP3kDetail = findViewById(R.id.tvP3kDetail);
```

```
        ImageView imgFavorite = findViewById(R.id.imgfavorite);
        WebView webViewVideo = findViewById(R.id.webView);

        p3k = getIntent().getParcelableExtra(ITEM_EXTRA);

        if (p3k != null) {

Glide.with(this).load(p3k.getPhoto()).into(imgP3k);
            tvP3kName.setText(p3k.getName());
            tvP3kDetail.setText(p3k.getDetail());

            String videoUrl = p3k.getVideoUrl();
            if (videoUrl != null && !videoUrl.isEmpty()) {

webViewVideo.setVisibility(View.VISIBLE);
                WebSettings webSettings = webViewVideo.getSettings();

webSettings.setJavaScriptEnabled(true);

webSettings.setDomStorageEnabled(true);

webSettings.setMediaPlaybackRequiresUserGesture(false);

                if (videoUrl.contains("watch?v=")) {
                    videoUrl = videoUrl.replace("watch?v=", "embed/");
                }

                String iframe = "<html><body style='margin:0;padding:0;'><iframe width=\"100%\" height=\"100%\" src=\"" + videoUrl + "\" frameborder=\"0\" allowfullscreen></iframe></body></html>";
                webViewVideo.loadData(iframe, "text/html", "utf-8");

webViewVideo.setWebViewClient(new WebViewClient());
            } else {

webViewVideo.setVisibility(View.GONE);
            }
        }

        imgFavorite.setOnClickListener(v -> {
```

```
        if (p3k != null) {
            addToFavorites(p3k);
        } else {
            Toast.makeText(this, "Error: Data tidak tersedia", Toast.LENGTH_SHORT).show();
        }
    });

    if (getSupportActionBar() != null) {
        getSupportActionBar().setTitle("Detail P3K");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

private void addToFavorites(P3k item) {
    SharedPreferences sharedPref = getSharedPreferences("user_prefs", Context.MODE_PRIVATE);
    String userId = sharedPref.getString("user_id", "default_id");

    OkHttpClient client = new OkHttpClient();
    MediaType JSON = MediaType.parse("application/json; charset=utf-8");

    JSONObject jsonData = new JSONObject();
    try {
        jsonData.put("user_id", userId);
        jsonData.put("name", item.getName());
        jsonData.put("detail", item.getDetail());
        jsonData.put("image", item.getPhoto());
        jsonData.put("video", item.getVideoUrl());
    } catch (JSONException e) {
        e.printStackTrace();
    }

    RequestBody body = RequestBody.create(JSON, jsonData.toString());

    Request request = new Request.Builder()
```

```
.url("https://proyek.kedaiaisyiyah.com/p3k/add_to_favorites.php")
            .post(body)
            .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                runOnUiThread(() -> {
                    Log.e(TAG, "Failed to add to favorites", e);
                    Toast.makeText(DetailP3kActivity.this, "Gagal menambahkan favorit: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                });
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                String responseBody = response.body().string();
                Log.d(TAG, "Server Response: " + responseBody);
                try {
                    JSONObject jsonResponse = new JSONObject(responseBody);
                    boolean success = jsonResponse.getBoolean("success");
                    String message = jsonResponse.getString("message");

                    runOnUiThread(() -> Toast.makeText(DetailP3kActivity.this, message, Toast.LENGTH_SHORT).show());

                    if (!success) {
                        Log.e(TAG, "Error: " + message);
                    }
                } catch (JSONException e) {
                    Log.e(TAG, "JSON Parsing Error", e);
                    runOnUiThread(() -> Toast.makeText(DetailP3kActivity.this, "Error parsing response", Toast.LENGTH_SHORT).show());
                }
            }
        });
    }
```

```
    return true;
  }
}
```

@Override

```
public boolean onSupportNavigateUp() {
    finish();
```

EditProfileActivity.Java

```
package com.dicoding.firs_aid;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.material.bottomnavigation.BottomNavigationView;

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

import java.io.IOException;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class EditProfileActivity extends AppCompatActivity {

  private EditText etUsername, etEmail, etPhone, etPassword;
  private Button btnSave;

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

    BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);

    bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
      @Override
      public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_favorite) {
          // Handle Favorite action
          Intent intent = new Intent(EditProfileActivity.this, FavoriteActivity.class);
          startActivity(intent);
          return true;
        } else if (id == R.id.nav_home) {
          // Handle Beranda action
          Intent intent = new Intent(EditProfileActivity.this, MainActivity.class);
          startActivity(intent);
          return true;
        } else if (id == R.id.nav_profile) {
          // Handle Profile action
          Intent intent = new Intent(EditProfileActivity.this, ProfileActivity.class);
          startActivity(intent);
          return true;
        }
        return false;
      }
    });

    etUsername =
```

```
findViewById(R.id.etUsername);
        etEmail = findViewById(R.id.etEmail);
        etPhone = findViewById(R.id.etPhone);
        etPassword = findViewById(R.id.etPassword);
        btnSave = findViewById(R.id.btnSave);

        fetchUserProfile();

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

    private void fetchUserProfile() {
        SharedPreferences sharedPref = getSharedPreferences("user_prefs", Context.MODE_PRIVATE);
        String userId = sharedPref.getString("user_id", "default_id");

        OkHttpClient client = new OkHttpClient();
        Request request = new Request.Builder()

.url("https://proyek.kedaiaisyiyah.com/p3k/get_user_profile.php?user_id=" + userId)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                runOnUiThread(() -> Toast.makeText(EditProfileActivity.this, "Failed to fetch profile: " + e.getMessage(), Toast.LENGTH_SHORT).show());
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                if (!response.isSuccessful()) {
                    throw new IOException("Unexpected code " + response);
                }

                String responseData = response.body().string();
                runOnUiThread(() -> {
                    try {
                        JSONObject jsonResponse = new JSONObject(responseData);
                        if (jsonResponse.getBoolean("success")) {
                            JSONObject user = jsonResponse.getJSONObject("user");

etUsername.setText(user.getString("username"));

etEmail.setText(user.getString("email"));

etPhone.setText(user.getString("phone"));
                        } else {

Toast.makeText(EditProfileActivity.this, jsonResponse.getString("message"), Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                });
            }
        });
    }

    private void updateProfile() {
        SharedPreferences sharedPref = getSharedPreferences("user_prefs", Context.MODE_PRIVATE);
        String userId = sharedPref.getString("user_id", "default_id");

        String username = etUsername.getText().toString();
        String email = etEmail.getText().toString();
        String phone = etPhone.getText().toString();
        String password = etPassword.getText().toString();

        OkHttpClient client = new OkHttpClient();
        MediaType JSON = MediaType.parse("application/json; charset=utf-8");
        JSONObject jsonBody = new
```

```
JSONObject();
        try {
            jsonBody.put("user_id", userId);
            jsonBody.put("username", username);
            jsonBody.put("email", email);
            jsonBody.put("phone", phone);
            jsonBody.put("password", password);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        RequestBody body = RequestBody.create(JSON, jsonBody.toString());
        Request request = new Request.Builder()

                .url("http://10.10.10.249:8012/p3k/update_user_profile.php")
                .post(body)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                runOnUiThread(() -> Toast.makeText(EditProfileActivity.this, "Failed to update profile: " + e.getMessage(), Toast.LENGTH_SHORT).show());
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                if (!response.isSuccessful()) {
                    throw new IOException("Unexpected code " + response);
                }

                String responseData = response.body().string();
                runOnUiThread(() -> {
                    try {
                        JSONObject jsonResponse = new JSONObject(responseData);
                        if (jsonResponse.getBoolean("success")) {

                            Toast.makeText(EditProfileActivity.this, "Profile updated successfully!", Toast.LENGTH_SHORT).show();
                            finish();
                        } else {

                            Toast.makeText(EditProfileActivity.this, jsonResponse.getString("message"), Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                });
            }
        });
    }
}
```

FavoriteAdapter.Java

```
package com.dicoding.firs_aid;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import java.util.ArrayList;

public class FavoriteAdapter extends BaseAdapter {
    private Context context;
    private ArrayList<P3k> favoriteItems;
    private OnFavoriteDeleteListener deleteListener;

    public FavoriteAdapter(Context context, ArrayList<P3k> favoriteItems, OnFavoriteDeleteListener deleteListener) {
        this.context = context;
        this.favoriteItems = favoriteItems;
        this.deleteListener = deleteListener;
    }
```

```
    @Override
    public int getCount() {
        return favoriteItems.size();
    }

    @Override
    public Object getItem(int position) {
        return favoriteItems.get(position);
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder;
        if (convertView == null) {
            convertView = LayoutInflater.from(context).inflate(R.layout.list_item_favorite, parent, false);
            holder = new ViewHolder();
            holder.imageViewFavorite = convertView.findViewById(R.id.imageViewFavorite);
            holder.textViewNameFavorite = convertView.findViewById(R.id.textViewNameFavorite);
            holder.textViewDetailFavorite = convertView.findViewById(R.id.textViewDetailFavorite);
            holder.buttonRemoveFavorite = convertView.findViewById(R.id.buttonRemoveFavorite);
            holder.buttonDetail = convertView.findViewById(R.id.btnDetail);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }

        P3k item = (P3k) getItem(position);

        holder.textViewNameFavorite.setText(item.getName());

        holder.textViewDetailFavorite.setText(item.getDetail());

        // Load image using Picasso
        Picasso.get().load(item.getPhoto()).into(holder.imageViewFavorite);

        // Aksi tombol hapus
        holder.buttonRemoveFavorite.setOnClickListener(v -> {
            if (deleteListener != null) {
                deleteListener.onDelete(position);
            }
        });

        // Aksi tombol "Selengkapnya"
        holder.buttonDetail.setOnClickListener(v -> {
            Intent intent = new Intent(context, DetailActivity.class);
            intent.putExtra("name", item.getName());
            intent.putExtra("detail", item.getDetail());
            intent.putExtra("image", item.getPhoto());
            intent.putExtra("video", item.getVideoUrl()); // Kirim URL video
            context.startActivity(intent);
        });


        return convertView;
    }

    private static class ViewHolder {
        ImageView imageViewFavorite;
        TextView textViewNameFavorite;
        TextView textViewDetailFavorite;
        Button buttonRemoveFavorite;
        Button buttonDetail;
    }

    public interface OnFavoriteDeleteListener {
        void onDelete(int position);
    }
}
```

ListP3KAdapter.Java

```
package com.dicoding.firs_aid;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import
androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import
com.bumptech.glide.request.RequestOptions
;

import java.util.ArrayList;

public class ListP3kAdapter extends
RecyclerView.Adapter<ListP3kAdapter.List
ViewHolder>{

  private ArrayList<P3k> listP3k;
  private ArrayList<P3k> filteredList; //
Daftar sementara untuk hasil pencarian

  private OnItemClickCallback
onItemClickCallback;

  public ListP3kAdapter(ArrayList<P3k>
list){
    this.listP3k = list;
    this.filteredList = new
ArrayList<>(list); // Inisialisasi filteredList
  }

  public void
setOnItemClickCallback(OnItemClickCallba
ck onItemClickCallback){
    this.onItemClickCallback =
onItemClickCallback;
  }

  @NonNull
  @Override
  public ListViewHolder
onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).infl
ate(R.layout.item_row_p3k, parent, false);
    return new ListViewHolder(view);
  }

  @Override
  public void
onBindViewHolder(@NonNull
ListViewHolder holder, int position) {
    P3k p3k = filteredList.get(position);

Glide.with(holder.itemView.getContext())
        .load(p3k.getPhoto())
        .apply(new
RequestOptions().override(55, 55))
        .into(holder.imgPhoto);

    holder.tvName.setText(p3k.getName());

holder.tvDetail.setText(p3k.getDetail());

    holder.itemView.setOnClickListener(v -
>
onItemClickCallback.onItemClicked(filtered
List.get(holder.getAdapterPosition())));
  }

  @Override
  public int getItemCount() {
    return filteredList.size();
  }

  class ListViewHolder extends
RecyclerView.ViewHolder {
    ImageView imgPhoto;
    TextView tvName;
    TextView tvDetail;

    ListViewHolder(View itemview) {
      super(itemview);
      imgPhoto =
itemview.findViewById(R.id.img_item_p3k)
;
      tvName =
itemview.findViewById(R.id.tv_item_name)
;
      tvDetail =
itemview.findViewById(R.id.tv_detail);
    }
  }

  public interface OnItemClickCallback{
    void onItemClicked(P3k data);
  }

  public void filter(String text) {
    filteredList.clear();
    if (text.isEmpty()) {
      filteredList.addAll(listP3k);
    } else {
```

```
        text = text.toLowerCase();
        for (P3k item : listP3k) {
            if (item.getName().toLowerCase().contains(text) || item.getDetail().toLowerCase().contains(text)) {
                filteredList.add(item);
            }
        }
    }

    notifyDataSetChanged();
  }

  public void setListP3k(ArrayList<P3k> listP3k) {
    this.listP3k = listP3k;
    this.filteredList = new ArrayList<>(listP3k); // Update filtered list
  }
}
```

MainActivity.Java

```
package com.dicoding.firs_aid;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.SearchView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.DividerItemDecoration;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

  private RecyclerView rvP3k;
  private ListP3kAdapter listP3kAdapter;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);

setContentView(R.layout.activity_main);

    BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
      @Override
      public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_favorite) {
          // Handle Favorite action
          Intent intent = new Intent(MainActivity.this, FavoriteActivity.class);
          startActivity(intent);
          return true;
        } else if (id == R.id.nav_home) {
          // Handle Beranda action

Toast.makeText(MainActivity.this, "Anda sekarang Di Menu Beranda", Toast.LENGTH_SHORT).show();
          return true;
        } else if (id == R.id.nav_profile) {
          // Handle Profile action
          Intent intent = new Intent(MainActivity.this, ProfileActivity.class);
          startActivity(intent);
        }
        return false;
      }

    });
```

```
        rvP3k = findViewById(R.id.rv_p3k);
        rvP3k.setHasFixedSize(true);

        // Setup Adapter
        listP3kAdapter = new ListP3kAdapter(new ArrayList<>());
        rvP3k.setLayoutManager(new LinearLayoutManager(this));
        rvP3k.setAdapter(listP3kAdapter);

        // Add Divider
        DividerItemDecoration dividerItemDecoration = new DividerItemDecoration(rvP3k.getContext(), LinearLayoutManager.VERTICAL);
        rvP3k.addItemDecoration(dividerItemDecoration);

        // Set item click listener
        listP3kAdapter.setOnItemClickCallback(p3k -> {
            Intent moveIntent = new Intent(MainActivity.this, DetailP3kActivity.class);
            moveIntent.putExtra(DetailP3kActivity.ITEM_EXTRA, p3k);
            startActivity(moveIntent);
        });

        // Setup SearchView
        SearchView searchView = findViewById(R.id.search_view);
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                return false;
            }

            @Override
            public boolean onQueryTextChange(String newText) {
                listP3kAdapter.filter(newText);
                return false;
            }
        });

        // Load data from API
        loadP3kData();
    }

    private void loadP3kData() {
        P3kData.getListData(new P3kData.OnDataFetchedCallback() {
            @Override
            public void onDataFetched(ArrayList<P3k> data) {
                listP3kAdapter.setListP3k(data);
                listP3kAdapter.notifyDataSetChanged();
            }
        });
    }
}
```

P3KData.java

```
package com.dicoding.firs_aid;

import android.os.AsyncTask;

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

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;

public class P3kData {
    private static final String API_URL = "https://proyek.kedaiaisyiyah.com/p3k/api.php";

    public static void getListData(OnDataFetchedCallback callback) {
        new FetchDataTask(callback).execute(API_URL);
    }

    private static class FetchDataTask extends AsyncTask<String, Void, ArrayList<P3k>> {
        private OnDataFetchedCallback callback;
```

```
FetchDataTask(OnDataFetchedCallback callback) {
        this.callback = callback;
    }

    @Override
    protected ArrayList<P3k> doInBackground(String... params) {
        String apiUrl = params[0];
        ArrayList<P3k> list = new ArrayList<>();
        StringBuilder result = new StringBuilder();
        try {
          URL url = new URL(apiUrl);
          HttpURLConnection connection = (HttpURLConnection) url.openConnection();
          BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
          String line;
          while ((line = reader.readLine()) != null) {
            result.append(line);
          }
          reader.close();

          JSONArray jsonArray = new JSONArray(result.toString());
          for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject obj = jsonArray.getJSONObject(i);
            P3k p3k = new P3k();

p3k.setName(obj.getString("name"));

p3k.setDetail(obj.getString("detail"));
```

ProfileAcitivity.Java

```
package com.dicoding.firs_aid;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import org.json.JSONException;
import org.json.JSONObject;
```

```
            String imageName = obj.getString("image_name");

p3k.setPhoto("https://proyek.kedaiaisyiyah.com/p3k/images/" + imageName); // Full URL to image

            // Mengambil URL video dari database
            if (obj.has("video") && !obj.isNull("video")) {

p3k.setVideoUrl(obj.getString("video"));
            } else {
              p3k.setVideoUrl(""); // Default kosong jika tidak ada video
            }


            list.add(p3k);
          }
        } catch (Exception e) {
          e.printStackTrace();
        }
        return list;
    }

    @Override
    protected void onPostExecute(ArrayList<P3k> result) {
        if (callback != null) {
          callback.onDataFetched(result);
        }
    }
  }

  public interface OnDataFetchedCallback {
    void onDataFetched(ArrayList<P3k> data);
  }
}
```

```
import java.io.IOException;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class ProfileActivity extends AppCompatActivity {

    private TextView tvUsername, tvEmail, tvPhone;
    private Button btnUpdateProfile;

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

        setContentView(R.layout.activity_profile);

        BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);

        bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                int id = item.getItemId();
                if (id == R.id.nav_favorite) {
                    // Handle Favorite action
                    Intent intent = new Intent(ProfileActivity.this, FavoriteActivity.class);
                    startActivity(intent);
                    return true;
                } else if (id == R.id.nav_home) {
                    // Handle Beranda action
                    Intent intent = new Intent(ProfileActivity.this, MainActivity.class);
                    startActivity(intent);
                    return true;
                } else if (id == R.id.nav_profile) {
                    // Handle Profile action

                    Toast.makeText(ProfileActivity.this, "Anda sekarang Di Menu Profile", Toast.LENGTH_SHORT).show();
                    return true;
                }
                return false;
            }

        });

        tvUsername = findViewById(R.id.tvUsername);
        tvEmail = findViewById(R.id.tvEmail);
        tvPhone = findViewById(R.id.tvPhone);
        btnUpdateProfile = findViewById(R.id.btnUpdateProfile);

        fetchUserProfile();


        btnUpdateProfile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(ProfileActivity.this, EditProfileActivity.class);
                startActivity(intent);
            }
        });
    }

    private void fetchUserProfile() {
        SharedPreferences sharedPref = getSharedPreferences("user_prefs", Context.MODE_PRIVATE);
        String userId = sharedPref.getString("user_id", "default_id");

        OkHttpClient client = new OkHttpClient();
        Request request = new Request.Builder()

                .url("https://proyek.kedaiaisyiyah.com/p3k/get_user_profile.php?user_id=" + userId)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                runOnUiThread(() -> Toast.makeText(ProfileActivity.this, "Failed to fetch profile: " + e.getMessage(),
```

```
Toast.LENGTH_SHORT).show());
        }

        @Override
        public void onResponse(Call call, Response response) throws IOException {
            if (!response.isSuccessful()) {
                throw new IOException("Unexpected code " + response);
            }

            String responseData = response.body().string();
            runOnUiThread(() -> {
                try {
                    JSONObject jsonResponse = new JSONObject(responseData);
                    if (jsonResponse.getBoolean("success")) {
                        JSONObject user = jsonResponse.getJSONObject("user");
                        String username = user.getString("username");
                        String email = user.getString("email");
                        String phone = user.getString("phone");


                        tvUsername.setText(username);
                        tvEmail.setText(email);
                        tvPhone.setText(phone);
                    } else {

                        Toast.makeText(ProfileActivity.this, jsonResponse.getString("message"), Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            });
        }
    });
  }
}
```

RegisterActivity.Java

```
package com.dicoding.firs_aid;

import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;

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

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;

public class RegisterActivity extends AppCompatActivity {

  private TextInputEditText etUsername, etPassword, etEmail, etPhone;
  private Button btnRegister;
  private TextView tvLoginPrompt;

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

setContentView(R.layout.register_user);

      etUsername = findViewById(R.id.etUsername);
      etPassword = findViewById(R.id.etPassword);
      etEmail = findViewById(R.id.etEmail);
      etPhone = findViewById(R.id.etPhone);
      btnRegister = findViewById(R.id.btnRegister);
      tvLoginPrompt = findViewById(R.id.tvLoginPrompt);
```

```
        btnRegister.setOnClickListener(new
View.OnClickListener()                    {
            @Override
            public void onClick(View v) {
              new RegisterUser().execute();
            }
        });


tvLoginPrompt.setOnClickListener(new
View.OnClickListener()                    {
            @Override
            public void onClick(View v) {
              Intent intent = new
Intent(RegisterActivity.this,
LoginActivity.class);
              startActivity(intent);
            }
        });
    }

    private class RegisterUser extends
AsyncTask<Void, Void, String>  {

        @Override
        protected String
doInBackground(Void... voids)  {
            String username =
etUsername.getText().toString().trim();
            String password =
etPassword.getText().toString().trim();
            String email =
etEmail.getText().toString().trim();
            String phone =
etPhone.getText().toString().trim();

            try                                {
              URL url = new
URL("https://proyek.kedaiaisyiyah.com/p3k
/register.php");
              HttpURLConnection
httpURLConnection =
(HttpURLConnection)
url.openConnection();

httpURLConnection.setRequestMethod("PO
ST");

httpURLConnection.setDoOutput(true);

              String postData = "username=" +
username + "&password=" + password +
"&email=" + email + "&phone=" + phone;

              OutputStream outputStream =
httpURLConnection.getOutputStream();

outputStream.write(postData.getBytes());
              outputStream.flush();
              outputStream.close();

              BufferedReader bufferedReader =
new BufferedReader(new
InputStreamReader(httpURLConnection.get
InputStream()));
              String result =
bufferedReader.readLine();
              bufferedReader.close();

              return result;

            } catch (Exception e) {
              e.printStackTrace();
              return null;
            }
        }

        @Override
        protected void onPostExecute(String
result)                                    {
            super.onPostExecute(result);

            if (result != null)  {
              try                              {
                JSONObject jsonObject = new
JSONObject(result);
                boolean success =
jsonObject.getBoolean("success");
                if (success)  {

Toast.makeText(RegisterActivity.this,
"Register successful",
Toast.LENGTH_SHORT).show();
                  startActivity(new
Intent(RegisterActivity.this,
LoginActivity.class));
                  finish();
                }  else  {
                  String message =
jsonObject.getString("message");

Toast.makeText(RegisterActivity.this,
message, Toast.LENGTH_SHORT).show();
                }
              } catch (JSONException e) {
                e.printStackTrace();
              }
            }  else  {

Toast.makeText(RegisterActivity.this,
```

```
"Register failed",
Toast.LENGTH_SHORT).show();
        }
```

```
    }
  }
}
```

Login.php

```
<?php
include '../koneksi.php'; // Sesuaikan path ini dengan lokasi file koneksi Anda

session_start(); // Mulai session untuk menyimpan status login

header('Content-Type: application/json');
error_reporting(E_ALL);
ini_set('display_errors', 1);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  // Ambil data dari form
  $username = $_POST['username'];
  $password = $_POST['password'];

   // Cek apakah username dan password sudah diisi
        if (empty($username) || empty($password)) {
      echo json_encode(['status' => 'error', 'message' => 'Username and password are required']);
      exit();
  }

  // Hash password
    $passwordHash = md5($password); // Untuk keamanan lebih baik gunakan password_hash

  // Prepare dan execute query
    $stmt = $conn->prepare("SELECT * FROM admin WHERE username = ? AND password = ?");
     $stmt->bind_param("ss", $username, $passwordHash);

  if ($stmt->execute()) {
    $result = $stmt->get_result();

    // Cek hasil query
    if ($result->num_rows === 1) {
      // Login berhasil
            $_SESSION['username'] = $username;
          header('Location: index.php'); // Redirect ke index.php setelah login berhasil
        exit();
    } else {
      // Login gagal
      header('Location: pages-login.php');
    }
  } else {
      echo json_encode(['status' => 'error', 'message' => 'Query failed']);
  }
} else {
    echo json_encode(['status' => 'error', 'message' => 'Invalid request method']);
}
?>
```

Tambah_p3k.php

```
<?php
// Inisialisasi respon default
$response = [
  'status' => 'error',
  'message' => 'Terjadi kesalahan yang tidak diketahui.'
];

// Menghubungkan ke database
include '../koneksi.php';

// Cek jika form disubmit
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  // Mendapatkan data dari form
      $judul = isset($_POST['judul']) ? trim($_POST['judul']) : '';
   $deskripsi = isset($_POST['deskripsi']) ? trim($_POST['deskripsi']) : '';
  $video_url = isset($_POST['video_url']) ? trim($_POST['video_url']) : '';
  $image_name = '';

  // Validasi input
```

```
    if (empty($judul)) {
        $response['message'] = 'Judul harus diisi!';
        echo json_encode($response);
        exit;
    }

    if (empty($deskripsi)) {
        $response['message'] = 'Deskripsi harus diisi!';
        echo json_encode($response);
        exit;
    }

    // Validasi URL video
    if (!filter_var($video_url, FILTER_VALIDATE_URL)) {
        $response['message'] = 'URL video tidak valid!';
        echo json_encode($response);
        exit;
    }

    // Proses unggahan gambar
    if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
        $image_tmp = $_FILES['image']['tmp_name'];
        $image_name = basename($_FILES['image']['name']);
        $target_dir = "../images/"; // Folder tujuan untuk menyimpan gambar
        $target_file = $target_dir . $image_name;
        $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

        // Validasi tipe file gambar
        $allowed_types = ['jpg', 'jpeg', 'png', 'gif'];
        if (!in_array($imageFileType, $allowed_types)) {
            $response['message'] = 'Hanya file gambar dengan format JPG, JPEG, PNG, atau GIF yang diperbolehkan!';
            echo json_encode($response);
            exit;
        }

        // Validasi ukuran file (max 5MB)
        if ($_FILES['image']['size'] > 5000000) {
            $response['message'] = 'Ukuran gambar terlalu besar, maksimal 5MB!';
            echo json_encode($response);
            exit;
        }

        // Memindahkan file gambar ke folder tujuan
        if (!move_uploaded_file($image_tmp, $target_file)) {
            $response['message'] = 'Terjadi kesalahan saat mengupload gambar!';
            echo json_encode($response);
            exit;
        }
    }

    // Fungsi untuk mengambil ID video dari URL YouTube
    function getYouTubeEmbedUrl($url) {
        // Mencocokkan berbagai format URL YouTube
        preg_match('/(?:youtube\.com\/(?:[^\/\n]+\/\S+\/|(?:v|e(?:mbed)?)\/|\S+\/?[\?&]v=))([\w\-]+)/i', $url, $matches);
        if (!empty($matches[1])) {
            return 'https://www.youtube.com/embed/' . $matches[1]; // Format URL embed
        }

        // Mencocokkan URL short YouTube (misal youtube.com/shorts/)
        preg_match('/(?:youtube\.com\/shorts\/)([\w\-]+)/i', $url, $matches);
        if (!empty($matches[1])) {
            return 'https://www.youtube.com/embed/' . $matches[1]; // Format URL embed
        }

        // Jika tidak ditemukan ID video, return false
        return false;
    }

    // Mendapatkan URL embed
    $video_embed = getYouTubeEmbedUrl($video_url);

    if (!$video_embed) {
        $response['message'] = 'URL video YouTube tidak valid!';
        echo json_encode($response);
        exit;
    }
```

```
    // Menyimpan data ke database menggunakan prepared statement
    $query = $conn->prepare("INSERT INTO p3k (name, detail, video, image_name) VALUES (?, ?, ?, ?)");
    $query->bind_param("ssss", $judul, $deskripsi, $video_embed, $image_name);

    if ($query->execute()) {
        $response['status'] = 'success';
        $response['message'] = "Data berhasil ditambahkan!";
    } else {
        $response['message'] = "Gagal menyimpan data! Error: " . $query->error;
    }

    // Menutup koneksi dan query
    $query->close();
    $conn->close();
}

// Mengembalikan response dalam format JSON
echo json_encode($response);
?>
```

Tambah_user.php

```
<?php
include '../koneksi.php';

$response = array('status' => '', 'message' => '');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get form data
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = md5($_POST['password']); // Encrypt password using MD5
    $phone = $_POST['phone'];

    // Check if username or email already exists
    $checkUserSql = "SELECT * FROM users WHERE username='$username' OR email='$email'";
    $result = $conn->query($checkUserSql);

    if ($result->num_rows > 0) {
        $response['status'] = 'error';
        $response['message'] = "Username or Email already exists.";
    } else {
        // Insert data into database
        $sql = "INSERT INTO users (username, password, email, phone) VALUES ('$username', '$password', '$email', '$phone')";

        if ($conn->query($sql) === TRUE) {
            $response['status'] = 'success';
            $response['message'] = "User successfully added.";
        } else {
            $response['status'] = 'error';
            $response['message'] = "Error: " . $sql . "<br>" . $conn->error;
        }
    }

    $conn->close();
    echo json_encode($response);
}
?>
```

Update_p3k.php

```
<?php
include '../koneksi.php';

// Fungsi untuk mengubah URL YouTube menjadi embed link
function convertToEmbedLink($url) {
    // Cek apakah URL adalah link dari YouTube Shorts atau video biasa (youtube.com)
    if (preg_match('/youtube\.com\/(?:shorts\/|watch\?v=)([a-zA-Z0-9_-]{11})/', $url, $matches)) {
        return "https://www.youtube.com/embed/" . $matches[1];
    }
```

```
    // Cek apakah URL adalah link pendek dari YouTube (youtu.be)
    if (preg_match('/youtu\.be\/([a-zA-Z0-9_-]{11})/', $url, $matches)) {
                                                    return "https://www.youtube.com/embed/" . $matches[1];
    }

    // Jika bukan URL YouTube, kembalikan URL yang asli
    return $url;
}

$id = $_POST['id'];
$name = $_POST['name'];
$detail = $_POST['detail'];
$video = $_POST['video']; // URL video yang dimasukkan oleh user

// Mengonversi link video ke dalam format embed
if (!empty($video)) {
    $video = convertToEmbedLink($video);
} else {
     $video = NULL; // Jika tidak ada URL video yang dimasukkan, set ke NULL
}

// Proses upload gambar jika ada
if (!empty($_FILES['image']['name'])) {
    $image = $_FILES['image']['name'];
    $target = '../images/' . basename($image);
     move_uploadedUp_file($_FILES['image']['tmp_name'], $target);

    // Update data di database dengan URL video dan gambar
     $stmt = $conn->prepare('UPDATE p3k SET name = ?, detail = ?, image_name = ?, video = ? WHERE id = ?');
    $stmt->bind_param('ssssi', $name, $detail, $image, $video, $id);
} else {
    // Jika gambar tidak diubah, hanya update data video dan detail
     $stmt = $conn->prepare('UPDATE p3k SET name = ?, detail = ?, video = ? WHERE id = ?');
    $stmt->bind_param('sssi', $name, $detail, $video, $id);
}

if ($stmt->execute()) {
    echo 'P3K berhasil diperbarui';
} else {
    echo 'Terjadi kesalahan saat memperbarui data: ' . $stmt->error;
}

$stmt->close();
$conn->close();
?>
```

Update_user.php

```
<?php
include '../koneksi.php';

header('Content-Type: application/json');
error_reporting(E_ALL);
ini_set('display_errors', 1);

try {
     if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $id = $_POST['id'];
        $username = $_POST['username'];
        $password = $_POST['password'];
        $email = $_POST['email'];
        $phone = $_POST['phone'];

        // Hash password
        $hashedPassword = md5($password);

            $stmt = $conn->prepare("UPDATE users SET username = ?, password = ?, email = ?, phone = ? WHERE id = ?");
        if (!$stmt) {
                throw new Exception("Prepare failed: " . $conn->error);
        }

        $stmt->bind_param("ssssi", $username, $hashedPassword, $email, $phone, $id);
        $success = $stmt->execute();

        if (!$success) {
                throw new Exception("Execute failed: " . $stmt->error);
        }

         echo 'Users updated successfully';     } else {
```

```
        echo json_encode(['status' => 'Invalid request method']);
    }
} catch (Exception $e) {
        echo json_encode(['status' => 'error', 'message' => $e->getMessage()]);
}
?>
```

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : BAIDURI MUSFIRA SAAD	**Pembimbing I : MARLINA,S.KOM.,M.KOM**
NIM : 220280186	**Pembimbing II : MUGHAFFIR YUNUS, ST.,MT**
Judul Skripsi : APLIKASI PANDUAN P3K BERBASIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Semua Bahasa Asing dicetak miring. - Beri Narasi pada desain Sistem		Konsultasi 1 - JUDUL JANGAN DISINGKAT	
Konsultasi 2 - Urut berdasarkan Abjad. A - Z.		Konsultasi 2 - Penulisan Pertolongan Pertama pada kecelakaan diganti, disesuaikan dengan buku	
Konsultasi 3 2		Konsultasi 3 -	
Konsultasi 4 Acc Ujian Proposal		Konsultasi 4 Acc	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : BAIDURI MUSFIRA SAAD	**Pembimbing I : Marlina, S.Kom.,M.Kom**
NIM : 220280186	**Pembimbing II : Mughaffir Yunus,S.T.,M.T**
Judul Skripsi : APLIKASI PANDUAN PERTOLONGAN PERTAMA PADA KEGAWATDARURATAN (P3K) BERBAIS ANDROID	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Bahasa Asing dicetak Miring		Konsultasi 1	
Konsultasi 2 gunakan spasi 1 pd tabel		Konsultasi 2	
Konsultasi 3 Beri penjelasan pd Desain Sistem yg berjalan		Konsultasi 3	
Konsultasi 4 Perbaiki gambar		Konsultasi 4 Memberikan Tambahan Vidio	
Konsultasi 5 Acc Ujian Hasil		Konsultasi 5 Acc seminar Hasil	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton A4 berwarna Hijau muda dan dicetak timbal balik

>

SERTIFIKAT

Nomor : 413/PRK/LB-TI/II.3.AU/D/2023

Diberikan Kepada :

Baiduri Musfira Saad

NIM.220280186

Telah Menyelesaikan

PRAKTIKUM III

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 16 Januari 2023 - 29 Maret 2023

Parepare, 15 Juni 2023

KETUA PRODI
TEKNIK INOFORMATIKA

Marlina. S.kom., M.kom
NBM. 1162 680

KEPALA
LABORATORIUM

Ir. Untung Suwardoyo, S.Kom., MT., IPP.
NBM. 1288 973

Nilai Praktikum laboratorium Teknik Informatika
Fakultas Teknik Universitas Muhammadiyah Parepare Tahun 2023

MATERI	NILAI ANGKA
PEMROGRAMAN WEB 2	77,5
PEMROGRAMAN JAVA	67,5
PEMROGRAMAN DELPHI 10	96,5
JARINGAN KOMPUTER 2	40
RATA-RATA	70,38
NILAI HURUF	B



SERTIFIKAT

Nomor :201/PRK/LB-TI/II.3.AU/D/2022

diberikan kepada :

BAIDURI MUSFIRA SAAD

NIM.220280186

telah menyelesaikan

PRAKTIKUM IV

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 14 September 2022 - 25 Desember 2022

Parepare, 02 Januari 2023

KETUA PRODI TEKNIK INFORMATIKA

MARLINA. S.Kom., M.Kom

KEPALA LABORATORIUM

Ir. UNTUNG SUWARDOYO, S.Kom., MT.,IPP.

Nilai Praktikum laboratorium Teknik Informatika
Fakultas Teknik Universitas Muhammadiyah Parepare Tahun 2022

MATERI	NILAI ANGKA
DESAIN 3D	80
UI/UX DESIGN	90
PEMROGRAMAN PYTHON	80
MATEMATIKA INFORMATIKA	80
RATA-RATA	82,5
NILAI HURUF	A



SERTIFIKAT

Nomor : 843/PRK/LB-TI/II.3.AU/D/2023

Diberikan Kepada :

Baiduri Musfira Saad

NIM.220280186

Telah Menyelesaikan

PRAKTIKUM V

Laboratorium Teknik Informatika
Program Studi Teknik Informatika Fakultas Teknik
Universitas Muhammadiyah Parepare

Pada tanggal 18 September - 16 Desember 2023

Parepare, 03 Januari 2024

KETUA PRODI
TEKNIK INOFORMATIKA

Marlina. S.kom., M.kom
NBM. 1162 680

KEPALA
LABORATORIUM

Ir. Untung Suwardoyo, S.Kom., MT., IPP.
NBM. 1288 973