DAFTAR PUSTAKA

Adisetya, E., Krisdiarto, A. W., & Partha, I. B. B. (2022). Pengaruh Kondisi Penyadapan Terhadap Kualitas Nira Kelapa (*Cocos Nucifera*). *Prosiding Seminar Nasional Instiper*, *1*(1), 271–278. https://doi.org/10.55180/pro.v1i1.263

Agrarian, R. P., Suprayogi, A., & Yuwono, B. D. (2015). Pembuatan Aplikasi Mobile Gis Berbasis Android Untuk Informasi Pariwisata Di Kabupaten Gunungkidul. *Jurnal Geodesi Undip*, *April*, 241–247.

Aini, A. (2007). Sistem Informasi Pengertian Dan Aplikasinya. *Angewandte Chemie International Edition, 6(11), 951–952.*, 5–24.

Ariyan, F. R., Rokhmawati, R. I., & Brata, K. C. (2019). Pengembangan Antarmuka Website E-Learning untuk Meningkatkan Minat Belajar Pemrograman Dasar Dalam Bahasa Pemrograman Java bagi Mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(10), 9920–9929. http://j-ptiik.ub.ac.id

Ariyanti, M., Suherman, C., Maxiselly, Y., & Rosniawaty, S. (2018). Pertumbuhan Tanaman Kelapa (*Cocos nucifera L.*) Dengan Pemberian Air Kelapa. *Jurnal Hutan Pulau-Pulau Kecil*, *2*(2), 201–212. https://doi.org/10.30598/jhppk.2018.2.2.201

Erawati, W. (2019). Perancangan Sistem Informasi Penjualan Dengan Pendekatan Metode Waterfall. *Jurnal Media Informatika Budidarma*, *3*(1), 1. https://doi.org/10.30865/mib.v3i1.987

Fauziyah, S., & Sugiarti, Y. (2022). Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web. *Jurnal Ilmiah Ilmu Komputer*, *8*(2), 87–93. https://doi.org/10.35329/jiik.v8i2.229

Firly, F., Dewi, I. P., Mursyida, L., & Samala, A. D. (2021). Dasar-dasar Android Studio Dan Membuat Aplikasi Mobile Sederhana. In *Angewandte Chemie International Edition, 6(11), 951–952.* (Vol. 10).

Irfana, W. R., Nugraha, A. L., & Awaluddin, M. (2019). Pembuatan Aplikasi Peta Rute Bus Rapid Transit (BRT) Kota Semarang Berbasis Mobile Gis Menggunakan Smartphone Android. *Jurnal Geodesi Undip*, *8*(1), 328–337.

Irsa, R., Budiarni, R., & Budiman, A. (2020). Pemetaan tempat pembuangan sampah di kota payakumbuh menggunakan mobile gis. *Jurnal SIMTIKA*, *3*(2), 13–20. https://undhari.ac.id/jurnal/index.php/simtika/article/view/77

Kraugusteeliana, K., Nasution, H. A., Triwahyono, B., Ikhwani, M., Ardian, Z., & Bintoro, A. (2023). Aplikasi Pemilihan Lapangan Futsal Menggunakan Mobile-GIS dan GPS Dengan Metode Algoritma Dijkstra. *Jurnal Informasi Dan Teknologi*, *5*(4), 59–66. https://doi.org/10.60083/jidt.v5i4.417

Minan, K. (2021). Analisis Pendekatan Metode TAM Pada Penggunaan Aplikasi E-Commerce. *Ekonomi, Keuangan, Investasi Dan Syariah (EKUITAS)*, *3*(2), 181–187. https://doi.org/10.47065/ekuitas.v3i2.1118

Mubarak, A. (2019). Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek. *JIKO (Jurnal Informatika Dan Komputer)*, *2*(1), 19–25. https://doi.org/10.33387/jiko.v2i1.1052

Naufal Ramadhan, A., Apriyanti, D., & Putro, R. W. (2023). Pemanfaatan Aplikasi Mobile GIS Menggunakan Plugin Mergin Maps Untuk Mendukung Kegiatan Survey Aset Jaringan Listrik Serta Pembuatan Peta Aset Jaringan Listrik (Studi Kasus: Kelurahan Cabean, Kecamatan Semarang Barat, Kota Semarang). *Jurnal Ilmiah Geomatika*, *3*(2), 43–58.

Perrina, M. G. (2021). Literature Review Sistem Informasi Geografis (SIG). *Journal of Information Technology and Computer Science*, *10*(10), 1–4. https://www.researchgate.net/publication/354704876

Regita Pramesti, N. (2023). Peranan Hasil Perkebunan Kelapa Terhadap Pendapatan Keluarga Perspektif Ekonomi Syariah (Studi Di Desa Sibedi Kecamatan Marawola Kabupaten Sigi).

Wowor, A. (2013). Satelit Sensor Stasiun Pengolah Analisis Data Ranah SIG Pemakai. *Jurnal Teknik Informatika*, *2*(1). https://ejournal.unsrat.ac.id/index.php/informatika/article/view/1986

Yanti, T. R., & Sularno. (2024). Perancangan Aplikasi Mobile GIS Berbasis Android Untuk Informasi Wisata Di Kabupaten Solok. *Journal of Informatics and Busisnes*, *01*(04), 307–312.

LAMPIRAN

A. Codingan

1. *Login*

```
package com.davidzebua.iman.ui.auth;

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

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.davidzebua.iman.DbContract;
import com.davidzebua.iman.MainActivity;
import com.davidzebua.iman.R;
import com.davidzebua.iman.VolleyConnection;
```

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

import java.util.HashMap;
import java.util.Map;

public class LoginActivity extends AppCompatActivity {

    EditText email, password;
    Button login;
    TextView register;
    ProgressDialog progressDialog;

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

        email = findViewById(R.id.edt_emailLogin);
        password = findViewById(R.id.edt_passwordLogin);
        login = findViewById(R.id.btn_loginL);
        register = findViewById(R.id.txt_registerL);
        progressDialog = new ProgressDialog(LoginActivity.this);

        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent registerIntent = new Intent(LoginActivity.this,
RegisterActivity.class);
                startActivity(registerIntent);
```

```
        }
    });

    login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String semail = email.getText().toString().trim();
            String spassword = password.getText().toString().trim();
            if (!semail.isEmpty() && !spassword.isEmpty()) {
                checkLogin(semail, spassword);
            } else {
                Toast.makeText(LoginActivity.this, "Email and Password cannot be
empty", Toast.LENGTH_SHORT).show();
            }
        }
    });
  }

  public void checkLogin(final String email, final String password) {
    if (checkNetworkConnection()) {
        progressDialog.setMessage("Logging in...");
        progressDialog.show();

        StringRequest stringRequest = new StringRequest(Request.Method.POST,
DbContract.SERVER_LOGIN_URL,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        progressDialog.dismiss();
                        try {
                            Log.d("LoginResponse", response);
                            JSONObject jsonObject = new JSONObject(response);
```

```
                    JSONArray serverResponseArray =
jsonObject.getJSONArray("server_response");
                    JSONObject serverResponse =
serverResponseArray.getJSONObject(0);

                    String status = serverResponse.getString("status");

                    if (status.equals("OK")) {
                        JSONObject userObject =
serverResponse.getJSONObject("user");

                        String userName = userObject.getString("nama");
                        String userEmail = userObject.getString("email");

                        SharedPreferences sharedPreferences =
getSharedPreferences("user_prefs", MODE_PRIVATE);
                        SharedPreferences.Editor editor = sharedPreferences.edit();
                        editor.putString("email", userEmail);
                        editor.putString("nama", userName);
                        editor.apply();

                        Toast.makeText(LoginActivity.this, "Login Successfully",
Toast.LENGTH_SHORT).show();
                        Intent dashboard = new Intent(LoginActivity.this,
MainActivity.class);
                        startActivity(dashboard);
                        finish();
                    } else {
                        Toast.makeText(LoginActivity.this, "Invalid Login",
Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
```

```
                    e.printStackTrace();
                    Toast.makeText(LoginActivity.this, "JSON Error: " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            progressDialog.dismiss();
            Toast.makeText(LoginActivity.this, "Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }) {
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("email", email);
            params.put("password", password);
            return params;
        }
    };


VolleyConnection.getInstance(LoginActivity.this).addToRequestQueue(stringReq
uest);
    } else {
        Toast.makeText(LoginActivity.this, "Tidak Ada Koneksi Internet",
Toast.LENGTH_SHORT).show();
    }
  }
```

```
    public boolean checkNetworkConnection() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
this.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
        return (networkInfo != null && networkInfo.isConnected());
    }
}
```

2. *Register*

```
package com.davidzebua.iman.ui.auth;

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

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
```

```
import com.davidzebua.iman.DbContract;
import com.davidzebua.iman.R;
import com.davidzebua.iman.VolleyConnection;

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

import java.util.HashMap;
import java.util.Map;

public class RegisterActivity extends AppCompatActivity {

    EditText nama, email, password, confPasswrd;
    Button register;
    TextView login;
    ProgressDialog progressDialog;

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

        nama = findViewById(R.id.edt_name);
        email = findViewById(R.id.edt_email);
        password = findViewById(R.id.edt_password);
        confPasswrd = findViewById(R.id.edt_confPassword);
        register = findViewById(R.id.btn_registerR);
        login = findViewById(R.id.txt_loginR);
        progressDialog = new ProgressDialog(RegisterActivity.this);

        login.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {
                Intent loginIntent = new Intent(RegisterActivity.this,
LoginActivity.class);
                startActivity(loginIntent);
            }
        });

        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String snama = nama.getText().toString();
                String semail = email.getText().toString();
                String spassword = password.getText().toString();
                String sConfPasswd = confPasswrd.getText().toString();

                if (spassword.equals(sConfPasswd) && !spassword.equals("")) {
                    Register(snama, semail, spassword);
                } else {
                    Toast.makeText(getApplicationContext(), "Invalid Credentials",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    public void Register(final String nama, final String email, final String
password) {
        if (checkNetworkConnection()) {
            progressDialog.show();
            StringRequest stringRequest = new StringRequest(Request.Method.POST,
DbContract.SERVER_REGISTER_URL,
                    new Response.Listener<String>() {
```

```
                @Override
                public void onResponse(String response) {
                    progressDialog.dismiss();
                    try {
                        JSONObject jsonObject = new JSONObject(response);
                        String resp = jsonObject.getString("server_response");
                        if (resp.equals("[{\"status\":\"OK\"}]")){
                            Toast.makeText(getApplicationContext(), "Registration
Successful", Toast.LENGTH_SHORT).show();
                            Intent loginIntent = new Intent(RegisterActivity.this,
LoginActivity.class);
                            startActivity(loginIntent);
                            finish();
                        } else {
                            Toast.makeText(getApplicationContext(), "Registration
Successful", Toast.LENGTH_SHORT).show();
                        }
                    } catch (JSONException e){
                        e.printStackTrace();
                    }
                }
            }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                progressDialog.dismiss();
                Toast.makeText(getApplicationContext(), "Error: " +
error.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }) {
            @Nullable
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
```

```
                Map<String, String> params = new HashMap<>();
                params.put("nama", nama);
                params.put("email", email);
                params.put("password", password);
                return params;
            };
        };


VolleyConnection.getInstance(RegisterActivity.this).addToRequestQueue(stringR
equest);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                progressDialog.cancel();
            }
        }, 2000);
    } else {
        Toast.makeText(getApplicationContext(), "Tidak Ada Koneksi Internet",
Toast.LENGTH_SHORT).show();
    }
  }

  public boolean checkNetworkConnection() {
    ConnectivityManager connectivityManager = (ConnectivityManager)
this.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
    return (networkInfo != null && networkInfo.isConnected());
  }
}
```

3. *Home*

```
package com.davidzebua.iman.ui.home;

import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import com.davidzebua.iman.DbContract;
import com.davidzebua.iman.R;

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

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class HomeFragment extends Fragment {

    private TextView textViewLocationCount;
    private TextView textViewBlockCount;

    public View onCreateView(@NonNull LayoutInflater inflater,
```

```
                        ViewGroup container, Bundle savedInstanceState) {

        View root = inflater.inflate(R.layout.fragment_home, container, false);

        textViewLocationCount =
root.findViewById(R.id.textViewLocationCount);
        textViewBlockCount = root.findViewById(R.id.textViewBlockCount);

        // Memanggil AsyncTask untuk mengambil data dari kedua URL secara
berurutan
        new
FetchDataAsyncTask().execute(DbContract.SERVER_COUNT_LOKASI,
DbContract.SERVER_COUNT_BLOK);

        return root;
    }

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

        private String locationCount;
        private String blockCount;

        @Override
        protected Void doInBackground(String... urls) {
            // Untuk setiap URL, lakukan pengambilan data
            for (int i = 0; i < urls.length; i++) {
                HttpURLConnection urlConnection = null;
                BufferedReader reader = null;
                try {
                    URL url = new URL(urls[i]);
                    urlConnection = (HttpURLConnection) url.openConnection();
```

```
                urlConnection.setRequestMethod("GET");
                urlConnection.connect();

                InputStream inputStream = urlConnection.getInputStream();
                StringBuilder buffer = new StringBuilder();
                if (inputStream == null) {
                    return null;
                }
                reader = new BufferedReader(new
InputStreamReader(inputStream));
                String line;
                while ((line = reader.readLine()) != null) {
                    buffer.append(line).append("\n");
                }
                if (buffer.length() == 0) {
                    return null;
                }
                String jsonStr = buffer.toString();

                JSONObject jsonObject = new JSONObject(jsonStr);
                String count = jsonObject.getString("count");

                // Memilih lokasi atau blok berdasarkan indeks i
                if (i == 0) {
                    locationCount = count;
                } else if (i == 1) {
                    blockCount = count;
                }

            } catch (IOException | JSONException e) {
                e.printStackTrace();
            } finally {
```

```
                if (urlConnection != null) {
                    urlConnection.disconnect();
                }
                if (reader != null) {
                    try {
                        reader.close();
                    } catch (final IOException e) {
                        e.printStackTrace();
                    }
                }
            }
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
        // Set text untuk textViewLocationCount dan textViewBlockCount
        textViewLocationCount.setText(locationCount);
        textViewBlockCount.setText(blockCount);
    }
  }
}
```

4. ***Main activity***

```
package com.davidzebua.iman;

import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
```

```
import android.view.MenuItem;
import android.view.Menu;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import com.davidzebua.iman.ui.auth.LoginActivity;
import com.google.android.material.navigation.NavigationView;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;
import com.davidzebua.iman.databinding.ActivityMainBinding;
public class MainActivity extends AppCompatActivity {
   private AppBarConfiguration mAppBarConfiguration;
   private ActivityMainBinding binding;
   ProgressDialog progressDialog;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

      binding = ActivityMainBinding.inflate(getLayoutInflater());
      setContentView(binding.getRoot());

      setSupportActionBar(binding.appBarMain.toolbar);

      DrawerLayout drawer = binding.drawerLayout;
      NavigationView navigationView = binding.navView;
      mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_home, R.id.nav_gallery, R.id.nav_profile)
```

```
            .setOpenableLayout(drawer)
            .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
        NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
        progressDialog = new ProgressDialog(MainActivity.this);
        progressDialog.setMessage("Logging out...");
        progressDialog.setCancelable(false);
        SharedPreferences sharedPreferences = getSharedPreferences("user_prefs",
MODE_PRIVATE);
        String name = sharedPreferences.getString("nama", "No Name");
        String email = sharedPreferences.getString("email", "No Email");
        View headerView = navigationView.getHeaderView(0);
        TextView navUserName = headerView.findViewById(R.id.nav_user_name);
        TextView navUserEmail = headerView.findViewById(R.id.nav_user_email);
        navUserName.setText(name);
        navUserEmail.setText(email);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.sign_out) {
            progressDialog.show();
            // Handle sign out
```

```
            new Handler().postDelayed(new Runnable() {
                @Override
                public void run() {
                    progressDialog.dismiss();

                    // Clear shared preferences to remove login status
                    SharedPreferences sharedPreferences =
getSharedPreferences("user_prefs", MODE_PRIVATE);
                    SharedPreferences.Editor editor = sharedPreferences.edit();
                    editor.clear();  // Hapus semua data yang disimpan di
SharedPreferences
                    editor.apply();

                    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                    startActivity(intent);
                    Toast.makeText(MainActivity.this, "Logout Successfully",
Toast.LENGTH_SHORT).show();
                    finish();
                }
            }, 2000);
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
```

```
            || super.onSupportNavigateUp();
    }
}
```

5. Volley connection

```
package com.davidzebua.iman;

import android.content.Context;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.Volley;

public class VolleyConnection {
    private static VolleyConnection vInstance;
    private RequestQueue requestQueue;
    private static Context vCtx;

    private VolleyConnection (Context context){
        vCtx = context;
        requestQueue = getRequestQueue();
    }

    private RequestQueue getRequestQueue(){
        if (requestQueue == null){
            requestQueue = Volley.newRequestQueue(vCtx.getApplicationContext());
        }
        return requestQueue;
    }

    public static synchronized VolleyConnection getInstance(Context context){
        if (vInstance == null){
```

```
            vInstance = new VolleyConnection(context);
        }
        return vInstance;
    }

    public<T> void addToRequestQueue(Request<T> request){
        getRequestQueue().add(request);
    }
}
```

6. Db contract

```
package com.davidzebua.iman;

public class DbContract {
    public static final String SERVER_LOGIN_URL =
"https://api.mappingdavid.site/checklogin.php";
    public static final String SERVER_REGISTER_URL =
"https://api.mappingdavid.site/register.php";

    public static final String
SERVER_COUNT_BLOK="https://api.mappingdavid.site/countBlok.php";
    public static final String
SERVER_COUNT_LOKASI="https://api.mappingdavid.site/countLokas.php";
    public static final String
SERVER_FETCH_ADDRESS="https://api.mappingdavid.site/fetch_address.php"
;
    public static final String
SERVER_FETCH_LOCATION="https://api.mappingdavid.site/fetch_location.ph
p";
    public static final String
SERVER_FETCH_BLOK="https://api.mappingdavid.site/fetch_blok.php";
}
```

B. Data Tabulasi Kuesioner

Data Tabulasi Kuesioner

Prsepsi Kegunaan (PU) (X1)					Total	Persepsi Kemudahan Penggunaan (PEOU) (X2)					Total	Intensi Penggunaan (IU) (X3)				Total	Pengguna Nyata		Total
X1.1	X1.2	X1.3	X1.4	X1.5		X2.1	X2.2	X2.3	X2.4	X2.5		X3.1	X3.2	X3.3	X3.4		X4.1	X4.2	
4	3	4	3	3	17	4	4	4	4	4	20	4	4	4	4	16	4	3	7
4	4	4	3	4	19	3	4	4	4	3	18	4	3	3	4	14	3	4	7
4	3	3	3	4	17	3	3	3	3	4	16	4	3	3	3	13	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	3	3	3	3	15	2	2	2	2	2	10	3	3	3	3	12	3	3	6
4	4	4	4	4	20	4	4	4	4	4	20	4	4	4	4	16	4	4	8
4	4	4	4	4	20	4	3	3	4	4	18	4	4	4	4	16	3	4	7
3	3	3	3	3	15	3	3	3	3	3	15	3	4	4	3	14	3	3	6
3	4	3	4	3	17	4	4	4	4	4	20	4	4	4	4	16	4	4	8
3	3	3	3	3	15	3	4	4	4	3	18	4	3	4	3	14	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	4	3	3	4	17	4	3	4	3	4	18	4	3	3	4	14	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	2	2	1	2	7	3	3	6
3	3	4	3	3	16	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
4	4	3	3	4	18	3	3	3	3	3	15	3	3	4	4	14	3	4	7
4	3	3	4	4	18	3	4	3	4	4	18	3	3	3	3	12	3	3	6
3	3	3	2	3	14	3	2	3	3	3	14	3	3	3	3	12	3	3	6
3	3	4	3	3	16	3	4	4	3	4	18	3	3	3	4	13	4	4	8
4	3	3	3	4	17	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	4	3	3	4	17	3	3	3	3	3	15	3	3	3	3	12	3	3	6
4	4	4	4	4	20	4	4	4	3	4	19	4	3	4	3	14	3	4	7
3	4	4	4	3	18	4	3	3	4	3	17	4	3	4	3	14	4	4	8
3	4	3	4	4	18	3	2	2	3	3	13	4	4	3	4	15	3	4	7
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	4	13	4	4	8
4	3	4	3	3	17	4	4	3	4	3	18	3	4	4	3	14	4	4	8
4	4	3	4	3	18	4	4	3	4	4	19	3	4	4	4	15	4	3	7
4	4	4	4	4	20	4	4	4	4	4	20	4	4	4	4	16	4	4	8
4	4	4	4	4	20	3	3	3	3	3	15	3	3	3	3	12	3	3	6
4	4	4	3	3	18	4	4	4	4	4	20	4	4	4	4	16	4	4	8
4	3	4	4	3	18	4	4	3	4	4	19	4	3	4	4	15	3	4	7
4	4	4	4	4	20	4	3	4	4	4	19	4	4	4	4	16	3	4	7
4	4	3	4	4	19	4	4	3	4	4	19	4	3	4	4	15	4	4	8
4	4	4	4	4	20	4	4	4	4	4	20	4	4	4	4	16	4	4	8
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
3	3	3	3	3	15	3	3	3	3	3	15	3	3	3	3	12	3	3	6
4	4	4	4	4	20	4	4	4	4	4	20	4	4	4	4	16	4	4	8
3	4	4	4	4	19	3	3	2	3	3	14	3	3	3	3	12	3	3	6
3	3	3	3	4	16	3	2	3	3	3	14	3	2	3	3	11	4	4	8

Sumber : Data Kuesioner excel

Keterangan :

X1 = Item Pertanyaan tentang Prsepsi Kegunaan (PU)

X2 = Item Pertanyaan tentang Persepsi Kemudahan Penggunaan (PEOU)

X3 = Item Pertanyaan tentang Intensi Penggunaan (IU)

X4 = Item Pertanyaan tentang Penguna Nyata (PU)

C. Hasil Olah Data di SPSS

1. Uji validitas

a. Persepsi Kegunaan

Correlations

		x1	x2	x3	x4	x5
x1	Pearson Correlation	1	.398*	.499**	.458**	.499**
	Sig. (2-tailed)		.011	.001	.003	.001
	N	40	40	40	40	40
x2	Pearson Correlation	.398*	1	.397*	.645**	.599**
	Sig. (2-tailed)	.011		.011	.000	.000
	N	40	40	40	40	40
x3	Pearson Correlation	.499**	.397*	1	.439**	.195
	Sig. (2-tailed)	.001	.011		.005	.228
	N	40	40	40	40	40
x4	Pearson Correlation	.458**	.645**	.439**	1	.458**
	Sig. (2-tailed)	.003	.000	.005		.003
	N	40	40	40	40	40
x5	Pearson Correlation	.499**	.599**	.195	.458**	1
	Sig. (2-tailed)	.001	.000	.228	.003	
	N	40	40	40	40	40

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

b. Persepsi kemudahan penggunaan

Correlations

		x1	x2	x3	x4	x5
x1	Pearson Correlation	1	.622**	.585**	.783**	.783**
	Sig. (2-tailed)		.000	.000	.000	.000
	N	40	40	40	40	40
x2	Pearson Correlation	.622**	1	.672**	.740**	.667**
	Sig. (2-tailed)	.000		.000	.000	.000
	N	40	40	40	40	40
x3	Pearson Correlation	.585**	.672**	1	.559**	.639**
	Sig. (2-tailed)	.000	.000		.000	.000
	N	40	40	40	40	40
x4	Pearson Correlation	.783**	.740**	.559**	1	.655**
	Sig. (2-tailed)	.000	.000	.000		.000
	N	40	40	40	40	40
x5	Pearson Correlation	.783**	.667**	.639**	.655**	1
	Sig. (2-tailed)	.000	.000	.000	.000	
	N	40	40	40	40	40

**. Correlation is significant at the 0.01 level (2-tailed).

c. Intensi penggunaan

Correlations

		x1	x2	x3	x4
x1	Pearson Correlation	1	.533**	.679**	.660**
	Sig. (2-tailed)		.000	.000	.000
	N	40	40	40	40
x2	Pearson Correlation	.533**	1	.702**	.617**
	Sig. (2-tailed)	.000		.000	.000
	N	40	40	40	40
x3	Pearson Correlation	.679**	.702**	1	.605**
	Sig. (2-tailed)	.000	.000		.000
	N	40	40	40	40
x4	Pearson Correlation	.660**	.617**	.605**	1
	Sig. (2-tailed)	.000	.000	.000	
	N	40	40	40	40

**. Correlation is significant at the 0.01 level (2-tailed).

d. Pengguna nyata

Correlations

		x1	x2
x1	Pearson Correlation	1	.562**
	Sig. (2-tailed)		.000
	N	40	40
x2	Pearson Correlation	.562**	1
	Sig. (2-tailed)	.000	
	N	40	40

**. Correlation is significant at the 0.01 level (2-tailed).

2. Uji realibilitas

a. Persepsi keguaan

Reliability Statistics

Cronbach's Alpha	N of Items
.810	5

b. Persepsi kemudahan penggunaan

Reliability Statistics

Cronbach's Alpha	N of Items
.908	5

c. Intensi penggunaan

Reliability Statistics

Cronbach's Alpha	N of Items
.873	4

d. Pengguna nyata

Reliability Statistics

Cronbach's Alpha	N of Items
.719	2

3. Analisis deskriptif

a. Persepsi keguaan

Descriptive Statistics

	N	Mean	Std. Deviation
x1	40	3.4750	.50574
x2	40	3.4750	.50574
x3	40	3.4250	.50064
x4	40	3.3750	.54006
x5	40	3.4750	.50574
Valid N (listwise)	40		

b. Persepsi kemudahan penggunaan

Descriptive Statistics

	N	Mean	Std. Deviation
x1	40	3.3750	.54006
x2	40	3.3000	.64847
x3	40	3.2500	.58835
x4	40	3.4000	.54538
x5	40	3.4000	.54538
Valid N (listwise)	40		

c. Intensi penggunaan

Descriptive Statistics

	N	Mean	Std. Deviation
x1	40	3.4250	.54948
x2	40	3.2750	.55412
x3	40	3.4000	.63246
x4	40	3.4250	.54948
Valid N (listwise)	40		

d. Pengguna nyata

Descriptive Statistics

	N	Mean	Std. Deviation
x1	40	3.3500	.48305
x2	40	3.4750	.50574
Valid N (listwise)	40		