

BAB V. IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

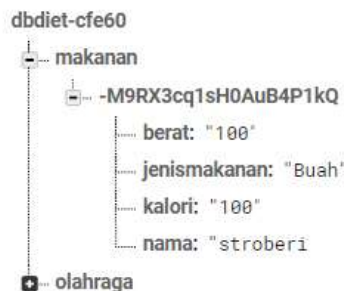
Setelah melalui tahap perancangan, pada bab ini akan dibahas penerapan dan implementasi dari perancangan. Tahapan implementasi adalah tahapan desain yang sudah dibuat merubah menjadi pengembangan sistem pendukung keputusan penentuan olahraga dan perhitungan kalori menggunakan metode *fuzzy tsukamoto*.

1.1.1. Implementasi Data

Implementasi data dilakukan sesuai dengan perancangan yang telah dilakukan pada bab sebelumnya. Basis data yang dibuat menggunakan media yaitu *Firestore*. Implementasi basis data adalah sebagai berikut :

a. Tabel Makanan

Tabel makanan digunakan untuk menyimpan data makanan yang berisi tentang nama makanan, jenis makanan, kalori makanan, dan berat makanan tersebut.

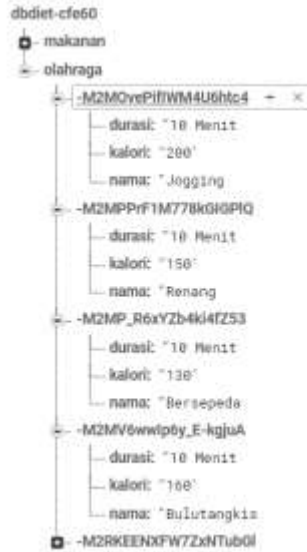


Gambar 5. 1 Data Makanan

Gambar 5.1 adalah gambar yang menunjukkan tentang tabel makanan yang berisi nama makanan, jenis makanan, kalori makanan, dan berat makanan.

b. Tabel Olahraga

Tabel olahraga digunakan untuk menyimpan data olahraga yang berisi tentang nama olahraga, kalori yang terbakar saat berolahraga, dan durasi olahraga tersebut.



Gambar 5. 2 Data Olahraga

Gambar 5.2 adalah gambar yang menunjukkan tentang tabel olahraga yang berisi nama olahraga, durasi olahraga, dan kalori yang terbakar saat olahraga.

1.1.2. Implementasi Tampilan Sistem

Pada tahap ini merupakan tahap dimana setelah selesai melakukan desain program akan ditampilkan kedalam kode program dengan menggunakan *platform mobile*

a. *Splash Screen*



Gambar 5. 3 Splash Screen

b. Tampilan *Home*



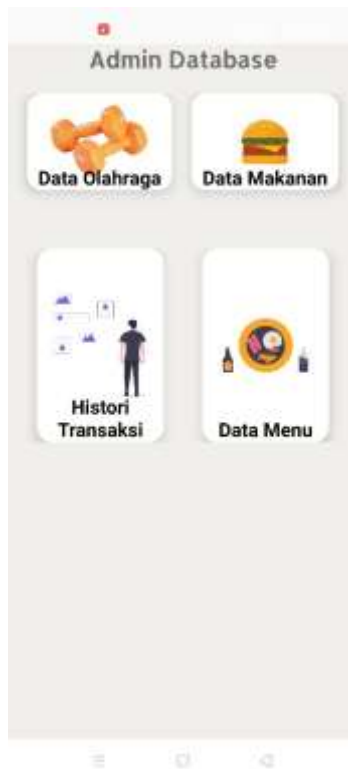
Gambar 5. 4 Gambar Tampilan Home

c. Tampilan *Login*



Gambar 5. 5 Tampilan Login Admin

d. Tampilan *Admin*



Gambar 5. 6 Tampilan Home Admin

e. Tampilan *List Olahraga*



Gambar 5. 7 List Olahraga

f. Tampilan *Input* Olahraga



Input Data Olahraga

Nama Olahraga

Nama Olahraga

Durasi

Durasi

Kalori Terbakar

Kalori Terbakar

SUBMIT

KEMBALI

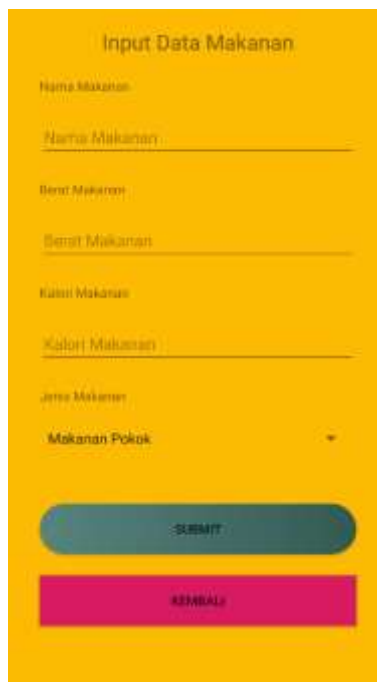
Gambar 5. 8 Input Olahraga

g. Tampilan *List* Makanan



Gambar 5. 9 List Makanan

h. Tampilan *Input* Makanan



Gambar 5. 10 Input Makanan

i. Tampilan List Menu



Gambar 5. 11 List Menu

j. Tampilan Input Menu

Input Menu

Makanan Pokok
Nasi

Lauk
Ayam Goreng

Sayur
Terong

Buah
Apel

Minuman
Susu Sapi

Submit

Gambar 5. 12 Input Menu

k. Tampilan Input Data

Input Data User

Jenis Kelamin
 Laki - Laki
 Perempuan

Umur

Berat Badan

Tinggi Badan

---Petunjuk Pengisian Aktivitas---

Aktivitas
 Aktivitas Ringan

SUBMIT

Gambar 5. 13 Input Data

1. Tampilan Hasil Data



Gambar 5. 14 Hasil Data

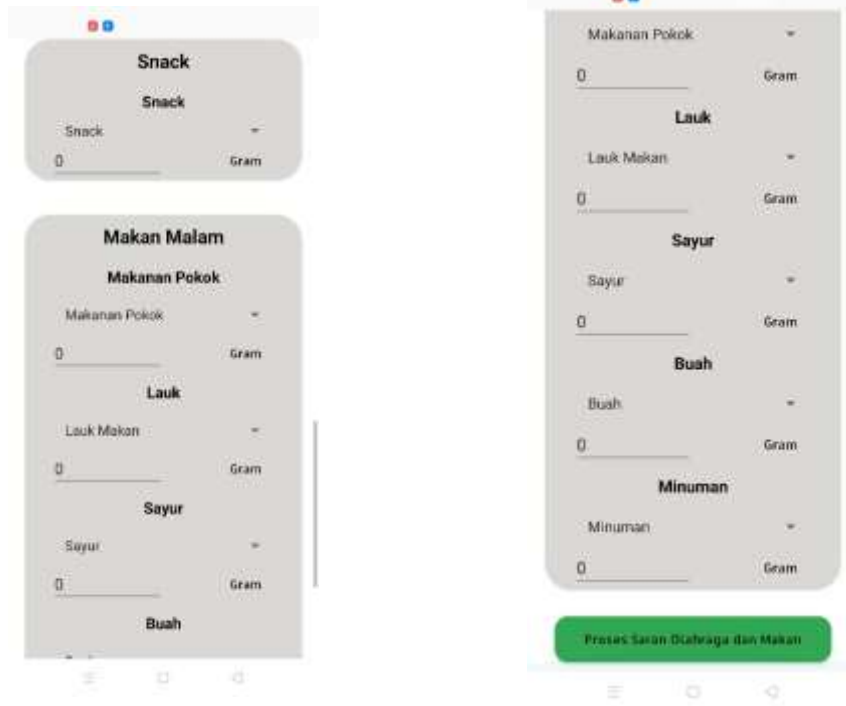
m. Tampilan Detail Perhitungan Hasil



Gambar 5. 15 Detail Perhitungan Sistem

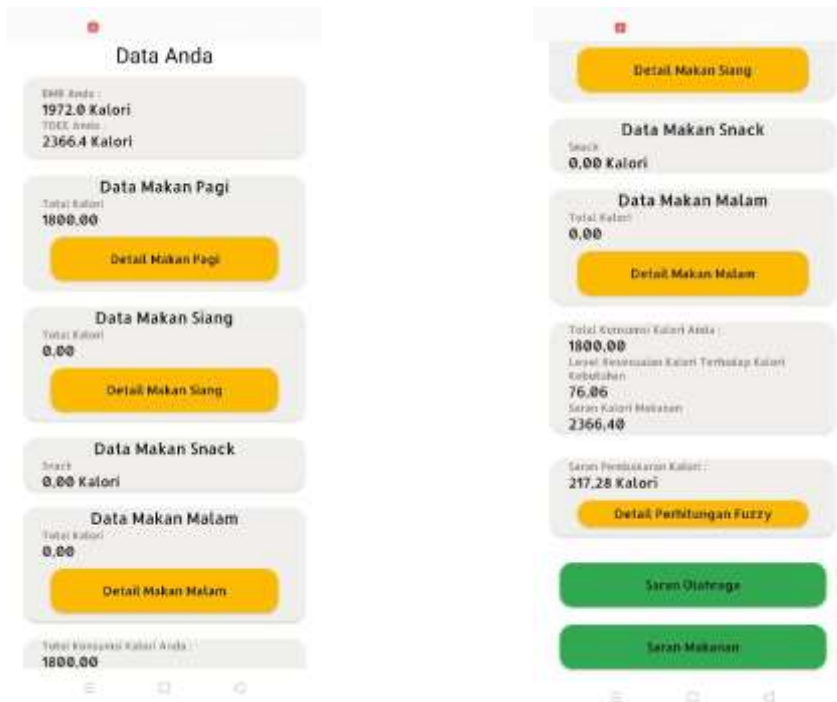
n. Tampilan Input Makanan





Gambar 5. 16 Input Makanan User

o. Tampilan Saran Pembakaran Kalori



Gambar 5. 17 Saran Pembakaran Kalori

p. Tampilan Saran Olahraga



Gambar 5. 18 Saran Olahraga

q. Tampilan Saran Menu Makanan



Gambar 5. 19 Saran Menu Makanan

1.1.3. Implementasi Pengkodean

Source code dari sistem ini akan dijelaskan sebagai inti dari Sistem Pendukung Keputusan Penentuan Olahraga dan Perhitungan Kalori menggunakan *Fuzzy Tsukamoto*.

a. Implementasi *code* perhitungan BMI, BMR, TDEE, dan BBI

Fungsi dari class ini adalah untuk menghitung BMI, BMR, TDEE, dan BBI

Inputdata.java

```
bbi = (Double.parseDouble(edttb.getText().toString()) - 100) -
((Double.parseDouble(edttb.getText().toString()) - 100)/100 *
10);

    if(rb.getText().toString().equals("Perempuan"))
    {
        bmr = 655 + (9.6 *
Double.parseDouble(edtbb.getText().toString()) + (1.8 *
Double.parseDouble(edttb.getText().toString()) - (4.7 *
Double.parseDouble(edtumur.getText().toString())));
    }
    else
    {
        bmr = 66 + (13.7 *
Double.parseDouble(edtbb.getText().toString()) + (5 *
Double.parseDouble(edttb.getText().toString()) - (6.8 *
Double.parseDouble(edtumur.getText().toString())));
    }

    bmi = Double.parseDouble(edtbb.getText().toString()) /
((Double.parseDouble(edttb.getText().toString())/100) *
(Double.parseDouble(edttb.getText().toString())/100));

    if (spnaktivitas.getSelectedItemAt().toString().equals("Aktivitas
Ringan"))
    {
```

```

        tdee = bmr * 1.2;
    }
    else
    if(spnaktivitas.getSelectedItem().toString().equals("Aktivitas
    Sedang"))
        {
            tdee = bmr * 1.5;
        }
    else
    if(spnaktivitas.getSelectedItem().toString().equals("Aktivitas
    Berat"))
        {
            tdee = bmr * 1.9;
        }
    }

```

b. Implementasi *code* perhitungan kalori input makan

Fungsi dari code ini adalah untuk menjumlahkan kalori dengan berat makan yang dikonsumsi user

Makananuser.java

```

// spinner pagi
spnpagi.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view,
int i, long l) {
        MakananModels makananModels = listmakan.get(i);
        kalorimakanpagi = makananModels.getKalori();
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

```

```

        kalorimakanpagi = "0";
        etberatpagi.setText("0");
    }
});

```

c. Implementasi *input* dan *update* data olahraga ke firebase

Berikut adalah code implementasi untuk input dan update data olahraga ke firebase

Inputor.java

```

void updateData()
{
    myRef.child(mbundle.getString("key"))
        .setValue(new OlahragaModels(jenisor.getText().toString(),
            durasi.getText().toString(),
            kalori.getText().toString()))
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(Inputor.this, "Success",
                    Toast.LENGTH_SHORT).show();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(Inputor.this, "Fail"+e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        });
    Intent i = new Intent(Inputor.this, ListOlahraga.class);
    startActivity(i);
    finish();
}

```

```

    }

    void newData(){
        myRef.push()
            .setValue(new OlahragaModels(jenisor.getText().toString(),
                durasi.getText().toString(),
                kalori.getText().toString()))
            .addOnSuccessListener(new OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Toast.makeText(Inputor.this, "Success",
                        Toast.LENGTH_SHORT).show();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(Inputor.this, "Fail"+e.getMessage(),
                        Toast.LENGTH_SHORT).show();
                }
            });
        Intent i = new Intent(Inputor.this, ListOlahraga.class);
        startActivity(i);
        finish();
    }
}

```

d. Implementasi input dan update data makanan ke firebase

Berikut adalah code implementasi untuk input dan update data makanan ke firebase

Inputmakanan.java

```

void newData()
{

```



```

myRef.push()
        .setValue(new MakananModels(etNama.getText().toString(),
        etBerat.getText().toString(),

etKalori.getText().toString(),jenismakanan.getSelectedItemAt(0).toString()))
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(Inputmakanan.this, "Success",
Toast.LENGTH_SHORT).show();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(Inputmakanan.this, "Fail"+e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
Intent i = new Intent(Inputmakanan.this, ListMakanan.class);
startActivity(i);
finish();
}

void loadData(){
    if (mBundle.getString("status").equals("update"))
    {
        etNama.setText(mBundle.getString("namamakanan"));
        etBerat.setText(mBundle.getString("beratmakanan"));
        etKalori.setText(mBundle.getString("kalorimakanan"));
        btndelete.setVisibility(View.VISIBLE);
    }
}

```

```
    else
    {
        btndelete.setVisibility(View.GONE);
    }
}
```

e. Implementasi code fuzzy tsukamoto

Code dibawah ini adalah fuzzy tsukamoto yang diterapkan dalam sistem pendukung keputusan penentuan olahraga dan perhitungan kalori.

ActivitySaran.java

```
double A1 = Math.min(bmiunderweight, xsangatrendah);
double A2 = Math.min(bmiunderweight, xrendah);
double A3 = Math.min(bmiunderweight, xringan);
double A4 = Math.min(bmiunderweight, xnormal);
double A5 = Math.min(bmiunderweight, xlebih);
double A6 = Math.min(bminormal, xsangatrendah);
double A7 = Math.min(bminormal, xrendah);
double A8 = Math.min(bminormal, xringan);
double A9 = Math.min(bminormal, xnormal);
double A10 = Math.min(bminormal, xlebih);
double A11 = Math.min(bmioverweight, xsangatrendah);
double A12 = Math.min(bmioverweight, xrendah);
double A13 = Math.min(bmioverweight, xringan);
double A14 = Math.min(bmioverweight, xnormal);
double A15 = Math.min(bmioverweight, xlebih);
double A16 = Math.min(bmiobesitas, xsangatrendah);
double A17 = Math.min(bmiobesitas, xrendah);
double A18 = Math.min(bmiobesitas, xringan);
double A19 = Math.min(bmiobesitas, xnormal);
double A20 = Math.min(bmiobesitas, xlebih);
```

```
double A21 = Math.min(bmiobesitasextreme, xsangatrendah);  
double A22 = Math.min(bmiobesitasextreme, xrendah);  
double A23 = Math.min(bmiobesitasextreme, xringan);  
double A24 = Math.min(bmiobesitasextreme, xnormal);  
double A25 = Math.min(bmiobesitasextreme, xlebih);
```

```
Log.d(TAG, "A1 : " + A1);  
Log.d(TAG, "A2 : " + A2);  
Log.d(TAG, "A3 : " + A3);  
Log.d(TAG, "A4 : " + A4);  
Log.d(TAG, "A5 : " + A5);  
Log.d(TAG, "A6 : " + A6);  
Log.d(TAG, "A7 : " + A7);  
Log.d(TAG, "A8 : " + A8);  
Log.d(TAG, "A9 : " + A9);  
Log.d(TAG, "A10 : " + A10);  
Log.d(TAG, "A11 : " + A11);  
Log.d(TAG, "A12 : " + A12);  
Log.d(TAG, "A13 : " + A13);  
Log.d(TAG, "A14 : " + A14);  
Log.d(TAG, "A15 : " + A15);  
Log.d(TAG, "A16 : " + A16);  
Log.d(TAG, "A17 : " + A17);  
Log.d(TAG, "A18 : " + A18);  
Log.d(TAG, "A19 : " + A19);  
Log.d(TAG, "A20 : " + A20);  
Log.d(TAG, "A21 : " + A21);  
Log.d(TAG, "A22 : " + A22);  
Log.d(TAG, "A23 : " + A23);  
Log.d(TAG, "A24 : " + A24);  
Log.d(TAG, "A25 : " + A25);
```

double penyebut = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9
+ A10 + A11 + A12 + A13 + A14 + A15 + A16 + A17 + A18 + A19 + A20
+ A21 + A22 + A23 + A24 + A25;

// Pembakaran Sangat Rendah

double Z1 = 100 - (A1 * (100-0));

double Z2 = 100 - (A2 * (100-0));

double Z3 = 100 - (A3 * (100-0));

double Z4 = 100 - (A4 * (100-0));

double Z5 = 100 - (A5 * (100-0));

// Pembakaran Rendah

double Z6 = 200 - (A6 * (200-100));

double Z7 = 200 - (A7 * (200-100));

double Z8 = 200 - (A8 * (200-100));

double Z9 = 200 - (A9 * (200-100));

double Z10 = 200 - (A10 * (200-100));

// Pembakaran Normal

double Z11 = 300 - (A11 * (300-200));

double Z12 = 300 - (A12 * (300-200));

double Z13 = (A13 * (300-200)) + 200;

double Z14 = (A14 * (300-200)) + 200;

double Z15 = (A15 * (300-200)) + 200;

// Pembakaran Tinggi

double Z16 = 1000 - (A16 * (400-300));

double Z17 = 1000 - (A17 * (400-300));

double Z18 = (A18 * (400-300)) + 300;

double Z19 = (A19 * (400-300)) + 300;

double Z20 = (A20 * (400-300)) + 300;

// Pembakaran Sangat Tinggi

double Z21 = 500 - (A21 * (500-400));

```

double Z22 = (A22 * (500-400)) + 400;
double Z23 = (A23 * (500-400)) + 400;
double Z24 = (A24 * (500-400)) + 400;
double Z25 = (A25 * (500-400)) + 400;

double pembilang = (Z1 * A1) + (Z2 * A2) + (Z3 * A3) + (Z4 *
A4) + (Z5 * A5) + (Z6 * A6) + (Z7 * A7) + (Z8 * A8) + (Z9 * A9)
+ (Z10 * A10) + (Z11 * A11) + (Z12 * A12) + (Z13 * A13) +
(Z14 * A14) + (Z15 * A15) + (Z16 * A16) + (Z17 * A17) + (Z18 * A18)
+ (Z19 * A19)
+ (Z20 * A20) + (Z21 * A21) + (Z22 * A22) + (Z23 * A23) +
(Z24 * A24) + (Z25 * A25);

saran = pembilang/penyebut;
String saran1 = String.format("%.2f", saran);
txtsaran.setText(saran1 + " Kalori");
kebutuhankalori = Double.parseDouble(extras.getString("tdee"));
String kebutuhankalori1 = String.format("%.2f", kebutuhankalori);
txtsarankonsumsi.setText(kebutuhankalori1);

```

f. Implementasi *code* saran olahraga

Code berikut digunakan untuk memunculkan saran olahraga yang akan ditampilkan kepada user.

ListSaranOlahraga.java

```

Bundle extras = getIntent().getExtras();
double kal = Double.parseDouble(extras.getString("saran"));
holder.nama.setText("Nama : "+model.getNama());
double kalori = Double.parseDouble(model.getKalori())/15;
double duration = kal/kalori;
String duration1 = String.format("%.2f", duration);

```

```

String kal1 = String.format("%.2f", kal);
holder.durasi.setText("Durasi : "+duration1 + " Menit");

holder.itemView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(ListSaranOlahraga.this, Inputor.class);
        i.putExtra("nama", model.getNama().toString());
        i.putExtra("durasi", model.getDurasi().toString());
        i.putExtra("key", getRef(position).getKey());
        i.putExtra("status", "update");
        startActivity(i);
        finish();
    }
});

```

g. Implementasi *code* saran menu makanan

ListSaranMenu.java

```

Bundle extras = getIntent().getExtras();
double          kalorikebutuhan          =
Double.parseDouble(extras.getString("tdee"));
double kalpokok = (kalorikebutuhan / 6) * 2;
double kallauk = (kalorikebutuhan / 6) * 1;
double kalsayur = (kalorikebutuhan / 6) * 1;
double kalbuah = (kalorikebutuhan / 6) * 1;
double kalminum = (kalorikebutuhan / 6) * 1;
double          kaloripokok          =
Double.parseDouble(model.getKaloripokok());

```

```
double          kalorilauk          =
Double.parseDouble(model.getKalorilauk());
double          kalorisayur         =
Double.parseDouble(model.getKalisayur());
double          kaloribuah          =
Double.parseDouble(model.getKaloribuah());
double          kaloriminum         =
Double.parseDouble(model.getKaloriminum());

holder.namapokok.setText(model.getNamapokok());
holder.kaloripokok.setText(model.getKaloripokok());
holder.namalauk.setText(model.getNamalauk());
holder.kalorilauk.setText(model.getKalorilauk());
holder.namasayur.setText(model.getNamasayur());
holder.kalorisayur.setText(model.getKalisayur());
holder.namabuah.setText(model.getNamabuah());
holder.kaloribuah.setText(model.getKaloribuah());
holder.namaminum.setText(model.getNamaminuman());
holder.kaloriminum.setText(model.getKaloriminum());
```