

DAFTAR PUSTAKA

- Alhassan, S. M., Ahmed, H. G., Almutlaq, B. A., & dkk. (2017). Risk Factors Associated with Acute Coronary Syndrome in Northern Saudi Arabia. *Journal of Cardiology & Current Research*.
- Atikah, A. N., & Widayati, T. A. (2019). HUBUNGAN INDEKS MASSA TUBUH (IMT) DENGAN STATUS PERIODONTAL PADA PEMUDA-PEMUDI DUKUH PENEGAR KEBUMEN .
- Ayani, D. D., Pratiwi, H. S., & Muhandi, H. (2019). Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace. *Jurnal Sistem dan Teknologi Informasi*, 257-262.
- Casas, R., Barquero, S. C., Estruch, R., & Sacanella, E. (2018). Nutrition and Cardiovascular Health. *International Journal of Molecular Sciences*.
- Dafinci, F. (2010). PEMBANGUNAN APLIKASI INFORMASI, PENGADUAN, KRITIK, DAN SARAN SEPUTAR KOTA CIMAHI PADA PLATFORM ANDROID.
- Djafri, D., Monalisa, Elytha, F., & Machmud, R. (2017). EFEK MODIFIKASI FAKTOR RISIKO MODIFIABLE PENYAKIT JANTUNG KORONER: A HOSPITAL-BASED MATCHED CASE CONTROL STUDY. *Jurnal Kesehatan Masyarakat Andalas*, 93-99.
- Fauzan, W. (2013). *Pembangunan Aplikasi Game Puzzle and The Solver*. Bandung: Universitas Komputer Indonesia.
- Herdiana, Y. (2014). APLIKASI RUMUS MATEMATIKA SMA BERBASIS MOBILE. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- Ikhsan, B. K. (2020). *Indeks Massa Tubuh (IMT), Kadar Insuline-Like Growth Factor-1 (IGF-1), Dan Fungsi Memori Jangka Pendek pada Mahasiswa Program Studi Pendidikan Dokter Umum Universitas Hasanuddin Angkatan 2017*. Makassar: Universitas Hasanuddin.
- Indonesia, K. K. (2019, 12). *panganku*. Retrieved from Data Komposisi Pangan Indonesia: <https://www.panganku.org/id-ID/beranda>
- Josi, A., Abdillah, L. A., & Suryayusra. (2014). PENERAPAN TEKNIK WEB SCRAPING. *Jurnal Sistem Informasi*, 159-164.
- Kandou, G. D. (2009). Makanan Etnik Minahasa dan Kejadian. *Jurnal Kesehatan Masyarakat Nasional*.

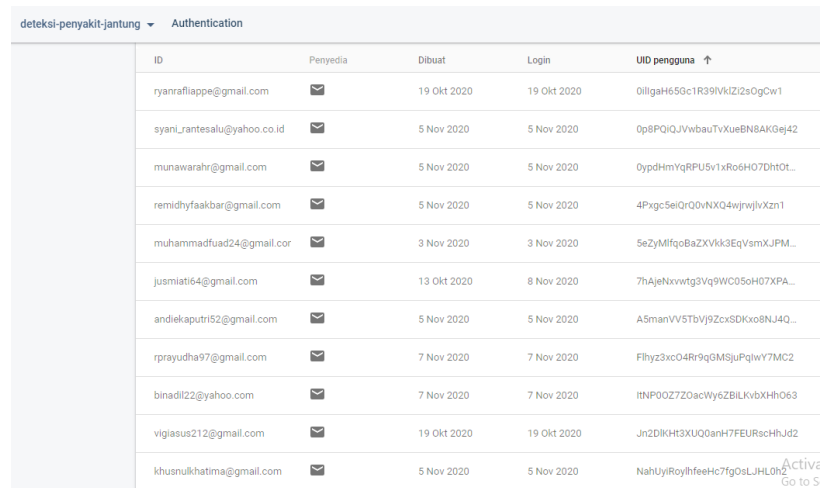
- Khazanah, W., Mulyani, N. S., Ramadhaniah, & Rahma, C. N. (2019). Konsumsi Natrium, Lemak Jenuh dan Serat Berhubungan dengan Kejadian Penyakit Jantung Koroner di Rumah Sakit dr. Zainoel Abidin, Banda Aceh. *Jurnal Kesehatan*.
- Kusuma, A. P., & Setiawan, B. (2018). White Box Testing pada Sistem Pemesanan Desain Sablon Berbasis Web. *Jurnal TEKNIKa*, 1040-1044.
- Lestari, E. S., S, L. D., & S, H. S. (2014). Faktor Risiko Penyakit Kardiovaskuler (Studi Pada Mahasiswa Perokok Fakultas Teknik Jurusan Mesin Universitas Diponegoro Semarang). *JURNAL KESEHATAN MASYARAKAT*.
- Lu, H. T., & Nordin, R. B. (2013). Ethnic differences in the occurrence of acute coronary syndrome: results of the Malaysian National Cardiovascular Disease (NCVD) Database Registry (March 2006 - February 2010). *BMC Cardiovascular Disorders*.
- Mozaffarian, D. (2016). Dietary and Policy Priorities for Cardiovascular Disease, Diabetes, and Obesity. *American Heart Association*, 187-225.
- Nainggolan, E. R., & Susafa'ati. (2018). Pengujian Usability Sistem Informasi Pelayanan Rukun Warga Menggunakan USE Questionnaire. *Jurnal AKRAB JUARA*, 27-34.
- Organization, W. H. (2020, 6). *Cardiovascular Diseases*. Retrieved from World Health Organization: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1
- Patel, A., Thanvi, S., & Patel, U. (2018). Analysis of prevalence of risk factors in patients aged ≤ 45 years presenting with acute coronary syndrome. *International Journal of Biomedical Research*, 309-319.
- Patriyani, R. H., & Purwanto, D. F. (2016). FAKTOR DOMINAN RISIKO TERJADINYA PENYAKIT JANTUNG KORONER (PJK). *Jurnal Keperawatan Global*, 01-54.
- Payara, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *Jurnal Teknik Informatika dan Sistem Informasi*.
- Putri, M. M. (2016). ANALISA KEBIASAAN MAKAN, RIWAYAT ASUPAN SATURATED FATTY ACIDS (SAFA), MONOUNSATURATED FATTY ACIDS (MUFA), POLYUNSATURATED FATTY ACIDS (PUFA) DAN

SERAT PADA PASIEN PENYAKIT JANTUNG KORONER. *ADLN-PERPUSTAKAAN UNIVERSITAS AIRLANGGA*.

- Ralapanawa, U., Kumarasiri, P. V., Jayawickreme, K. P., & dkk. (2019). Epidemiology and risk factors of patients with types of acute coronary syndrome presenting to a tertiary care hospital in Sri Lanka. *BMC Cardiovascular Disorders*.
- Ramadhani, M. F. (2015). PEMBANGUNAN APLIKASI INFORMASI, PENGADUAN, KRITIK, DAN SARAN SEPUTAR KOTA CIMAHU PADA PLATFORM ANDROID. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- Sandy, L. A., Januar, R., & Hariadi, R. R. (2017). Rancang Bangun Aplikasi Chat pada Platform Android dengan Media Input berupa Canvas dan Shareable Canvas untuk Bekerja Dalam Satu Canvas secara Online. *JURNAL TEKNIK ITS*.
- Tanari, A., Handojo, A., & Andjarwirawan, J. (2019). Aplikasi Pencarian Jurnal Ilmiah dengan Term Frequency-Inverse Document Frequency. *JURNAL INFRA*.
- Utami, F. N., Satoto, K. I., & Martono, K. T. (2016). Rancang Bangun Aplikasi Sistem Pakar Diagnosis Gangguan Emosional Pada Anak Berbasis Aplikasi Website. *Jurnal Teknologi dan Sistem Komputer*.

LAMPIRAN 1

- Tampilan *Database Akun Users*



ID	Penyedia	Dibuat	Login	UID pengguna ↑
ryanrafiappe@gmail.com	✉	19 Okt 2020	19 Okt 2020	0lllgaH65Gc1R39lVkiZl2sOgCw1
syani_rantesalu@yahoo.co.id	✉	5 Nov 2020	5 Nov 2020	0p8PQIQJvVbauTVxueBN8AKGej42
munawarahr@gmail.com	✉	5 Nov 2020	5 Nov 2020	0ypdHmYqRPU5v1xRo6H07Dht0T...
remidhyfaakbar@gmail.com	✉	5 Nov 2020	5 Nov 2020	4Pxc5eiQrQ0vNXQ4vjrvjVXzn1
muhammadfud24@gmail.com	✉	3 Nov 2020	3 Nov 2020	5eZyMfqcBaZXVkk3EqVsmXJPM...
jusmiati64@gmail.com	✉	13 Okt 2020	8 Nov 2020	7hAjeNxvvtg3Vq9WC05oH07XPA...
andiekaputri52@gmail.com	✉	5 Nov 2020	5 Nov 2020	A5manVV5Tbv9ZcxSDkxo8NJ4Q...
rprayudha97@gmail.com	✉	7 Nov 2020	7 Nov 2020	FhYz3xcO4Rr9qGMSjuPqIwY7MC2
binadl2@yahoo.com	✉	7 Nov 2020	7 Nov 2020	lHNp00Z7Z0acWY6ZBLIKvbxHhO63
vigasus212@gmail.com	✉	19 Okt 2020	19 Okt 2020	Jn2DlKH3XUQ0anH7FEURscHhJd2
khusnulhatima@gmail.com	✉	5 Nov 2020	5 Nov 2020	NahUyR0yllfheeHc7fg0sLJHL0K...

- Tampilan *Database User*

```
user
  -MKc3AIB6yNAtwMQGHdX
    agama: "Islam"
    alamat: "Gowa"
    email: "jusmiati64@gmail.co
    gender: "perempuan"
    hp: "082196794451"
    hpKeluarga: ""
    konsumsiAlkohol: "tidak"
    namaLengkap: "Jusmiati"
    pekerjaan: "mahasiswa"
    pendidikanTerakhir: "SMA/Sederajat"
    penghasilanBulanan: "Kurang dari 1 Juti"
    riwayat_jantung_keluarga: "tidak"
    statusMerokok: "Bukan Perokok"
    suku: "Bugis"
```

- Tampilan *Database IMT*

[deteksi-penyakit-jantung](#) > [imt](#) > [7NE4Y7sR09XHyOwkj7ovSZGk10X2](#)

7NE4Y7sR09XHyOwkj7ovSZGk10X2

-MT13hoTrriVSnCS_vXD

```
berat_badan: "50"  
email: "jusmiatic@gmail.com"  
hasil: "19.777699\n\nNormal\n"  
tanggal_input: -1612795017132  
tinggi_badan: "159"
```

- Tampilan *Database* Riwayat Penyakit

riwayat_penyakit

+ -MKm__M4ybi3HwYckDOJ

-MLBBYzXXd5h3mwiUSCN

```
email: "jusmiati64@gmail.com"  
hasil: "tidak ada\n"
```

- Tampilan *Database* Pola Hidup

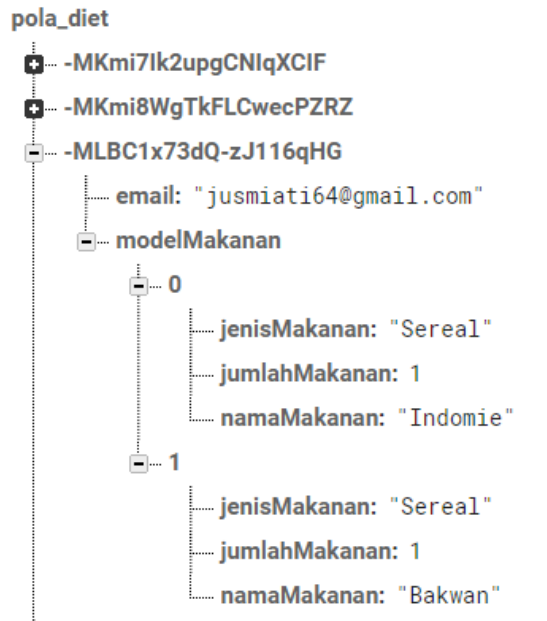
[deteksi-penyakit-jantung](#) > [aktivitas_fisik](#) > [0ojkPLHWukNjZKmY68s76RfETt93](#)

0ojkPLHWukNjZKmY68s76RfETt93

-MTfSrNHTm_UfSmdirZ

```
durasi_olahraga: "Lebih dari 3 x Seminggu"  
email: "jusmiati64@gmail.com"  
hasil: "120-200 menit = anda aktif"  
hasilDalamAngka: 180  
lama_olahraga: "30-45 menit/berolahraga"  
tanggal_input: -1613489473890
```

- Tampilan *Database* Pola Diet

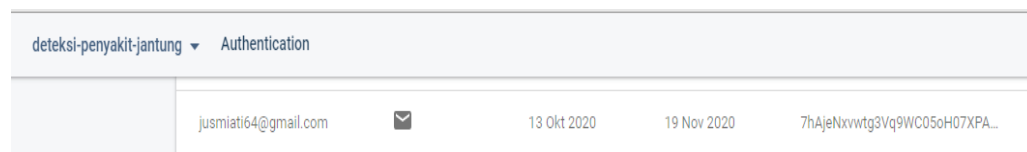
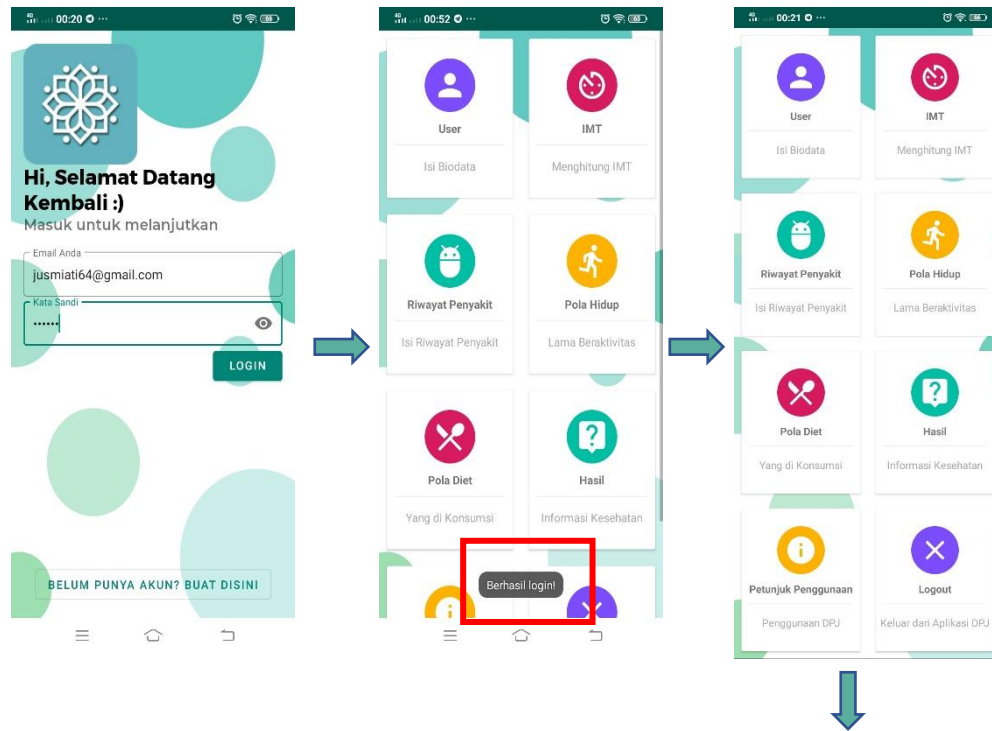


LAMPIRAN 2

- *Screenshot* pengujian *Black Box*

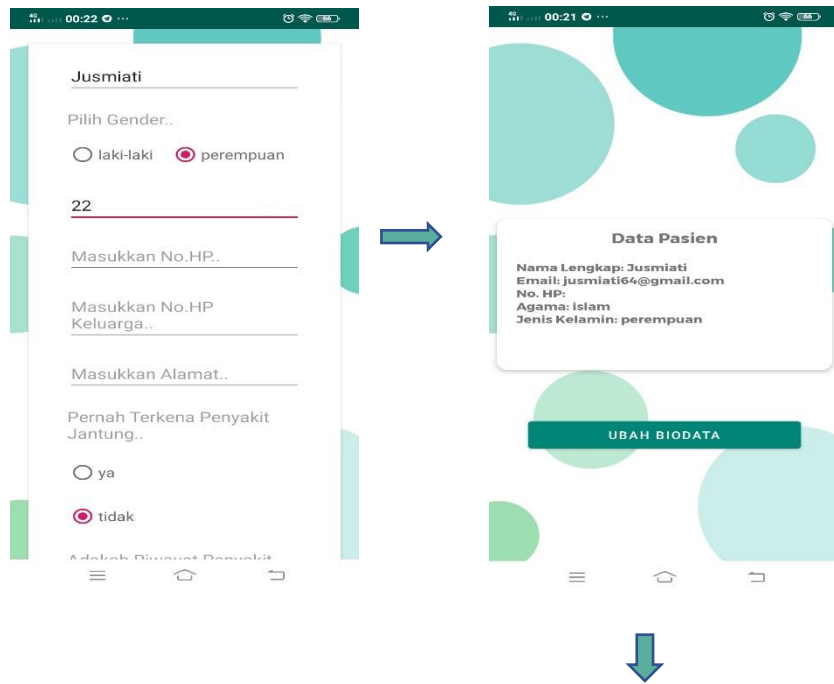
Hasil Yang Diharapkan	Input	Output
User dapat melakukan <i>login</i> pada halaman <i>login</i>	User menginput email dan <i>password</i>	User berhasil <i>login, user</i> dapat masuk pada halaman utama dan data login tersimpan pada <i>Authentication Database</i>

1. *Screenshot* hasil pengujian pada halaman *login*



Hasil Yang Diharapkan	Input	Output
User dapat menginput data Biodata pada halaman <i>User</i>	<i>User</i> menginput data biodata	Data biodata tampil di halaman Data Pasien dan data tersimpan di <i>Realtime Database</i>

2. Screenshot hasil pengujian pada halaman User



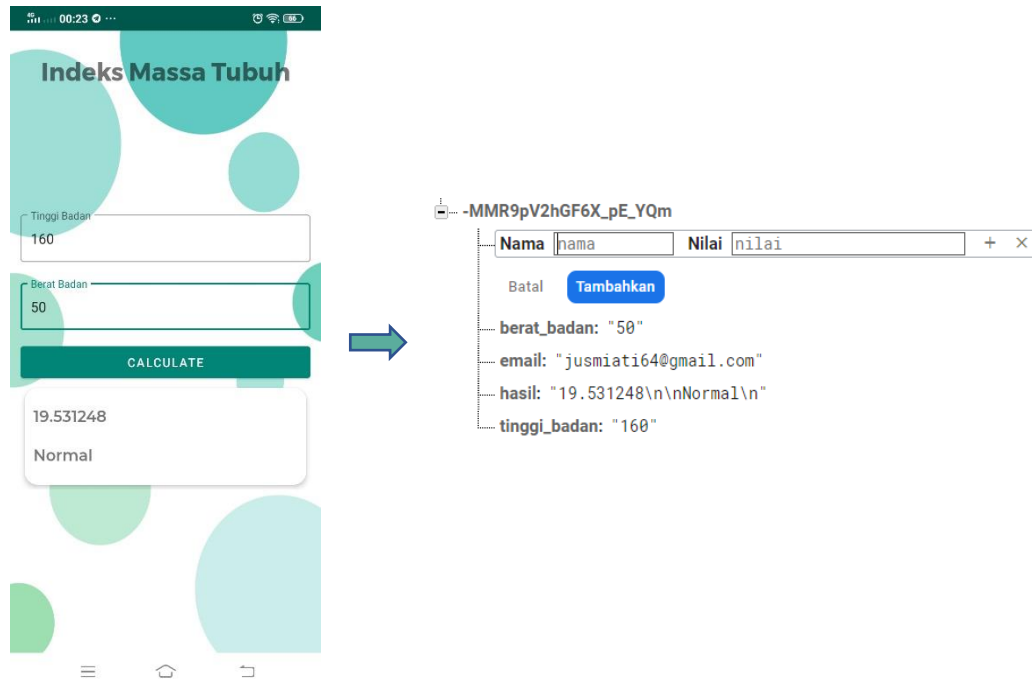
```

- -MMQt5p3ione34u6xC5-
  Nama | nama | Nilai | nilai | + | x
  Bata | Tambahkan
  agama: "islam"
  alamat: ""
  email: "jusmiati64@gmail.com"
  gender: "perempuan"
  hp: ""
  hpKeluarga: ""
  konsumsiAlkohol: "tidak"
  namaLengkap: "Jusmiati"
  pekerjaan: ""
  pendidikanTerakhir: "SMA/Sederajat"

```

Hasil Yang Diharapkan	Input	Output
User dapat menginput indeks massa tubuh (IMT) pada halaman Indeks Massa Tubuh	User menginput tinggi badan dan berat badan	Data hasil kalkulasi indeks massa tubuh tampil di halaman indeks massa tubuh dan data tersimpan di <i>Realtime Database</i>

3. *Screenshot* hasil pengujian pada halaman IMT



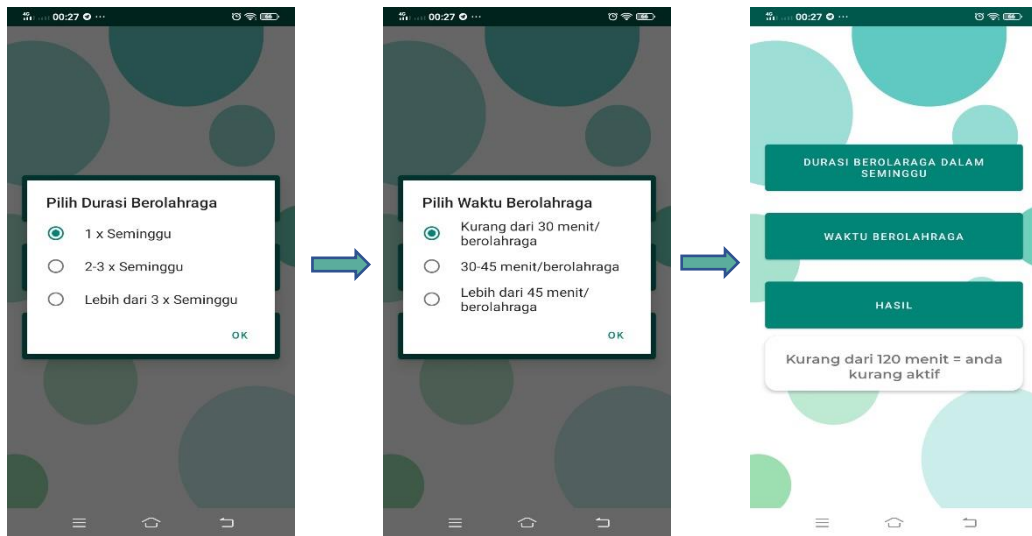
Hasil Yang Diharapkan	Input	Output
<i>User</i> dapat memilih riwayat penyakit pada halaman riwayat penyakit	<i>User</i> memilih riwayat penyakit	Data riwayat penyakit tampil di halaman riwayat penyakit dan data tersimpan di <i>Realtime Database</i>

4. *Screenshot* hasil pengujian pada halaman riwayat penyakit



Hasil Yang Diharapkan	Input	Output
<i>User</i> dapat mengetahui pola hidup pada halaman pola hidup	<i>User</i> memilih durasi berolahraga dan memilih waktu berolahraga	Data hasil kalkulasi tampil dan data tersimpan di <i>Realtime Database</i>

5. *Screenshot* hasil pengujian pada halaman pola hidup



```

-MMRAeN-ljUOa_wxu_pf
  Nama |nama      Nilai |nilai
  Batal  Tambahkan
  durasi_olahraga: "1 x Seminggu"
  email: "jusmiati64@gmail.com"
  hasil: "Kurang dari 120 menit = anda kurang aktif"
  hasilDalamAngka: 30
  lama_olahraga: "Kurang dari 30 menit/berolahraga"

```



Hasil Yang Diharapkan	Input	Output
User dapat memilih jenis makanan yang dikonsumsi pada halaman pola diet	User memilih jenis makanan	Data makanan tersimpan di <i>Realtime Database</i>

6. *Screenshot* hasil pengujian pada halaman pola diet

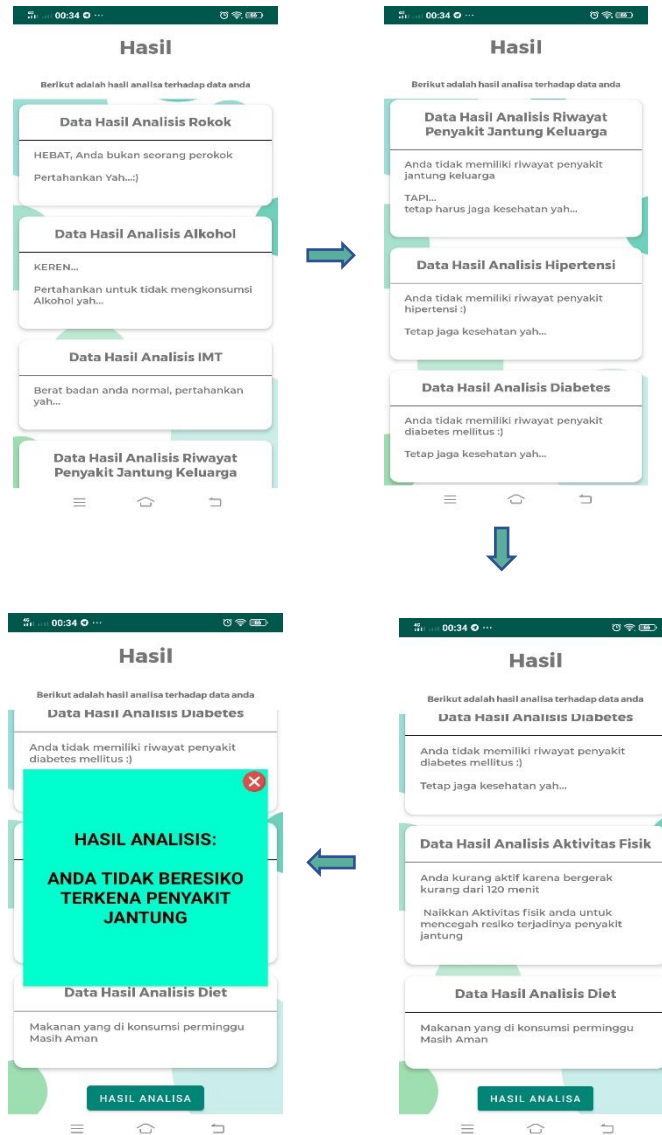


```

- MMRSea_zytLkZal-2Fy
  email: "jusmiati64@gmail.com"
  modelMakanan
    0
      jenisMakanan: "Sereal"
      jumlahMakanan: 1
      namaMakanan: "Bihun goreng instan"
    1
      Nama | nama | Nilai | nilai
      Batal | Tambahkan
      jenisMakanan: "Sereal"
      jumlahMakanan: 2
      namaMakanan: "Indomie"
  
```

Hasil Yang Diharapkan	Input	Output
User dapat mengetahui hasil analisa pada halaman hasil	Menginput semua data pada menu	Menampilkan hasil kalkulasi setiap menu dan menampilkan hasil analisis

7. Screenshot hasil pengujian pada halaman hasil



Hasil Yang Diharapkan	Input	Output
User dapat Menampilkan petunjuk penggunaan pada halaman petunjuk penggunaan	User menekan menu petunjuk penggunaan	Menampilkan petunjuk penggunaan

8. Screenshot hasil pengujian pada halaman petunjuk penggunaan



Cara Menggunakan Aplikasi DPJ

- Membuat akun, jika belum memiliki akun
- Kemudian melakukan Login

Anda telah berada di Halaman Utama Aplikasi DPJ, yang harus anda lakukan menginput semua data di menu pada aplikasi DPJ untuk menghasilkan data analisis

Langkah pertama :

- Pilih Menu User
- Pilih isi biodata
- Masukkan biodata anda
- Klik tombol Simpan Biodata

Langkah kedua :

- Pilih Menu IMT
- Masukkan tinggi badan
- Masukkan berat badan
- Klik tombol Calculate

Langkah ketiga :

- Pilih Menu Riwayat Penyakit
- Klik tombol Riwayat Penyakit
- Pilih Riwayat Penyakit Anda
- Klik tombol Simpan

Langkah keempat :

- Pilih Menu Pola Hidup
- Klik Durasi Berolahraga Dalam Seminggu
- Pilih Durasi Berolahraga
- Klik tombol OK
- Selanjutnya, Klik Waktu Berolahraga
- Pilih Waktu Berolahraga
- Klik tombol OK.
- Kemudian Klik tombol HASIL

Langkah kelima :



Hasil Yang Diharapkan	Input	Output
<i>User dapat melakukan logout pada halaman logout</i>	<i>User menekan menu logout</i>	Keluar dari halaman utama

9. Screenshot hasil pengujian pada halaman petunjuk penggunaan



Hi, Selamat Datang Kembali :)

Masuk untuk melanjutkan

LOGIN

[BELUM PUNYA AKUN? BUAT DISINI](#)

LAMPIRAN 3

Tabel *Database* Aplikasi (Menu User dan IMT)

No	Data User							Indeks Massa Tubuh	
	Nama	Jenis Kelamin	Umur	Penyakit jantung?(Ya/Tidak)	Ada riwayat penyakit jantung dalam keluarga?(Ya/Tidak)	Status Merokok	Konsumsi Alkohol?	Tinggi Badan	Berat Badan
1	muhammadfuad24@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak		
2	khunulkhatima09@gmail.com	perempuan	24	tidak	tidak	bukan perokok	tidak	147	33
3	ryanrafiappe@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak	169	53
4	Lauratalia231@gmail.com	perempuan	22	tidak	tidak	bukan perokok	tidak	171	65
5	ardiyadwirahmasari@gmail.com	perempuan	23	tidak	tidak	bukan perokok	tidak	162	46
6	umnihuun@gmail.com	perempuan	23	tidak	ya	bukan perokok	tidak	154	44
7	rregina2209@gmail.com	perempuan	22	tidak	ya	bukan perokok	tidak	150	51
8	Nazilarizaaprisa09@gmail.com	perempuan	23	tidak	tidak	bukan perokok	tidak	163	47
9	fhirastikanisha@gmail.com	perempuan	31	tidak	tidak	bukan perokok	tidak	165	57
10	syani_rantesalu@yahoo.co.id	perempuan	31	tidak	tidak	bukan perokok	tidak	152	60
11	rinanovianty36@gmail.com	perempuan	33	tidak	tidak	bukan perokok	tidak	146	45
12	Andiekaputri52@gmail.com	perempuan	23	tidak	tidak	bukan perokok	tidak	160	45
13	Fathur8wolf@gmail.com	laki-laki	22	tidak	ya	bukan perokok	tidak	183	86
14	Munawarahr@gmail.com	perempuan	35	tidak	tidak	bukan perokok	tidak	144	66
15	fuadkhairi.ifb@gmail.com	laki-laki	23	tidak	ya	bukan perokok	tidak	167	62
16	fadelr35@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak	165	75
17	Gyustika@gmail.com	perempuan	21	tidak	tidak	bukan perokok	tidak	164	51
18	rprayudha97@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak	169	68
19	nurul_fatwa5@yahoo.co.id	perempuan	24	tidak	tidak	bukan perokok	tidak	150	45
20	irhawatyirwan@gmail.com	perempuan	23	tidak	tidak	bukan perokok	tidak	155	55
21	saidsyamil86@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak	171	74
22	Sabtian8@gmail.com	laki-laki	23	tidak	tidak	bukan perokok	tidak	163	54
23	musfirahamka470@mail.com	perempuan	24	tidak	tidak	bukan perokok	tidak	155	39

Tabel *Database* Aplikasi (Menu Riwayat Penyakit, Pola Hidup dan Pola Diet)

Riwayat Penyakit (Checklist)								Pola Hidup (Checklist)						Pola Diet
Asam Urat	Asma	Diabetes Mellitus	Gagal Ginjal	Hipertensi	Kolesterol	Stroke	Lainnya	Durasi Berolahraga (/minggu)			Waktu Berolahraga (/olahraga)			
								1 kali	2-3 kali	> 3 kali	< 30 menit	30 - 45 menit	> 45 menit	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	indomie 3, tahu telur 1
-	-	-	-	-	-	-	-	-	√	-	-	-	√	tahu goreng 3, ikan gorong
-	-	-	-	-	-	-	tipes	√	-	-	√	-	-	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	-
-	-	-	-	-	-	-	-	√	-	-	-	-	√	indomie 1, nasi minyak 1
-	-	-	-	-	-	-	Demam berdarah	√	-	-	√	-	-	indomie 1, roti putih 3, ap
-	-	-	-	-	-	-	-	-	√	-	-	√	-	Indomie 1, Roti putih 5, A
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	indomie 3, roti putih 4, bis
-	-	-	-	-	-	-	-	√	-	-	√	-	-	roti putih 4, biskuit 4
-	-	-	-	-	-	-	-	√	-	-	√	-	-	-
-	-	-	-	-	-	-	-	-	-	√	-	√	-	indomie 1
-	-	-	-	-	-	-	Tipes	√	-	-	√	-	-	Indomie 7
-	-	-	-	-	√	-	-	√	-	-	-	√	-	kelepon 1, ketoprak 1, mie
-	-	-	-	-	-	-	-	-	√	-	-	√	-	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	indomie 3, roti putih 1, ba
-	-	-	-	-	-	-	-	√	-	-	√	-	-	bihun goreng instan 9, in
-	-	-	-	-	-	-	-	√	-	-	√	-	-	indomie 8
-	-	-	-	-	-	-	-	-	√	-	-	-	√	-
-	-	-	-	-	-	-	-	√	-	-	√	-	-	indomie 2, roti putih 2, bis

Tabel Database Aplikasi (Menu Hasil)

Hasil								Hasil Analisa (ceklis)	
Analisis Rokok	Analisis Alkohol	Analisis IMT	Analisis RPJK	Analisis Hipertensi	Analisis Diabetes	Analisis Aktivitas Fisik	Analisis Diet	Berisiko	Tidak Berisiko
tidak	tidak		tidak ada	-	-	-	-		√
tidak	tidak	Berat badan kurang	tidak ada	tidak	tidak	Kurang aktif	tidak berisiko		√
tidak	tidak	Normal	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Normal	tidak ada	tidak		Aktif	berisiko	√	
tidak	tidak	Berat badan kurang	tidak ada	tidak	tidak	Kurang aktif	-		√
tidak	tidak	Normal	ya	tidak	tidak	Kurang aktif	tidak berisiko		√
tidak	tidak	Normal	ya	tidak	tidak	Kurang aktif	tidak berisiko		√
tidak	tidak	Berat badan kurang	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Normal	tidak ada	tidak	tidak	Aktif	berisiko	√	
tidak	tidak	Obesitas 1	tidak ada	tidak	tidak	-	-		√
tidak	tidak	Normal	tidak ada	tidak	tidak	-	-		√
tidak	tidak	Berat badan kurang	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Obesitas I	ya	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Obesitas II	tidak ada	tidak	tidak	Kurang aktif	-		√
tidak	tidak	Normal	ya	tidak	tidak	Aktif	tidak berisiko		√
tidak	tidak	Obesitas I	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Normal	tidak ada	tidak	tidak	Kurang aktif	tidak berisiko		√
tidak	tidak	Beresiko	tidak ada	tidak	tidak	Aktif	tidak berisiko		√
tidak	tidak	Normal	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Normal	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Obesitas I	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	
tidak	tidak	Normal	tidak ada	tidak	tidak	Aktif	-		√
tidak	tidak	Berat badan kurang	tidak ada	tidak	tidak	Kurang aktif	berisiko	√	

LAMPIRAN 4

- Isi pertanyaan USE *Questionnaire*

No	Pertanyaan	Sangat tidak setuju / Sangat Setuju				
		1	2	3	4	5
1.	Menu pada aplikasi DPJ dapat berfungsi dengan baik					
2.	Aplikasi DPJ tidak mengalami kendala dalam penggunaan					
3.	Aplikasi DPJ memberikan informasi mengenai faktor-faktor yang dapat menimbulkan penyakit jantung					
4.	Aplikasi DPJ mudah untuk di <i>install</i>					
5.	Aplikasi DPJ dapat digunakan tanpa instruksi tertulis					
6.	Penggunaan aplikasi DPJ mudah di pelajari					

7.	Aplikasi DPJ dapat dipelajari dengan cepat					
8.	Saya mudah mengingat bagaimana menggunakan aplikasi DPJ					
9.	Tata letak menu yang terdapat di layar <i>handphone</i> sangat jelas					
10.	Aplikasi DPJ mudah digunakan (<i>user friendly</i>)					

LAMPIRAN 5

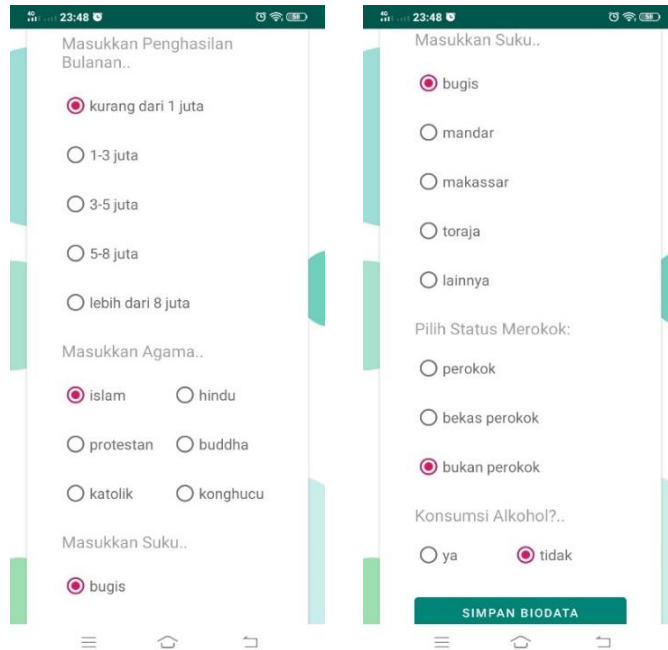
Tampilan pada Aplikasi DPJ :

- Halaman Buat Akun

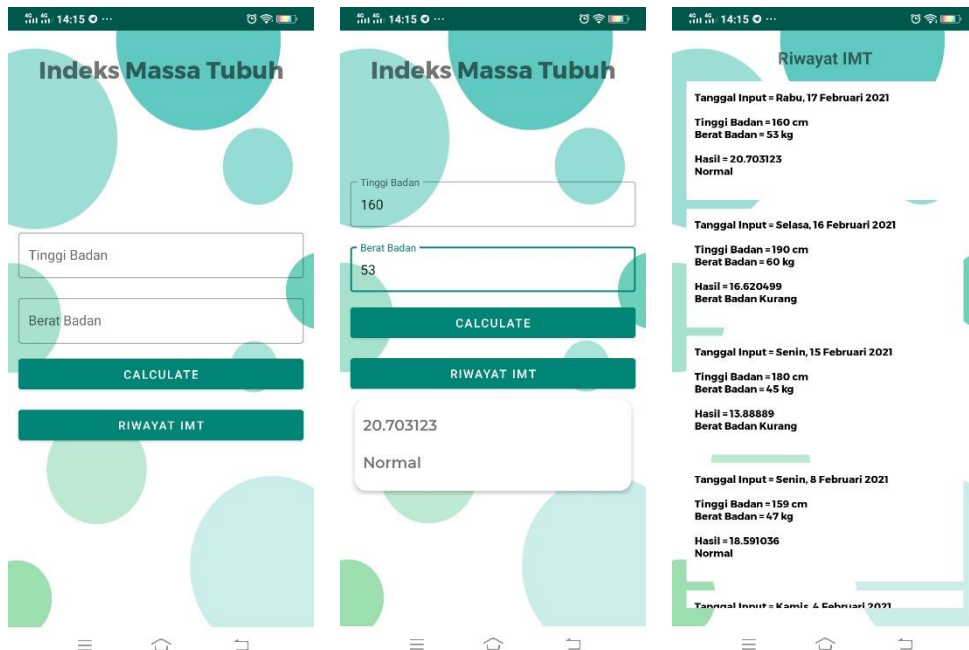
Screenshot of the 'Buat Akun' (Create Account) screen in the DPJ application. The screen features a teal header with a white geometric logo. Below the header, the text 'Selamat Datang' (Welcome) is displayed, followed by 'Buat Akun untuk Memulai Aplikasi' (Create Account to start the application). The form includes three input fields: 'Email Anda' (Your Email), 'Kata Sandi' (Password), and 'Ulangi Kata Sandi' (Repeat Password). A teal button labeled 'BUAT AKUN' is positioned below the fields. At the bottom, a link reads 'SUDAH PUNYA AKUN? MASUK SEKARANG' (ALREADY HAVE AN ACCOUNT? LOGIN NOW). The Android navigation bar is visible at the bottom.

- Halaman User

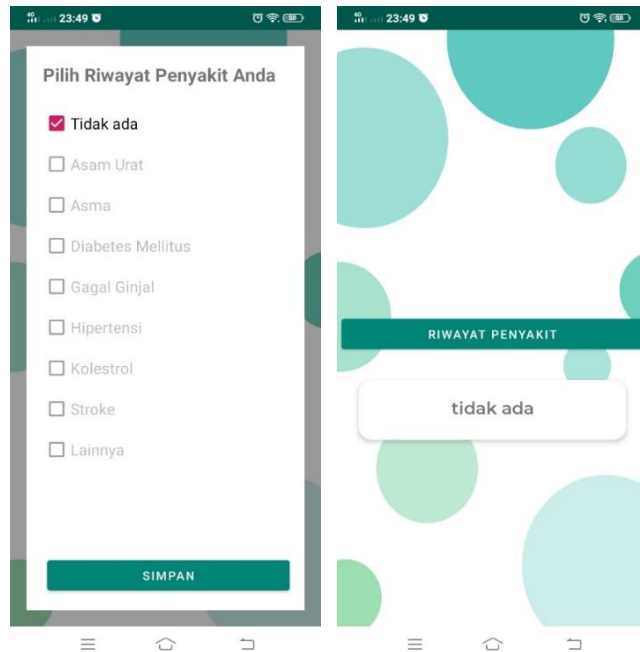
Two side-by-side screenshots of the 'Halaman User' (User Page) in the DPJ application. The left screenshot shows a user profile for 'Jusmiati' with fields for 'Pilih Gender..' (selected 'perempuan'), '23', 'Masukkan No.HP..', 'Masukkan No.HP Keluarga..', 'Gowa', and 'Pernah Terkena Penyakit Jantung..' (selected 'tidak'). The right screenshot shows a questionnaire titled 'Adakah Riwayat Penyakit Jantung Dalam Keluarga?..' with radio button options for 'ya' and 'tidak' (selected). Below are sections for 'Masukkan Pendidikan Terakhir..' (selected 'SMA/Sederajat'), 'Masukkan Pekerjaan..' (selected 'Mahasiswa'), and 'Masukkan Penghasilan Bulanan..'.



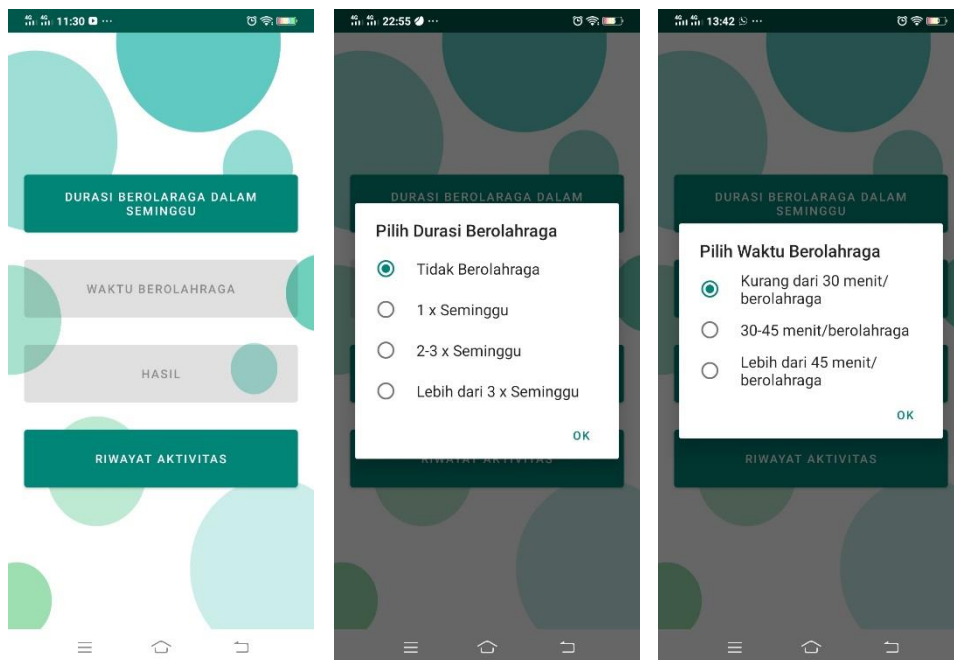
- Halaman IMT (indeks massa tubuh)

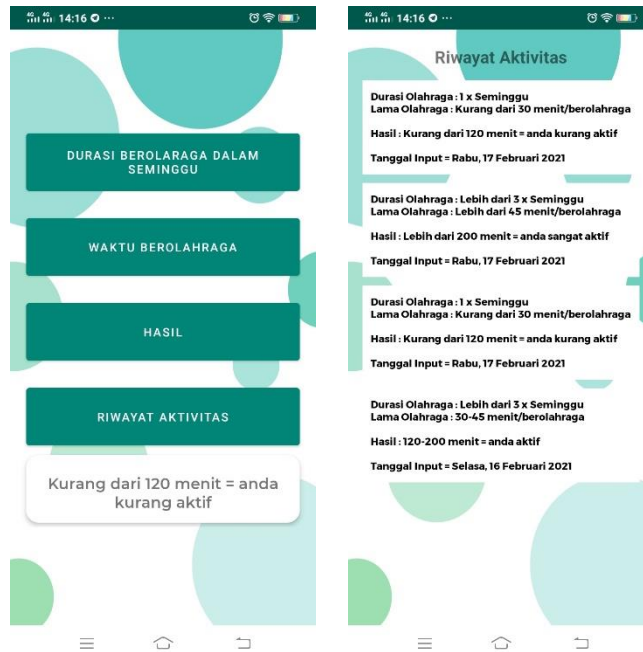


- Halaman Riwayat Penyakit

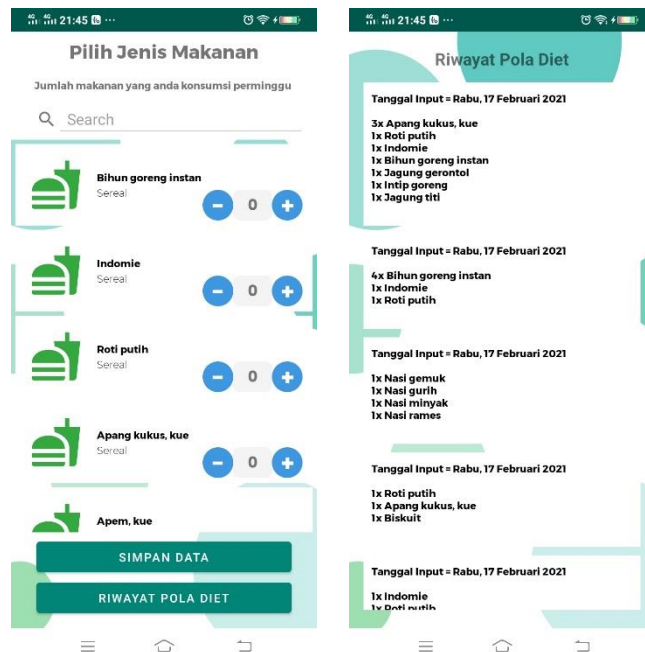


- Halaman Pola Hidup





- Halaman Pola Diet



- Halaman Petunjuk Penggunaan

Cara Menggunakan Aplikasi DPJ

Anda telah berada di Halaman Utama Aplikasi DPJ, yang harus anda lakukan menginput semua data di menu pada aplikasi DPJ untuk menghasilkan data analisis

Langkah pertama :
- Pilih Menu User
- Pilih Isi biodata
- Masukkan biodata anda
- Klik tombol Simpan Biodata

Langkah kedua :
- Pilih Menu IMT
- Masukkan tinggi badan
- Masukkan berat badan
- Klik tombol Calculate

Langkah ketiga :
- Pilih Menu Riwayat Penyakit
- Klik tombol Riwayat Penyakit
- Pilih Riwayat Penyakit Anda
- Klik tombol Simpan

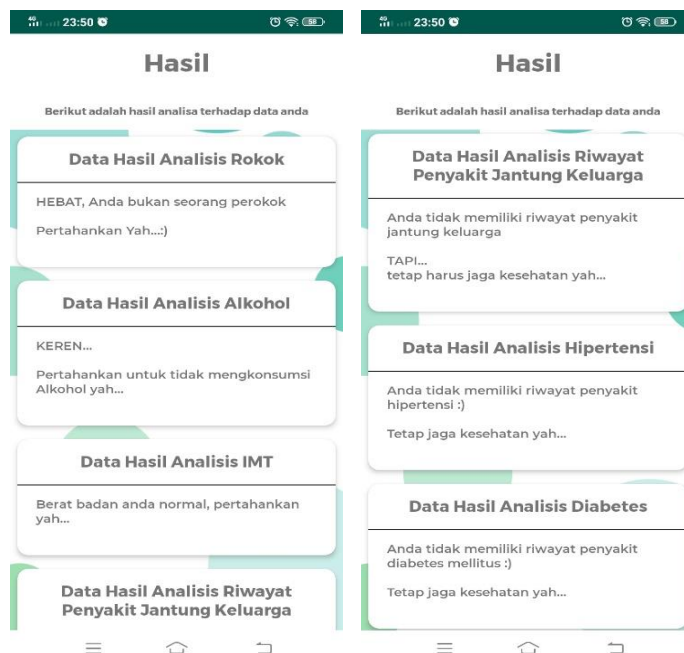
Langkah keempat :
- Pilih Menu Pola Hidup
- Pilih Durasi Berolahraga Dalam Seminggu
- Pilih Durasi Berolahraga
- Klik tombol OK
- Selanjutnya, Klik Waktu Berolahraga
- Pilih Waktu Berolahraga
- Klik tombol OK
- Kemudian Klik tombol HASIL

Langkah kelima :
- Pilih Menu Pola Diet
- Pilih Jenis Makanan
- Klik tombol Simpan Data

Langkah keenam :
- Pilih Menu Hasil. "Anda dapat melihat hasil analisis setiap menu".
- Kemudian Klik tombol Hasil Analisa "Untuk mengetahui anda memiliki kategori beresiko terkena penyakit jantung atau kategori tidak beresiko terkena penyakit jantung"

"Terima Kasih Telah Menggunakan Aplikasi DPJ"

• Halaman Hasil





LAMPIRAN 6

1.1. IsiBiodataActivity.java

```
public class IsiBiodataActivity extends AppCompatActivity {
    //private RadioGroup radioGroup;
    private RadioGroup radioRokok, radioGender, radioRiwayat, radioAlkohol,
    radioPendidikan, radioPenghasilan, radioAgama, radioSuku,
    radioPekerjaan, radioTerkena;
    String rokokData = "";
    String genderData = "";
    String riwayatData = "";
    String alkoholData = "";
    String pendidikanData = "";
    String penghasilanData = "";
    String agamaData = "";
    String sukuData = "";
    String pekerjaanData = "";
    String terkenaData = "";
    //Firebase Database
    private BiodataModel biodataModel;
    private DatabaseReference databaseReference;
    private FirebaseDatabase firebaseDatabase;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_isi_biodata);
        radioGender = findViewById(R.id.radiogender);
        radioRiwayat = findViewById(R.id.radiorj);
        radioRokok = findViewById(R.id.radio);
        radioAlkohol = findViewById(R.id.radioalkohol);
        radioPendidikan = findViewById(R.id.radiopt);
        radioPenghasilan = findViewById(R.id.radiopb);
        radioAgama = findViewById(R.id.radioagama);
        radioSuku = findViewById(R.id.radiosuku);
        radioPekerjaan = findViewById(R.id.radiokerja);
        radioTerkena = findViewById(R.id.radioterkena);
        radioGender.setOnCheckedChangeListener((radioGroup, i) -> {
            RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
            boolean isChecked = rB.isChecked();
            if (isChecked) {
                genderData = rB.getText().toString();
                Toast.makeText(getBaseContext(), genderData,
Toast.LENGTH_SHORT).show();
            }
        });
        radioRokok.setOnCheckedChangeListener((radioGroup, i) -> {
            RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
            boolean isChecked = rB.isChecked();
            if (isChecked) {
                rokokData = rB.getText().toString();
                Toast.makeText(getBaseContext(), rokokData,
```

```

Toast.LENGTH_SHORT).show();
    }
    });
    radioRiwayat.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        if (isChecked) {
            riwayatData = rB.getText().toString();
            Toast.makeText(getBaseContext(), riwayatData,
Toast.LENGTH_SHORT).show();
        }
    });
    radioAlkohol.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        if (isChecked) {
            alkoholData = rB.getText().toString();
            Toast.makeText(getBaseContext(), alkoholData,
Toast.LENGTH_SHORT).show();
        }
    });
    radioPendidikan.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        if (isChecked) {
            pendidikanData = rB.getText().toString();
            Toast.makeText(getBaseContext(), pendidikanData,
Toast.LENGTH_SHORT).show();
        }
    });
    radioPenghasilan.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        if (isChecked) {
            penghasilanData = rB.getText().toString();
            Toast.makeText(getBaseContext(), penghasilanData,
Toast.LENGTH_SHORT).show();
        }
    });
    radioAgama.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        if (isChecked) {
            agamaData = rB.getText().toString();
            Toast.makeText(getBaseContext(), agamaData,
Toast.LENGTH_SHORT).show();
        }
    });
    radioSuku.setOnCheckedChangeListener((radioGroup, i) -> {

```

```

        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        EditText sukuET = findViewById(R.id.etsuku);
        if (isChecked) {
            sukuData = rB.getText().toString();
            Toast.makeText(getApplicationContext(), sukuData,
Toast.LENGTH_SHORT).show();
            Log.d("kirim", sukuData);
            if (sukuData.equals("lainnya")) {
                sukuET.setVisibility(View.VISIBLE);
            } else {
                sukuET.setVisibility(View.GONE);
            }
        }
    });
    radioPekerjaan.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        EditText pekerjaanET = findViewById(R.id.etkerja);
        if (isChecked) {
            pekerjaanData = rB.getText().toString();
            Toast.makeText(getApplicationContext(), pekerjaanData,
Toast.LENGTH_SHORT).show();
            Log.d("kirim", pekerjaanData);
            if (pekerjaanData.equals("lainnya")) {
                pekerjaanET.setVisibility(View.VISIBLE);
            } else {
                pekerjaanET.setVisibility(View.GONE);
            }
        }
    });
    radioTerkena.setOnCheckedChangeListener((radioGroup, i) -> {
        RadioButton rB =
radioGroup.findViewById(radioGroup.getCheckedRadioButtonId());
        boolean isChecked = rB.isChecked();
        EditText tahunjantungET = findViewById(R.id.etjantung);
        if (isChecked) {
            terkenaData = rB.getText().toString();
            Toast.makeText(getApplicationContext(), terkenaData,
Toast.LENGTH_SHORT).show();
            Log.d("kirim", terkenaData);
            if (terkenaData.equals("ya")) {
                tahunjantungET.setVisibility(View.VISIBLE);
            } else {
                tahunjantungET.setVisibility(View.GONE);
            }
        }
    });
    EditText namaET = findViewById(R.id.nameET);
    EditText umurET = findViewById(R.id.umurET);
    EditText nohpET = findViewById(R.id.nohpET);
    EditText nohpkeluargaET = findViewById(R.id.nohpkeLuargaET);

```



```

biodataModel.setStatusMerokok(rokokData);
biodataModel.setKonsumsiAlkohol(alkoholData);
databaseReference.push().setValue(biodataModel);
Toast.makeText(this, "Berhasil menyimpan data",
Toast.LENGTH_SHORT).show();
Intent intent = new Intent(this, MainMenuActivity.class);
startActivity(intent);
} else {
Toast.makeText(this, "Konsumsi Alkohol tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Status Merokok tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Suku tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Agama tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {

Toast.makeText(this, "Penghasilan tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}

} else {
Toast.makeText(this, "Pekerjaan tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Pendidikan Terakhir tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Riwayat penyakit jantung keluarga tidak boleh
kosong!", Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "Pernah terkena penyakit jantung tidak boleh
kosong!", Toast.LENGTH_SHORT).show();
}
} else {
alamatET.setError("Alamat tidak boleh kosong!");
Toast.makeText(this, "Alamat tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
nohpkeluargaET.setError("No Hp keluarga tidak boleh kosong!");
Toast.makeText(this, "No Hp keluarga tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}

```

```

}

} else {
nohpET.setError("No Hp tidak boleh kosong!");
Toast.makeText(this, "No Hp tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
umurET.setError("Umur tidak boleh kosong!");
Toast.makeText(this, "umur tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
Toast.makeText(this, "gender tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
} else {
namaET.setError("Nama Tidak Boleh Kosong!");
Toast.makeText(this, "Nama tidak boleh kosong!",
Toast.LENGTH_SHORT).show();
}
});
}
}
}

```

1.2. ImtActivity.java

```

public class ImtActivity extends AppCompatActivity {
    TextInputEditText height, weight;
    TextView result;
    Button calculate, history;
    LinearLayout view;
    String email, userId;
    private CardView imtHolder;
    private ImtModel imtModel;
    private DatabaseReference databaseReference;
    // private FirebaseDatabase firebaseDatabase;
    // TODO: Bikin cabang tiap user
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_imt);
        email =
SharedPreferencesManager.getStringPreferences(getApplicationContext()
, "user_email");
        userId =
SharedPreferencesManager.getStringPreferences(getApplicationContext()
, "user_id");
        height = findViewById(R.id.height);
        weight = findViewById(R.id.weight);
        result = findViewById(R.id.dataHasilIMT);
        calculate = findViewById(R.id.calculate);
        history = findViewById(R.id.historyBtn);
        view = findViewById(R.id.L_Hasil);
    }
}

```



```

        imtHolder = findViewById(R.id.hasilIMT);
        imtModel = new ImtModel();
        databaseReference =
FirebaseDatabase.getInstance().getReference("imt");
        calculate.setOnClickListener(view -> calculateBMI());
        history.setOnClickListener(view -> openHistory());
    }
    private void calculateBMI(){
        String heightStr =
Objects.requireNonNull(height.getText()).toString();
        String weightStr =
Objects.requireNonNull(weight.getText()).toString();
        if (heightStr != null && !"".equals(heightStr) && weightStr
!= null && !"".equals(weightStr)){
            float heightValue = Float.parseFloat(heightStr) / 100;
            float weightValue = Float.parseFloat(weightStr);
            float bmi = weightValue / (heightValue * heightValue);
            imtModel.setTinggi_badan(heightStr);
            imtModel.setBerat_badan(weightStr);
            displayBMI(bmi);
            imtHolder.setVisibility(View.VISIBLE);
        } else {
            imtHolder.setVisibility(View.GONE);
        }
    }
    private void displayBMI(float bmi){
        String bmiLabel = "";
        if (Float.compare(bmi, 18.5f) <= 0) {
            bmiLabel = "Berat Badan Kurang\n";
        } else if (Float.compare(bmi, 18.5f) > 0 &&
Float.compare(bmi, 22.9f) <= 0) {
            bmiLabel = "Normal\n";
        } else if (Float.compare(bmi, 22.9f) > 0 &&
Float.compare(bmi, 23f) <= 0) {
            bmiLabel = "Berat Badan Lebih\n";
        } else if (Float.compare(bmi, 23f) > 0 &&
Float.compare(bmi,24.9f) <= 0) {
            bmiLabel = "Beresiko\n";
        } else if (Float.compare(bmi, 25f) > 0 && Float.compare(bmi,
29.9f) <= 0) {
            bmiLabel = "Obesitas I\n";
        } else {
            bmiLabel = "Obesitas II\n";
        }
        bmiLabel = bmi + "\n\n" + bmiLabel;
        result.setText(bmiLabel);
        imtModel.setHasil(bmiLabel);
        imtModel.setEmail(email);
        imtModel.setTanggal_input(System.currentTimeMillis() * -1);
        // Submit ke salah satu cabang dengan e-mail sebagai pembeda
        DatabaseReference child = databaseReference.child(userId);
        child.push().setValue(imtModel);
        // databaseReference.push().setValue(imtModel);
        Toast.makeText(this,"Berhasil Menyimpan

```

```

Data", Toast.LENGTH_SHORT).show();
    }
    private void openHistory() {
        Intent historyIntent = new Intent(this,
RiwayatIMTActivity.class);
        startActivity(historyIntent);
    }
}

```

1.3. FragmentJenisPenyakit.java

```

public class FragmentJenisPenyakit extends Fragment {
    CheckBox cbTidak, cbUrat, cbAsma, cbDiabet, cbGinjal, cbHiper,
cbKolesterol, cbStroke, cbLainnya;
    EditText etLainnya;
    Button btnKembali, btnSimpan;
    LinearLayout linerHasil;
    private RiwayatPenyakitModel riwayatPenyakitModel;
    private DatabaseReference databaseReference;
    private FirebaseDatabase firebaseDatabase;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_jenis_penyakit,
container, false);
    }
    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        riwayatPenyakitModel=new RiwayatPenyakitModel();
        firebaseDatabase=FirebaseDatabase.getInstance();
        databaseReference=firebaseDatabase.getReference("riwayat_penyakit");
        cbTidak = view.findViewById(R.id.cb_tidak);
        cbUrat = view.findViewById(R.id.cb_urat);
        cbAsma = view.findViewById(R.id.cb_asma);
        cbDiabet = view.findViewById(R.id.cb_diabetes);
        cbGinjal = view.findViewById(R.id.cb_ginjal);
        cbHiper = view.findViewById(R.id.cb_hipertensi);
        cbKolesterol = view.findViewById(R.id.cb_kolestrol);
        cbStroke = view.findViewById(R.id.cb_stroke);
        cbLainnya = view.findViewById(R.id.cb_Lainnya);
        etLainnya = view.findViewById(R.id.etLainnya);
        btnSimpan = view.findViewById(R.id.btn_rp_oke);
        linerHasil = view.findViewById(R.id.L_HasilPenyakit);
        cbTidak.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton
compoundButton, boolean b) {
                if (cbTidak.isChecked()) {
                    cbUrat.setEnabled(false);

```

```

        cbUrat.setChecked(false);
        cbAsma.setEnabled(false);
        cbAsma.setChecked(false);
        cbDiabet.setEnabled(false);
        cbDiabet.setChecked(false);
        cbGinjal.setEnabled(false);
        cbGinjal.setChecked(false);
        cbHiper.setEnabled(false);
        cbHiper.setChecked(false);
        cbKolesterol.setEnabled(false);
        cbKolesterol.setChecked(false);
        cbStroke.setEnabled(false);
        cbStroke.setChecked(false);
        cbLainnya.setEnabled(false);
        cbLainnya.setChecked(false);
    } else {
        cbUrat.setEnabled(true);
        cbAsma.setEnabled(true);
        cbDiabet.setEnabled(true);
        cbGinjal.setEnabled(true);
        cbHiper.setEnabled(true);
        cbKolesterol.setEnabled(true);
        cbStroke.setEnabled(true);
        cbLainnya.setEnabled(true);
    }
    }
});

        cbLainnya.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton
compoundButton, boolean b) {
        if (cbLainnya.isChecked()) {
            etLainnya.setVisibility(View.VISIBLE);
        } else {
            etLainnya.setVisibility(View.GONE);
        }
    }
});
btnSimpan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        StringBuilder daftarPenyakit = new StringBuilder();
        if (cbTidak.isChecked()) {
            daftarPenyakit.append("tidak ada\n");
        }
        if (cbUrat.isChecked()) {
            daftarPenyakit.append("asam urat\n");
        }
        if (cbAsma.isChecked()) {
            daftarPenyakit.append("asma\n");
        }
        if (cbDiabet.isChecked()) {

```

```

        daftarPenyakit.append("diabetes mellitus\n");
    }
    if (cbGinjal.isChecked()) {
        daftarPenyakit.append("gagal ginjal\n");
    }
    if (cbHiper.isChecked()) {
        daftarPenyakit.append("hipertensi\n");
    }
    if (cbKolesterol.isChecked()) {
        daftarPenyakit.append("kolestrol\n");
    }
    if (cbStroke.isChecked()) {
        daftarPenyakit.append("stroke\n");
    }
    if (cbLainnya.isChecked()) {
daftarPenyakit.append(etLainnya.getText().toString());
    }
    String email=
SharedPreferences.getStringPreferences(getContext(),"user_email");
riwayatPenyakitModel.setEmail(email);
riwayatPenyakitModel.setHasil(daftarPenyakit.toString());
Toast.makeText(getContext(), "Berhasil Menyimpan
Data",Toast.LENGTH_SHORT).show();
databaseReference.push().setValue(riwayatPenyakitModel);
RiwayatPenyakitHomeFragment fragment = new
RiwayatPenyakitHomeFragment();
Bundle mBundle = new Bundle();
mBundle.putString(RiwayatPenyakitHomeFragment.LIST_PENYAKIT,
String.valueOf(daftarPenyakit));
    int isLinear = 1;
    fragment.setArguments(mBundle);
    fragment.setIsLinear(isLinear);
    FragmentManager fragmentManager =
getFragmentManager();
    if (fragmentManager != null) {
        fragmentManager
            .beginTransaction()
            .replace(R.id.rl_riwayat_penyakit,
fragment, RiwayatPenyakitHomeFragment.class.getSimpleName())
            .commit();
    }
    });
}
}
}

```

1.4. AktifitasFisikActivity.java

```

public class AktifitasFisikActivity extends AppCompatActivity
implements SingleChoiceDialogFragment.SingleChoiceListener {
    public static String[] list;
    private TextView tvDurasi;

```

```

private TextView tvLamaOlahraga;
private TextView tvHasil;
LinearLayout view;
CardView cardHasil;
String userId;
Button btnSelectChoicetwo;
Button btnHasil;
int durasiOlahraga=0;
int lamaOlahraga=0;
private PolaHidupModel polaHidupModel;
private DatabaseReference databaseReference;
private FirebaseDatabase firebaseDatabase;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_aktifitas_fisik);
    userId =
SharedPreferencesManager.getStringPreferences(getApplicationContext()
, "user_id");
    tvHasil = findViewById(R.id.tvHasil);
    view = findViewById(R.id.layout_Hasil);
    cardHasil = findViewById(R.id.hasildurasi);
    polaHidupModel=new PolaHidupModel();
    firebaseDatabase=FirebaseDatabase.getInstance();
    databaseReference=firebaseDatabase.getReference("aktivitas_fisik");
    Button btnSelectChoice = findViewById(R.id.btnSelectChoice);
    btnSelectChoicetwo = findViewById(R.id.btnSelectChoicetwo);
    btnHasil = findViewById(R.id.btnHasil);
    btnSelectChoicetwo.setEnabled(false);
    btnHasil.setEnabled(false);
    btnSelectChoice.setOnClickListener(view -> {
        DialogFragment singleChoiceDialog = new
SingleChoiceDialogFragment();
        singleChoiceDialog.setCancelable(true);
        singleChoiceDialog.show(getSupportFragmentManager(),
"Choice Dialog");
    });
    btnSelectChoicetwo.setOnClickListener(view -> {
        DialogFragment myDialog = new DuoChoiceDialogFragment();
        myDialog.setCancelable(true);
        myDialog.show(getSupportFragmentManager(), "Choice
Dialog");
    });
    btnHasil.setOnClickListener(view -> {
        int hasil = durasiOlahraga * lamaOlahraga;
        String hasilRumus = "";
        if (hasil == 0) {
            hasilRumus = "Tidak Berolahraga";
            lamaOlahraga = 0;
        } else if (hasil > 0 && hasil <= 120) {
            hasilRumus = "Kurang dari 120 menit = anda kurang
aktif";
        } else if (hasil > 120 && hasil < 200) {
            hasilRumus = "120-200 menit = anda aktif";
        }
    });
}

```

```

        } else if (hasil >= 200) {
            hasilRumus = "Lebih dari 200 menit = anda sangat
aktif";
        }
        tvHasil.setText(hasilRumus);
        polaHidupModel.setHasil(hasilRumus);
        polaHidupModel.setHasilDalamAngka(hasil);

        polaHidupModel.setTanggal_input(System.currentTimeMillis() * -1);
        String email=
        SharedPreferences.getStringPreferences(this,"user_email");
        polaHidupModel.setEmail(email);
        DatabaseReference childAktivitasFisik =
        databaseReference.child(userId);
        childAktivitasFisik.push().setValue(polaHidupModel);
        Toast.makeText(this,"Berhasil Menyimpan
Data",Toast.LENGTH_SHORT).show();
        cardHasil.setVisibility(View.VISIBLE);
    });
    Button btnHistory = findViewById(R.id.historyBtnAktivitas);
    btnHistory.setOnClickListener(view1 -> {
        Intent intent = new Intent(this,
RiwayatAktivitasActivity.class);
        startActivity(intent);
    });
}

@Override
public void onPositiveButtonClicked(String[] list, int position,
boolean isSingleChoice) {
    if (isSingleChoice) {
        polaHidupModel.setDurasi_olahraga(list[position]);
        if (position==0) {
            durasiOlahraga = 0;
            btnSelectChoicetwo.setEnabled(false);
            btnHasil.setEnabled(true);
        } else if (position==1) {
            durasiOlahraga = 1;
            btnSelectChoicetwo.setEnabled(true);
            btnHasil.setEnabled(true);
        } else if (position==2) {
            durasiOlahraga = 3;
            btnSelectChoicetwo.setEnabled(true);
            btnHasil.setEnabled(true);
        } else if (position == 4) {
            durasiOlahraga = 4;
            btnSelectChoicetwo.setEnabled(true);
            btnHasil.setEnabled(true);
        }
    }
} else {
    // tvLamaOlahraga.setText("Lama = " + list[position]);
    polaHidupModel.setLama_olahraga(list[position]);
    if (position==0) {
        lamaOlahraga = 30;
    }
}

```

```

        } else if (position==1) {
            lamaOlahraga = 45;
        } else if (position==2) {
            lamaOlahraga = 50;
        } else {
            lamaOlahraga = 0;
        }
    }
}
@Override
public void onNegativeButtonClicked() {
    tvDurasi.setText("Select Item");
}
}

```

1.5. JenisMakananActivity.java

```

public class JenisMakananActivity extends AppCompatActivity {
    ArrayAdapter<String> adapter;
    SearchView searchView;
    private List<ModelMakanan> listMakanan = new ArrayList<>();
    private Map<String, String> mapJenis = new LinkedHashMap<>();
    private Map<String, Integer> mapJumlah = new LinkedHashMap<>();
    RecyclerView listMakananRV;
    Button btnSimpan, btnHistory;
    int totalQuantity;
    private DatabaseReference databaseReference;
    private FirebaseDatabase firebaseDatabase;
    private Handler mHandler;
    String userId;
    ProgressDialog progressDialog;
    Boolean isSearching = false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_jenismakanan);
        userId =
SharedPreferencesManager.getStringPreferences(getApplicationContext()
, "user_id");
        mHandler = new Handler();
        listMakananRV = findViewById(R.id.makananRV);
        btnSimpan = findViewById(R.id.btnSimpan);
        btnHistory = findViewById(R.id.btnRiwayatJenisMakanan);
        // tambahkanDataMakanan();
        searchView = findViewById(R.id.titleSearch);
        firebaseDatabase = FirebaseDatabase.getInstance();
        databaseReference =
firebaseDatabase.getReference("pola_diet");
        btnSimpan.setOnClickListener(view -> {
            simpanData();
        });
        btnHistory.setOnClickListener(view -> {
            Intent intent = new Intent(this,

```

```

RiwayatPolaDietActivity.class);
        startActivity(intent);
    });
    searchView.setOnQueryTextListener(new
    SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            String searchQuery = query.trim();
            // Log.d("JenisMakananActivity", "Query: " +
searchQuery);
            // Log.d("JenisMakananActivity", "Query length: " +
searchQuery.length());
            if (searchQuery.length() > 0 && !isSearching) {
                tutupKeyboard();
                searchMakanan(query);
            } /*else {
                Log.d("JenisMakananActivity", "Search is
empty!");
                Toast.makeText(context, "Silahkan mengetik
pencarian anda.", Toast.LENGTH_SHORT).show();
            }*/
            return false;
        }
        @Override
        public boolean onQueryTextChange(String query) {
            String searchQuery = query.trim();
            mHandler.removeCallbacksAndMessages(null);
            mHandler.postDelayed(() -> {
                if (!isSearching) {
                    tutupKeyboard();
                    if (searchQuery.length() > 0) {
                        searchMakanan(query);
                    } else {
                        getAllMakanan();
                    }
                }
            }, 1000);
            return false;
        }
    });
    getAllMakanan();
}
void tambahkanDataMakanan() {
    ModelMakanan makanan1 = new ModelMakanan();
    makanan1.setNamaMakanan("Nasi");
    makanan1.setJenisMakanan("Karbohidrat");
    listMakanan.add(makanan1);
    ModelMakanan makanan2 = new ModelMakanan();
    makanan2.setNamaMakanan("Daging");
    makanan2.setJenisMakanan("Protein");
    listMakanan.add(makanan2);
    ModelMakanan makanan3 = new ModelMakanan();
    makanan3.setNamaMakanan("Ikan Goreng");
    makanan3.setJenisMakanan("Protein");
}

```



```

        listMakanan.add(makanan3);
        setRecyclerView(listMakanan);
    }
    void setRecyclerView(List<ModelMakanan> listMakanan) {
        MakananAdapter makananAdapter = new MakananAdapter(
            this,
            listMakanan,
            (nama, jenis) -> {
                if (mapJenis == null) {
                    mapJenis = new LinkedHashMap<>();
                }
                mapJenis.put(nama, jenis);
            },
            (nama, jumlah) -> {
                if (mapJumlah == null) {
                    mapJumlah = new LinkedHashMap<>();
                }
                mapJumlah.put(nama, jumlah);
            },
            mapJenis,
            mapJumlah,
            totalQuantity
        );
        LinearLayoutManager layoutManager = new
        LinearLayoutManager(JenisMakananActivity.this);
        listMakananRV.setLayoutManager(layoutManager);
        listMakananRV.setItemViewCacheSize(listMakanan.size());
        listMakananRV.setAdapter(makananAdapter);
    }
    void simpanData() {
        List<ModelMakanan> listMakanan = new ArrayList<>();
        for (Map.Entry<String, String> entry : mapJenis.entrySet())
        {
            ModelMakanan modelMakanan = new ModelMakanan();
            modelMakanan.setNamaMakanan(entry.getKey());
            modelMakanan.setJenisMakanan(entry.getValue());
            for (Map.Entry<String, Integer> quantityEntry :
            mapJumlah.entrySet()) {
                if (quantityEntry.getKey().equals(entry.getKey())) {
                    modelMakanan.setJumlahMakanan(quantityEntry.getValue());
                }
            }
            if (modelMakanan.getJumlahMakanan() > 0) {
                listMakanan.add(modelMakanan);
            }
        }
        PolaDiet polaDiet = new PolaDiet();
        String email =
        SharedPreferences.getStringPreferences(this, "user_email");
        polaDiet.setEmail(email);
        polaDiet.setModelMakanan(listMakanan);
        polaDiet.setTanggal_input(System.currentTimeMillis() * -1);
        DatabaseReference childJenisMakanan =
        databaseReference.child(userId);
    }

```

```

        childJenisMakanan.push().setValue(polaDiet);
        Toast.makeText(this, "Berhasil Menyimpan Data",
Toast.LENGTH_SHORT).show();
        onBackPressed();
    }
    public void getAllMakanan() {
        progressDialog = new
ProgressDialog(JenisMakananActivity.this);
        progressDialog.setTitle("Tunggu Sebentar");
        progressDialog.setMessage("Data sedang diproses");
        progressDialog.setCancelable(true);
        progressDialog.show();
        ApiInterface api = new ApiConfig().instance();
        Call<ArrayList<ModelMakanan>> call = api.getListMakanan();
        call.enqueue(new Callback<ArrayList<ModelMakanan>>() {
            @Override
            public void onResponse(Call<ArrayList<ModelMakanan>>
call, Response<ArrayList<ModelMakanan>> response) {
                // pasang data dan matikan progress dialog kalau
sudah ada responnya server
                setRecyclerView(response.body());
                progressDialog.dismiss();
            }
            @Override
            public void onFailure(Call<ArrayList<ModelMakanan>>
call, Throwable t) {
                // kalo ada error ini yang terpanggil
                Toast.makeText(JenisMakananActivity.this,
t.toString(), Toast.LENGTH_LONG).show();
                progressDialog.dismiss();
            }
        });
    }
    private void searchMakanan(String namaMakanan) {
        isSearching = true;
        progressDialog = new
ProgressDialog(JenisMakananActivity.this);
        progressDialog.setTitle("Tunggu Sebentar");
        progressDialog.setMessage("Pencarian sedang dilakukan..");
        progressDialog.setCancelable(true);
        progressDialog.show();
        ApiInterface api = new ApiConfig().instance();
        Call<ArrayList<ModelMakanan>> call =
api.searchMakanan(namaMakanan);
        call.enqueue(new Callback<ArrayList<ModelMakanan>>() {
            @Override
            public void onResponse(Call<ArrayList<ModelMakanan>>
call, Response<ArrayList<ModelMakanan>> response) {
                // pasang data dan matikan progress dialog kalau
sudah ada responnya server
                setRecyclerView(response.body());
                progressDialog.dismiss();
                isSearching = false;
            }
        });
    }
}

```

```

        @Override
        public void onFailure(Call<ArrayList<ModelMakanan>>
call, Throwable t) {
            // kalo ada error ini yang terpanggil
            Toast.makeText(JenisMakananActivity.this,
t.toString(), Toast.LENGTH_LONG).show();
            progressDialog.dismiss();
            isSearching = false;
        }
    });
}

private void tutupKeyboard() {
    View view = this.getCurrentFocus();
    if (view != null) {
        InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
    }
}
//override = onCreateOptionsMenu(..)-- is to create a menu
}

```

1.6. HasilActivity.java

```

public class HasilActivity extends AppCompatActivity {
    String email, userId;
    TextView biodata;
    TextView imt;
    TextView datapenyakit;
    TextView dataHasilaktivitas;
    TextView dataPolaDiet;
    TextView dataHasilRokok;
    TextView dataHasilAlkohol;
    TextView dataHasilIMT;
    TextView dataHasilKeluarga;
    TextView dataHasilHipertensi;
    TextView dataHasilDiabetes;
    TextView dataHasilDiet;
    private Context mContext;
    private Activity mActivity;
    private RelativeLayout mRelativeLayout;
    private Button mButton;
    private PopupWindow mPopupWindow;
    BiodataModel biodataModel;
    ImtModel imtModel;
    RiwayatPenyakitModel riwayatPenyakitModel;
    PolaHidupModel polaHidupModel;
    List<ModelMakanan> polaDiet;
    String dataHasilAnalisis;
    ArrayList<String> diagnosa = new ArrayList<>();
    ArrayList<String> penyakit = new ArrayList<>();
    boolean obesitas;
}

```

```

boolean merokok;
boolean alkohol;
boolean riwayat_jantung_keluarga;
boolean hipertensi;
boolean diabetes;
boolean aktivitas;
boolean jenisMakanan;
boolean sudahIsiData;
boolean isDiet = false;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hasil);
    mContext = getApplicationContext();
    mActivity = HasilActivity.this;
    mRelativeLayout = (RelativeLayout) findViewById(R.id.rl);
    mButton = (Button) findViewById(R.id.btn);
    //tambahkan semua jenis penyakit ke list penyakit
    penyakit.add("obesitas");
    penyakit.add("merokok");
    penyakit.add("alkohol");
    penyakit.add("riwayat_jantung_keluarga");
    penyakit.add("hipertensi");
    penyakit.add("diabetes");
    penyakit.add("aktivitas");
    penyakit.add("jenisMakanan");
    email = SharedPreferences.getStringPreferences(this,
"user_email");
    userId =
SharedPreferences.getStringPreferences(getApplicationContext()
, "user_id");
    biodata = findViewById(R.id.dataTV);
    dataHasilaktivitas = findViewById(R.id.dataAktivitasFisik);
    dataHasilRokok = findViewById(R.id.dataHasilRokok);
    dataHasilAlkohol = findViewById(R.id.dataHasilAlkohol);
    dataHasilIMT = findViewById(R.id.dataHasilIMT);
    dataHasilKeluarga = findViewById(R.id.dataHasilKeluarga);
    dataHasilHipertensi =
findViewById(R.id.dataHasilHipertensi);
    dataHasilDiabetes = findViewById(R.id.dataHasilDiabetes);
    dataHasilDiet = findViewById(R.id.dataHasilDiet);
    DatabaseReference databaseBiodata =
FirebaseDatabase.getInstance().getReference("user");
    databaseBiodata.limitToLast(1).addChildEventListener(new
ChildEventListener() {
        @Override
        public void onChildAdded(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
            BiodataModel model =
snapshot.getValue(BiodataModel.class);
            if (model.getEmail().equals(email)) {
                sudahIsiData = true;
                biodataModel = model;
                setBiodata();
            }
        }
    });
}

```

```

        if
(biodataModel.getStatusMerokok().toLowerCase().equals("perokok")) {
            //merokok = true;
            diagnosa.add("merokok");
            dataHasilRokok.append("Wah, Anda perokok
aktif yah!\n\nHati-hati itu beresiko penyakit jantung\n\nKURANGI
KEBIASAAN MEROKOKNYA YAH...:\n\n");
        } else if
(biodataModel.getStatusMerokok().toLowerCase().equals("bekas
perokok")) {
            dataHasilRokok.append("Meskipun anda seorang
perokok pasif, tetap berhati-hati karena dapat beresiko penyakit
jantung loh\n\nkurangi bertemu dengan perokok aktif atau gunakan
masker\n\n");
        } else {
            dataHasilRokok.append("HEBAT, Anda bukan
seorang perokok\n\nPertahankan Yah...:\n\n");
        }
        if
(biodataModel.getKonsumsiAlkohol().toLowerCase().equals("ya")) {
            alkohol = true;
            diagnosa.add("alkohol");
            dataHasilAlkohol.append("Kurangi atau
berhentilah mengkonsumsi alkohol, karena itu beresiko terkena
penyakit jantung\n\nSEMANGAT\n\n");
        } else {
            dataHasilAlkohol.append("KEREN...\n\nPertahankan untuk tidak
mengkonsumsi Alkohol yah...\n\n");
        }
        if
(biodataModel.getRiwayat_jantung_keluarga().toLowerCase().equals("ya
")) {
            riwayat_jantung_keluarga = true;
            diagnosa.add("riwayat_jantung_keluarga");
            dataHasilKeluarga.append("jaga kesehatan dan
pola makan yah...\n\nanda memiliki riwayat penyakit jantung keluarga
loh... dan itu beresiko terkena penyakit jantung\n\n");
        } else {
            dataHasilKeluarga.append("Anda tidak
memiliki riwayat penyakit jantung keluarga\n\nTAPI...\ntetap harus
jaga kesehatan yah...\n\n");
        }
    }
}
@Override
public void onChildChanged(@NonNull DataSnapshot
snapshot, @Nullable String previousChildName) {
}
@Override
public void onChildRemoved(@NonNull DataSnapshot
snapshot) {
}
@Override
public void onChildMoved(@NonNull DataSnapshot snapshot,

```

```

@Nullable String previousChildName) {
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
DatabaseReference databaseIMT =
FirebaseDatabase.getInstance().getReference("imt").child(userId);
databaseIMT.limitToLast(1).addChildEventListener(new
ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
        if (snapshot.exists()) {
            ImtModel model =
snapshot.getValue(ImtModel.class);
            if (model.getEmail().equals(email)) {
                sudahIsiData = true;
                imtModel = model;
                setDataIMT();
                if
(imtModel.getHasil().toLowerCase().contains("Obesitas
II".toLowerCase())) {
                    //obesitas = true;
                    diagnosa.add("obesitas");
                    dataHasilIMT.append("Berat badan anda
sangat berlebihan, ubah diet anda dengan pola gizi seimbang\n\n");
                } else if
(imtModel.getHasil().toLowerCase().contains("Obesitas
I".toLowerCase())) {
                    //obesitas = true;
                    diagnosa.add("obesitas");
                    dataHasilIMT.append("Berat badan anda
berlebihan, ubah diet anda dengan pola gizi seimbang\n\n");
                } else if
(imtModel.getHasil().toLowerCase().contains("Beresiko".toLowerCase())
)) {
                    dataHasilIMT.append("Berat badan anda
beresiko, tetap jaga pola diet yah...\n\n");
                } else if
(imtModel.getHasil().toLowerCase().contains("Berat Badan
Lebih".toLowerCase())) {
                    dataHasilIMT.append("Berat badan anda
Berlebih, tetap jaga pola diet yah...\n\n");
                } else if
(imtModel.getHasil().toLowerCase().contains("Normal".toLowerCase()))
{
                    dataHasilIMT.append("Berat badan anda
normal, pertahankan yah...\n\n");
                } else {
                    dataHasilIMT.append("Berat badan anda
kurang \n\n");
                }
            }
        }
    }
}

```

```

        }
    }
    @Override
    public void onChildChanged(@NonNull DataSnapshot
snapshot, @Nullable String previousChildName) {
    }
    @Override
    public void onChildRemoved(@NonNull DataSnapshot
snapshot) {
    }
    @Override
    public void onChildMoved(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
DatabaseReference databaseRiwayatPenyakit =
FirebaseDatabase.getInstance().getReference("riwayat_penyakit");

databaseRiwayatPenyakit.limitToLast(1).addChildEventListener(new
ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
        RiwayatPenyakitModel model =
snapshot.getValue(RiwayatPenyakitModel.class);
        if (model.getEmail().equals(email)) {
            sudahIsiData = true;
            riwayatPenyakitModel = model;
            setRiwayatPenyakit();
            String[] listPenyakit =
getResources().getStringArray(R.array.jenis_penyakit);
            int jumlahPenyakitYangMemicu = 0;
            for (String penyakit : listPenyakit) {
                if
(riwayatPenyakitModel.getHasil().toLowerCase().contains(penyakit.toL
owerCase())) {
                    jumlahPenyakitYangMemicu++;
                }
            }
            if
(riwayatPenyakitModel.getHasil().toLowerCase().contains("hipertensi"
)) {
                hipertensi = true;
                diagnosa.add("hipertensi");
            }
            if
(riwayatPenyakitModel.getHasil().toLowerCase().contains("diabetes
mellitus")) {
                diabetes = true;
                diagnosa.add("diabetes");
            }
        }
    }
});

```

```

    }
    if (hipertensi) {
        dataHasilHipertensi.append("Anda memiliki
riwayat penyakit hipertensi dan itu beresiko penyakit
jantung\n\nJagalah tekanan darah anda dalam batas normal <120/80
mmHg!!!\n\n");
    } else {
        dataHasilHipertensi.append("Anda tidak
memiliki riwayat penyakit hipertensi :)\n\nTetap jaga kesehatan
yah...\n\n");
    }

    if (diabetes) {
        dataHasilDiabetes.append("Anda memiliki
riwayat penyakit diabetes mellitus dan itu beresiko penyakit
jantung\n\nTetap Perhatikan kadar gula anda dengan : \n- perhatikan
pola makan\n- pola aktifitas fisik, dan\nJANGAN LUPA KONSULTASI KE
DOKTER!\n\n");
    } else {
        dataHasilDiabetes.append("Anda tidak
memiliki riwayat penyakit diabetes mellitus :)\n\nTetap jaga
kesehatan yah...\n\n");
    }
}
}
@Override
public void onChildChanged(@NonNull DataSnapshot
snapshot, @Nullable String previousChildName) {
}
@Override
public void onChildRemoved(@NonNull DataSnapshot
snapshot) {
}
@Override
public void onChildMoved(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
DatabaseReference databaseAktivitasFisik =
FirebaseDatabase.getInstance().getReference("aktivitas_fisik").child
(userId);

databaseAktivitasFisik.limitToLast(1).addChildEventListener(new
ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
        PolaHidupModel model =
snapshot.getValue(PolaHidupModel.class);
        if (model.getEmail().equals(email)) {
            sudahIsiData = true;

```



```

        polaHidupModel = model;
        setAktivitasFisik();
        // Tampil di hasil analisis
        int hasil = model.getHasilDalamAngka();
        if (hasil <= 120) {
            //aktivitas = true;
            diagnosa.add("aktivitas");
            dataHasilaktivitas.append("Anda kurang aktif
karena bergerak kurang dari 120 menit\n\n Naikkan Aktivitas fisik
anda untuk mencegah resiko terjadinya penyakit jantung\n\n");
        } else if (hasil < 200) {
            dataHasilaktivitas.append("Selamat anda
tergolong aktif\n\nTetap bergerak aktif yah..\nUntuk mencegah resiko
terjadinya penyakit jantung\n\n");
        } else if (hasil >= 200) {
            dataHasilaktivitas.append("Wah.. Selamat
anda bergerak sangat aktif\n\nJaga kebugaran agar tidak cepat lelah
yah..\n\n");
        }
    }
}
@Override
public void onChildChanged(@NonNull DataSnapshot
snapshot, @Nullable String previousChildName) {
}
@Override
public void onChildRemoved(@NonNull DataSnapshot
snapshot) {
}
@Override
public void onChildMoved(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});

DatabaseReference databasePolaDiet =
FirebaseDatabase.getInstance().getReference("pola_diet").child(userI
d);
databasePolaDiet.limitToLast(1).addChildEventListener(new
ChildEventListener() {
@Override
public void onChildAdded(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
PolaDiet model = snapshot.getValue(PolaDiet.class);
if (model.getEmail().equals(email)) {
sudahIsiData = true;
polaDiet = model.getModelMakanan();
setPolaDiet();
//jika salah satu makanan lebih dari 3, maka
langsung beresiko

```

```

        for (ModelMakanan makanan : polaDiet) {
            if (makanan.getJumlahMakanan() >= 3) {
                isDiet = true;
                Log.i("isDiet", "TRUE");
                jenisMakanan = true;
                diagnosa.add("jenisMakanan");
            } else {
                Log.i("isDiet", "FALSE");
            }
        }
        stateHasilDiet(isDiet);
    }
}
@Override
public void onChildChanged(@NonNull DataSnapshot
snapshot, @Nullable String previousChildName) {
}
@Override
public void onChildRemoved(@NonNull DataSnapshot
snapshot) {
}
@Override
public void onChildMoved(@NonNull DataSnapshot snapshot,
@Nullable String previousChildName) {
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
//button popup window
mButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
    LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(LAYOUT_INFLATER_SERVICE);

    View customView =
inflater.inflate(R.layout.popup_window, null);

    mPopupWindow = new PopupWindow(
        customView,
RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT
    );
    if (Build.VERSION.SDK_INT >= 21) {
        mPopupWindow.setElevation(5.0f);
    }
    TextView dataAnalysis =
customView.findViewById(R.id.tv);
    dataAnalysis.setText(dataHasilAnalysis);
    ImageButton closeButton = (ImageButton)
customView.findViewById(R.id.ib_close);
    closeButton.setOnClickListener(new
View.OnClickListener() {

```

```

        @Override
        public void onClick(View view) {
            mPopupWindow.dismiss();
        }
    });
    mPopupWindow.showAtLocation(mRelativeLayout,
Gravity.CENTER,0,0);
    }
});
final Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        // Do something after 5s = 5000ms
        //checkKategoriBeresiko();
        checkTingkatResikoBerdasarkanDiagnosa();
    }
}, 3000);
}
void checkTingkatResikoBerdasarkanDiagnosa() {
    int tingkatResiko = 0;
    for (String diag : diagnosa) {
        for (String peny : penyakit) {
            if (diag == peny) {
                if (diag == "jenisMakanan") {
                    tingkatResiko += 3;
                } else {
                    tingkatResiko ++;
                }
            }
        }
    }
    if (sudahIsiData) {
        if (tingkatResiko >= 3) {
            dataHasilAnalisis = "\n\nHASIL ANALISIS:\n\n ANDA
MASUK KATEGORI BERESIKO TERKENA PENYAKIT JANTUNG\n\n";
        } else {
            dataHasilAnalisis = "\n\nHASIL ANALISIS:\n\nANDA
TIDAK BERESIKO TERKENA PENYAKIT JANTUNG\n\n";
        }
    } else {
        dataHasilAnalisis = "\n\nANDA BELUM MENGISI DATA APAPUN,
LENGKAPI DATA TERLEBIH DAHULU\n\n";
    }
}
void checkKategoriBeresiko() {
    int tingkatResiko = 0;
    if (obesitas) {
        tingkatResiko++;
    }
    if (merokok) {
        tingkatResiko++;
    }
}

```

```

        if (alkohol) {
            tingkatResiko++;
        }
        if (riwayat_jantung_keluarga) {
            tingkatResiko++;
        }
        if (hipertensi) {
            tingkatResiko++;
            // dataHasilAnalisis.append("\n\n- Anda memiliki riwayat
penyakit hipertensi sehingga meningkatkan resiko penyakit jantung,
disarankan jaga tekanan darah anda dalam batas normal <120/80
mmHg\n");
        }
        if (diabetes) {
            tingkatResiko++;
        }
        if (aktivitas) {
            tingkatResiko++;
        }
        if (jenisMakanan) {
            tingkatResiko+=3;
        }
        if (sudahIsiData) {
            if (tingkatResiko >= 3) {
                dataHasilAnalisis = "\n\nHASIL ANALISIS:\n\n ANDA
MASUK KATEGORI BERESIKO TERKENA PENYAKIT JANTUNG\n\n";
            } else {
                dataHasilAnalisis = "\n\nHASIL ANALISIS:\n\nANDA
TIDAK BERESIKO TERKENA PENYAKIT JANTUNG\n\n";
            }
        } else {
            dataHasilAnalisis = "\n\nANDA BELUM MENGISI DATA APAPUN,
LENGKAPI DATA TERLEBIH DAHULU\n\n";
        }
    }
    @SuppressWarnings("SetTextI18n")
    void setBiodata() {
    }
    @SuppressWarnings("SetTextI18n")
    void setDataIMT() {
    }
    @SuppressWarnings("SetTextI18n")
    void setRiwayatPenyakit() {
    }
    @SuppressWarnings("SetTextI18n")
    void setAktivitasFisik() {
    }
    @SuppressWarnings("SetTextI18n")
    void setPolaDiet() {
dataPolaDiet.setText(dataDiet.toString());
    }

    void stateHasilDiet(boolean isDiet) {
        if (isDiet) {

```

```
        dataHasildiet.append("Makanan yang anda konsumsi
perminggu berlebih, itu masuk kategori penyakit jantung\n\nTETAP
ATUR POLA MAKAN YAH...\n\n");
    } else {
        dataHasildiet.append("Makanan yang di konsumsi perminggu
Masih Aman\n\n");
    }
}
}
```