

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap pembangunan aplikasi, tahap lanjut dari perancangan sistem. Tahap ini dilakukan untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dimengerti oleh komputer. Adapun pembahasan implementasi terdiri dari lingkungan implementasi, implementasi basis data dan implementasi antarmuka.

4.1.1 Lingkungan Implementasi

Lingkungan implementasi menjelaskan kebutuhan sistem yang disarankan dalam lingkungan implementasi, kebutuhan ini terbagi menjadi dua yaitu spesifikasi kebutuhan *hardware* / perangkat keras dan spesifikasi kebutuhan *software* / perangkat lunak. Berikut adalah lingkungan implementasi dari perangkat lunak *marathon trainer disertai dengan indikator level kebugaran berdasarkan denyut jantung dan suhu tubuh*.

a) Spesifikasi Kebutuhan *Hardware*

Spesifikasi kebutuhan *hardware* / perangkat keras minimum yang digunakan untuk menggunakan sistem adalah sebagai berikut

Tabel 4.1 Spesifikasi kebutuhan *hardware*

Nama perangkat keras	Spesifikasi
Internal Memory	8 GB
Ram	1 GB
Layar	TFT LCD
Kamera	5 Megapixel

b) Spesifikasi Kebutuhan *Software*

Spesifikasi kebutuhan *software* / perangkat lunak minimum yang digunakan untuk menggunakan sistem adalah sebagai berikut

Tabel 4.2 Spesifikasi kebutuhan software

Nama perangkat lunak	Spesifikasi
Sistem Operasi	Android 5.0
Mesin Pencarian	Google Chrome

4.1.2 Implementasi Teknologi

Implementasi Teknologi merupakan implementasi dari analisis teknologi kedalam bentuk aplikasi yang nyata. Adapun implementasi teknologi yang digunakan pada aplikasi yang dibangun adalah sebagai berikut:

4.1.2.1 Implementasi Flash Kamera

Implementasi flash kamera pada sistem yang dibangun pada penelitian ini adalah untuk pendeteksian denyut nadi/jantung. Adapun implementasi flash kamera adalah sebagai berikut

```
private fun startWithPermissionCheck() {
    if (!hasPermission(Manifest.permission.CAMERA)) {
        checkPermissions(REQUEST_CAMERA_PERMISSION,
Manifest.permission.CAMERA)
        return
    }
    val kalman = JKalman(2, 1)
    // measurement [x]
    val m = Matrix(1, 1)
    // transitions for x, dx
    val tr = arrayOf(doubleArrayOf(1.0, 0.0), doubleArrayOf(0.0, 1.0))
    kalman.transition_matrix = Matrix(tr)
    // Is somewhere?
    kalman.error_cov_post = kalman.error_cov_post.identity()
    val bpmUpdates = HeartRateOmeter()
        .withAverageAfterSeconds(3)
        .setFingerDetectionListener(this::onFingerChange)
```

```

.bpmUpdates(preview)
.subscribe({
  if (it.value == 0)
    return@subscribe
  m.set(0, 0, it.value.toDouble())
  // state [x, dx]
  val s = kalman.Predict()
  // corrected state [x, dx]
  val c = kalman.Correct(m)

  val bpm = it.copy(value = c.get(0, 0).toInt())
  Log.v("HeartRateOmeter", "[onBpm] ${it.value} => ${bpm.value}")
  onBpm(bpm)
}, Throwable::printStackTrace)
subscription?.add(bpmUpdates)
}

```

4.1.2.2 Implementasi Sensor Proximity

Implementasi sensor proximity pada sistem yang dibangun pada penelitian ini adalah untuk menghitung gerakan push up. Adapun implementasi sensor proximity adalah sebagai berikut

```

SensorEventListener proximitySensorEventListener
    = new SensorEventListener() {
    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // TODO Auto-generated method stub
    }
    @Override
    public void onSensorChanged(SensorEvent event) {
        // TODO Auto-generated method stub
        if (event.sensor.getType() == Sensor.TYPE_PROXIMITY) {
            if (event.values[0] == 0) {
                mCounter++;
                aq.id(R.id.data).text(Integer.toString(mCounter));
            } else {
                //aq.id(R.id.data).text("Away");
            }
        }
    }
};

```

4.1.2.3 Implementasi API Sosial Media

Implementasi API Sosial Media pada sistem yang dibangun pada penelitian ini adalah untuk pencarian even marathon. Adapun implementasi API Sosial Media adalah sebagai berikut

Intent	intent	=
<pre>getPackageManager().getLaunchIntentForPackage("com.instagram.android"); if (intent != null) { Intent shareIntent = new Intent(); shareIntent.setAction(Intent.ACTION_SEND); shareIntent.setPackage("com.instagram.android"); try { shareIntent.putExtra(Intent.EXTRA_STREAM, Uri.parse(MediaStore.Images.Media.insertImage(getContentResolver(), mCurrentPhotoPath, "I am Happy", "Share happy !"))); } catch (FileNotFoundException e) { // TODO Auto-generated catch block e.printStackTrace(); } shareIntent.setType("image/jpeg"); startActivity(shareIntent); } else { // bring user to the market to download the app. // or let them choose an app? intent = new Intent(Intent.ACTION_VIEW); intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK); intent.setData(Uri.parse("market://details?id="+ "com.instagram.android")); startActivity(intent); }</pre>		

4.1.3 Implementasi Basis Data

Implementasi basis data menjelaskan tahap pembuatan basis data dan tabel-tabel yang dibutuhkan dalam aplikasi yang sedang dibangun. Implementasi basis data yang dilakukan adalah sebagai berikut

a) Pembuatan Basis Data

Untuk Source code implementasi pembuatan basis data adalah sebagai berikut

```
Create database marathoner_db
```

b) Pembuatan Tabel counter

Tabel counter digunakan untuk menyimpan data counter push up yang digunakan pengguna untuk menghitung gerakan push up yang akan dilakukan. Untuk pembuatan tabel counter adalah sebagai berikut

```
CREATE TABLE `counter` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `jumlah` int(11) NOT NULL,
  `tanggal` date NOT NULL,
  `waktu` time NOT NULL,
  `email` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  CONSTRAINT `counter_ibfk_1` FOREIGN KEY (`email`) REFERENCES
  `pengguna` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

c) Pembuatan Tabel kebugaran

Tabel kebugaran digunakan untuk menyimpan data kebugaran yang berisi indikator level kebugaran para pengguna. Untuk pembuatan tabel kebugaran adalah sebagai berikut

```

CREATE TABLE `kebugaran` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `denyut_jantung` double NOT NULL,
  `suhu` double NOT NULL,
  `email` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  CONSTRAINT `kebugaran_ibfk_1` FOREIGN KEY (`email`)
REFERENCES `pengguna` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

d) Pembuatan Tabel log_lari

Tabel log_lari digunakan untuk menyimpan data log lari dari masing masing pengguna. Untuk pembuatan tabel log_lari adalah sebagai berikut

```

CREATE TABLE `log_lari` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `jarak` double NOT NULL,
  `tanggal` date NOT NULL,
  `waktu` time NOT NULL,
  `email` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  CONSTRAINT `log_lari_ibfk_1` FOREIGN KEY (`email`) REFERENCES
`pengguna` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

e) Pembuatan Tabel pengguna

Tabel pengguna digunakan untuk menyimpan data para pengguna. Untuk pembuatan tabel pengguna adalah sebagai berikut

```
CREATE TABLE `pengguna` (
  `id_pengguna` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(50) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `nama_toko` varchar(200) NOT NULL,
  `telepon` int(11) NOT NULL,
  `password` varchar(50) NOT NULL,
  PRIMARY KEY (`id_pengguna`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.1.4 Implementasi Class

Implementasi class berisi pemaparan setiap class dari fungsionalitas yang dibangun. Adapun implementasi class perangkat lunak yang dibangun terdiri dari nama class atau file yang mewakilinya.

Tabel 4.3 Implementasi Class

Menu	Deskripsi	Nama File
Login	Merupakan class untuk login	MasukActivity.java
Registrasi	Merupakan class untuk registrasi akun baru	DaftarActivity.java
Lupa Password	Merupakan class untuk melakukan lupa password	LupaActivity.java
Mencari Event	Merupakan class untuk mencari event	EventActivity.java
Melakukan Posting Marathon	Merupakan class untuk melakukan posting marathon	LariActivity.java
Melakukan Deteksi Denyut Jantung	Merupakan class untuk melakukan pendeteksian denyut jantung	OmiActivity.java
Menghitung	Merupakan class untuk	LariActivity.java

Menu	Deskripsi	Nama File
Jarak Tempuh	menghitung jarak tempuh	
Mengubah Profile	Merupakan class untuk mengubah profile pengguna	ProfilActivity.java
Melakukan Counter Push Up	Merupakan class untuk menghitung gerakan push up	PushupActivity.java
Menambah Data Latihan	Merupakan class untuk menambah data latihan	AdminActivity.java
Mengubah Data Latihan	Merupakan class untuk mengubah data latihan	AdminActivity.java

4.1.5 Implementasi Antarmuka

Implementasi antarmuka yang dilakukan pada perangkat lunak yang dibangun terdiri dari beberapa tampilan yaitu nama antarmuka beserta file xml sebagai layout dari masing-masing antarmuka. Berikut ini adalah beberapa tampilan antarmuka perangkat lunak yang dibangun.

Tabel 4.4 Implementasi Antar Muka

Menu	Deskripsi	Nama File
Login	Merupakan halaman untuk login	Masuk_activity.xml
Registrasi	Merupakan halaman untuk registrasi akun baru	daftar_activity.xml
Lupa Password	Merupakan halaman untuk melakukan lupa password	lupa_activity.xml
Mencari Event	Merupakan halaman untuk mencari event	event_activity.xml
Melakukan Posting Marathon	Merupakan halaman untuk melakukan posting marathon	lari_activity.xml
Melakukan	Merupakan halaman untuk	omni_activity.xml

Deteksi Denyut Jantung	melakukan pendeteksian denyut jantung	
Menghitung Jarak Tempuh	Merupakan halaman untuk menghitung jarak tempuh	lari_activity.xml
Mengubah Profile	Merupakan halaman untuk mengubah profile pengguna	profil_activity.xml
Melakukan Counter Push Up	Merupakan halaman untuk menghitung gerakan push up	pushup_activity.xml
Menambah Data Latihan	Merupakan halaman untuk menambah data latihan	admin_activity.xml

4.2 Pengujian Sistem

Pengujian dilakukan dengan tujuan untuk menemukan kesalahan dan kekurangan pada perangkat lunak yang diuji. Pengujian tersebut dimaksudkan untuk mengetahui perangkat lunak yang dibuat apakah sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak.

4.2.1 Pengujian Fungsionalitas

Pada pengujian ini semua fitur pada perangkat lunak yang dibangun akan diuji menggunakan metode black box. Pengujian *Black box* dilakukan yaitu menguji fungsional dari sebuah sistem yang dibangun dan menemukan kesalahan program. Berikut ini adalah hasil dari pengujian yang telah dirancang.

Tabel 4.5 Pengujian Fungsionalitas Perangkat Lunak

Kelas Uji	Poin Pengujian	Jenis Pengujian
Login	Input data login	<i>Black Box</i>
	Validasi data login	<i>Black Box</i>
Registrasi	Input data registrasi	<i>Black Box</i>
	Validasi data registrasi	<i>Black Box</i>
	Menyimpan data registrasi ke database	<i>Black Box</i>
Lupa Password	Input data lupa password	<i>Black Box</i>

Kelas Uji	Poin Pengujian	Jenis Pengujian
	Validasi data lupa password	<i>Black Box</i>
	Menyimpan data lupa password ke database	<i>Black Box</i>
Mencari Event	Input data pencarian event	<i>Black Box</i>
	Validasi data mencari event	<i>Black Box</i>
	Menampilkan data pencarian event	<i>Black Box</i>
Melakukan Posting Marathon	Input data posting marathon	<i>Black Box</i>
	Validasi data posting marathon	<i>Black Box</i>
	Menyimpan data posting marathon di database	<i>Black Box</i>
Melakukan Deteksi Denyut Jantung	Input data deteksi denyut jantung	<i>Black Box</i>
	Validasi data deteksi denyut jantung	<i>Black Box</i>
	Menyimpan data deteksi denyut jantung ke database	<i>Black Box</i>
Menghitung Jarak Tempuh	Input data menghitung jarak tempuh	<i>Black Box</i>
	Validasi data menghitung jarak tempuh	<i>Black Box</i>
	Menampilkan hasil menghitung jarak tempuh	<i>Black Box</i>
Mengubah Profil	Input data mengubah profil	<i>Black Box</i>
	Validasi data mengubah profil	<i>Black Box</i>
	Menyimpan data mengubah profil ke database	<i>Black Box</i>
Melakukan Counter Push Up	Input data counter push up	<i>Black Box</i>
	Validasi data counter push up	<i>Black Box</i>
	Menampilkan data counter push up	<i>Black Box</i>

Kelas Uji	Poin Pengujian	Jenis Pengujian
Menambah Data Latihan	Input data menambah data latihan	<i>Black Box</i>
	Validasi data menambah data latihan	<i>Black Box</i>
	Menyimpan data menambah data latihan ke database	<i>Black Box</i>
Mengubah Data Latihan	Input data mengubah data latihan	<i>Black Box</i>
	Validasi data mengubah data latihan	<i>Black Box</i>
	Menyimpan data mengubah data latihan ke database	<i>Black Box</i>

4.2.2 Pengujian Acceptance

Pengujian acceptance dilakukan untuk mengetahui respon pengguna mengenai perangkat lunak yang dibangun. Pengujian ini dilakukan dengan cara menyebarkan kuesioner kepada 30 responden. Berdasarkan hasil kuesioner tersebut maka dapat diambil kesimpulan apakah perangkat lunak yang dibangun telah sesuai dengan tujuan atau tidak.

Adapun pertanyaan dari kuesioner pengujian acceptance adalah sebagai berikut:

- a) Apakah dengan adanya aplikasi ini mempermudah anda dalam mengatur ritme gerakan saat melakukan lari marathon?
- b) Apakah dengan adanya aplikasi ini membantu anda dalam menghitung denyut jantung dan suhu tubuh dalam lari marathon?
- c) Apakah dengan adanya aplikasi ini membantu anda mengetahui informasi mengenai even-even yang akan berlangsung?
- d) Apakah dengan adanya aplikasi ini mempermudah anda dalam menghitung gerakan push up?
- e) Apakah dengan adanya aplikasi ini membantu anda mengetahui level kebugaran?
- f) Apakah aplikasi ini mudah digunakan?

g) Apakah aplikasi ini memiliki tampilan yang menarik?

Jawaban untuk kuesioner :

- a. Sangat setuju
- b. Setuju
- c. Cukup setuju
- d. Tidak setuju
- e. Sangat tidak setuju

Perhitungan kuesioner dilakukan dengan menggunakan skala likert. Menurut Sugiyono, skala likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang tentang fenomena sosial. Adapun rumus perhitungan yang digunakan adalah sebagai berikut:

1. Untuk menghitung jumlah maksimum skor kriterium jumlah penilaian = 5, dan jumlah responden 30 yaitu dengan rumus:

$$\textit{Kriterium} = \textit{Jumlah Penilaian} \times \textit{Jumlah Responden}$$

Maka nilai dari kuesioner yang dibuat $5 \times 30 = 150$

2. Untuk mengetahui jumlah jawaban dari responden dalam bentuk persentase, di gunakan rumus sebagai berikut:

$$P = \frac{\textit{Total nilai}}{\textit{Skor ideal}} \times 100\%$$

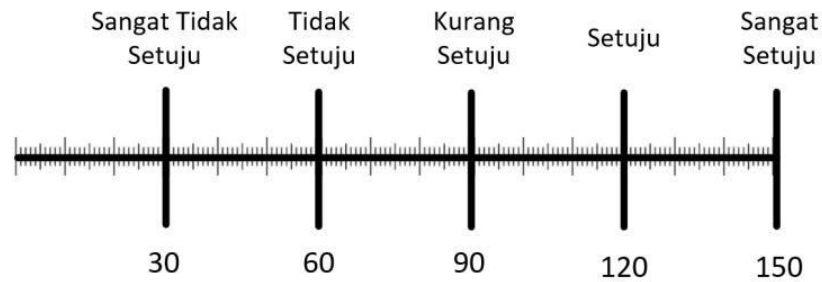
Keterangan:

P = Nilai persentase yang dicari

Total nilai = Jumlah frekuensi dikalikan nilai yang di tetapkan jawaban

Skor Ideal = Nilai tertinggi dilakukan dengan jumlah sampel

3. Skor yang telah didapat selanjutnya dimasukkan ke dalam bentuk interval rating scale sebagai berikut :



Gambar 4.1 Interval Rating Scale

4.3 Hasil Pengujian

Hasil pengujian menampilkan hasil dari pengujian yang dilakukan. Adapun hasil pengujian yang dilakukan antara lain:

4.3.1 Hasil Pengujian Fungsionalitas

Hasil pengujian fungsionalitas dengan menggunakan *Black box* dilakukan yaitu untuk menguji fungsional dari sebuah sistem yang dibangun dan menemukan kesalahan program. Berikut ini adalah hasil dari pengujian yang telah dirancang.

1) Hasil Pengujian Login

Tabel 4.6 Hasil Pengujian Login

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email: Password:	Sistem Menampilkan menu utama	Tampil Menu Utama	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email:{kosong} Password:{kosong}	Sistem menampilkan pesan "Email atau password tidak boleh kosong".	Tampil pesan "Email atau password tidak boleh kosong"	Diterima

2) Hasil Pengujian Registrasi

Tabel 4.7 Hasil Pengujian Registrasi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama Pengguna: Email: Password:	Sistem menampilkan pesan “Data akun berhasil tersimpan”	Tampil pesan “Data akun berhasil tersimpan”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama Pengguna: {kosong} Email: {kosong} Password: {kosong}	Sistem menampilkan pesan “Lengkapi data akun”	Tampil pesan “Lengkapi data akun”	Diterima

3) Hasil Pengujian Lupa Password

Tabel 4.8 Hasil Pengujian Lupa Password

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email:	Sistem menampilkan pesan “Data berhasil terkirim”	Tampil pesan “Data berhasil terkirim”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email: {kosong}	Sistem menampilkan pesan “Email harus diisi”	Tampil pesan “Email harus diisi”	Diterima

4) Hasil Pengujian Mencari Event

Tabel 4.9 Hasil Pengujian Mencari Event

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data event : {terisi}	Sistem menampilkan data event yang dicari	Tampil data event yang dicari	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data event : {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

5) Hasil Pengujian Posting Marathon

Tabel 4.10 Hasil Pengujian Posting Marathon

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Marathon : {terisi}	Sistem menampilkan pesan “Data berhasil terkirim”	Tampil pesan “Data berhasil terkirim”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Marathon : {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

6) Hasil Pengujian Deteksi Denyut Jantung

Tabel 4.11 Hasil Pengujian Deteksi Denyut Jantung

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Denyut : {terisi}	Sistem menampilkan data denyut	Tampil data event denyut	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Denyut : {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

7) Hasil Pengujian Menghitung Jarak Tempuh

Tabel 4.12 Hasil Pengujian Menghitung Jarak Tempuh

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Jarak : {terisi}	Sistem menampilkan data jarak tempuh	Tampil data event jarak tempuh	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Jarak: {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

8) Hasil Pengujian Mengubah Profile

Tabel 4.13 Hasil Pengujian Mengubah Profile

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Profil : {terisi}	Sistem menampilkan pesan “Data berhasil terkirim”	Tampil pesan “Data berhasil terkirim”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Profil : {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

9) Hasil Pengujian Counter Push Up

Tabel 4.14 Hasil Pengujian Counter Push Up

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Counter : {terisi}	Sistem menampilkan data counter push up	Tampil data counter push up	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Counter: {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

10) Hasil Pengujian Menambah Data Latihan

Tabel 4.15 Hasil Pengujian Menambah Data Latihan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Latihan : {terisi}	Sistem menampilkan pesan “Data berhasil terkirim”	Tampil pesan “Data berhasil terkirim”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Latihan: {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

11) Hasil Pengujian Mengubah Data Latihan

Tabel 4.16 Hasil Pengujian Mengubah Data Latihan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Latihan : {terisi}	Sistem menampilkan pesan “Data berhasil terkirim”	Tampil pesan “Data berhasil terkirim”	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data Latihan: {kosong}	Sistem menampilkan pesan “Data harus diisi”	Tampil pesan “Data harus diisi”	Diterima

4.3.2 Hasil Pengujian Acceptance

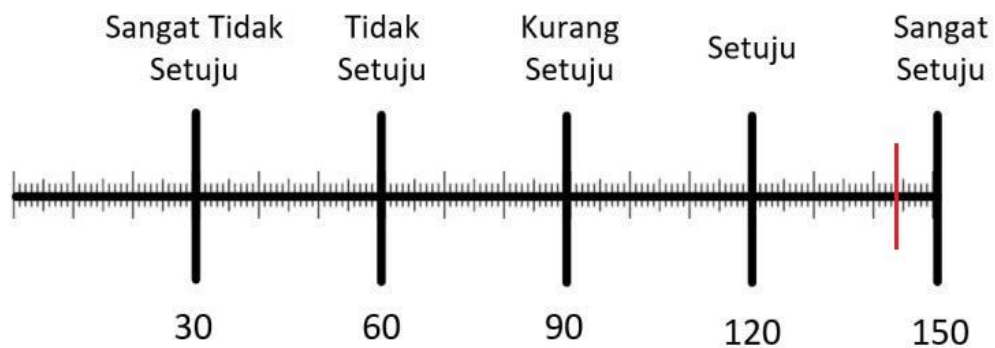
Hasil perhitungan persentase masing-masing jawaban kuesioner adalah sebagai berikut :

- a) Apakah dengan adanya aplikasi ini mempermudah anda dalam mengatur ritme gerakan saat melakukan lari marathon?

Tabel 4.17 Hasil Kuesioner Pertanyaan 1

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	19	95
Setuju	4	11	44
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	139

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 139 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini dapat mempermudah dalam mengatur ritme gerakan saat melakukan lari marathon. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



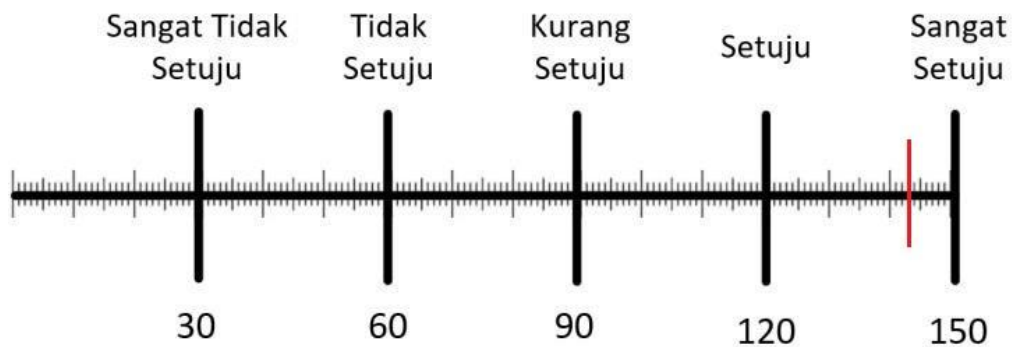
Gambar 4.2 Interval Rating Scale Pertanyaan 1

- b) Apakah dengan adanya aplikasi ini membantu anda dalam menghitung denyut jantung dan suhu tubuh dalam lari marathon?

Tabel 4.18 Hasil Kuesioner Pertanyaan 2

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	18	90
Setuju	4	12	48
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	138

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 138 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu dalam menghitung denyut jantung dan suhu tubuh dalam lari marathon. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



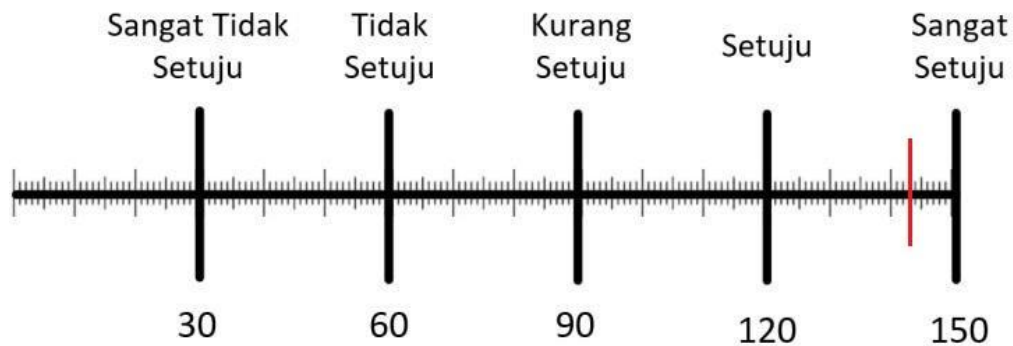
Gambar 4.3 Interval Rating Scale Pertanyaan 2

- c) Apakah dengan adanya aplikasi ini membantu anda mengetahui informasi mengenai even-even yang akan berlangsung?

Tabel 4.19 Hasil Kuesioner Pertanyaan 3

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	18	90
Setuju	4	12	48
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	138

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 138 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu dalam mengetahui informasi mengenai even-even yang akan berlangsung. Hasil perhitungan di atas dapat digambarkan sebagai berikut:

**Gambar 4.4 Interval Rating Scale Pertanyaan 3**

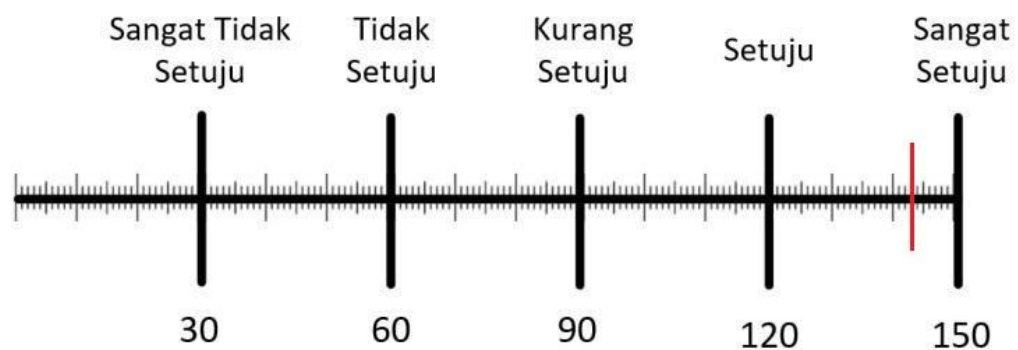
- d) Apakah dengan adanya aplikasi ini mempermudah anda dalam menghitung gerakan push up?

Tabel 4.20 Hasil Kuesioner Pertanyaan 4

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	18	90
Setuju	4	12	48

Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	138

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 138 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini dapat mempermudah dalam menghitung gerakan push up. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



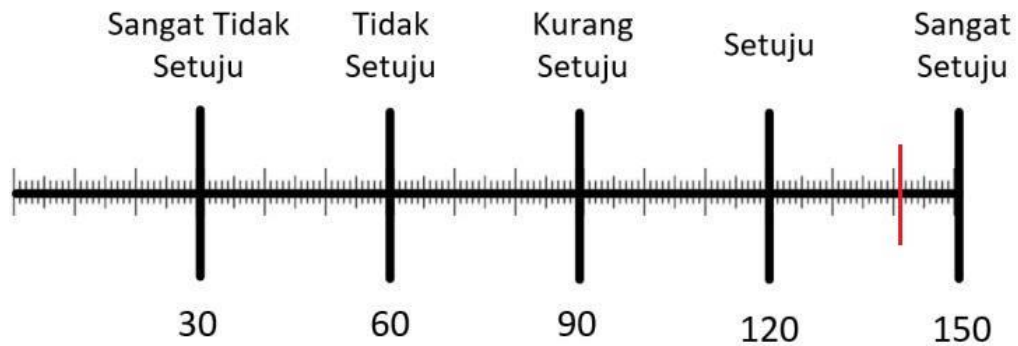
Gambar 4.5 Interval Rating Scale Pertanyaan 4

- e) Apakah dengan adanya aplikasi ini membantu anda mengetahui level kebugaran?

Tabel 4.21 Hasil Kuesioner Pertanyaan 5

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	16	80
Setuju	4	14	56
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	136

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 136 skor dengan skala kategori jawaban sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini dapat membantu dalam mengetahui level kebugaran. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



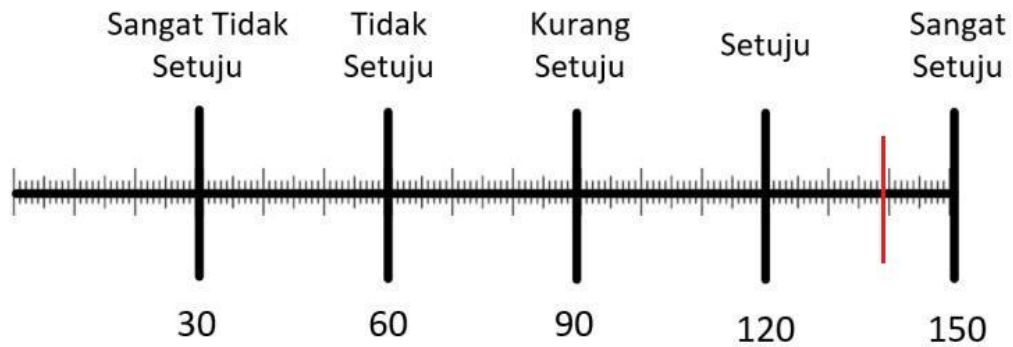
Gambar 4.6 Interval Rating Scale Pertanyaan 5

f) Apakah aplikasi ini mudah digunakan?

Tabel 4.22 Hasil Kuesioner Pertanyaan 6

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	14	70
Setuju	4	16	64
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	134

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 134 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini mudah digunakan oleh pengguna. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



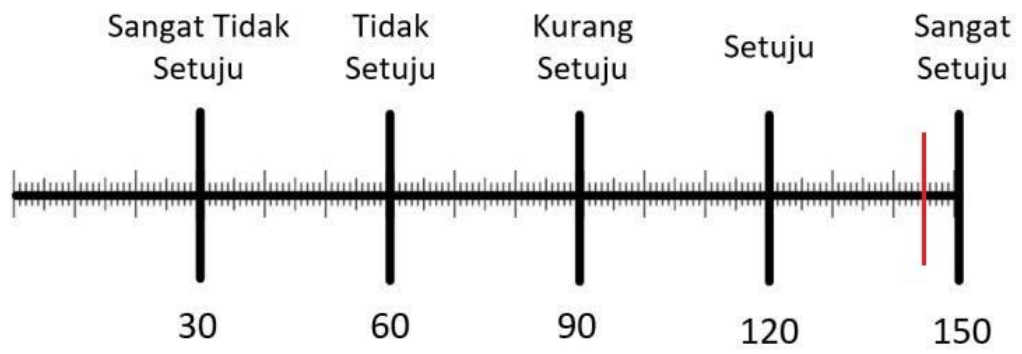
Gambar 4.7 Interval Rating Scale Pertanyaan 6

g) Apakah aplikasi ini memiliki tampilan yang menarik?

Tabel 4.23 Hasil Kuesioner Pertanyaan 7

Kategori Jawaban	Bobot	Frekuensi Jawaban	Total Skor
Sangat setuju	5	20	100
Setuju	4	10	40
Ragu-ragu	3	0	0
Tidak Setuju	2	0	0
Sangat tidak setuju	1	0	0
Total		30	140

Berdasarkan hasil perhitungan di atas, total skor yang didapat adalah sebanyak 140 skor dengan skala kategori jawaban mendekati sangat setuju. Maka dapat disimpulkan bahwa aplikasi ini memiliki tampilan yang menarik yang disukai oleh pengguna. Hasil perhitungan di atas dapat digambarkan sebagai berikut:



Gambar 4.8 Interval Rating Scale Pertanyaan 7