

BAB IV

HASIL DAN PEMBAHASAN

4.1 Jumlah Pelanggan Setiap Feeder Rayon Pekalongan di Gardu Induk Pekalongan

Berdasarkan data yang saya dapatkan dari PLN APJ Pekalongan yang beralamat di Jalan Manggis No.2 Sampangan, Pekalongan Timur, Kota Pekalongan, Jawa Tengah 51126 menyatakan bahwa Rayon distribusi PLN Area Kota Pekalongan mempunyai feeder yang berjumlah 12 buah feeder.

Mengacu pada data yang diberikan PLN APJ Pekalongan, jumlah total pelanggan dari seluruh feeder rayon Kota Pekalongan mencapai 197139 pelanggan pada tahun 2018. Pada tabel 4.1 di bawah ini ditunjukkan rincian dari jumlah pelanggan setiap feeder di Rayon Distribusi Area Kota Pekalongan tahun 2018.

Tabel 4.1 Jumlah Pelanggan PLN Rayon Kota Pekalongan

No.	Nama Feeder	Jumlah Pelanggan
1	PKL01	19432
2	PKL02	9958
3	PKL04	17510
4	PKL06	22976
5	PKL07	32546
6	PKL08	6414
7	PKL09	23111
8	PKL10	31825
9	PKL11	1235
10	PKL12	25155
11	PKL14	4858
12	PKL15	2119
Total Pelanggan		197139

4.2 Data Gangguan Feeder Gardu Induk Batang Tahun 2018

Berdasarkan data yang saya dapatkan dari PLN APJ Pekalongan, jumlah data trip PMT pada feeder rayon Kota Pekalongan selama tahun 2018 yang mencakup data waktu pemadaman dan waktu terhubung kembali pada tiap-tiap feeder ditunjukkan pada tabel 4.2 dan 4.3 dibawah ini. Perhitungan frekuensi dan durasi lama padam di setiap feeder dapat diketahui dari data di bawah ini. Frekuensi padam didapat dari perhitungan berapa kali trip atau padam pada masing-masing feeder setiap bulannya dan ditotal dalam satu tahun, sedangkan lamanya durasi padam dihasilkan dalam satuan menit dari berapa lama waktu pemadaman.

Tabel 4.2 Data Gangguan Pada Tiap Feeder PLN Rayon Kota Pekalongan

Tahun 2018					
No.	Nama Feeder	Waktu Padam	Waktu Nyala	Lama Padam (menit)	Bulan Terjadi
1	PKL11	9:39	9:48	9	Januari
2	PKL08	19:12	20:03	51	Januari
3	PKL12	17:13	18:01	48	Januari
4	PKL04	3:54	4:02	8	Januari
5	PKL15	18:34	20:52	48	Februari
6	PKL02	3:31	6:33	156	Maret
7	PKL14	18:42	20:44	113	April
8	PKL12	19:15	20:29	25	April
9	PKL01	15:24	16:38	36	Mei
10	PKL01	21:25	21:54	5	Juni
11	PKL15	22:04	1:58	116	Juni
12	PKL07	3:06	3:13	7	Juli
13	PKL04	19:52	20:17	25	Agustus
14	PKL07	4:02	4:08	6	Agustus
15	PKL14	10:44	10:51	7	Agustus
16	PKL11	14:13	17:43	210	Oktober
17	PKL06	22:43	22:50	7	Oktober
18	PKL08	0:17	2:53	156	Nopember
19	PKL06	11:16	11:26	10	Nopember
20	PKL07	6:32	7:30	58	Nopember
21	PKL10	15:27	16:29	62	Nopember

Tabel 4.3 Data Gangguan Pada Tiap Feeder PLN Rayon Kota Pekalongan (Lanjutan)

Tahun 2018					
No.	Nama Feeder	Waktu Padam	Waktu Nyala	Lama Padam (menit)	Bulan Terjadi
22	PKL08	18:19	19:22	63	Nopember
23	PKL02	0:40	0:53	13	Nopember
24	PKL11	3:12	3:18	6	Desember
25	PKL06	23:57	0:56	59	Desember

4.3 Perhitungan Indeks Keandalan Tiap Feeder Secara Manual

Perhitungan Indeks keandalan jaringan distribusi tenaga listrik secara manual dilakukan dengan menginputkan dan mengolah data – data yang telah didapatkan. Dalam kasus ini peneliti menggunakan data dari jaringan distribusi di rayon distribusi kota pekalongan. Perhitungan dilakukan satu per satu secara berurutan menggunakan kalkulator sebagai alat bantu hitungnya. Hasil dari perhitungan secara manual ini dimaksudkan untuk menjadi acuan atau patokan dari perangkat lunak android yang akan peneliti buat guna memudahkan dalam proses perhitungannya.

4.3.1 Perhitungan Indeks SAIFI

Berdasarkan data yang saya dapatkan dari PLN APJ Pekalongan diperoleh hasil frekuensi gangguan setiap feeder pada area distribusi rayon kota pekalongan. Data tersebut dipakai guna melakukan perhitungan, pengamatan dan analisis terhadap indeks keandalan jaringan distribusi listrik area pekalongan khususnya pada rayon kota pekalongan. Guna menghasilkan nilai indeks SAIFI, maka diperlukan beberapa data yang dipakai dalam melakukan perhitungan. Diantara data tersebut adalah jumlah pelanggan yang terkena gangguan per feeder, jumlah total pelanggan yang terkena gangguan, dan berapa kali frekuensi terjadinya gangguan yang masih dalam satu jaringan distribusi. Selanjutnya guna menghitung data yang telah tersajikan diperlukan sebuah formula dari perhitungan indeks SAIFI seperti pada persamaan 2.4. Formula indeks SAIFI bisa dilihat dibawah ini.

$$SAIFI = \frac{\lambda_i \cdot Ni}{Nt}$$

Keterangan :

λ_i = Frekuensi Pemadaman Feeder i (Kali/Tahun)

N_i = Jumlah Pelanggan Pada Feeder i

N_t = Jumlah Total Pelanggan

Kemudian data dan hasil perhitungan yang telah dilakukan sesuai formula diatas tersajikan dalam tabel 4.4 dibawah ini.

Tabel 4.4 Data & Hasil Perhitungan Indeks SAIFI

No.	Nama Feeder	Frekuensi Gangguan	Jumlah Pelanggan	Nilai Indeks SAIFI (kali/pelanggan/tahun)
1	PKL01	3	19432	0.295
2	PKL02	2	9958	0.101
3	PKL04	4	17510	0.355
4	PKL06	3	22976	0.349
5	PKL07	3	32546	0.495
6	PKL08	3	6414	0.098
7	PKL09	0	23111	0
8	PKL10	1	31825	0.161
9	PKL11	4	1235	0.025
10	PKL12	3	25155	0.383
11	PKL14	2	4858	0.049
12	PKL15	2	2119	0.021
Jumlah Total Pelanggan			197139	

Dibawah ini adalah contoh perhitungan indeks SAIFI dari tabel yang sudah tersaji diatas dengan menggunakan formula dan data yang ada.

$$\text{SAIFI Feeder PKL01} = \frac{\lambda_i \cdot N_i}{N_t} = \frac{3 \times 19432}{197139} = 0,295 \text{ Kali/Pelanggan/Tahun}$$

4.3.2 Perhitungan Indeks SAIDI

Indeks SAIDI adalah indeks yang menyatakan lama pemadaman yang dirasakan dari sudut pandang pelanggan. Ada beberapa data yang diperlukan dalam melakukan kalkulasi perhitungan indeks SAIDI, diantaranya adalah data durasi pemadaman dalam satuan jam tiap feeder atau penyulang, jumlah pelanggan yang

terdampak pemadaman per feeder, dan total jumlah pelanggan yang masih dalam satu jaringan distribusi. Selanjutnya guna menghitung data yang telah tersajikan diperlukan sebuah formula dari perhitungan indeks SAIDI seperti pada persamaan 2.5. Formula indeks SAIDI bisa dilihat dibawah ini.

$$SAIDI = \frac{U_i \cdot N_i}{N_t}$$

Keterangan:

U_i = Durasi Pemadaman Feeder i (Jam/Tahun)

N_i = Jumlah Gangguan Feeder i

N_t = Total Seluruh Pelanggan dalam jaringan

Kemudian data dan hasil perhitungan yang telah dilakukan sesuai formula diatas tersajikan dalam tabel 4.5.

Tabel 4.5 Data & Hasil Perhitungan Indeks SAIDI

No.	Nama Feeder	Jumlah Pelanggan N_i	Durasi Padam (Menit)	Durasi Padam U_i (Jam)	Nilai SAIDI (jam/pelanggan/tahun)
1	PKL01	19432	113	1.883	0.185
2	PKL02	9958	192	3.2	0.162
3	PKL04	17510	133	2.216	0.197
4	PKL06	22976	76	1.26	0.147
5	PKL07	32546	71	1.183	0.195
6	PKL08	6414	270	4.5	0.146
7	PKL09	23111	0	0	0
8	PKL10	31825	62	1.03	0.166
9	PKL11	1235	252	4.2	0.026
10	PKL12	25155	140	2.3	0.293
11	PKL14	4858	129	2.15	0.053
12	PKL15	2119	372	6.2	0.067
Total		197139			

Dibawah ini adalah contoh perhitungan indeks SAIDI dari tabel yang sudah tersaji diatas dengan menggunakan formula dan data yang ada.

$$SAIDI \text{ Feeder PKL01} = \frac{U_i \cdot N_i}{N_t} = \frac{1.883 \times 19432}{197139} = 0.185 \text{ Jam/Pelanggan/Tahun}$$

4.3.3 Perhitungan Indeks CAIDI

Indeks CAIDI adalah indeks yang menyatakan sebuah perbandingan antara indeks SAIDI dibandingkan dengan indeks SAIFI. Indeks CAIDI menggambarkan rata – rata lamanya durasi pemadaman yang terjadi di suatu jaringan distribusi. Sesuai uraian tersebut maka data yang dibutuhkan adalah nilai indeks SAIDI dan nilai indeks SAIFI. Selanjutnya guna menghitung data yang telah tersajikan diperlukan sebuah formula dari perhitungan indeks CAIDI seperti pada persamaan 2.6. Formula indeks SAIFI bisa dilihat dibawah ini.

$$CAIDI = \frac{SAIDI}{SAIFI}$$

Keterangan :

SAIDI = Total Durasi Pemadaman pada pelanggan (Jam/Tahun)

SAIFI = Total Frekuensi Pemadaman pada Pelanggan

Kemudian data dan hasil perhitungan yang telah dilakukan sesuai formula diatas tersajikan dalam tabel 4.6 dibawah ini.

Tabel 4.6 Data & Hasil Perhitungan Indeks CAIDI

No.	Nama Feeder	Nilai SAIDI (jam/pelanggan/tahun)	Nilai SAIFI (kali/pelanggan/tahun)	Nilai CAIDI (jam/kali/tahun)
1	PKL01	0.185	0.295	0.627
2	PKL02	0.162	0.101	0.604
3	PKL04	0.197	0.355	0.553
4	PKL06	0.147	0.349	0.421
5	PKL07	0.195	0.495	0.39
6	PKL08	0.146	0.097	0.505
7	PKL09	0	0	0
8	PKL10	0.166	0.161	1.03
9	PKL11	0.026	0.025	1.04
10	PKL12	0.29	0.383	0.757
11	PKL14	0.053	0.049	1.082
12	PKL15	0.067	0.021	3.19

Dibawah ini adalah contoh perhitungan indeks CAIDI dari tabel yang sudah tersaji diatas dengan menggunakan formula dan data yang ada.

$$\text{CAIDI Feeder PKL01} = \frac{\text{SAIDI}}{\text{SAIFI}} = \frac{0.185}{0.295} = 0.627 \text{ Jam/Kali/Tahun}$$

4.3.4 Perhitungan Indeks ASAI

ASAI merupakan kependekan dari Average Service Availability Index, yaitu indeks yang menyatakan ketersediaan tenaga listrik yang diterima oleh pelanggan. Data yang dibutuhkan pada perhitungan ASAI adalah data nilai dari indeks SAIDI. Selanjutnya guna menghitung data yang telah tersajikan diperlukan sebuah formula dari perhitungan indeks ASAI seperti pada persamaan 2.7. Formula indeks ASAI bisa dilihat dibawah ini.

$$\text{ASAI} = \frac{8760 - \text{SAIDI}}{8760} \times 100$$

Keterangan:

8760 = Jumlah Total Jam dalam Setahun

Tabel 4.7 Data & Hasil Perhitungan Indeks ASAI

No.	Nama Feeder	Nilai SAIDI (jam/pelanggan/tahun)	Nilai ASAI (%)
1	PKL01	0.185	99.99
2	PKL02	0.162	99.99
3	PKL04	0.197	99.99
4	PKL06	0.147	99.99
5	PKL07	0.195	99.99
6	PKL08	0.146	99.99
7	PKL09	0	100
8	PKL10	0.166	99.99
9	PKL11	0.026	99.99
10	PKL12	0.293	99.99
11	PKL14	0.053	99.99
12	PKL15	0.067	99.99

Dibawah ini adalah contoh perhitungan indeks ASAI dari tabel yang sudah tersaji diatas dengan menggunakan formula dan data yang ada.

$$\text{ASAI Feeder PKL01} = \frac{8760 - 0.185}{8760} = 0.9999 \times 100 = 99.99 \%$$

4.4 Desain Antarmuka Pengguna Program Aplikasi

Desain *user interface* atau antarmuka pengguna yakni sebuah desain bagian visual yang dapat dilihat dan berinteraksi dengan pengguna dari suatu program aplikasi. Oleh sebab itu sebuah user interface diharapkan bisa memberikan tampilan yang menarik dan interaksi yang mudah dimengerti oleh pengguna (*user friendly*).

4.4.1 Halaman *On Boarding Screen*

Halaman *On Boarding Screen* adalah halaman yang hanya sekali tampil, yaitu pada saat aplikasi diinstall untuk pertama kalinya saja. Halaman ini berfungsi memberi tahu secara singkat kegunaan dan fitur yang ada pada sebuah aplikasi. Halaman *On Boarding Screen* dapat digeser-geser kekanan atau kekiri untuk menampilkan beberapa fitur unggulan yang ada pada aplikasi ini. Jika dirasa tidak perlu melihat maka pengguna bisa menekan tombol lanjut yang ada di pojok kanan bawah untuk melewatinya. Gambar halaman *On Boarding Screen* seperti pada gambar 4.1, 4.2, dan 4.3.



Gambar 4.1 Halaman *On Boarding Screen* (1)



Gambar 4.2 Halaman *On Boarding Screen (2)*



Gambar 4.3 Halaman *On Boarding Screen (3)*

4.4.2 Halaman *Home*

Halaman *Home* adalah tampilan yang menampilkan menu utama dari aplikasi, berisi daftar menu utama yang ada pada aplikasi. Gambar halaman *Home* seperti terlihat pada gambar 4.5. Namun sebelum muncul layar *Home* terlebih dahulu muncul halaman *splash screen* pada awal membuka aplikasi ini. *Splash screen* yakni halaman pertama dari program aplikasi yang muncul untuk sementara waktu sebelum masuk ke menu *Home* atau menu utamanya. Gambar Halaman *splash screen* ada pada gambar 4.4.



Gambar 4.4 Halaman *Splash Screen*



Gambar 4.5 Halaman *Home*

4.4.3 Halaman Menu INTRODUCTION

Halaman menu INTRODUCTION adalah halaman menu yang menampilkan definisi singkat keandalan sistem distribusi listrik dan beberapa indeks keandalan yang digunakan serta beberapa parameter yang digunakan dalam pengevaluasian keandalan sistem distribusi listrik. Isi definisi tersebut ditampilkan dalam format PDF yang bisa digeser dan di zoom seperti layaknya penampil PDF umumnya. Gambar halaman menu INTRODUCTION bisa dilihat pada gambar 4.6.

← INTRODUCTION

Keandalan Sistem Tenaga Listrik merupakan kemampuan sebuah sistem tenaga listrik yang terdiri dari Pusat Pembangkit Listrik, Saluran Transmisi, dan Sistem Distribusi untuk melaksanakan suatu fungsi sesuai standar (waktu kegagalan) dalam keadaan yang ditentukan untuk jangka waktu tertentu.

Pengevaluasian dari suatu keandalan sistem tenaga listrik yang digunakan pada sistem distribusi mempunyai parameter-parameter sebagai berikut:

1. pemadaman rata-rata (rs),
2. kegagalan rata-rata (G),
3. waktu pemadaman rata-rata (US).

Beberapa indeks keandalan yang umum digunakan dalam menentukan nilai keandalan suatu sistem distribusi adalah seperti SAIFI, SAIDI, CAIDI, dan ASAI.

A. SAIFI (System Average Interruption Frequency Index)

Indeks SAIFI memberikan informasi tentang keandalan sistem distribusi berupa frekuensi rata-rata pemadaman per pelanggan. Melalui indeks SAIFI dapat diketahui nilai mengenai frekuensi kegagalan rata-rata pada suatu sistem. Sehingga bisa dilakukan evaluasi dan dikelompokkan sesuai dengan tingkat keandalannya. Satuan dari indeks SAIFI yaitu pemadaman per pelanggan. Besarnya nilai SAIFI bisa digunakan sebagai penggambaran besarnya *future rate* atau tingkat kegagalan (λ) sistem distribusi keseluruhan dirinjau dari sisi pelanggan. Indeks ini dirumuskan dengan:

$$SAIFI = \frac{\text{Total Frekuensi Pemadaman}}{\text{Total Jumlah Pelanggan yang dilayani}}$$

$$SAIFI = \frac{\sum_i N_i}{Nt}$$

Keterangan:

N_i = Frekuensi Pemadaman Per Pelanggan (kalibahan)

Nt = Jumlah Pelanggan Pada Persebaran i

Nt = Jumlah Total Pelanggan

Gambar 4.6 Halaman Menu INTRODUCTION

4.4.4 Halaman Menu DATA ENTRY

Halaman menu DATA ENTRY berisi sebuah form untuk mengisikan nilai dari sejumlah parameter yang digunakan dalam pengevaluasian keandalan sistem distribusi listrik yang nantinya akan ditampilkan dalam bentuk tabel. Gambar halaman tabel data entry seperti pada gambar 4.7. Sedangkan gambar halaman form data entry seperti gambar 4.8 di dalamnya mencakup:

- 1) Tombol "TAMBAH" guna menambahkan data yang diminta.
- 2) Tombol "HITUNG" guna memproses perhitungan.
- 3) Tombol "edit/sunting" guna merubah atau menghapus data. Sentuh data yang dimaksud, sentuh nilai parameter yang akan di ubah, sentuh "UBAH" untuk mengubah data (menyimpan data kembali) atau sentuh "HAPUS" jika ingin menghapus data tersebut. Gambar halaman form edit ada pada gambar 4.9.

← DATA ENTRY

No.	Nama Feeder	Frekuensi Gangguan	Durasi Gangguan (jam)	Jumlah Pelanggan
1	PKL01	3	1.883	19432
2	PKL02	2	3.2	9958
3	PKL04	4	2.216	17510
4	PKL06	3	1.26	22976
5	PKL07	3	1.183	32546
6	PKL08	3	4.5	6414
7	PKL09	0	0.0	23111
8	PKL10	1	1.03	31825
9	PKL11	4	4.2	1235
10	PKL12	3	2.3	25155
11	PKL14	2	2.15	4858
12	PKL15	2	6.2	2119

TAMBAH HITUNG

Gambar 4.7 Halaman Tabel Data Entry

← Kalkulator Indeks Keandalan

Nama Feeder

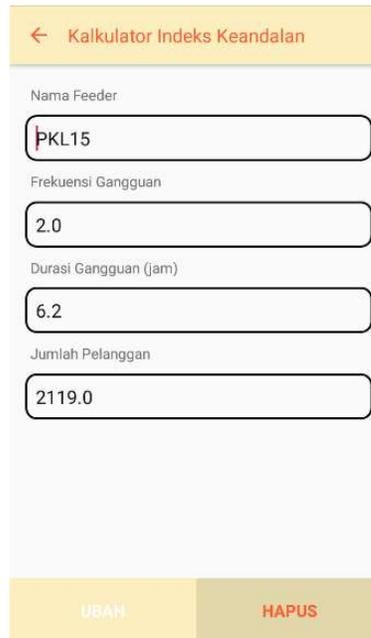
Frekuensi Gangguan

Durasi Gangguan (jam)

Jumlah Pelanggan

TAMBAH

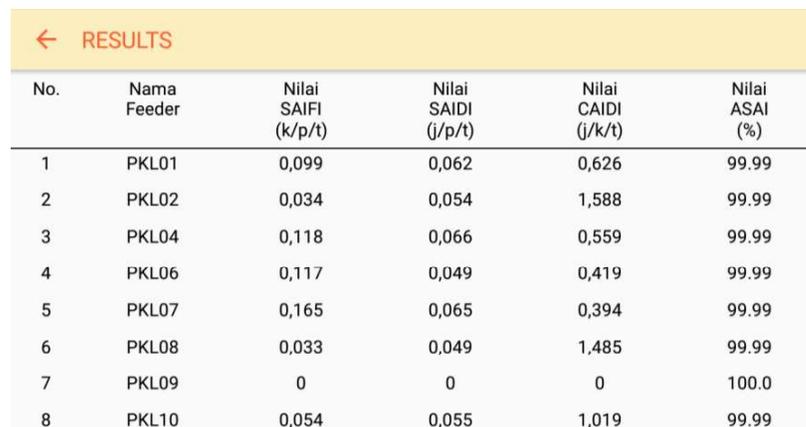
Gambar 4.8 Halaman Form Data Entry



Gambar 4.9 Halaman Form Edit

4.4.5 Halaman Menu RESULTS

Halaman dari menu RESULTS adalah halaman yang digunakan untuk menampilkan hasil dari perhitungan yang telah dilakukan pada menu "DATA ENTRY" dalam bentuk tabel. Gambar halaman hasil ada pada gambar 4.10 dan 4.11.



No.	Nama Feeder	Nilai SAIFI (k/p/t)	Nilai SAIDI (j/p/t)	Nilai CAIDI (j/k/t)	Nilai ASAI (%)
1	PKL01	0,099	0,062	0,626	99.99
2	PKL02	0,034	0,054	1,588	99.99
3	PKL04	0,118	0,066	0,559	99.99
4	PKL06	0,117	0,049	0,419	99.99
5	PKL07	0,165	0,065	0,394	99.99
6	PKL08	0,033	0,049	1,485	99.99
7	PKL09	0	0	0	100.0
8	PKL10	0,054	0,055	1,019	99.99

Gambar 4.10 Halaman RESULTS (1)

← RESULTS					
No.	Nama Feeder	Nilai SAIFI (k/p/t)	Nilai SAIDI (j/p/t)	Nilai CAIDI (j/k/t)	Nilai ASAI (%)
5	PKL07	0,165	0,065	0,394	99,99
6	PKL08	0,033	0,049	1,485	99,99
7	PKL09	0	0	0	100,0
8	PKL10	0,054	0,055	1,019	99,99
9	PKL11	0,008	0,009	1,125	99,99
10	PKL12	0,128	0,098	0,766	99,99
11	PKL14	0,016	0,018	1,125	99,99
12	PKL15	0,007	0,022	3,143	99,99

Gambar 4.11 Halaman RESULTS (2)

4.4.6 Halaman Menu ANALYSIS

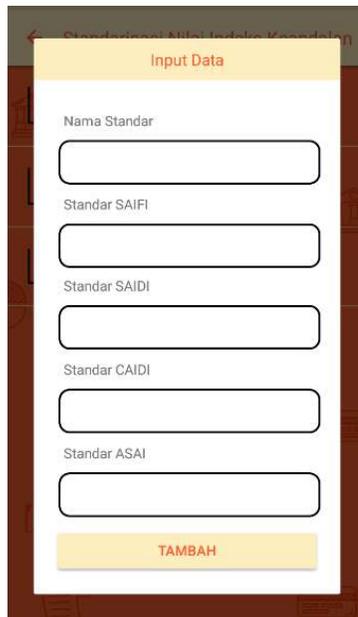
Halaman menu ANALYSIS adalah halaman yang akan menampilkan hasil dari analisis yang terbagi dalam tiga standar yaitu standar custom, standar SPLN dan standar IEEE. Gambar halaman sub menu analysis ini ada pada gambar 4.12.



Gambar 4.12 Halaman Sub Menu Analysis

Penjelasan ketiga jenis standar tersebut yaitu:

- 1) Standar Custom, adalah standar yang dimiliki oleh masing-masing wilayah, mengingat target nilai indeks keandalan dari tiap wilayah tidak sama. Pengguna diharapkan mengisi form yang tersedia untuk melihat hasil analisisnya. Form tersebut berisi nama standar dan nilai dari masing-masing indeks seperti pada gambar 4.13. Standar Custom bisa diisi jika diperlukan dan dilewati jika tidak dibutuhkan. Di bawah pada gambar 4.14, 4.15, 4.16, dan 4.17 adalah gambar halaman hasil analisis standar custom dalam hal ini standar yang digunakan yakni standar PLN Rayon Kota Pekalongan.



The image shows a mobile application interface for data entry. At the top, there is a title bar with a back arrow and the text 'Standardisasi Nilai Indeks Keandalan'. Below this is a yellow header with the text 'Input Data'. The main content area is white and contains five input fields, each with a label above it: 'Nama Standar', 'Standar SAIFI', 'Standar SAIDI', 'Standar CAIDI', and 'Standar ASAI'. At the bottom of the form is a yellow button with the text 'TAMBAH'.

Gambar 4.13 Halaman Form Input Standar Custom

← Data Analisis					
Standarisasi Nilai Indeks Keandalan berdasarkan PLN Rayon Kota Pekalongan					
No.	Nama Feeder	Nilai SAIFI	Nilai SAIDI	SAIFI < 3.62 (k/p/t)	SAIDI < 9.77 (j/p/t)
1	PKL01	0,296	0,186	✓	✓
2	PKL02	0,101	0,162	✓	✓
3	PKL04	0,355	0,197	✓	✓
4	PKL06	0,35	0,147	✓	✓
5	PKL07	0,495	0,195	✓	✓
6	PKL08	0,098	0,146	✓	✓

Gambar 4.14 Halaman Hasil Analisis Standar Custom (1)

← Data Analisis					
7	PKL09	0	0	✓	✓
8	PKL10	0,054	0,055	✓	✓
9	PKL11	0,008	0,009	✓	✓
10	PKL12	0,128	0,098	✓	✓
11	PKL14	0,016	0,018	✓	✓
12	PKL15	0,007	0,022	✓	✓

Hasil Analisis

- Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL01 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

Gambar 4.15 Halaman Hasil Analisis Standar Custom (2)

← Data Analisis					
2. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL02 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					
3. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL04 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					
4. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL06 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					
5. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL07 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					
6. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL08 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					
7. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL09 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.					

Gambar 4.16 Halaman Hasil Analisis Standar Custom (3)

← Data Analisis

7. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL09 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

8. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL10 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

9. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL11 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

10. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL12 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

11. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL14 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

12. Berdasarkan standar PLN Rayon Kota Pekalongan, Feeder PKL15 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

Gambar 4.17 Halaman Hasil Analisis Standar Custom (4)

- 2) Standar SPLN, sesuai namanya standar ini adalah standar yang dibuat oleh pihak PLN (persero) yang digunakan sebagai patokan indeks keandalan untuk seluruh Indonesia. Kemudian di bawah pada gambar 4.18, 4.19, 4.20, dan 4.21 adalah halaman hasil analisis standar SPLN.

← Data Analisis

Standarisasi Nilai Indeks Keandalan berdasarkan SPLN

No.	Nama Feeder	Nilai SAIFI	Nilai SAIDI	SAIFI < 3.2 (k/p/t)	SAIDI < 21.09 (j/p/t)
1	PKL01	0,296	0,186	✓	✓
2	PKL02	0,101	0,162	✓	✓
3	PKL04	0,355	0,197	✓	✓
4	PKL06	0,35	0,147	✓	✓
5	PKL07	0,495	0,195	✓	✓
6	PKL08	0,098	0,146	✓	✓

Gambar 4.18 Halaman Hasil Analisis Standar SPLN (1)

← Data Analisis						
7	PKL09	0	0	✓	✓	
8	PKL10	0,054	0,055	✓	✓	
9	PKL11	0,008	0,009	✓	✓	
10	PKL12	0,128	0,098	✓	✓	
11	PKL14	0,016	0,018	✓	✓	
12	PKL15	0,007	0,022	✓	✓	

Hasil Analisis

1. Berdasarkan standar SPLN, Feeder PKL01 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.

Gambar 4.19 Halaman Hasil Analisis Standar SPLN (2)

← Data Analisis						
2. Berdasarkan standar SPLN, Feeder PKL02 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
3. Berdasarkan standar SPLN, Feeder PKL04 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
4. Berdasarkan standar SPLN, Feeder PKL06 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
5. Berdasarkan standar SPLN, Feeder PKL07 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
6. Berdasarkan standar SPLN, Feeder PKL08 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
7. Berdasarkan standar SPLN, Feeder PKL09 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						

Gambar 4.20 Halaman Hasil Analisis Standar SPLN (3)

← Data Analisis						
7. Berdasarkan standar SPLN, Feeder PKL09 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
8. Berdasarkan standar SPLN, Feeder PKL10 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
9. Berdasarkan standar SPLN, Feeder PKL11 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
10. Berdasarkan standar SPLN, Feeder PKL12 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
11. Berdasarkan standar SPLN, Feeder PKL14 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						
12. Berdasarkan standar SPLN, Feeder PKL15 dinyatakan handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.						

Gambar 4.21 Halaman Hasil Analisis Standar SPLN (4)

- 3) Standar IEEE adalah standar yang dibuat oleh Institute of Electrical and Electronics Engineers, yaitu sebuah organisasi internasional yang beranggotakan para insinyur. Standar ini berlaku untuk seluruh dunia. Kemudian di bawah pada gambar 4.22, 4.23, 4.24, dan 4.25 adalah halaman hasil analisis standar IEEE.

← Data Analisis									
Standarisasi Nilai Indeks Keandalan berdasarkan IEEE									
No.	Nama Feeder	Nilai SAIFI	Nilai SAIDI	Nilai CAIDI	Nilai ASAI	SAIFI < 1.45 (k/p/t)	SAIDI < 2.3 (j/p/t)	CAIDI < 1.47 (j/k/t)	ASAI > 99.92 (%)
1	PKL01	0,296	0,186	0,628	99.99	✓	✓	✓	✓
2	PKL02	0,101	0,162	1,604	99.99	✓	✓	X	✓
3	PKL04	0,355	0,197	0,555	99.99	✓	✓	✓	✓
4	PKL06	0,35	0,147	0,42	99.99	✓	✓	✓	✓
5	PKL07	0,495	0,195	0,394	99.99	✓	✓	✓	✓
6	PKL08	0,098	0,146	1,49	99.99	✓	✓	X	✓

Gambar 4.22 Halaman Hasil Analisis Standar IEEE (1)

← Data Analisis									
7	PKL09	0	0	0	100.0	✓	✓	✓	✓
8	PKL10	0,054	0,055	1,019	99.99	✓	✓	✓	✓
9	PKL11	0,008	0,009	1,125	99.99	✓	✓	✓	✓
10	PKL12	0,128	0,098	0,766	99.99	✓	✓	✓	✓
11	PKL14	0,016	0,018	1,125	99.99	✓	✓	✓	✓
12	PKL15	0,007	0,022	3,143	99.99	✓	✓	X	✓

Hasil Analisis

1. Berdasarkan standar IEEE, Feeder PKL01 dinyatakan **handal** karena seluruh indeks keandalan tidak melebihi dan kurang dari batas yang ditetapkan.

Gambar 4.23 Halaman Hasil Analisis Standar IEEE (2)



Gambar 4.24 Halaman Hasil Analisis Standar IEEE (3)



Gambar 4.25 Halaman Hasil Analisis Standar IEEE (4)

4.4.7 Halaman Menu EXPORT

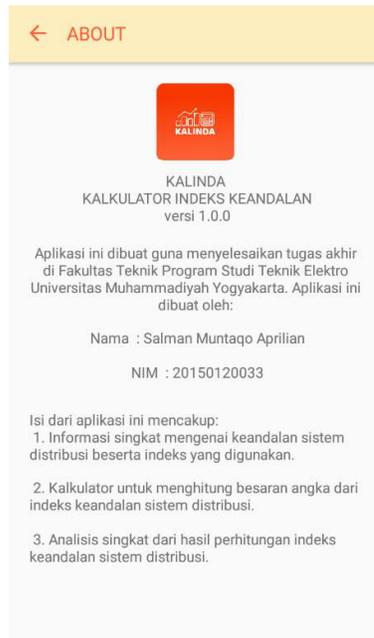
Halaman menu EXPORT adalah halaman menu yang digunakan untuk mengubah format suatu file ke dalam bentuk tertentu agar dapat dibaca oleh software lain dalam hal ini ke format file microsoft excel (.xls). File yang diexport hanya hasil data perhitungan saja, tidak termasuk analisis. Ketika sudah menekan tombol export maka file hasil export dapat ditemukan di media penyimpanan telepon File Manager dalam folder Download. Pada halaman ini diperlukan izin supaya aplikasi bisa mengakses media penyimpanan telepon guna meletakkan file excel tadi. Hasil dari export ada pada halaman lamiran. Gambar halaman izin akses media penyimpanan ada pada gambar 4.26.



Gambar 4.26 Halaman Izin Akses Media Penyimpanan

4.4.8 Halaman Menu ABOUT

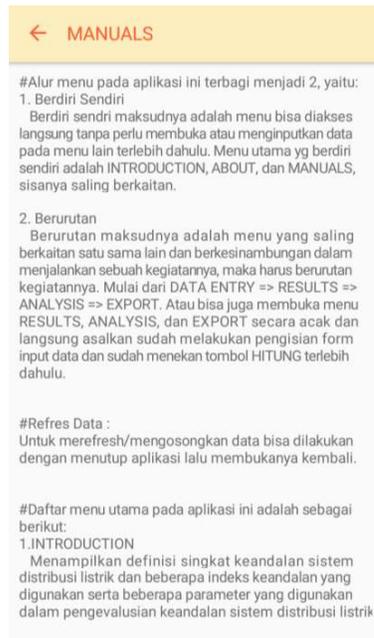
Halaman ABOUT adalah halaman yang berisikan mengenai perihal aplikasi itu sendiri. Mulai dari nama aplikasi, versi aplikasi, kegunaan aplikasi, dan yang menjadi latar belakang pembuatan aplikasi. Gambar halaman menu ABOUT bisa dilihat pada gambar 4.27.



Gambar 4.27 Halaman Menu ABOUT

4.4.9 Halaman Menu MANUALS

Halaman MANUALS adalah halaman yang menampilkan tata cara penggunaan dari aplikasi agar penggunaan tidak bingung dalam mengoperasikan aplikasi ini. Mulai dari definisi singkat dari beberapa menu utama, refresh data sampai alur kerja aplikasi ini. Gambar halaman menu MANUALS bisa dilihat pada gambar 4.28.



Gambar 4.28 Halaman Menu MANUALS

4.5 Kode Sumber dan *User Manual*

Source code atau yang dalam bahasa Indonesia berarti kode sumber, adalah bagian dasar dari program yang berupa beberapa baris deklarasi pernyataan atau perintah yang dituliskan oleh seorang programmer agar dapat berkomunikasi dengan komputer. Sedangkan *user manuals* adalah sebuah teks komunikasi teknis yang berguna memberikan panduan atau bantuan untuk penggunaan suatu sistem, dalam hal ini penggunaan aplikasi KALINDA. Di bawah akan dibahas mengenai kode sumber dari masing-masing halaman pada aplikasi KALINDA dan *user manual* nya.

4.5.1 Kode Sumber *On Boarding Screen*

On Boarding Screen adalah halaman yang muncul pada pertama kali dilakukan install aplikasi. Pada halaman ini yang ada hanya *On Boarding Screen* yang dapat di geser - geser untuk memberi tahu informasi fitur utama aplikasi. Kode sumber halaman *On Boarding Screen* terlihat pada gambar 4.29.

```

public void initView() {
    tabLayout = findViewById(R.id.pageIndicatorView);
    viewPager = findViewById(R.id.viewpager);
    btStart = findViewById(R.id.btStart);

    btStart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(BoardingActivity.this,
                MainActivity.class);
            startActivity(intent);
            finish();});

    List<Boarding> splashPic = new ArrayList<>();
    splashPic.add(new Boarding(R.drawable.bg_1,R.drawable.petir_1,
        "Intro", "Penjelasan singkat tentang keandalan jaringan
        distribusi listrik.));
    splashPic.add(new Boarding(R.drawable.bg_2,R.drawable.petir_2,"
        Perhitungan indeks keandalan", "Menghitung indeks keandalan
        jaringan distribusi listrik.));
    splashPic.add(new Boarding(R.drawable.bg_3,R.drawable.petir_3,"
        Analisa", "Menganalisa hasil indeks keandalan jaringan
        distribusi listrik.));
    final BoardingPagerAdapter adapterPager = new
    BoardingPagerAdapter(this, splashPic);
    viewPager.setAdapter(adapterPager);
    tabLayout.setAnimationType(AnimationType.DROP);
    tabLayout.setViewPager(viewPager);
}

```

Gambar 4.29 Kode Sumber Halaman *On Boarding Screen*

Pada kode sumber di atas digunakan fungsi viewPager dan tablayout. ViewPager ditugaskan guna membungkus isi konten yang mau ditampilkan dalam hal ini *On Boarding Screen*, juga untuk swipe atau geser dari kanan ke kiri atau sebaliknya. Sedangkan tab layout ditugaskan guna menjadi navigasi tabnya. Fungsi intent juga digunakan untuk melakukan perpindahan halaman dari *boardingactivity* yakni halaman *On Boarding Screen* ke *mainactivity* yakni halaman *home* atau menu utama.

4.5.1 Kode Sumber *Home*

Kode sumber pada menu *Home* berisi fungsi penampil teks dan gambar, dan beberapa tombol yang mengarahkan user pada berbagai menu utama pada aplikasi. Kode sumber halaman *Home* terlihat pada gambar 4.31. Namun sebelum masuk ke halaman *Home*, terlebih dahulu masuk ke halaman *splash screen*. Kode sumber halaman *splash screen* terlihat pada gambar 4.30.

```
public void run()
{if (text.isEmpty()) {startActivity(new Intent(SplashActivity.this,
BoardingActivity.class));}
else {startActivity(new
Intent(SplashActivity.this, MainActivity.class));}
    finish();}
}, 2500);
```

Gambar 4.30 Kode Sumber Halaman *Splash Screen*

Pada kode sumber diatas menjelaskan adanya perpindahan class oleh fungsi intent. Perpindahan dari class *splashactivity* untuk menampilkan halaman splash ke class *boardingactivity* untuk menampilkan halaman awal install jika sebelumnya belum pernah tertampil. Tetapi jika sudah pernah tertampil maka class *splashactivity* akan berpindah ke class *mainactivity* yang mana merupakan halaman utama dari layar *Home*. Waktu tahan atau delay untuk halaman *splash screen* adalah 2500 milidetik atau 2,5 detik.

```

viewIntro.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {Intent intent = new Intent
(MainActivity.this, IntroActivity.class) intent.putExtra("title",
"Intro"); startActivity(intent);} });

viewInputan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {Intent intent = new Intent
(MainActivity.this, InputActivity.class); intent.putExtra("title",
"Inputan"); startActivity(intent);} });

viewHasil.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (hasilDataList.size() > 0) {Intent intent = new
Intent(MainActivity.this, HasilActivity.class);
startActivity(intent); }
    else { Toast.makeText(MainActivity.this, "Belum melakukan input
data", Toast.LENGTH_SHORT).show();} } });

viewSaifiSaidi.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (hasilDataList.size() > 0) {Intent intent = new
Intent(MainActivity.this, MenuAnalisaActivity.class);
startActivity(intent);}
    else { Toast.makeText(MainActivity.this, "Belum melakukan input
data", Toast.LENGTH_SHORT).show();} } });

viewExport.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    if (hasilDataList.size() > 0) {permissionReq(); }
    else { Toast.makeText(MainActivity.this, "Belum melakukan input
data", Toast.LENGTH_SHORT).show(); } } });

viewAbout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {Intent intent = new
Intent(MainActivity.this, AboutActivity.class);
startActivity(intent); } });

viewGuide.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { Intent intent = new Intent
(MainActivity.this, UserGuideActivity.class); startActivity(intent)
;} });

```

Gambar 4.31 Kode Sumber Halaman *Home*

Pada kode sumber diatas menjelaskan adanya fungsi perpindahan halaman oleh fungsi intent. Perpindahan halaman yang dilakukan adalah dari class *mainactivity* atau halaman *Home* ke halaman introduction, data entry, results, analysis, export, about, dan manuals. Namun khusus perpindahan ke halaman result, analysis, dan export akan ditolak dan muncul *toast text* atau teks kecil dengan tulisan “Belum melakukan input data” jika pengguna belum melakukan input data serta perhitungan di menu data entry.

4.5.2 Kode Sumber INTRODUCTION

Kode sumber pada menu INTRODUCTION berisi fungsi dari penampil file PDF yang berisi informasi singkat mengenai keandalan jaringan distribusi. Kode sumber penampil file PDF terlihat pada gambar 4.32.

```
pdfView = findViewById(R.id.pdfView);  
  
pdfView.fromAsset(PDF_FILE)  
        .defaultPage(1)  
        .onPageChange(null)  
        .load();
```

Gambar 4.32 Kode Sumber Penampil File PDF

Pada kode sumber di atas menjelaskan adanya fungsi penampil file PDF oleh fungsi *pdfView*. Sedangkan *defaultPage(1)* menunjukkan halaman awalnya adalah halaman satu atau pertama.

4.5.3 Kode Sumber DATA ENTRY

Kode sumber pada menu DATA ENTRY berisi fungsi memasukan data atau input data yang terlihat pada gambar 4.33. Serta fungsi operasi hitung dari masing – masing indeks keandalan yang terlihat pada gambar 4.34.

```
etName.setText(inputData.getNamaFeeder());  
etFrekuensi.setText(inputData.getFrekuensiGangguan() + "");  
etDurasi.setText(inputData.getDurasiGangguan() + "");  
etJumlah.setText(inputData.getJumlahPelanggan() + "");
```

Gambar 4.33 Kode Sumber Input Data

Pada kode sumber di atas menjelaskan adanya fungsi pengambilan data input. Data input yang diambil diantaranya adalah nama feeder, frekuensi gangguan, durasi gangguan, dan jumlah pelanggan.

```
saifi = Double.parseDouble(new
DecimalFormat("##.###").format((frekuensiGangguan *
jumlahPelanggan) / totalJumlahPelanggan).replaceAll(",", "."));

saidi = Double.parseDouble(new
DecimalFormat("##.###").format((durasiGangguan *
jumlahPelanggan) / totalJumlahPelanggan).replaceAll(",", "."));

caidi = saidi / saifi;

asai = ((8760 - saidi) / 8760) * 100;
```

Gambar 4.34 Kode Sumber Operasi Hitung

Pada kode sumber di atas menjelaskan adanya fungsi operasi perhitungan dari indeks SAIFI, SAIDI, CAIDI, dan ASAI. Terdapat fungsi untuk mengubah data string menjadi data double yang ditunjukkan oleh sintak *Double.parseDouble*. kemudian bisa dilihat pada kode sumber *DecimalFormat* ("##.###") menunjukkan bahwa hanya diambil tiga angka dibelakang koma. Kemudian tanda koma diganti dengan tanda titik ditunjukkan oleh sintak *replaceAll* ("", ".").

4.5.4 Kode Sumber RESULTS

Kode sumber dari menu RESULTS berisi fungsi dari penampil data hasil perhitungan yang sudah dilakukan perhitungannya pada menu DATA ENTRY, jadi pada menu RESULTS ini hanya tinggal menampilkan datanya saja. Kode sumber penampil hasil perhitungan terlihat pada gambar 4.35.

```
public void initView() {rvData = findViewById(R.id.rvData);

LinearLayoutManager layoutManager = new
LinearLayoutManager(context:HasilActivity.this,
LinearLayoutManager.VERTICAL, reverseLayout:false);
adapter = new HasilAdapter(HasilActivity.this,hasilDataList);
rvData.setLayoutManager(layoutManager);
rvData.setAdapter(adapter);
adapter.notifyDataSetChanged(); }
```

Gambar 4.35 Kode Sumber Penampil Hasil Perhitungan

Pada kode sumber di atas menjelaskan adanya fungsi recycle view ditunjukkan oleh sintak *rvData* yang berfungsi untuk menampilkan data dalam jumlah yang besar dan banyak. Ada juga *layoutManager* berfungsi mengatur halaman tampilan dalam kasus ini di atur pada mode vertikal dari atas ke bawah. Adapula fungsi adapter guna menghubungkan data dengan recycle view, dimana kumpulan data yang akan ditampilkan bernama *hasilDataList*.

4.5.5 Kode Sumber ANALYSIS

Kode sumber dari menu ANALYSIS terdiri dari banyak kode sumber, diantaranya adalah kode sumber sub menu analysis seperti pada gambar 4.36 yang berisi standar custom, standar SPLN, dan standar IEEE. Gambar 4.37 yaitu Kode Sumber Input Standar Custom yaitu input data yang digunakan untuk standar sesuai target keandalan di wilayahnya masing-masing. Gambar 4.38 tentang kode sumber standar SPLN yaitu standar yang digunakan di Indonesia dan ditetapkan oleh pihak PLN. Gambar 4.39 tentang kode sumber standar IEEE yaitu standar internasional yang digunakan di seluruh negara yang ditetapkan oleh IEEE. Gambar 4.40 tentang kode sumber analisis data yaitu kode sumber yang digunakan untuk membandingkan antara data nilai hasil perhitungan dengan data nilai indeks keandalan yang tercantum dalam standarnya masing-masing. Gambar 4.41 tentang kode sumber teks analisis data dan gambar 4.42 sebagai lanjutannya yang membahas teks analisis singkat mengenai indeks keandalan yang akan ditampilkan, apakah sudah handal, kurang handal atau tidak handal.

```

viewStandar1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {showInput();
Intent intent = new Intent(MenuAnalisaActivity.this
AnalisisActivity.class);intent.putExtra("standar", "custom");
startActivity(intent);} });

viewStandar2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {Intent intent = new
Intent(MenuAnalisaActivity.this, AnalisisActivity.class);
intent.putExtra("standar", "SPLN");startActivity(intent); });

viewStandar3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {Intent intent = new
Intent(MenuAnalisaActivity.this, AnalisisActivity.class);
intent.putExtra("standar", "IEEE");startActivity(intent); } });

```

Gambar 4.36 Kode Sumber Sub Menu Analysis

Pada kode sumber sub menu analysis di atas yang terjadi hanya perpindahan halaman *MenuAnalisaActivity* ke halaman *AnalisisActivity* yang dilakukan oleh fungsi intent dengan bantuan *OnClickListener*. Adanya penambahan *putExtra* pada intent berfungsi guna mengirimkan data ke activity yang dituju dalam hal ini ke standar custom, standar SPLN, dan standar IEEE.

```

btnSubmit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
dialog.dismiss();
String name = etName.getText().toString();
String standarSaidi =
etStandarSaidi.getText().toString().replaceAll(",", ".");
String standarSaifi =
etStandarSaifi.getText().toString().replaceAll(",", ".");
String standarCaidi =
etStandarCaidi.getText().toString().replaceAll(",", ".");
String standarAsai =
etStandarAsai.getText().toString().replaceAll(",", ".");

```

Gambar 4.37 Kode Sumber Input Standar Custom

Pada kode sumber input standar custom diatas menjelaskan bahwa *btnSubmit* digunakan sebagai tombol tambah pada halaman input standar custom yang berguna memasukan data – data yang diperlukan seperti nama standar yang ditunjukkan oleh variabel *name*, nilai masing – masing standar dengan variabel *standarSaifi*, *standarSaidi*, *standarCaidi*, dan *standarAsai*. Kemudian penggantian tanda koma diganti dengan tanda titik ditunjukkan oleh sintak *replaceAll(",",".")*.

```
if (standar.equals("SPLN")) {
    viewHeader.setVisibility(View.VISIBLE);
    tvText.setText("Standarisasi Nilai Indeks Keandalan
berdasarkan SPLN");
    shownSaidi = true;
    shownSaifi = true;
    for (int i = 0; dataSaifiSaidis1.size() > i; i++) {
        dataSaifiSaidis1.get(i).setStandartNilaiSaifi(3.2);
        dataSaifiSaidis1.get(i).setStandartNilaiSaidi(21.09) }
    tvHeader1Saifi.append("\n< " +
dataSaifiSaidis1.get(0).getStandartNilaiSaifi() +
"\n(k/p/t)");
    tvHeader1Saidi.append("\n< " +
dataSaifiSaidis1.get(0).getStandartNilaiSaidi() +
"\n(j/p/t)");
```

Gambar 4.38 Kode Sumber Standar SPLN

Pada kode sumber standar SPLN di atas menjelaskan bahwa data yang ditampilkan hanya data SAIFI dan SAIDI, ditunjukkan pada sintak *shownSaidi=true*; dan *shownSaifi=true*; karena data yang digunakan pada standar SPLN hanya data SAIFI dan SAIDI saja. Kemudian dilakukan pengaturan terhadap nilai standar yang ditetapkan, ditunjukkan pada sintak *setStandartNilaiSaifi(3.2)* dan *setStandartNilaiSaidi(21.09)*.

```

if (standar.equals("IEEE")) {
    tvText.setText("Standarisasi Nilai Indeks Keandalan
berdasarkan IEEE");
    viewHeader2.setVisibility(View.VISIBLE);

    tvStandarSaidi.setVisibility(View.VISIBLE);
    tvSaidi.setVisibility(View.VISIBLE);

    tvStandarSaifi.setVisibility(View.VISIBLE);
    tvSaifi.setVisibility(View.VISIBLE);

    tvStandarCaidi.setVisibility(View.VISIBLE);
    tvCaidi.setVisibility(View.VISIBLE);

    tvStandarAsai.setVisibility(View.VISIBLE);
    tvAsai.setVisibility(View.VISIBLE);

    shownSaidi = true;
    shownCaidi = true;
    shownSaifi = true;
    shownAsai = true;
    for (int i = 0; dataSaifiSaidis1.size() > i; i++) {
        dataSaifiSaidis1.get(i).setStandartNilaiSaifi(1.45);
        dataSaifiSaidis1.get(i).setStandartNilaiSaidi(2.30);
        dataSaifiSaidis1.get(i).setStandartNilaiCaidi(1.47);
        dataSaifiSaidis1.get(i).setStandartNilaiAsai(99.92);
    }
}

```

Gambar 4.39 Kode Sumber Standar IEEE

Pada kode sumber standar IEEE di atas menjelaskan bahwa data yang ditampilkan adalah semuanya mulai dari data SAIFI, SAIDI, CAIDI, dan ASAI, ditunjukkan pada sintak *shownSaifi=true; shownSaidi=true; shownCaidi=true; shownAsai=true;*. Kemudian dilakukan pengaturan terhadap nilai standar yang ditetapkan, ditunjukkan pada sintak *setStandartNilaiSaifi(1.45), setStandartNilaiSaidi(2.30), setStandartNilaiCaidi(1.47), dan setStandartNilaiAsai(99.92).*

```

if (inputData.getNilaiSaidi() < standarSaidi)
{holder.tvStandarSaidi.setText(context.getString(R.string.done));}
else
{holder.tvStandarSaidi.setText(context.getString(R.string.cross));}

if (inputData.getNilaiSaifi() < standarSaifi)
{holder.tvStandarSaifi.setText(context.getString(R.string.done));}
else
{holder.tvStandarSaifi.setText(context.getString(R.string.cross));}

if (inputData.getNilaiCaidi() < standarCaidi)
{holder.tvStandarCaidi.setText(context.getString(R.string.done));}
else
{holder.tvStandarCaidi.setText(context.getString(R.string.cross));}

if (inputData.getNilaiAsai() > standarAsai)
{holder.tvStandarAsai.setText(context.getString(R.string.done));}
else
{holder.tvStandarAsai.setText(context.getString(R.string.cross));}

```

Gambar 4.40 Kode Sumber Analisis Data

Pada kode sumber hasil analisis diatas yang dilakukan adalah membandingkan data, yaitu data nilai masing – masing indeks dengan data nilai standar yang ditetapkan. Kemudian apabila data telah dibandingkan dan hasilnya sudah sesuai nilai standar maka menampilkan simbol centang atau checklist yang pada sintak diatas diwakili oleh variabel *done*. Sedangkan apabila data nilai indeks tidak sesuai dengan data nilai standar yang ditetapkan maka akan menampilkan simbol silang yang pada sintak diatas diwakili oleh variabel *cross*.

```

for (int i = 0; dataSaifiSaidis1.size() > i; i++) {
    DataSaifiSaidi dataSaifiSaidi = dataSaifiSaidis1.get(i);
    AnalisaData analisaData = new AnalisaData();
    analisaData.setNamaFeeder(dataSaifiSaidi.getNamaFeeder());
    analisaData.setAnalisaSaifi("");
    analisaData.setAnalisaSaidi("");
    analisaData.setAnalisaCaidi("");
    analisaData.setAnalisaAsai("");
    if (dataSaifiSaidi.getStandartNilaiSaidi() != 0 &&
dataSaifiSaidi.getNilaiSaidi() >=
dataSaifiSaidi.getStandartNilaiSaidi()){analisaData.setAnalisaSaidi
("true");}
else if (dataSaifiSaidi.getStandartNilaiSaidi() != 0 &&
dataSaifiSaidi.getNilaiSaidi() <=
dataSaifiSaidi.getStandartNilaiSaidi()){analisaData.setAnalisaSaidi
("false"); }
    if (dataSaifiSaidi.getStandartNilaiSaifi() != 0 &&
dataSaifiSaidi.getNilaiSaifi() >=
dataSaifiSaidi.getStandartNilaiSaifi()){analisaData.setAnalisaSaifi
("true");}
else if (dataSaifiSaidi.getStandartNilaiSaifi() != 0 &&
dataSaifiSaidi.getNilaiSaifi() <=
dataSaifiSaidi.getStandartNilaiSaifi()){analisaData.setAnalisaSaifi
("false");}
    if (dataSaifiSaidi.getStandartNilaiCaidi() != 0 &&
dataSaifiSaidi.getNilaiCaidi() >=
dataSaifiSaidi.getStandartNilaiCaidi()){analisaData.setAnalisaCaidi
("true");}
else if (dataSaifiSaidi.getStandartNilaiCaidi() != 0 &&
dataSaifiSaidi.getNilaiCaidi() <=
dataSaifiSaidi.getStandartNilaiCaidi()){analisaData.setAnalisaCaidi
("false");}
    if (dataSaifiSaidi.getStandartNilaiAsai() != 0 &&
dataSaifiSaidi.getNilaiAsai() <=
dataSaifiSaidi.getStandartNilaiAsai()){analisaData.setAnalisaAsai
("true");}
else if (dataSaifiSaidi.getStandartNilaiAsai() != 0 &&
dataSaifiSaidi.getNilaiAsai() >=
dataSaifiSaidi.getStandartNilaiAsai()){analisaData.setAnalisaAsai
("false");}
    dataList.add(analisaData);}
for (int i = 0; dataList.size() > i; i++) {
    int pos = i + 1;
    String data = handleAnalisa(dataList.get(i).getAnalisaSaifi(),
dataList.get(i).getAnalisaSaidi(),
dataList.get(i).getAnalisaCaidi(),
dataList.get(i).getAnalisaAsai());
    if (i == 0) {tvAnalisa.setText(pos + ". Berdasarkan standar " +
standar + ", Feeder " + dataSaifiSaidis1.get(i).getNamaFeeder() + "
dinyatakan " + data);}
    else {tvAnalisa.append("\n\n" + pos + ". Berdasarkan
standar " + standar + ", Feeder " +
dataSaifiSaidis1.get(i).getNamaFeeder() + " dinyatakan " + data);}
}

```

Gambar 4.41 Kode Sumber Teks Analisis Data

```

if (falseData.size() == stringList.size())
{if (dataSaifiSaidis1.get(0).getStandartNilaiAsai() != 0)
{result = "handal karena seluruh indeks keandalan tidak melebihi
dan kurang dari batas yang ditetapkan.";}
else {result = "handal karena seluruh indeks keandalan tidak
melebihi dari batas yang ditetapkan.";}}

else if (trueData.size() == stringList.size()) {if
(dataSaifiSaidis1.get(0).getStandartNilaiAsai() != 0) {result =
"tidak handal karena semua indeks keandalan melebihi dan kurang
dari batas yang ditetapkan.";}
else {result = "tidak handal karena semua indeks keandalan
melebihi dari batas yang ditetapkan.";}}

else {if (dataSaifiSaidis1.get(0).getStandartNilaiAsai() != 0)
{result = "kurang handal karena ada indeks keandalan yang
melebihi atau kurang dari batas yang ditetapkan yaitu ";}
else {result = "kurang handal karena ada indeks keandalan yang
melebihi dari batas yang ditetapkan yaitu "};

    if (trueData.size() > 1) {
        for (int i = 0; trueData.size() > i; i++) {
            if (i < trueData.size() - 1)
{result = result + trueData.get(i).getNamaIndeks() + ", "};
            else if (i == trueData.size() - 1)
{result = result + "dan " + trueData.get(i).getNamaIndeks();}
        }
    }
    else {result = result + trueData.get(0).getNamaIndeks();}
}
}

```

Gambar 4.42 Kode Sumber Teks Analisis Data (lanjutan)

Pada gambar kode sumber teks analisis data diatas, dilakukan perulangan pengecekan data dari indeks SAIFI, SAIDI, CAIDI, dan ASAI yang nantinya data tersebut akan digolongkan pada list data *true* atau list data *false*. *True* berarti data bermasalah sedangkan *false* artinya data tidak bermasalah. Masalah yang dimaksud adalah apakah indeks tersebut sesuai standar yang ditetapkan atau tidak. Setelahnya list data tadi digunakan untuk analisa data. Analisa data dilakukan dengan menampilkan teks untuk menyatakan analisis (kalimat warna hijau) yang diulang oleh fungsi *for* sampai perulangan selesai. Terakhir adalah memasukan variabel *data* yang ada dalam fungsi *for* dengan logika yang ada.

Logika pertama jika semua data adalah *false* maka hasil datanya akan menampilkan *result* yang jika ada nilai asai maka akan berbunyi “handal karena seluruh indeks keandalan tidak melebihi dan kurang dari batas yang ditetapkan.”, jika tidak maka “handal karena seluruh indeks keandalan tidak melebihi dari batas yang ditetapkan.”.

Logika kedua jika semua data adalah *true* maka hasil datanya akan menampilkan *result* yang jika ada nilai asai maka akan berbunyi "tidak handal karena semua indeks keandalan melebihi dan kurang dari batas yang ditetapkan.", jika tidak maka "tidak handal karena semua indeks keandalan melebihi dari batas yang ditetapkan."

Logika ketiga jika ada data bernilai *true* dan *false* maka hasil datanya akan menampilkan *result* yang jika ada nilai asai maka akan berbunyi "kurang handal karena ada indeks keandalan yang melebihi atau kurang dari batas yang ditetapkan yaitu " jika tidak maka "kurang handal karena ada indeks keandalan yang melebihi dari batas yang ditetapkan yaitu ". Selanjutnya di belakang kata “yaitu” ada penyebutan nilai indeks yang tidak sesuai standar yang ditetapkan yang diambil dari list data bermasalah yaitu *trueData* yang kemudian dilakukan perulangan dari data pertama hingga terakhir untuk mencari data indeks apa saja yang bermasalah untuk ditampilkan. Jika indeks hanya satu maka akan langsung menampilkan teks indeks yang bermasalah tapi jika lebih dari satu indeks maka akan menampilkan teks yang mana ada ketambahan kata “dan” untuk yang mendekati satu sebelum terakhir.

4.5.6 Kode Sumber EXPORT

Kode sumber menu EXPORT berisi fungsi izin aplikasi untuk mengakses media penyimpanan, kode sumbernya seperti terlihat pada gambar 4.41. Fungsi database juga digunakan sebagai media untuk untuk mengubah format data hasil perhitungan ke dalam bentuk format file microsoft excel (.xls), kode sumbernya seperti terlihat pada gambar 4.43.

```

private void permissionReq() {
    Dexter.withActivity(MainActivity.this)

    .withPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE)
        .withListener(new PermissionListener() {
@Override public void onPermissionDenied(PermissionDeniedResponse
response) {
    Toast.makeText(MainActivity.this, "Permission ditolak, export
gagal", Toast.LENGTH_SHORT).show();
}
}
}

```

Gambar 4.43 Kode Sumber Izin Akses Media Penyimpanan

Pada kode sumber di atas menjelaskan adanya izin permisi untuk menuliskan data ke penyimpanan eksternal dari aplikasi dalam hal ini media penyimpanan telepon. Jika izin ditolak maka akan memunculkan teks kecil dengan kalimat “permission ditolak, export gagal”, maka jika ada perizinan seperti ini sebaiknya dituruti saja agar aplikasi bisa mengakses penyimpanan telepon dan menuliskan data export ke ruang yang dituju.

```

String directory_path =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS).getPath();
SQLiteToExcel sqliteToExcel = new
SQLiteToExcel(context:MainActivity.this,
dbName:"indeks_keandalan", directory_path);
sqliteToExcel.exportSingleTable(table:"hasil",
fileName"hasil.xls", new SQLiteToExcel.ExportListener() {
}
}

```

Gambar 4.44 Kode Sumber Fungsi *Export to Excel*

Pada kode sumber di atas menjelaskan adanya penggunaan database SQLite guna media perantara agar data bisa diexport dalam file excel. Bisa dilihat bahwa hasil dari file export akan ditempatkan di file download, kemudian untuk nama dari filenya adalah hasil.xls.

4.5.7 Kode Sumber ABOUT

Kode sumber menu ABOUT berisi fungsi penampil gambar, fungsi penampil text dan fungsi center teks. Berikut adalah kode sumber xml nya seperti terlihat pada gambar 4.45, namun hanya ditampilkan sebagian saja sebagai contoh. Untuk kode sumber yang lebih lengkap bisa dilihat di halaman lampiran.

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical">
    <ImageView
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:src="@drawable/kalinda3" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="KALINDA\n
        KALKULATOR INDEKS KEANDALAN\n
        versi 1.0.0\n" />
</LinearLayout>

```

Gambar 4.45 Kode Sumber Menu ABOUT

Kode sumber diatas adalah sebagian contoh kode sumber bahasa XML pada menu ABOUT. Sedikit penjelasan untuk gambar diatas, *LinearLayout* adalah tampilan yang hanya bisa menambahkan komponen dengan orientasi vertikal atau horizontal. *Layout_width* adalah variable untuk mengatur lebar tampilan sedangkan *match_parent* adalah nilai dari variable yang berisi lebar sesuai ukuran *handphone*. Sama halnya dengan *layout_height* hanya beda pada mengatur panjang tampilan saja. *Layout_margin* untuk mengatur jarak konten berdasarkan posisi atau sudutnya. Variabel *orientation* guna mengatur oreientasi dari tampilan apakah akan horizontal atau vertikal. *ImageView* untuk menampilkan gambar sedangkan *TextView* untuk menampilkan tulisan atau teks.

4.5.8 Kode Sumber MANUALS

Kode sumber menu MANUALS berisi fungsi penampil file (.txt) seperti terlihat pada gambar 4.46. Serta fungsi teks rata kanan kiri seperti terlihat pada gambar 4.47.

```
reader = new BufferedReader(  
    new InputStreamReader(getAssets().open("Panduan.txt")));
```

Gambar 4.46 Kode Sumber Pembaca File TXT

Pada kode sumber di atas menjelaskan adanya fungsi pembaca teks oleh fungsi *reader* yang mana file yang akan ditampilkan berada pada folder *asset*.

```
JustifyTextView output = findViewById(R.id.tvText);  
output.setText(text);
```

Gambar 4.47 Kode Sumber Justify Teks

justifyTextView output berfungsi untuk mengatur teks yang akan ditampilkan dalam format rata kanan kiri atau biasa disebut juga justify yang akan memberikan kesan rapi pada dokumen tersebut.

4.5.9 User Manual

User manual merupakan sebuah panduan yang ditujukan untuk pengguna dari aplikasi KALINDA. Di bawah ini adalah beberapa panduan mulai dari *install*, penggunaan, dan *uninstall*.

Cara *Install* Aplikasi :

1. Klik Apk yang akan di *install* dalam hal ini yaitu KALINDA.apk.
2. User akan diarahkan ke pengaturan untuk mengizinkan instalasi aplikasi dari sumber selain Play Store. Kemudian centangkan atau perbolehkan.
3. Klik izinkan instalasi kali ini saja
4. Lalu klik OK.
5. Klik install atau pasang
6. Setelah itu selesai klik *launch* atau buka.

Cara Penggunaan Aplikasi :

1. Buka aplikasi KALINDA
2. Lakukan perhitungan data terlebih dahulu pada menu DATA ENTRY

3. Lihat hasil perhitungan pada menu RESULTS
4. Lihat analisis data pada menu ANALYSIS
5. Untuk export hasil data bisa dengan menekan menu EXPORT, ketika file sudah terexport maka bisa dilihat pada media penyimpanan telepon dalam folder download dengan nama hasil.xls
6. Untuk menu INTRODUCTION, ABOUT, dan MANUALS bisa langsung diakses tanpa perlu melakukan perhitungan terlebih dahulu seperti menu-menu diatas.

Cara *Uninstall* Aplikasi :

1. Pergi ke pengaturan
2. Pilih *installed apps*
3. Cari aplikasi KALINDA dan klik
4. Klik *uninstall* terletak di bagian bawah
5. Lalu klik ok
6. Setelah selesai klik ok lagi pada aplikasi sudah terhapus

4.6 Validasi Hasil dan Analisis

Pada tahap akhir dari pembahasan tugas akhir ini adalah validasi hasil atau membandingkan hasil dari perhitungan aplikasi KALINDA dengan hasil perhitungan secara manual. Setelah dibandingkan langkah selanjutnya yakni dilakukan analisis terhadap data yang ada.

4.6.1 Validasi Hasil

Dalam pembahasan ini objek yang dibandingkan adalah nilai hasil dari perhitungan indeks keandalan. Komparasi hasil perhitungan dari dua metode perhitungan yaitu metode manual dengan menggunakan kalkulator biasa dan metode otomatis dengan menggunakan aplikasi KALINDA. Hal tersebut dilakukan guna melakukan pengecekan apakah data dari sumber yang sama hasilnya akan sama atau tidak. Di bawah ini akan ditampilkan tabel 4.12 tentang validasi hasil perhitungan dari dua metode tersebut.

Tabel 4.8 Validasi Hasil Perhitungan

No.	Nama Feeder	Perhitungan Manual				Perhitungan KALINDA				Keterangan
		SAIFI (k/p/t)	SAIDI (j/p/t)	CAIDI (j/k/t)	ASAI (%)	SAIFI (k/p/t)	SAIDI (j/p/t)	CAIDI (j/k/t)	ASAI (%)	
1	PKL01	0.295	0.185	0.628	99.99	0.296	0.186	0.628	99.99	Valid
2	PKL02	0.101	0.162	1.604	99.99	0.101	0.162	1.604	99.99	Valid
3	PKL04	0.355	0.197	0.553	99.99	0.355	0.197	0.555	99.99	Valid
4	PKL06	0.349	0.147	0.421	99.99	0.35	0.147	0.42	99.99	Valid
5	PKL07	0.495	0.195	0.39	99.99	0.495	0.195	0.394	99.99	Valid
6	PKL08	0.098	0.146	1.49	99.99	0.098	0.146	1.49	99.99	Valid
7	PKL09	0	0	0	100	0	0	0	100	Valid
8	PKL10	0.161	0.166	1.03	99.99	0.161	0.166	1.031	99.99	Valid
9	PKL11	0.025	0.026	1.04	99.99	0.025	0.026	1.04	99.99	Valid
10	PKL12	0.383	0.293	0.765	99.99	0.383	0.293	0.765	99.99	Valid
11	PKL14	0.049	0.053	1.082	99.99	0.049	0.053	1.082	99.99	Valid
12	PKL15	0.021	0.067	3.19	99.99	0.021	0.067	3.19	99.99	Valid

Berdasarkan data yang telah tersaji di atas, yaitu data hasil perbandingan antara perhitungan manual dengan perhitungan aplikasi menunjukkan angka yang relatif sama, sebagai contoh pada feeder PKL01. Pada feeder PKL01 nilai indeks SAIFI 0.295 untuk perhitungan manual dan 0.256 untuk perhitungan KALINDA. SAIDI 0.185 pada perhitungan manual dan 0.186 untuk perhitungan KALINDA. CAIDI 0.628 pada perhitungan manual dan 0.628 untuk perhitungan KALINDA. ASAI 99.99 pada perhitungan manual dan 99.99 untuk perhitungan KALINDA. Pada data hasil tersebut ada sedikit perbedaan angka dibelakang koma yang disebabkan oleh adanya pembulatan angka pada aplikasi KALINDA, namun hal tersebut tidak mengurangi keakuratan perhitungan yang dihasilkan. Maka dapat disimpulkan bahwa perhitungan pada aplikasi KALINDA sudah sesuai dengan perhitungan manual yang artinya perhitungan KALINDA sudah terverifikasi valid. Sehingga aplikasi KALINDA bisa dikatakan lolos uji dari segi keakuratan perhitungan.

4.6.2 Analisis Hasil

Pada analisis hasil indeks keandalan akan ditampilkan dalam dua tabel yaitu pertama tabel 4.9 tentang analisis menurut standar PLN Rayon Kota Pekalongan dan SPLN. Kedua tabel 4.10 tentang analisis menurut standar IEEE.

Tabel 4.9 Analisis PLN Rayon Kota Pekalongan dan SPLN

No.	Feeder	Nilai SAIFI	Nilai SAIDI	PLN Rayon Kota Pekalongan		SPLN	
				SAIFI 3.62 (k/p/t)	SAIDI 9.77 (j/p/t)	SAIFI 3.2 (k/p/t)	SAIDI 21.09 (j/p/t)
1	PKL01	0.295	0.185	✓	✓	✓	✓
2	PKL02	0.101	0.162	✓	✓	✓	✓
3	PKL04	0.355	0.197	✓	✓	✓	✓
4	PKL06	0.349	0.147	✓	✓	✓	✓
5	PKL07	0.495	0.195	✓	✓	✓	✓
6	PKL08	0.097	0.146	✓	✓	✓	✓
7	PKL09	0	0	✓	✓	✓	✓
8	PKL10	0.161	0.166	✓	✓	✓	✓
9	PKL11	0.025	0.026	✓	✓	✓	✓
10	PKL12	0.383	0.29	✓	✓	✓	✓
11	PKL14	0.049	0.053	✓	✓	✓	✓
12	PKL15	0.021	0.067	✓	✓	✓	✓

Berdasarkan tabel 4.9 diatas bisa didapat kesimpulan bahwa nilai indeks SAIFI dan SAIDI pada semua feeder pada sistem distribusi Rayon Kota Pekalongan menurut standar PLN Rayon Kota Pekalongan yaitu 3.62 (kali/pelanggan/tahun) dan 9.77 (jam/pelanggan/tahun) dan standar SPLN yaitu 3.2 (kali/pelanggan/tahun) dan 21.09 (jam/pelanggan/tahun), menunjukkan hasil centang semua yang menandakan nilai dari tiap indeks SAIFI dan SAIDI tidak ada yang melebihi batas yang ditetapkan dari kedua standar yang telah tersebut di atas. Oleh karena itu maka setiap feeder yang melayani kebutuhan listrik di Kota Pekalongan dikategorikan handal menurut dua macam standar yaitu standar PLN Rayon Kota Pekalongan dan standar SPLN.

Tabel 4.10 Analisis IEEE

No.	Feeder	Nilai SAIFI	Nilai SAIDI	Nilai CAIDI	Nilai ASAI	IEEE			
						SAIFI < 1.45 (k/p/t)	SAIDI < 2.3 (j/p/t)	CAIDI < 1.47 (j/k/t)	ASAI > 99.92 (%)
1	PKL01	0.295	0.185	0.628	99.99	✓	✓	✓	✓
2	PKL02	0.101	0.162	1.604	99.99	✓	✓	×	✓
3	PKL04	0.355	0.197	0.555	99.99	✓	✓	✓	✓
4	PKL06	0.349	0.147	0.42	99.99	✓	✓	✓	✓
5	PKL07	0.495	0.195	0.394	99.99	✓	✓	✓	✓
6	PKL08	0.097	0.146	1.49	99.99	✓	✓	×	✓
7	PKL09	0	0	0	100	✓	✓	✓	✓
8	PKL10	0.161	0.166	1.031	99.99	✓	✓	✓	✓
9	PKL11	0.025	0.026	1.04	99.99	✓	✓	✓	✓
10	PKL12	0.383	0.29	0.765	99.99	✓	✓	✓	✓
11	PKL14	0.049	0.053	1.082	99.99	✓	✓	✓	✓
12	PKL15	0.021	0.067	3.19	99.99	✓	✓	×	✓

Dengan melihat data analisis pada tabel 4.10 diatas bisa didapat kesimpulan bahwa mayoritas setiap feeder pada sistem distribusi Rayon Kota Pekalongan dinyatakan handal kecuali tiga feeder yaitu feeder PKL02, PKL08, PKL15. Ketiga feeder tersebut dinyatakan dalam status kurang handal karena suatu hal yang sama yaitu nilai indeks CAIDI yang telah melebihi batas standar yang telah ditetapkan oleh pihak IEEE yaitu 1.47 jam/kali/tahun. Sedangkan bisa dilihat pada data feeder PKL02 nilai CAIDI menunjukkan angka 1.604 jam/kali/tahun, feeder PKL08 menunjukkan angka 1.49 jam/kali/tahun, feeder PKL15 menunjukkan angka 3.19 jam/kali/tahun. Oleh karena hal tersebut, berdasarkan standar IEEE feeder yang dinyatakan handal adalah PKL01, PKL04, PKL06, PKL07, PKL09, PKL10, PKL11, PKL12, PKL14 karena nilai indeks SAIFI, SAIDI, CAIDI tidak melebihi batas dan nilai indeks ASAI tidak kurang dari batas yang ditetapkan oleh pihak IEEE. Sedangkan feeder yang lainnya yaitu tiga feeder yang telah tersebut tadi dalam kategori kurang handal.