

BAB IV

HASIL DAN PEMBAHASAN

4.1 Jumlah Pelanggan Setiap Penyulang di Gardu Induk Batang

Pada gardu induk Batang memiliki 12 buah penyulang, penyulang tersebut yaitu BTG01, BTG02, BTG03, BTG04, BTG06, BTG07, BTG08, BTG09, BTG10, BTG11, BTG12, BTG13. Total pelanggan dari seluruh penyulang gardu induk Batang adalah sebesar 147849 Jiwa pada tahun 2018.

Pada tabel 4.1 berikut menunjukkan rincian jumlah pelanggan setiap penyulang Gardu Induk Batang tahun 2018.

Tabel 4.1 Data Pelanggan pada Setiap Penyulang di GI Batang Tahun 2018

No	Penyulang	Jumlah Pelanggan
1	BTG01	19130
2	BTG02	22335
3	BTG03	5903
4	BTG04	1
5	BTG06	23720
6	BTG07	9432
7	BTG08	2
8	BTG09	24554
9	BTG10	18785
10	BTG11	1
11	BTG12	1
12	BTG13	23985
Total		147849

4.2 Data Gangguan Penyulang Gardu Induk Batang Tahun 2018

Jumlah data trip trafo selama tahun 2018 meliputi data waktu pematama dan waktu masuk kembali pada masing-masing penyulang ditunjukkan pada tabel 4.2 dan tabel 4.3. Perhitungan frekuensi dan durasi lama padam di setiap penyulang dapat diketahui dari data di bawah ini.

Tabel 4.2 Data gangguan penyulang di GI Batang tahun 2018

No	Penyulang	Jam Padam	Jam Nyala	Lama Padam (menit)	Bulan
1	BTG01	00.26	00.29	3	Januari
2	BTG07	19.58	22.08	130	Februari
3	BTG01	13.58	15.07	69	Februari
4	BTG03	13.58	15.05	67	Februari
5	BTG01	17.33	18.18	45	Februari
6	BTG13	02.17	02.43	26	Maret
7	BTG07	10.47	12.36	109	Maret
8	BTG07	14.02	15.49	107	Maret
9	BTG09	10.38	10.58	20	April
10	BTG04	12.14	12.55	41	April
11	BTG02	15.46	20.37	291	April
12	BTG06	08.45	09.46	61	April
13	BTG07	12.04	12.47	43	April
14	BTG01	18.37	20.21	104	April
15	BTG03	18.37	20.57	140	April
16	BTG02	14.43	16.24	101	Mei
17	BTG01	23.30	02.46	196	Juni
18	BTG01	22.14	03.06	292	Juni

Tabel 4.3 Data gangguan penyulang di GI Batang tahun 2018 (lanjutan)

No	Penyulang	Jam Padam	Jam Nyala	Lama Padam (menit)	Bulan
19	BTG12	09.34	09.36	2	Agustus
20	BTG01	03.28	03.38	10	Agustus
21	BTG08	10.06	10.48	42	September
22	BTG03	20.27	22.39	132	September
23	BTG12	04.21	09.49	328	Oktober
24	BTG03	13.03	13.38	35	Desember
25	BTG07	01.58	03.10	72	Desember
26	BTG01	10.22	11.14	52	Desember
27	BTG03	10.23	12.12	109	Desember

4.3 Perhitungan Nilai Indeks Keandalan dengan Cara *Manual*

Pada penelitian ini perhitungan nilai indeks keandalan dilakukan dengan cara manual yang mana masing – masing nilai SAIFI, SAIDI, dan CAIDI dihitung satu persatu dengan menggunakan rumus yang digunakan pada indeks keandalan.

4.3.1 Perhitungan SAIFI pada Setiap Penyulang

Nilai SAIFI didapatkan dari jumlah pelanggan yang mengalami, frekuensi gangguan, dan jumlah total dari semua pelanggan. Data frekuensi gangguan dan jumlah pelanggan setiap penyulang di Gardu Induk Batang pada tabel 4.4 di bawah ini untuk memudahkan pengamatan dan perhitungan.

Tabel 4.4 Data Frekuensi Gangguan Penyulang di Gardu Induk Batang 2018

No	Penyulang	Frekuensi Gangguan	Pelanggan
1	BTG01	7	19130
2	BTG02	2	22335
3	BTG03	5	5903
4	BTG04	1	1
5	BTG06	1	23720
6	BTG07	5	9432
7	BTG08	1	2
8	BTG09	2	24554
9	BTG10	0	18785
10	BTG11	0	1
11	BTG12	2	1
12	BTG13	1	23985
Total Pelanggan			147849

Perhitungan nilai SAIFI dari masing-masing penyulang dengan menggunakan rumus perhitungan sebagai berikut:

$$SAIFI = \frac{\lambda i \times Ni}{Nt}$$

Keterangan

λi = Frekuensi pemadaman / Angka kegagalan

Ni = Jumlah pelanggan yang terkena gangguan

Nt = Jumlah pelanggan

Pada tabel 4.5 berikut ini menunjukkan hasil perhitungan nilai SAIFI setiap penyulang Gardu Induk Batang pada tahun 2018 dengan jumlah total pelanggan pada Gardu Induk Batang yaitu 147849 pelanggan.

Tabel 4.5 Nilai SAIFI setiap Penyulang Gardu Induk Batang

No	Penyulang	Frekuensi Gangguan	Pelanggan	Nilai SAIFI (Kali/Pelanggan/Tahun)
1	BTG01	7	19130	0.906
2	BTG02	2	22335	0.302
3	BTG03	5	5903	0.199
4	BTG04	1	1	0.0000068
5	BTG06	1	23720	0.160
6	BTG07	5	9432	0.319
7	BTG08	1	2	0.000013
8	BTG09	2	24554	0.332
9	BTG10	0	18785	0
10	BTG11	0	1	0
11	BTG12	2	1	0.000013
12	BTG13	1	23985	0.162
Total			147849	2.301

4.3.2 Perhitungan SAIDI pada Setiap Penyulang

Nilai SAIDI didapatkan dari jumlah pelanggan yang mengalami, durasi gangguan, dan jumlah total dari semua pelanggan. Data durasi gangguan dan jumlah pelanggan setiap penyulang di Gardu Induk Batang pada tabel 4.6 di bawah ini untuk memudahkan pengamatan dan perhitungan.

Perhitungan nilai SAIDI dari masing-masing penyulang dengan menggunakan rumus perhitungan sebagai berikut:

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

Keterangan

U_i = Durasi gangguan

N_i = Jumlah pelanggan yang terkena gangguan

N_t = Jumlah pelanggan

Pada tabel 4.6 berikut ini menunjukkan hasil perhitungan nilai SAIDI setiap penyulang Gardu Induk Batang pada tahun 2018 dengan jumlah total pelanggan pada Gardu Induk Batang yaitu 147849 pelanggan.

Tabel 4.6 Nilai SAIDI setiap Penyulang di Gardu Induk Batang

No	Penyulang	Durasi Gangguan (jam)	Pelanggan	Nilai SAIDI (Jam/Pelanggan/Tahun)
1	BTG01	12.8	19130	1.656
2	BTG02	6.53	22335	0.986
3	BTG03	8.05	5903	0.26
4	BTG04	0.68	1	0.0000046
5	BTG06	1.02	23720	0.164
6	BTG07	7.68	9432	0.49
7	BTG08	0.7	2	0.0000095
8	BTG09	0.38	24554	0.0631
9	BTG10	0	18785	0
10	BTG11	0	1	0
11	BTG12	5.5	1	0.000037
12	BTG13	0.43	23985	0.07
Total		43.77	147849	3.750

4.3.3 Perhitungan CAIDI pada Setiap Penyulang

Nilai CAIDI didapatkan dari nilai SAIFI dan SAIDI pada setiap penyulang.

Untuk memudahkan pengamatan dan perhitungan, nilai SAIFI dan SAIDI setiap penyulang di Gardu Induk Batang ditulis pada tabel 4.7 di bawah ini.

Perhitungan nilai CAIDI dari masing-masing penyulang dengan menggunakan rumus perhitungan sebagai berikut:

$$CAIDI = \frac{\sum U_i \times N_i}{\sum \lambda_i \times N_i}$$

Keterangan

U_i = Durasi gangguan

λ_i = Frekuensi pemadaman / Angka kegagalan

N_i = Jumlah pelanggan

Tabel 4.7 Nilai CAIDI setiap Penyulang di Gardu Induk Batang

No	Penyulang	Nilai SAIFI (Kali/Pwlanggan/Tahun)	Nilai SAIDI (Jam/Pelanggan/Tahun)	Nilai CAIDI (Jam/Kali/Tahun)
1	BTG01	0.906	1.656	1.828
2	BTG02	0.302	0.986	3.265
3	BTG03	0.199	0.26	1.61
4	BTG04	0.0000068	0.0000046	0.68
5	BTG06	0.160	0.164	1.02
6	BTG07	0.319	0.49	1.536
7	BTG08	0.000013	0.0000095	0.7
8	BTG09	0.332	0.0631	0.19
9	BTG10	0	0	NaN
10	BTG11	0	0	NaN
11	BTG12	0.000013	0.000037	2.75
12	BTG13	0.162	0.07	0.43

4.3.4 Perhitungan ASAI pada Setiap Penyulang

Nilai ASAI didapatkan dari nilai CAIDI dan pada setiap penyulang. Untuk memudahkan pengamatan dan perhitungan, nilai SAIDI setiap penyulang di Gardu Induk Batang ditulis pada tabel 4.8 di bawah ini.

Perhitungan nilai ASAI dari masing-masing penyulang dengan menggunakan rumus perhitungan sebagai berikut:

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

Keterangan

8760 = Jumlah Total Jam dalam Setahun

Tabel 4.8 Nilai ASAI setiap Penyulang di Gardu Induk Batang

No	Penyulang	Nilai SAIDI (Jam/Kali/Tahun)	Nilai ASAI (%)
1	BTG01	1.656	99.98
2	BTG02	0.986	99.98
3	BTG03	0.26	99.99
4	BTG04	0.0000046	100
5	BTG06	0.164	99.99
6	BTG07	0.49	99.99
7	BTG08	0.0000095	100
8	BTG09	0.0631	99.99
9	BTG10	0	100
10	BTG11	0	100
11	BTG12	0.000037	100
12	BTG13	0.07	99.99

4.3.5 Perhitungan ASUI pada Setiap Penyulang

Nilai ASUI didapatkan dari nilai ASAI pada setiap penyulang. Untuk memudahkan pengamatan dan perhitungan, nilai ASAI setiap penyulang di Gardu Induk Batang ditulis pada tabel 4.9 di bawah ini.

Perhitungan nilai ASUI dari masing-masing penyulang dengan menggunakan rumus perhitungan sebagai berikut:

$$ASUI = 1 - ASAI$$

Tabel 4.9 Nilai ASUI setiap Penyulang di Gardu Induk Batang

No	Penyulang	Nilai ASAI (%)	Nilai ASUI
1	BTG01	99.98	-98.98
2	BTG02	99.98	0.0
3	BTG03	99.99	-98.99
4	BTG04	100	-99.0
5	BTG06	99.99	-98.99
6	BTG07	99.99	-98.99
7	BTG08	100	-99.0
8	BTG09	99.99	-98.99
9	BTG10	100	-99.0
10	BTG11	100	-99.0
11	BTG12	100	-99.0
12	BTG13	99.99	-98.99

4.3.6 Analisis Hasil Perhitungan

Berdasarkan indeks nilai keandalan PLN Rayon Batang penyulang di Gardu Induk Batang nilai total SAIFI dan SAIDI tidak melebihi batas maksimum yang sudah ditentukan, maka dapat dikategorikan handal. Setiap penyulang di

Gardu Induk dapat dikatakan handal karena nilai SAIFI nya tidak melebihi dari batas maksimum SPLN 68 – 2 : 1986 yaitu 3.2 kali/pelanggan/tahun. Selain itu nilai SAIDI dari Gardu Induk Batang tidak melebihi batas maksimum SPLN 68 – 2 : 1986 yaitu 21.09 jam/pelanggan/tahun. Nilai – nilai CAIDI yang diperoleh dapat dikategorikan handal kecuali pada penyulang BTG01, BTG02, BTG03, BTG07, BTG12 karena nilainya belum mencapai target yang telah ditentukan oleh IEEE std 1366 – 2003 yaitu sebesar 1.47 jam/kali/tahun. Validasi nilai Keandalan Gardu Induk Batang dengan standar SPLN 68 – 2 : 1986 lebih jelasnya dapat dilihat pada tabel 4.10.

Tabel 4.10 Perbandingan Nilai Keandalan dengan Standar SPLN

No	Penyulang	Nilai SAIFI	Nilai SAIDI	Nilai CAIDI	SPLN	
					SAIFI 3.2 (K/P/T)	SAIDI 21.09 (J/P/T)
1	BTG01	0.906	1.656	1.828	✓	✓
2	BTG02	0.302	0.986	3.265	✓	✓
3	BTG03	0.199	0.26	1.61	✓	✓
4	BTG04	0.0000068	0.0000046	0.68	✓	✓
5	BTG06	0.160	0.164	1.02	✓	✓
6	BTG07	0.319	0.49	1.536	✓	✓
7	BTG08	0.000013	0.0000095	0.7	✓	✓
8	BTG09	0.332	0.0631	0.19	✓	✓
9	BTG10	0	0	NaN	✓	✓
10	BTG11	0	0	NaN	✓	✓
11	BTG12	0.000013	0.000037	2.75	✓	✓
12	BTG13	0.162	0.07	0.43	✓	✓

Tabel 4.11 Perbandingan Nilai Keandalan dengan Standar IEEE

No	Feeder	Nilai SAIFI	Nilai SAIDI	Nilai CAIDI	IEEE		
					SAIFI 1.45 (K/P/T)	SAIDI 2.3 (J/P/T)	CAIDI 1.47 (J/K/T)
1	BTG01	0.906	1.656	1.828	✓	✓	×
2	BTG02	0.302	0.986	3.265	✓	✓	×
3	BTG03	0.199	0.26	1.61	✓	✓	×
4	BTG04	0.0000068	0.0000046	0.68	✓	✓	✓
5	BTG06	0.160	0.164	1.02	✓	✓	✓
6	BTG07	0.319	0.49	1.536	✓	✓	×
7	BTG08	0.000013	0.0000095	0.7	✓	✓	✓
8	BTG09	0.332	0.0631	0.19	✓	✓	✓
9	BTG10	-	-	-	✓	✓	✓
10	BTG11	-	-	-	✓	✓	✓
11	BTG12	0.000013	0.000037	2.75	✓	✓	×
12	BTG13	0.162	0.07	0.43	✓	✓	✓

Keterangan :

(K/P/T) = Kali / Pelanggan / Tahun

(J/P/T) = Jam / Pelanggan / Tahun

(J/K/T) = Jam / Kali / Tahun

✓ = Memenuhi standar yang ditentukan

× = Tidak memenuhi standar yang ditentukan

- = Tidak ada

4.4 Desain *Software* Aplikasi Nilai Keandalan

Desain *software* aplikasi merupakan tampilan antarmuka yang menghubungkan aplikasi dengan user. Untuk mempermudah *user* maka tampilan aplikasi dirancang sesederhana mungkin. Berikut merupakan daftar tampilan desain yang ada pada aplikasi kalkulator nilai keandalan.

4.4.1 Tampilan Halaman *Start*

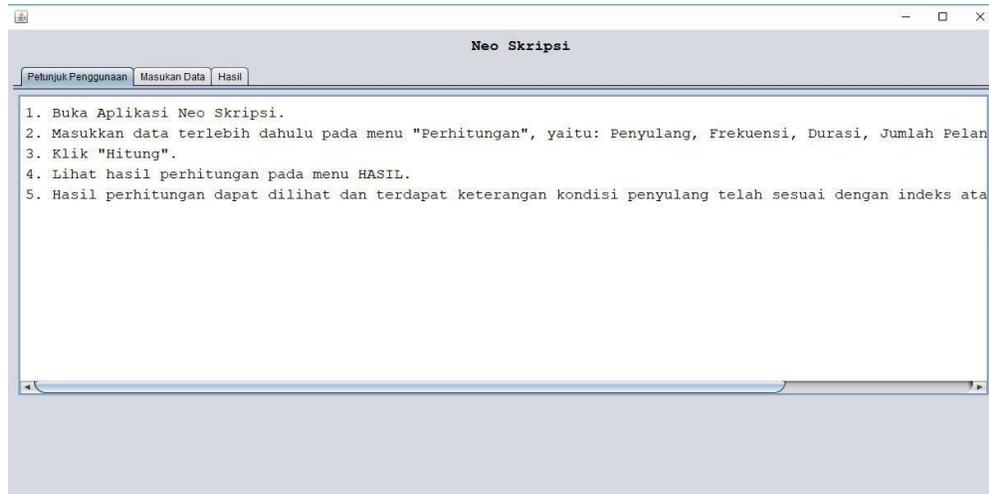
Tampilan halaman *start* merupakan halaman awal yang muncul pada saat aplikasi dijalankan oleh *user*. Halaman *start* digunakan untuk menunjukkan bahwa aplikasi siap dijalankan. Terdapat tombol “*START*” yang jika diklik oleh *user* maka akan menuju ke menu utama dari aplikasi ini. Tampilan halaman *start* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan Halaman *Start*

4.4.2 Tampilan Menu Utama

Pada saat aplikasi bekerja menu utama aplikasi ini terdapat 3 bar yang terbagi atas petunjuk penggunaan, masukan data, dan hasil. Pada bar petunjuk penggunaan hanya berisi petunjuk penggunaan aplikasi dari membuka aplikasi hingga hasil perhitungan dapat ditampilkan. Tampilan halaman petunjuk penggunaan dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan Petunjuk penggunaan

Setelah *user* mengetahui cara penggunaan ini maka *user* dapat langsung memasukkan data yang diperlukan dalam perhitungan yang meliputi nama penyulang, frekuensi terjadinya gangguan, durasi lamanya gangguan, jumlah pelanggan yang terkena gangguan pada setiap penyulang, dan total pelanggan. Lalu setelah selesai melakukan pengisian masukan data klik tombol "Hitung". Tampilan halaman masukan data dapat dilihat pada Gambar 4.3.

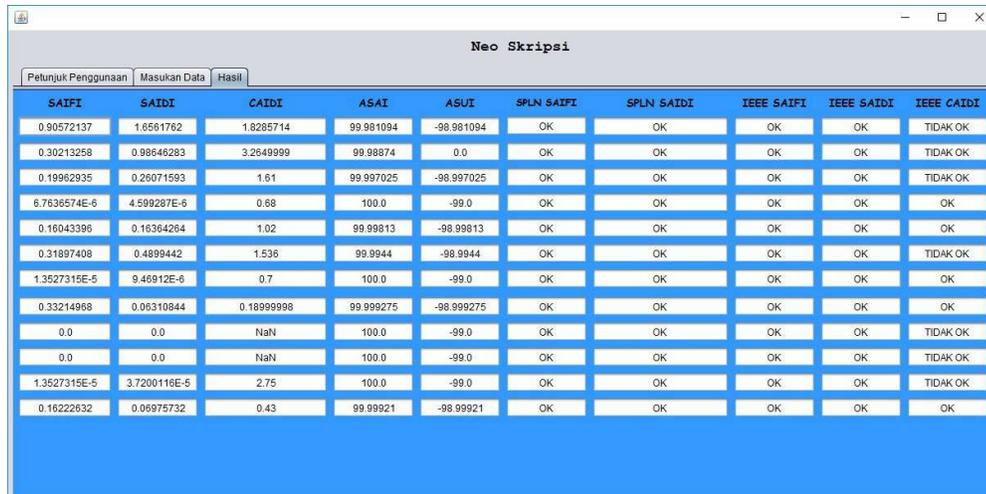
The screenshot shows the "Masukan Data" tab of the "Neo Skripsi" application. It contains a table with the following data:

Penyulang	Frekuensi	Durasi	Banyak Pelanggan
BTG01	7	12.8	19130
BTG02	2	6.53	22335
BTG03	5	8.05	5903
BTG04	1	0.68	1
BTG06	1	1.02	23720
BTG07	5	7.68	9432
BTG08	1	0.7	2
BTG09	2	0.38	24554
BTG10	0	0	18785
BTG11	0	0	1
BTG12	2	5.5	1
BTG13	1	0.43	23985

Below the table, there is a label "Masukkan Total Pelanggan" followed by an input field containing the value "147849" and a "Hitung" button.

Gambar 4.3 Tampilan Masukan Data

Pada saat selesai melakukan pengisian data yang diperlukan untuk di proses maka data dapat ditampilkan pada halaman Hasil. Keluaran pada aplikasi ini akan menampilkan nilai SAIFI, nilai SAIDI, nilai CAIDI, nilai ASAI, dan nilai ASUI di setiap penyulang dengan kondisi masing-masing penyulang. Tampilan hasil dapat dilihat pada Gambar 4.4.



SAIFI	SAIDI	CAIDI	ASAI	ASUI	SPLN SAIFI	SPLN SAIDI	IEEE SAIFI	IEEE SAIDI	IEEE CAIDI
0.90572137	1.6561762	1.8285714	99.981094	-98.981094	OK	OK	OK	OK	TIDAK OK
0.30213258	0.98646283	3.2649999	99.98874	0.0	OK	OK	OK	OK	TIDAK OK
0.19962935	0.26071593	1.61	99.997025	-98.997025	OK	OK	OK	OK	TIDAK OK
6.7636574E-6	4.599287E-6	0.68	100.0	-99.0	OK	OK	OK	OK	OK
0.16043396	0.16364264	1.02	99.99813	-98.99813	OK	OK	OK	OK	OK
0.31897408	0.4899442	1.536	99.9944	-98.9944	OK	OK	OK	OK	TIDAK OK
1.3527315E-5	9.46912E-6	0.7	100.0	-99.0	OK	OK	OK	OK	OK
0.33214968	0.06310844	0.1899998	99.999275	-98.999275	OK	OK	OK	OK	OK
0.0	0.0	NaN	100.0	-99.0	OK	OK	OK	OK	TIDAK OK
0.0	0.0	NaN	100.0	-99.0	OK	OK	OK	OK	TIDAK OK
1.3527315E-5	3.7200116E-5	2.75	100.0	-99.0	OK	OK	OK	OK	TIDAK OK
0.16222632	0.06975732	0.43	99.99921	-98.99921	OK	OK	OK	OK	OK

Gambar 4.4 Tampilan Hasil

4.4.3 Pembahasan Keluaran Aplikasi pada Penyulang Batang

Keluaran pada aplikasi ini menghasilkan angka dari hitungan yang telah diproses. Untuk mendapatkan keluaran pada aplikasi maka masukan pada aplikasi harus terisi semua, jika tidak maka aplikasi akan *error*. Pada saat semua masukan yang diperlukan telah terisi maka *user* dapat mengklik tombol “Hitung” yang nantinya aplikasi akan bekerja sehingga menghasilkan keluaran dari aritmatika yang telah dibentuk dari *source code*. Angka keluaran yang tertampil dapat dianalisis oleh *user* dengan cara membandingkan hasil dengan standart yang telah ditetapkan.

4.5 Source Code

Pada penelitian ini *source code* yang digunakan untuk membuat atau merancang aplikasi merupakan komponen dasar sebuah program komputer yang dirancang oleh penulis. *Source code* ini berupa teks yang nantinya akan memiliki fungsi pada setiap pakatnya dan kegunaannya. Beberapa macam *source code* yang digunakan oleh penulis dijabarkan sebagai berikut.

4.5.1 Halaman Start

Source code yang digunakan pada halaman *start* ini cukup sederhana karena hanya menampilkan satu label judul dan tombol “*START*” yang harus diklik untuk menuju menu utama. Berikut *source code* halaman *start* ditampilkan pada gambar 4.5.

```
Penghitung penghitung=new Penghitung();  
penghitung.setVisible(true);  
this.setVisible(false);
```

Gambar 4.5 *Source Code* Halaman *Start*

4.5.2 Pengambilan Nilai

Paket *source code* yang digunakan pengambilan nilai pada aplikasi ini memiliki inti yang sama hanya saja pada tiap baris berbeda alamat. Sebagai contoh bila nilai yang akan diambil merupakan nilai yang berada pada baris ke – 1 maka alamatnya akan a1, b1, c1, ta1, tb1, dan tc1. Begitu halnya dengan nilai yang akan diambil merupakan nilai yang berada pada baris ke – 2 maka alamatnya akan a2, b2, c2, ta2, tb2, dan tc2 dan akan berlanjut pada baris seterusnya. *Source code* pengambilan nilai ini dicontohkan pada Gambar 4.6.

```
a1=ta1.getText();  
b1=tb1.getText();  
c1=tc1.getText();
```

Gambar 4.6 *Source Code* Pengambilan Nilai

4.5.3 Pengubah Nilai Huruf Menjadi Nilai Angka

Pada pengubah nilai huruf menjadi nilai angka ini diperlukan “*Float.parseFloat*” pada *source code* karena berfungsi untuk mengubah nilai masukan huruf menjadi nilai angka. Hal tersebut diperlukan agar masukan dapat diproses karena masukan yang diperlukan oleh *user* berupa angka. *Source code* yang digunakan dari baris ke – 1 hingga baris ke – 12 sama hanya saja merubah alamat ketika yang akan dihitung berada pada baris ke berapa. Sebagai contoh bila nilai yang akan diambil merupakan nilai yang berada pada baris ke – 1 maka alamatnya akan va1, vb1, vc1, ta1, tb1, dan tc1. Hal tersebut berlaku pada setiap baris yang ada pada aplikasi. *Source code* pengubah nilai huruf menjadi nilai angka ini dicontohkan pada Gambar 4.7.

```
float va1=Float.parseFloat(ta1.getText());  
float vb1=Float.parseFloat(tb1.getText());  
float vc1=Float.parseFloat(tc1.getText());
```

Gambar 4.7 *Source Code* Pengubah Nilai Huruf Menjadi Nilai Angka

4.5.4 Pengoperasian Aritmatika

Pada pengoperasian aritmatika ini *source code* yang digunakan menggunakan rumus dari perhitungan yang digunakan untuk menghitung nilai indeks keandalan seperti SAIFI, SAIDI, dan CAIDI. Terdapat letak hasil yang *dicodingkan* yaitu saifi1, saidi1, dan caidi1, hal tersebut diketik sesuai alamat baris hasil pada aplikasi sehingga akan berlanjut hingga baris akhir. Aritmatika yang digunakan pada setiap *coding* juga sama hanya saja berbeda alamat baris

yang telah ditentukan seperti halnya *source code* yang lain, pada aritmatika juga sama hanya beda pada alamat baris saja hanya saja pada pengoperasian aritmatika ini *source code* digunakan untuk mengolah data yang telah dimasukkan oleh *user*. *Source code* pengoperasian aritmatika ini dicontohkan pada Gambar 4.8.

```
float saifi1=(va1*vc1)/vt1;  
float saidi1=(vb1*vc1)/vt1;  
float caidi1=((vb1*vc1)/vt1)/((va1*vc1)/vt1);  
float asai1=((8760-saidi1)/8760)*100;  
float asui1=(1-asai1);
```

Gambar 4.8 *Source Code* Pengoperasian Aritmatika

4.5.5 Tampilan Hasil

Paket yang digunakan untuk mencetak atau menampilkan hasil akhir dari perhitungan yang telah dioperasikan sebelumnya. *Source code* ini juga sama dengan *source code* lain yang berada pada aplikasi ini yang sama penulisannya hanya berbeda pada alamat per baris yang akan dipergunakan oleh *user* dari baris ke – 1 hingga baris akhir. Sehingga pada hasil yang akan tercetak tidak akan salah alamat ketika telah dilakukan perhitungan. *Source code* tampilan hasil ini dicontohkan pada Gambar 4.9.

```
tx1.setText(" "+ saifi1);  
ty1.setText(" "+ saidi1);  
tz1.setText(" "+ caidi1);  
asa1.setText(" "+ asai1);  
asu1.setText(" "+ asui1);
```

Gambar 4.9 *Source Code* Tampilan Hasil

4.5.6 Keterangan Batas Standar Nilai Keandalan

Pada tampilan ini *software* aplikasi dapat menampilkan keterangan bahwa nilai yang telah diolah pada *software* aplikasi tersebut telah memenuhi standar atau belum. Ketika nilai yang dihitung tidak melebihi batas standar SPLN 68 – 2 1986 dan IEEE std 1366 – 2003 maka akan memunculkan keterangan “OK”, sedangkan saat nilai yang dihitung melebihi batas standar indeks keandalan maka akan memunculkan keterangan “TIDAK OK”. *Source code* yang digunakan memiliki inti yang sama hanya dibedakan alamat pada setiap baris perhitungan. *Source code* tampilan hasil ini dicontohkan pada Gambar 4.10.



```
if (saifi1<3.2) {
ah1.setText("OK");
}
else {
ah1.setText("TIDAK OK");
}
if (saidi1<21.09) {
ah13.setText("OK");
}
else {
ah13.setText("TIDAK OK");
}
if (saifi1<1.45) {
isf1.setText("OK");
}
else {
isf1.setText("TIDAK OK");
}
if (saidi1<2.3) {
isd1.setText("OK");
}
else {
isd1.setText("TIDAK OK");
}
```

Gambar 4.10 *Source code* Keterangan Batas Standar Nilai Keandalan

4.6 Validasi Hasil Perhitungan

Validasi hasil perhitungan diperlukan untuk mengetahui bahwa aplikasi yang dibuat atau dirancang hasilnya akurat. Jika aplikasi yang dibuat hasilnya sama dengan perhitungan dengan cara *manual* maka aplikasi ini dapat dibilang berhasil dan layak digunakan untuk membantu PT. PLN dalam melakukan perhitungan tingkat keandalan suatu sistem distribusi. Untuk mengetahui tingkat keakuratan dari aplikasi ini digunakan 2 cara perhitungan yaitu perhitungan secara *manual* dan perhitungan yang dilakukan oleh aplikasi ini.

Pada tabel 4.12 berikut menunjukkan validasi hasil perhitungan dengan cara *manual* dan dengan menggunakan aplikasi yang telah dibuat.

Tabel 4.12 Validasi Hasil Perhitungan Cara *Manual* dan *Software* Aplikasi

No	Perhitungan Secara Manual			Perhitungan dengan Aplikasi			Keterangan
	SAIFI	SAIDI	CAIDI	SAIFI	SAIDI	CAIDI	
1	0.906	1.656	1.828	0.906	1.656	1.828	Akurat
2	0.302	0.986	3.265	0.302	0.986	3.265	Akurat
3	0.199	0.26	1.61	0.199	0.26	1.61	Akurat
4	0.0000068	0.0000046	0.68	0.0000068	0.0000046	0.68	Akurat
5	0.160	0.164	1.02	0.160	0.164	1.02	Akurat
6	0.319	0.49	1.536	0.319	0.49	1.536	Akurat
7	0.000013	0.0000095	0.7	0.000013	0.0000095	0.7	Akurat
8	0.332	0.0631	0.19	0.332	0.0631	0.19	Akurat
9	0	0	0	0	0	0	Akurat
10	0	0	0	0	0	0	Akurat
11	0.000013	0.000037	2.75	0.000013	0.000037	2.75	Akurat
12	0.162	0.07	0.43	0.162	0.07	0.43	Akurat