

## BAB V

### PENGUJIAN APLIKASI

#### 4.2.1 Testing

Pada tahapan ini, dilakukan pengujian terhadap sistem aplikasi presensi berbasis android, hal ini diperlukan untuk mengetahui kinerja aplikasi secara keseluruhan serta memastikan semua fungsi yang terdapat didalam aplikasi dapat berjalan dengan baik. Teknik pengujian yang digunakan adalah teknik pengujian menggunakan metode Blackbox, metode ini akan menguji semua proses inputan yang dilakukan oleh pengguna dan respon aplikasi terhadap inputan tersebut.

*Table 1. Pengujian Blackbox*

No	User Interface	Input	Output	Hasil Uji
<b>Splashscreen</b>				
1.	Membuka aplikasi presensi	-	Menampilkan splashscreen aplikasi	Komponen aplikasi berjalan dengan baik
<b>Login</b>				
1.	Login dengan email dan password akun dalam keadaan kosong	Email: - Password:-	Menampilkan pesan error pada kolom inputan email	Komponen aplikasi berjalan dengan baik
2.	Login dengan password dalam keadaan kosong	Email: valid Password:-	Menampilkan pesan error pada kolom inputan password	Komponen aplikasi berjalan dengan baik
3.	Login dengan email dalam keadaan kosong	Email: - Password: valid	Menampilkan pesan error pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
4.	Login dengan email dan password dalam keadaan salah	Email: invalid Password: invalid	Menampilkan toast singkat akun tidak terdaftar	Komponen aplikasi berjalan dengan baik
5.	Login dengan email benar dan password salah	Email:valid assword: invalid	Menampilkan toast singkat	Komponen aplikasi berjalan dengan baik

			password salah`	
6.	Login dengan email salah dan password benar	Email:invalid Password: valid	Menampilka n toast singkat akunn pengguna tidak ditemukan	Komponen aplikasi berjalan dengan baik
7.	Masuk kedalam halaman lupa password	Klik tombol lupa password	Berpindah kedalam halaman lupa password	Komponen aplikasi berjalan dengan baik
8.	Masuk kedalam halaman signup	Klik tombol belum punya akun	Berpindah kedalam halaman signup	Komponen aplikasi berjalan dengan baik
<b>Lupa password</b>				
1.	Reset password dengan kolom inputan email dalam keadaan kosong	Email: -	Menampilka n pesan email tidak boleh kosong pada kolom inputan	Komponen aplikasi berjalan dengan baik
2.	Reset password dengan email tidak terdaftar	Email: invalid	Menampilka n pesan email tidak terdaftar	Komponen aplikasi berjalan dengan baik
3.	Reset password dengan email terdaftar	Email: valid	Mengirmkan tautan reset password ke email terkait	Komponen aplikasi berjalan dengan baik
<b>Signup</b>				
1.	Signup dengan email dan password akun dalam keadaan kosong	Email: - Password:-	Menampilka n pesan error pada kolom inputan email	Komponen aplikasi berjalan dengan baik
2.	Signup dengan password dalam keadaan kosong	Email: valid Password:-	Menampilka n pesan error pada kolom inputan password	Komponen aplikasi berjalan dengan baik
3.	Signup dengan email dalam keadaan kosong	Email: - Password: valid	Menampilka n pesan error pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik

4.	Signup dengan email dan password dalam keadaan terdaftar	Email: invalid Password: invalid	Menampilka n toast singkat akun telah terdaftar	Komponen aplikasi berjalan dengan baik
5.	Masuk kedalam halaman login	Klik tombol punya akun	Berpindah kedalam halaman login	Komponen aplikasi berjalan dengan baik
<b>Beranda</b>				
1.	Melakukan aksi klik pada ikon fitur eksternal lapor gubernur	Klik ikon lapor gubernur	Membuka tautan laporgub.jate ngprov.go.id	Komponen aplikasi berjalan dengan baik
2.	Melakukan aksi klik pada ikon fitur eksternal lapor bupati	Klik ikon lapor gubernur	Membuka aplikasi playstore unntuk mengunduh aplikasi lapor bupati tegal	Komponen aplikasi berjalan dengan baik
3.	Melakukan aksi klik pada ikon fitur eksternal info covid-19	Klik ikon info covid-19	Membuka tautan covid19.tegal kab.go.id	Komponen aplikasi berjalan dengan baik
4.	Melakukan aksi klik pada ikon fitur eksternal PPID Kab. Tegal	Klik ikon PPID Kab. Tegal	Membuka tautan ppid.tegalkab .go.id	Komponen aplikasi berjalan dengan baik
5.	Melakukan aksi klik pada ikon fitur eksternal Portal Kab. Tegal	Klik ikon Portal Kab. Tegal	Membuka tautan utama.tegalk ab.go.id	Komponen aplikasi berjalan dengan baik
6.	Melakukan aksi klik pada ikon fitur eksternal Disdukcapil Kab. Tegal	Klik ikon Disdukcapil Kab. Tegal	Membuka tautan disdukcapil.t egalkab.go.id	Komponen aplikasi berjalan dengan baik
7.	Melakukan aksi klik pada daftar berita	Klik salah satu list daftar berita	Membuka tautan berita terkait	Komponen aplikasi berjalan dengan baik
<b>Presensi</b>				
1.	Melakukan presensi tanpa koneksi internet	Klik tombol presensi	Memunculka n toast singkat koneksi	Komponen aplikasi dapat berjalan dengan baik

			internet tidak tersedia	
2.	Melakukan presensi diluar waktu yang telah ditentukan	Klik tombol presensi	Muncul dialog presensi ditolak	Komponen aplikasi dapat berjalan dengan baik
3.	Melakukan presensi tanpa izin aplikasi	Klik tombol presensi	Muncul permintaan akses lokasi oleh sistem android	Komponen aplikasi dapat berjalan dengan baik
4.	Melakukan presensi dengan status GPS ponsel mati	Klik tombol presensi	Muncul dialog untuk mengaktifkan GPS ponsel	Komponen aplikasi dapat berjalan dengan baik
5.	Melakukan presensi diluar jarak yang telah ditentukan	Klik tombol presensi	Muncul dialog untuk mendekat ke area presensi.	Komponen aplikasi dapat berjalan dengan baik
6.	Melakukan presensi dengan koneksi internet tersedia, dalam waktu presensi, izin aplikasi, GPS menyala dan dalam lingkup jarak presensi yang diperbolehkan	Klik tombol presensi	Muncul dialog presensi dapat dilakukan	Komponen aplikasi dapat berjalan dengan baik
7.	Melakukan presensi pada dialog presensi	Klik tombol presensi sekarang	Muncul toast presensi berhasil	Komponen aplikasi dapat berjalan dengan baik
<b>Histori presensi</b>				
1.	Membuka menu presensi aplikasi ketika data dalam database tersedia	Klik navigasi presensi	Muncul data presensi dalam bentuk list yang berisi username, tanggal presensi serta ikon presensi	Komponen aplikasi dapat berjalan dengan baik
2.	Membuka menu presensi aplikasi ketika data dalam	Klik navigasi presensi	Muncul gambar yang menandakan	Komponen aplikasi dapat berjalan dengan baik

	database tidak tersedia		bahwa data tidak tersedia	
<b>Foto profil</b>				
1.	Mengganti foto profil pada menu akun aplikasi	Aksi klik pada foto profile	Proses unggah dan mengganti foto profil	Komponen aplikasi dapat berjalan dengan baik
2.	Memperbarui foto profil pada menu akun aplikasi	Aksi klik pada tombol update foto profil	Muncul toast foto profil berhasil diperbarui	Komponen aplikasi dapat berjalan dengan baik
<b>Ganti username</b>				
1.	Ganti username pada menu akun	Aksi klik pada tombol ganti username	Muncul kolom inputan username	Komponen aplikasi berjalan dengan baik
2.	Ganti username dengan kolom inputan dalam keadaan kosong	Username: -	Muncul error pada kolom inputan username	Komponen aplikasi berjalan dengan baik
3.	Ganti username dengan kolom inputan dalam keadaan terisi	Username: valid	Muncul toast singkat berisi informasi update username	Komponen aplikasi berjalan dengan baik
<b>Ganti email</b>				
1.	Ganti email dengan kolom inputan autentikasi dalam keadaan kosong	Password: -	Muncul error kolom harus diisi pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
2.	Ganti email dengan kolom inputan autentikasi dalam keadaan salah	Password: invalid	Muncul error password salah pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
3.	Ganti email dengan kolom inputan autentikasi dalam keadaan benar	Password: valid	Muncul kolom inputan mengganti email baru	Komponen aplikasi dapat berjalan dengan baik
4.	Ganti email dengan kolom inputan email	Email: -	Muncul error email tidak boleh kosong	Komponen aplikasi dapat

	dalam keadaan kosong		pada kolom inputan email	berjalan dengan baik
5.	Ganti email dengan kolom inputan email disini bukan bertipe email	Email: invalid	Muncul error email tidak valid pada kolom inputan email	Komponen aplikasi dapat berjalan dengan baik
6.	Ganti email dengan kolom inputan email diisi dengan tipe inputan email	Email: valid	Muncul toast singkat ketika email berhasil diganti	Komponen aplikasi dapat berjalan dengan baik
<b>Verifikasi email</b>				
1.	Verifikasi email dengan aksi klik pada tombol verifikasi email	Aksi klik pada tombol verifikasi email	Muncul toast verifikasi email dan mengirim tautan ke email terkait	Komponen aplikasi dapat berjalan dengan baik
<b>Ganti password</b>				
1.	Ganti password dengan kolom inputan autentikasi dalam keadaan kosong	Password:-	Muncul error kolom harus di isi pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
2.	Ganti password dengan kolom inputan autentikasi dalam keadaan salah	Password: invalid	Muncul error password salah pada kolom inputan password	Komponen aplikasi dapat berjalan dengan baik
3.	Ganti password dengan kolom inputan autentikasi dalam keadaan benar	Password: valid	Muncul kolom inputan mengganti password baru	Komponen aplikasi dapat berjalan dengan baik
4.	Ganti password dengan kolom input password dalam keadaan kosong	Password: - konfirmasi password: -	Muncul error password tidak boleh kosong	Komponen aplikasi berjalan dengan baik
5.	Ganti password dengan jumlah karakter kurang dari 8	Password: invalid (< 8 karakter)	Muncul error password kurang dari 8 karakter	Komponen aplikasi dapat berjalan dengan baik

		Konfirmasi password: invalid (< 8 karakter)		
6.	Ganti password dengan kolom inputan password dan konfirmasi password tidak sama	Password: invalid (tidak sama) Konfirmasi password: invalid (tidak sama)	Muncul error password tidak sama	Komponen aplikasi dapat berjalan dengan baik
7.	Ganti password dengan kolom inputan password dan konfirmasi password diisi dengan karakter sama serta dengan jumlah karakter > 8 karakter	Password: valid Konfirmasi password: valid	Muncul toast informasi ubah password dan berpindah ke halaman login aplikasi	Komponen aplikasi dapat berjalan dengan baik
<b>Laporkan masalah aplikasi</b>				
1.	Aksi klik pada tombol laporkan masalah aplikasi	Aksi klik pada tombol laporkan masalah aplikasi	Berpindah ke aplikasi email didalam sistem android	Komponen aplikasi dapat berjalan dengan baik
<b>Tentang aplikasi</b>				
1.	Aksi klik pada tombol tentang aplikasi	Aksi klik pada tombol tentang aplikasi	Berpindah kedalam halaman tentang aplikasi	Komponen aplikasi dapat berjalan dengan baik
<b>Keluar/Logout</b>				
1.	Aksi klik pada tombol keluar	Aksi klik pada tombol keluar aplikasi	Muncul dialog konfirmasi keluar aplikasi?	Komponen aplikasi dapat berjalan dengan baik
2.	Aksi klik pada tombol “ya” dialog konfirmasi keluar aplikasi	Aksi klik pada tombol “ya” dialog konfirmasi keluar aplikasi	Logout dari akun terkait dan berpindah kedalam halaman login aplikasi	Komponen aplikasi dapat berjalan dengan baik

3.	Aksi klik pada tombol “tidak” dialog konfirmasi keluar aplikasi	Aksi klik pada tombol “tidak” dialog konfirmasi keluar aplikasi	Dialog menghilang dan tetap pada menu akun aplikasi	Komponen aplikasi dapat berjalan dengan baik
----	---	---	---	--

Dibawah ini adalah pengujian Whitebox yang dilakukan pada aplikasi, pengujian Whitebox diperlukan untuk menguji lebih dekat tentang detail dari perangkat lunak dengan pendekatan *logical path* pada aplikasi yang akan dilakukan pengujian.

a. Pengujian Whitebox presensi aplikasi

Kasus yang akan diuji dalam pengujian Whitebox testing pada gambar nomor 51 adalah pengujian sistem presensi aplikasi. Gambar 51 berisikan baris kode yang akan di eksekusi oleh aplikasi khusus untuk menangani fitur presensi dan pengukuran jarak menggunakan metode haversine.

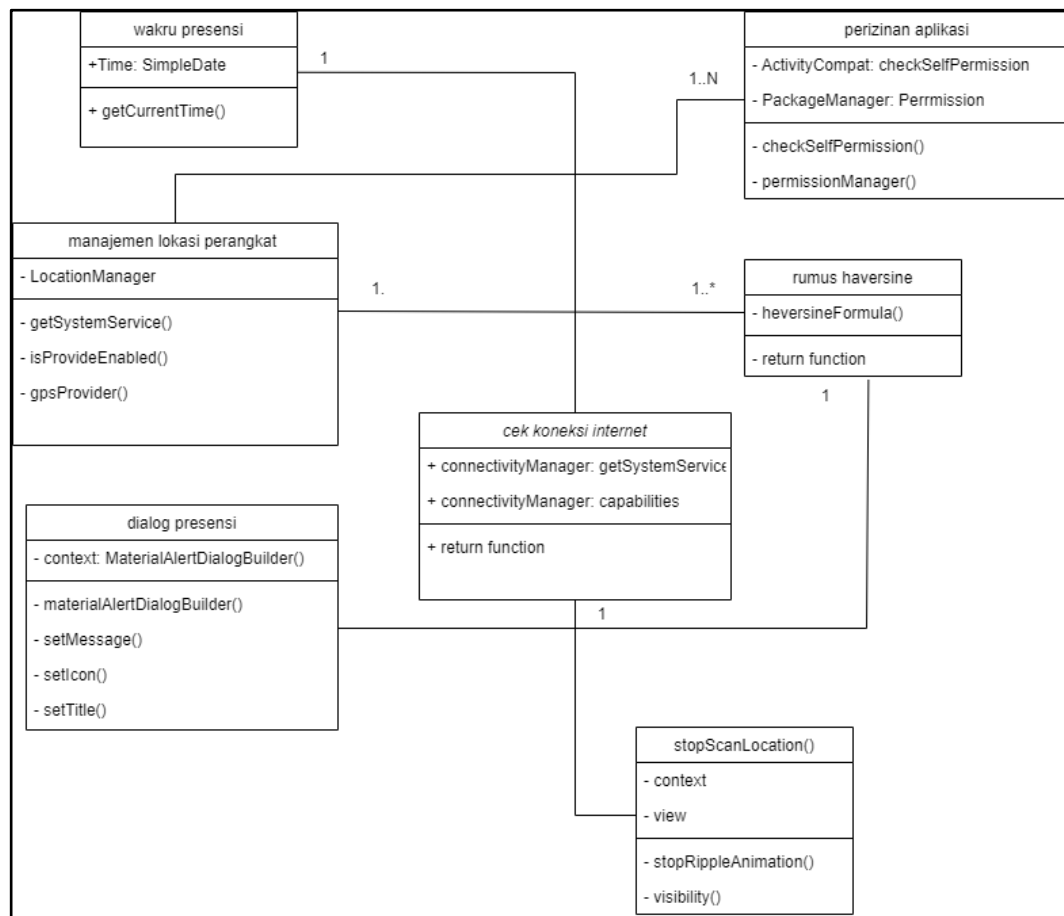


```

1 private fun getLastLocation() {
2     if (FunctionLibrary.checkConnection(requireContext())) {
3         if (FunctionLibrary.timeAttendance() || FunctionLibrary.timeAttendanceLate()) {
4             if (checkPermission()) {
5                 if (isLocationEnabled()) {
6                     val locationCallback = object : LocationCallback() {
7                         override fun onLocationResult(locationResult: LocationResult) {
8                             super.onLocationResult(locationResult)
9                             val location = locationResult.lastLocation
10                            val currentLat = location!!.latitude
11                            val currentLong = location.longitude
12                            val destinationLat = getAddress()[0].latitude
13                            val destinationLong = getAddress()[0].longitude
14                            lifecycleScope.launch(Dispatchers.Default) {
15                                val distance = FunctionLibrary.calculateDistance(
16                                    currentLat, currentLong, destinationLat, destinationLong
17                                ) * 1000
18                                Log.d(TAG, "$TAG_RESULT - $distance")
19                                withContext(Dispatchers.Main) {
20                                    if (distance < MEASURING_DISTANCE) {
21                                        showDialog()
22                                        FunctionLibrary.toast(
23                                            context as Activity,
24                                            TOAST_SUCCESS,
25                                            LOCATION_FOUND,
26                                            MotionToastStyle.SUCCESS,
27                                            MotionToast.GRAVITY_BOTTOM,
28                                            MotionToast.LONG_DURATION,
29                                            ResourcesCompat.getFont(
30                                                context as Activity,
31                                                R.font.helveticaBold
32                                            )
33                                        )
34                                    } else {
35                                        simpleDialog(
36                                            OUT_OF_RANGE,
37                                            OUT_OF_RANGE_MESSAGE
38                                        )
39                                        binding.tvCheckIn.visibility = View.VISIBLE
40                                    }
41                                }
42                            }
43                            fusedLocationProviderClient?.removeLocationUpdates(this)
44                            stopScanLocation()
45                        }
46                    }
47                    lifecycleScope.launch(Dispatchers.IO) {
48                        fusedLocationProviderClient?.requestLocationUpdates(
49                            locationRequest,
50                            locationCallback,
51                            Looper.getMainLooper()
52                        )
53                    }
54                } else {
55                    simpleDialog(
56                        GPS_STATUS,
57                        GPS_MESSAGE
58                    )
59                    stopScanLocation()
60                }
61            } else {
62                stopScanLocation()
63                requestPermission()
64            }
65        } else {
66            stopScanLocation()
67            simpleDialog(
68                ATTENDANCE_DENIED,
69                ATTENDANCE_TIME
70            )
71        }
72    } else {
73        FunctionLibrary.toast(
74            context as Activity,
75            TOAST_ERROR,
76            PERMISSION_INTERNET,
77            MotionToastStyle.ERROR,
78            MotionToast.GRAVITY_BOTTOM,
79            MotionToast.LONG_DURATION,
80            ResourcesCompat.getFont(context as Activity, R.font.helveticaBold)
81        )
82        stopScanLocation()
83    }
84 }

```

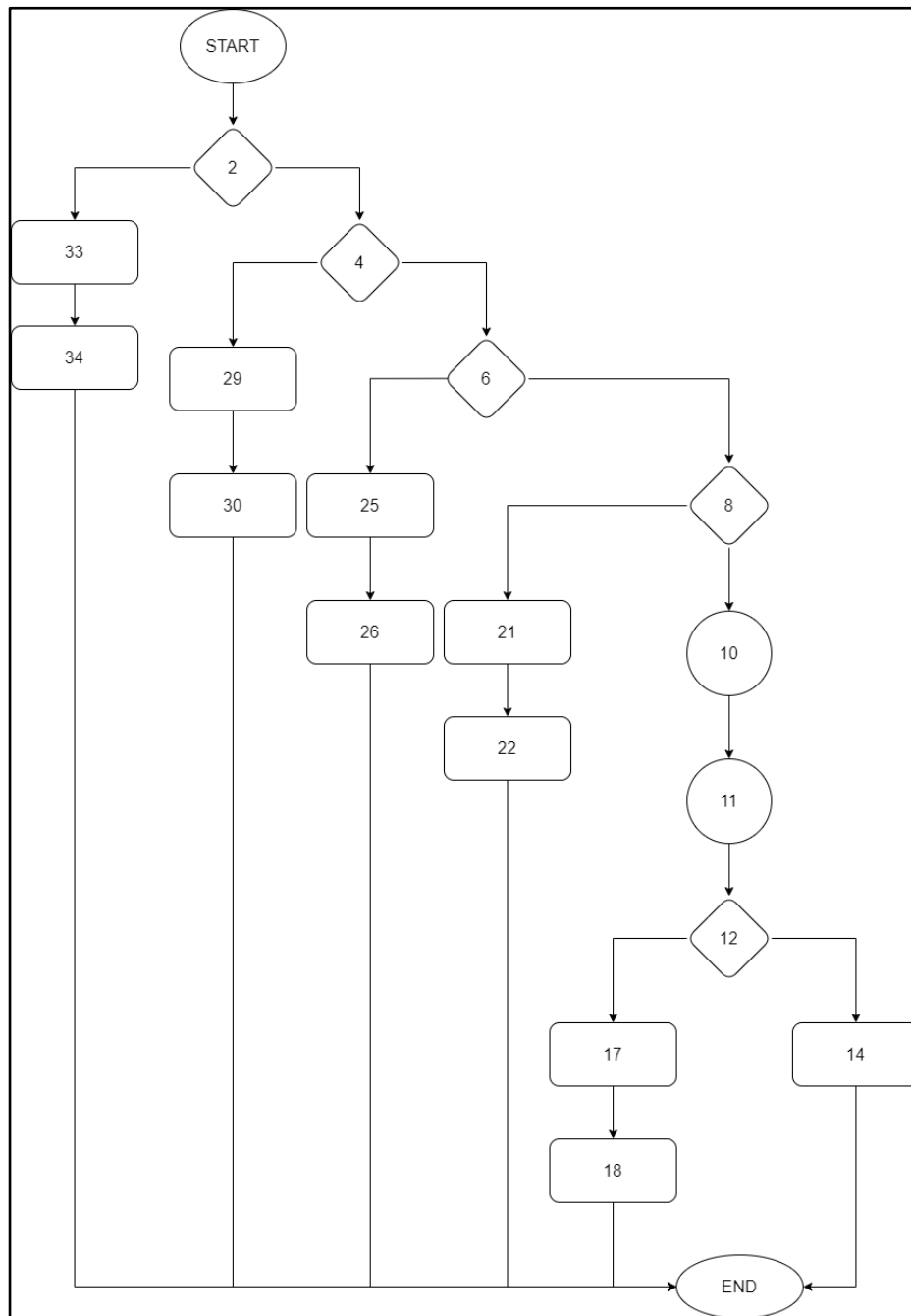
*Gambar 1. Pengujian Whitebox sistem presensi aplikasi*



*Gambar 2. Class diagram fungsi presensi aplikasi*

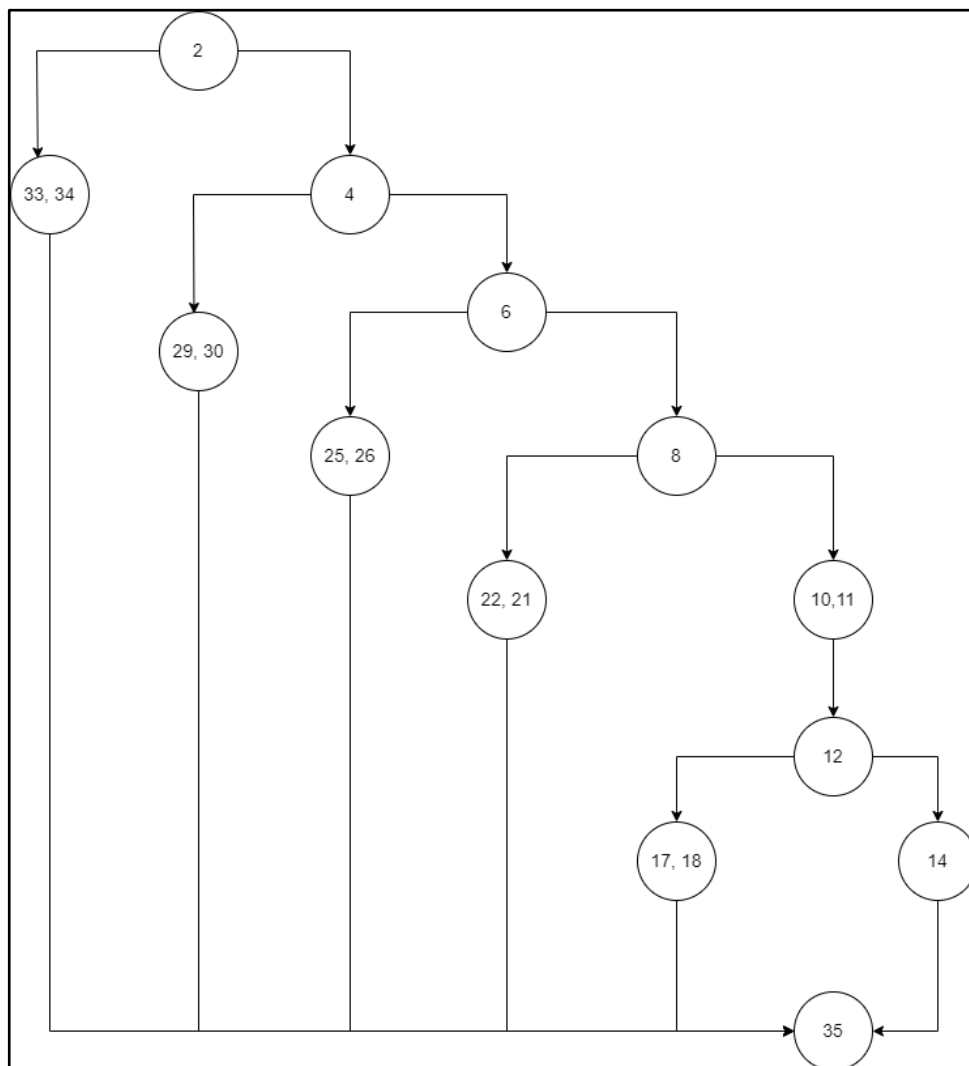
Pada gambar 52, class diagram akan menggambarkan atribut yang digunakan aplikasi untuk membuat fitur presensi dan pengukuran jarak, class diagram diatas juga merupakan fungsi yang digunakan dan keterkaitan antar fungsi yang terjadi.

Flow chart dari pengujian Whitebox pada gambar 52 adalah sebagai berikut:



*Gambar 3. Flowchart whitebox testing*

Sedangkan untuk flowgraph dari pengujian Whitebox terdapat pada gambar 54 dibawah ini



*Gambar 4. Flowgraph whiteBox testing sistem presensi*

Dengan adanya flowgraph, maka Cyclomatic Complexity dapat dilakukan dengan rumus sebagai berikut:

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 35 - 12 + 2 \\
 &= 25
 \end{aligned}$$

Dengan begitu, jumlah region dapat ditentukan dengan jumlah region keseluruhan adalah 25. Untuk menentukan independent path, dapat melihat flowgraph yang ada digambar 53, independent graph yang terbentuk berdasarkan flowgraph adalah sebagai berikut:

Path 1: 2, 33, 34, 35

Path 2: 2, 4, 29, 30, 35

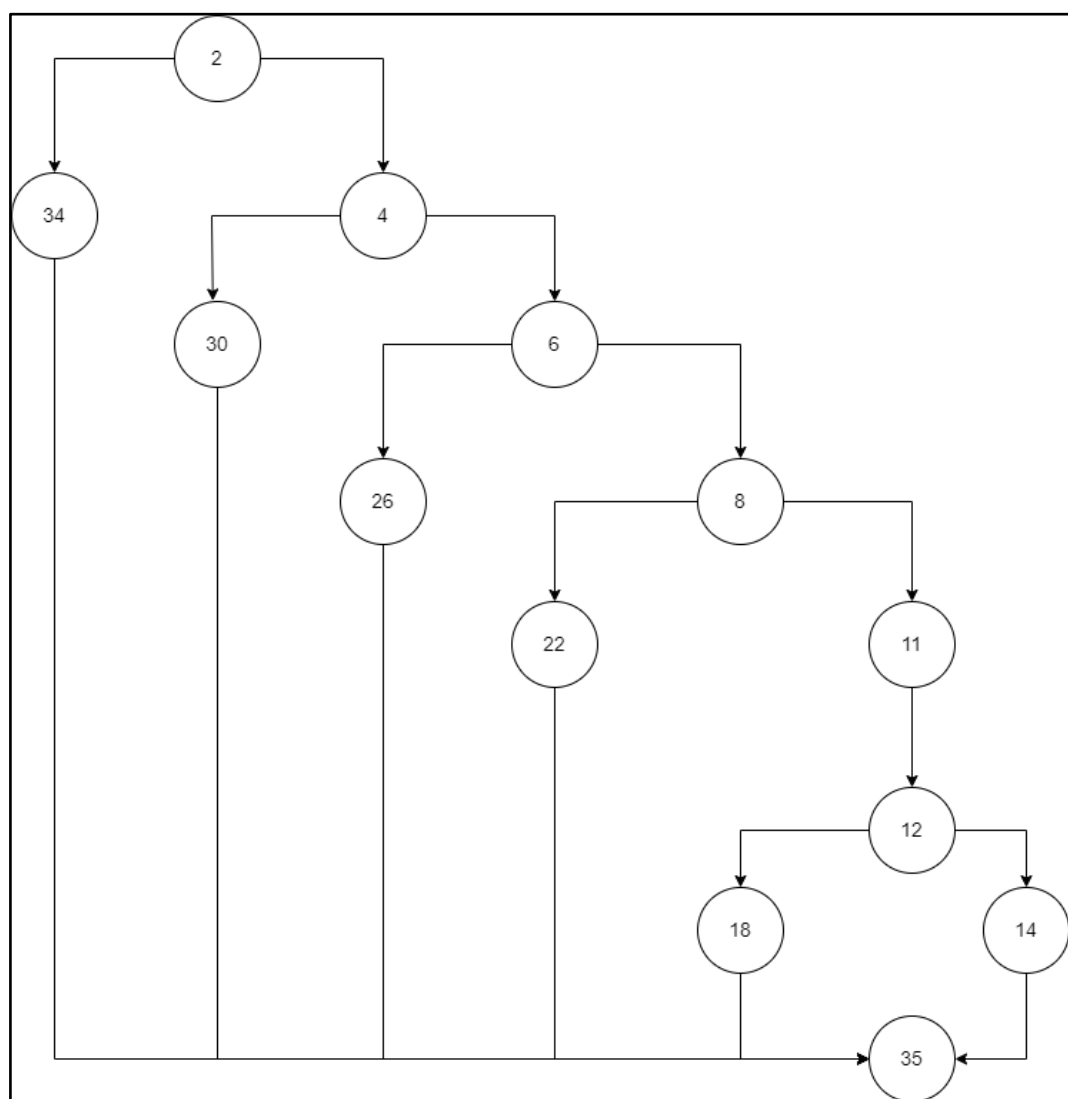
Path 3: 2, 4, 6, 25, 26, 35

Path 4: 2, 4, 6, 8, 22, 21, 35

Path 5: 2, 4, 6, 8, 10, 11, 12, 17, 18, 35

Path 6: 2, 4, 6, 8, 10, 11, 12, 14, 35

Untuk mendapatkan gambaran yang lebih jelas tentang graph matriks, maka perlu dibuat konversi dari flowgraph seperti berikut:



*Gambar 5. Konversi flowgraph Whitebox testing*

Tahapan selanjutnya dalam menentukan predicate node adalah membuat tabel matriks sesuai dengan konversi graph pada gambar nomor 54, detail dari tabel matriks adalah sebagai berikut:

Pada tabel lampiran nomor 1, didapatkan predicate node dengan jumlah 5 node, dengan begitu, predicate node dapat dihitung dengan rumus:

$$\begin{aligned} V(G) &= P + 1 \\ &= 5 + 1 \\ &= 6 \end{aligned}$$

Dengan predicate node yang terdeteksi adalah matriks 2, 4, 6, 8, 12

#### 4.1 Hasil Penelitian

Tahapan selanjutnya setelah implementasi aplikasi adalah uji coba system, untuk mendapatkan hasil uji coba yang akurat sesuai dengan metode yang telah ada sebelumnya, maka uji coba aplikasi akan menggunakan metode *System Usability Scale* (SUS), secara singkat, metode usability merupakan sebuah analisa yang bersifat kualitatif atau berorientasi pada kualitas pada objek yang diujikan, dengan metode usability ini akan menentukan seberapa mudah antarmuka aplikasi ketika digunakan oleh pengguna, salah satu indikator antarmuka aplikasi mudah digunakan antara lain adalah ketika semua fungsi yang terdapat didalam aplikasi efektif, memuaskan dan defisien ketika digunakan, konklusi dari usability testing adalah menentukan atau melakukan evaluasi antarmuka aplikasi apakah telah memenuhi kebutuhan pengguna atau belum memenuhi dan perlu diadakan perbaikan (Firmansyah, 2018) dan (Sharfina and Santoso, 2017).

Table 2. Kuisiонер System Usability Scale (SUS)

No	Pertanyaan	Rata-rata	Kategori
1	Saya merasa akan sering menggunakan aplikasi peduliPresensi	4.05	setuju
2	Saya menilai bahwa aplikasi peduliPresensi tidak perlu serumit ini	2.4	Tidak setuju
3	Saya merasa aplikasi peduliPresensi mudah untuk digunakan	4.2	setuju
4	Saya membutuhkan pendampingan teknis dalam menggunakan aplikasi ini	2.8	Netral
5	Saya menilai fitur yang terdapat pada aplikasi PeduliPresensi dapat berjalan dengan baik	2.8	Netral
6	Saya menilai banyak hal pada aplikasi PeduliPresensi yang tidak konsisten	2.4	Tidak setuju
7	Saya merasa bahwa orang lain dapat mempelajari aplikasi PeduliPresensi dengan cepat	4.15	Setuju
8	Saya menilai aplikasi PeduliPresensi ssangat rumit ketika digunakan	2.2	Tidak setuju
9	Saya tidak menemui hambatan dalam penggunaan aplikasi	3.85	Setuju
10	Saya perlu untuk beradaptasi dalam menggunakan aplikasi	2.2	Tidak setuju

#### 4.2 Pembahasan

Pembahasan hasil penelitian dilakukan dengan melakukan kalkulasi nilai yang terdapat pada tiap kuisioner yang telah di isi oleh pengguna.

*Table 3. Penilaian kuisioner SUS*

No	Jawaban	Skor
1	Sangat tidak setuju	1
2	Tidak setuju	2
3	Ragu-ragu	3
4	Setuju	4
5	Sangat setuju	5

Aturan perhitungan jawaban pada kuisioner System Usability Scale (SUS) adalah sebagai berikut:

1. Setiap pertanyaan yang bernomor ganjil, penilaian dilakukan dengan melakukan operasi pengurangan terhadap skor yang didapat dari pengguna dengan angka 1.
2. Setiap pertanyaan yang bernomor genap, penilaian dilakukan dengan melakukan operasi pengurangan angka 5 terhadap skor yang didapat dari pengguna.
3. Sedangkan untuk mendapatkan hasil akhir dari SUS didapatkan dari hasil penjumlahan skor setiap pertanyaan dan dikalikan dengan 2.5

Untuk mendapatkan skor akhir dari SUS, maka dilakukan perhitungan dengan rumus

$$\bar{x} = \frac{\sum x}{n}$$

$\bar{x}$  = skor rata-rata  
 $\sum x$  = jumlah skor SUS  
 $n$  = jumlah responden

*Gambar 6. Rumus skor rata-rata SUS*

Dengan begitu, semua data yang didapatkan dari hasil kuisioner akan dimasukkan dalam tabel dan dilakukan kalkulasi sesuai dengan ketentuan dari metode SUS.

*Table 4. Tabel data asli responden*

No	Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	Responden 1	4	2	4	3	5	4	4	2	5	2
2	Responden 2	3	3	4	4	4	3	3	2	2	1
3	Responden 3	4	3	4	4	5	2	4	2	4	3
4	Responden 4	3	3	4	4	5	3	4	3	4	2
5	Responden 5	4	3	3	4	3	4	2	4	3	2
6	Responden 6	3	4	5	3	4	3	5	2	5	3
7	Responden 7	4	3	4	3	4	2	5	3	5	1
8	Responden 8	4	2	4	2	4	2	4	2	4	3

9	Responden 9	4	4	4	4	4	3	4	3	3	3
10	Responden 10	4	3	4	2	4	4	4	2	4	2
11	Responden 11	4	2	5	4	4	2	3	3	2	2
12	Responden 12	4	2	5	3	4	2	4	2	4	2
13	Responden 13	4	1	3	3	5	2	5	2	4	3
14	Responden 14	5	2	4	2	5	2	5	1	3	3
15	Responden 15	5	2	5	2	4	2	4	2	5	3
16	Responden 16	4	2	4	2	5	2	5	2	4	2
17	Responden 17	4	2	4	2	4	2	4	2	4	2
18	Responden 18	5	1	5	1	4	1	4	1	4	1
19	Responden 19	4	2	4	2	5	2	5	2	4	2
20	Responden 20	5	2	5	2	5	1	5	2	4	2

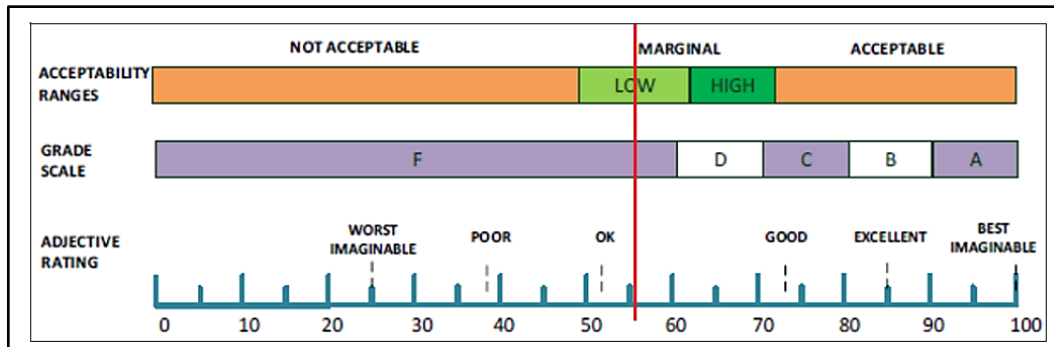
Data yang didapatkan pada tabel 10 selanjutnya akan dilakukan perhitungan dengan rumus SUS dan hasilnya akan ditampillkan pada tabel 11.

Table 5. Tabel hasil perhitungan dengan rumus SUS

[illegible]



Berdasarkan akhhir dari tabel 11, didapatkan skor rata-rata nilai 72, untuk mendapatkan konklusi yang lebih jelas, maka dibutuhkan analisa skor SUS.



Gambar 7. tingkat penerimaan metode SUS

Dari gambar 54 dan hasil akhir yang diperoleh, maka metode SUS pada aplikasi peduli presensi mendapatkan tingkat penerimaan “MARGINAL HIGH”, dengan adjective rating “GOOD” (Supriyadi, Thya Safitri and Kristiyanto, 2020).

Grade	SUS	Percentile range
A+	84.1 – 100	96 – 100
A	80.8 – 84.0	90 – 95
A-	78.9 – 80.7	85 – 89
B+	77.2 – 78.8	80 – 84
B	74.1 – 77.1	70 – 79
B-	72.6 – 74.0	65 – 69
C+	71.1 – 72.5	60 – 64
C	65.0 – 71.0	41 – 59
C-	62.7 – 64.9	35 – 40
D	51.7 – 62.6	15 – 34
F	0 – 51.6	0 – 14

Gambar 8. Curved Grade Scale metode SUS

Sedangkan untuk CGS ataaau Curved Grade Scale, aplikasi peduli presensi mendapatkan grade B dengan percentile range 72. Berdasarkan penilaian menggunakan metode SUS, maka pengembangan dan perbaikan pada sisi fungsionalitas aplikasi perlu untuk ditingkatkan, selain itu karena aplikasi ini terbilang baru pada lingkungan kantor balai desa Warureja, perlu juga dibuat sosialisasi mengenai penggunaan aplikasi baik berupa video tutorial ataupun *manual book* tentang tata cara penggunaan dan aplikasi bekerja.