

LAMPIRAN



Gambar 1. Penyerahan Surat Izin Penelitian.



Gambar 2. Penyerahan Surat Izin Penelitian.



Gambar 3. Penyerahan Surat Izin Penelitian.



Gambar 4. Kunjungan Penelitian.



Gambar 5. Kunjungan Penelitian.



Gambar 6. Kunjungan Penelitian.



Gambar 7. Wawancara Dengan Pemilik *Pet Shop*.



Gambar 8. Wawancara Dengan Konsumen.



Gambar 9. Wawancara dan Uji Coba Aplikasi Dengan Konsumen.



Gambar 10. Penyerahan dan Uji Coba Aplikasi Kepada Pemilik *Pet Shop*.

Tabel 1. Hasil Uji Penggunaan Algoritma *Selection Sort* dan *Insertion Sort*.

Proses	<i>Memory Usage Selection Sort</i>	Waktu Eksekusi <i>Selection Sort</i>	<i>Memory Usage Insertion Sort</i>	Waktu Eksekusi <i>Insertion Sort</i>
<i>Running Pertama</i>				
Keterangan	<p>Pada <i>running</i> pertama, algoritma <i>selection sort</i> mendapatkan 19451 KB pada 100 data.</p>	<p>Pada <i>running</i> pertama algoritma <i>selection sort</i> mendapatkan waktu eksekusi 4 ms pada 100 data.</p>	<p>Pada <i>running</i> pertama algoritma <i>insertion sort</i> mendapatkan 30354 KB pada 100 data.</p>	<p>Pada <i>running</i> pertama algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 8 ms pada 100 data.</p>
<i>Running Ke-2 (Dua)</i>				
Keterangan	<p>Pada <i>running</i> ke-2, algoritma <i>selection sort</i> mendapatkan 11663 KB pada 100 data.</p>	<p>Pada <i>running</i> ke-2 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 100 data.</p>	<p>Pada <i>running</i> ke-2 algoritma <i>insertion sort</i> mendapatkan 24356 KB pada 100 data.</p>	<p>Pada <i>running</i> ke-2 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 2 ms di 80 data dan 3 ms di 100 data.</p>

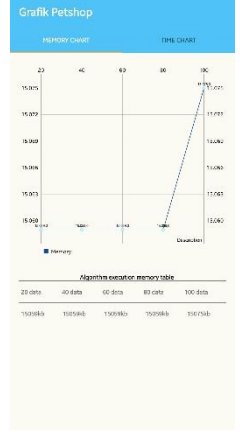
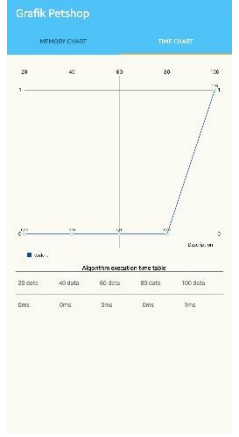
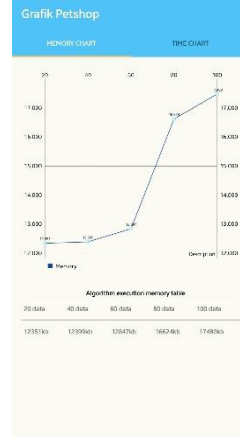
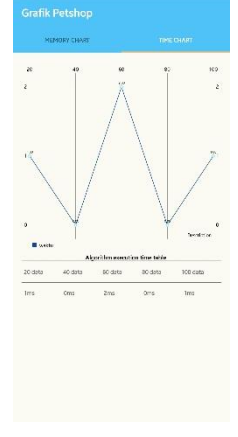
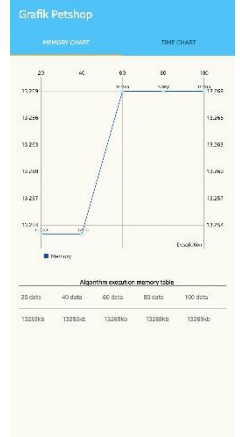
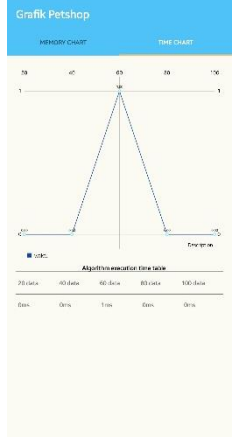
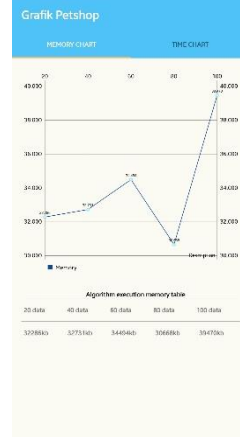
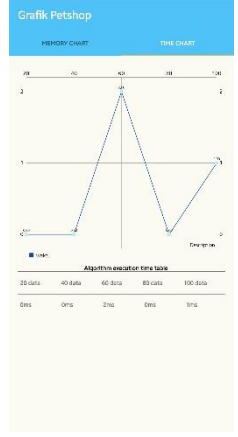
Tabel 1. Hasil Uji Penggunaan Algoritma *Selection Sort* dan *Insertion Sort* (Lanjutan).

Proses	<i>Memory Usage Selection Sort</i>	Waktu Eksekusi <i>Selection Sort</i>	<i>Memory Usage Insertion Sort</i>	Waktu Eksekusi <i>Insertion Sort</i>
Running Ke-3 (Tiga)				
Keterangan	Pada <i>running</i> ke-3, algoritma <i>selection sort</i> mendapatkan 29802 KB pada 100 data.	Pada <i>running</i> ke-3 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 80 data dan 1 ms pada 100 data.	Pada <i>running</i> ke-3 algoritma <i>insertion sort</i> mendapatkan 18791 KB pada 100 data.	Pada <i>running</i> ke-3 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 2 ms pada 80 data dan 2 ms pada 100 data.
Running Ke-4 (Empat)				
Keterangan	Pada <i>running</i> ke-4, algoritma <i>selection sort</i> mendapatkan 16359 KB pada 100 data.	Pada <i>running</i> ke-4 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 100 data.	Pada <i>running</i> ke-4 algoritma <i>insertion sort</i> mendapatkan 22905 KB pada 100 data.	Pada <i>running</i> ke-4 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 1 ms pada 100 data.

Tabel 1. Hasil Uji Penggunaan Algoritma *Selection Sort* dan *Insertion Sort* (Lanjutan).

Proses	<i>Memory Usage Selection Sort</i>	<i>Waktu Eksekusi Selection Sort</i>	<i>Memory Usage Insertion Sort</i>	<i>Waktu Eksekusi Insertion Sort</i>																																								
Running Ke-5 (Lima)	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>12227</td> <td>12229</td> <td>13229</td> <td>13229</td> <td>13344</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	12227	12229	13229	13229	13344	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>0ms</td> <td>1ms</td> <td>0ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	0ms	1ms	0ms	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>11134</td> <td>11134</td> <td>11134</td> <td>11134</td> <td>11800</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	11134	11134	11134	11134	11800	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>1ms</td> <td>1ms</td> <td>1ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	1ms	1ms	1ms
20 data	40 data	60 data	80 data	100 data																																								
12227	12229	13229	13229	13344																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	0ms	1ms	0ms																																								
20 data	40 data	60 data	80 data	100 data																																								
11134	11134	11134	11134	11800																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	1ms	1ms	1ms																																								
Keterangan	Pada <i>running</i> ke-5, algoritma <i>selection sort</i> mendapatkan 13345 KB pada 100 data.	Pada <i>running</i> ke-5 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 80 data.	Pada <i>running</i> ke-5 algoritma <i>insertion sort</i> mendapatkan 11134 KB pada 100 data.	Pada <i>running</i> ke-5 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 1 ms pada 60 data 1 ms pada 80 data dan 1 ms pada 100 data.																																								
Running Ke-6 (Enam)	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>18778</td> <td>18778</td> <td>19778</td> <td>19778</td> <td>22793</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	18778	18778	19778	19778	22793	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>2ms</td> <td>3ms</td> <td>1ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	2ms	3ms	1ms	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>24356</td> <td>24356</td> <td>24356</td> <td>42000</td> <td>42000</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	24356	24356	24356	42000	42000	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>1ms</td> <td>1ms</td> <td>2ms</td> <td>2ms</td> <td>3ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	1ms	1ms	2ms	2ms	3ms
20 data	40 data	60 data	80 data	100 data																																								
18778	18778	19778	19778	22793																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	2ms	3ms	1ms																																								
20 data	40 data	60 data	80 data	100 data																																								
24356	24356	24356	42000	42000																																								
20 data	40 data	60 data	80 data	100 data																																								
1ms	1ms	2ms	2ms	3ms																																								
Keterangan	Pada <i>running</i> ke-6, algoritma <i>selection sort</i> mendapatkan 22793 KB pada 100 data.	Pada <i>running</i> ke-6 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 100 data.	Pada <i>running</i> ke-6 algoritma <i>insertion sort</i> mendapatkan 24356 KB pada 100 data.	Pada <i>running</i> ke-6 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 1 ms pada 40 data, 2 ms pada 80 data dan 3 ms pada 100 data.																																								




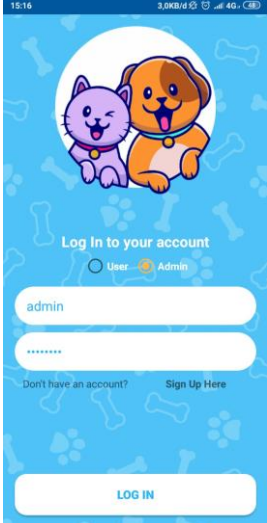
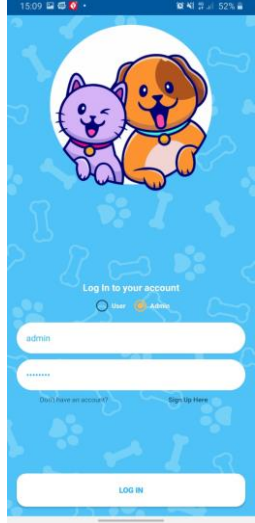
Tabel 1. Hasil Uji Penggunaan Algoritma *Selection Sort* dan *Insertion Sort* (Lanjutan).

Proses	<i>Memory Usage Selection Sort</i>	<i>Waktu Eksekusi Selection Sort</i>	<i>Memory Usage Insertion Sort</i>	<i>Waktu Eksekusi Insertion Sort</i>
Running Ke-7 (Tujuh)				
Keterangan	Pada <i>running</i> ke-7, algoritma <i>selection sort</i> mendapatkan 15075 KB pada 100 data.	Pada <i>running</i> ke-7 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 100 data.	Pada <i>running</i> ke-7 algoritma <i>insertion sort</i> mendapatkan 17480 KB pada 100 data.	Pada <i>running</i> ke-7 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 1 ms pada 20 data, 2 ms pada 60 data dan 1 ms pada 100 data.
Running Ke 8 (Delapan)				
Keterangan	Pada <i>running</i> ke-8, algoritma <i>selection sort</i> mendapatkan 13269 KB pada 100 data.	Pada <i>running</i> ke-8 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 60 data.	Pada <i>running</i> ke-8 algoritma <i>insertion sort</i> mendapatkan 39470 KB pada 100 data.	Pada <i>running</i> ke-8 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 2 ms pada 60 data dan 1 ms pada 100 data.

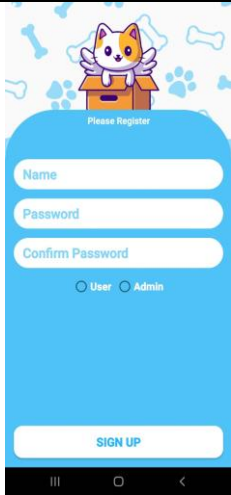
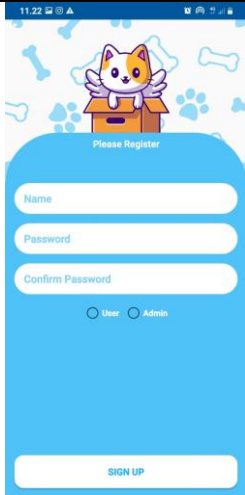
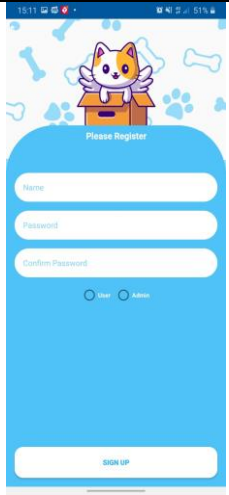

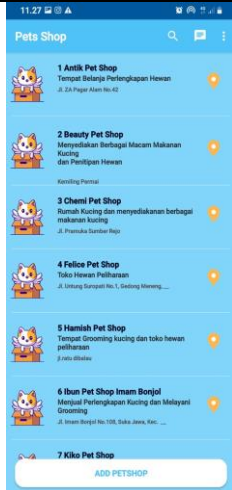
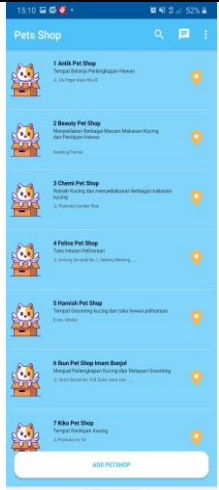
Tabel 1. Hasil Uji Penggunaan Algoritma *Selection Sort* dan *Insertion Sort* (Lanjutan).

Proses	<i>Memory Usage Selection Sort</i>	<i>Waktu Eksekusi Selection Sort</i>	<i>Memory Usage Insertion Sort</i>	<i>Waktu Eksekusi Insertion Sort</i>																																								
Running Ke-9 (Sembilan)	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>15102B</td> <td>15163B</td> <td>15162B</td> <td>15152B</td> <td>15179B</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	15102B	15163B	15162B	15152B	15179B	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>1ms</td> <td>1ms</td> <td>0ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	1ms	1ms	0ms	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>24622B</td> <td>24698B</td> <td>24690B</td> <td>34103B</td> <td>30827B</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	24622B	24698B	24690B	34103B	30827B	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>1ms</td> <td>1ms</td> <td>1ms</td> <td>0ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	1ms	1ms	1ms	0ms
20 data	40 data	60 data	80 data	100 data																																								
15102B	15163B	15162B	15152B	15179B																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	1ms	1ms	0ms																																								
20 data	40 data	60 data	80 data	100 data																																								
24622B	24698B	24690B	34103B	30827B																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	1ms	1ms	1ms	0ms																																								
Keterangan	Pada <i>running</i> ke-9, algoritma <i>selection sort</i> mendapatkan 15179 KB pada 100 data.	Pada <i>running</i> ke-9 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1ms pada 60 data dan 1 ms pada 80 data.	Pada <i>running</i> ke-9 algoritma <i>insertion sort</i> mendapatkan 30827 KB pada 100 data.	Pada <i>running</i> ke-9 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 1 ms pada 40 data, 1 ms pada 60 data dan 1 ms pada 80 data.																																								
Running Ke-10 (Sepuluh)	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>29672B</td> <td>29672B</td> <td>29672B</td> <td>29672B</td> <td>29688B</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	29672B	29672B	29672B	29672B	29688B	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>0ms</td> <td>1ms</td> <td>0ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	0ms	1ms	0ms	<p>Algorithm execution memory table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>31298B</td> <td>31348B</td> <td>31778B</td> <td>35958B</td> <td>36375B</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	31298B	31348B	31778B	35958B	36375B	<p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>20 data</th> <th>40 data</th> <th>60 data</th> <th>80 data</th> <th>100 data</th> </tr> </thead> <tbody> <tr> <td>0ms</td> <td>0ms</td> <td>0ms</td> <td>0ms</td> <td>3ms</td> </tr> </tbody> </table>	20 data	40 data	60 data	80 data	100 data	0ms	0ms	0ms	0ms	3ms
20 data	40 data	60 data	80 data	100 data																																								
29672B	29672B	29672B	29672B	29688B																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	0ms	1ms	0ms																																								
20 data	40 data	60 data	80 data	100 data																																								
31298B	31348B	31778B	35958B	36375B																																								
20 data	40 data	60 data	80 data	100 data																																								
0ms	0ms	0ms	0ms	3ms																																								
Keterangan	Pada <i>running</i> ke-10, algoritma <i>selection sort</i> mendapatkan 29688 KB pada 100 data.	Pada <i>running</i> ke-10 algoritma <i>selection sort</i> mendapatkan waktu eksekusi 1 ms pada 80 data.	Pada <i>running</i> ke-10 algoritma <i>insertion sort</i> mendapatkan 36375 KB pada 100 data.	Pada <i>running</i> ke-10 algoritma <i>insertion sort</i> mendapatkan waktu eksekusi 3 ms pada 100 data.																																								



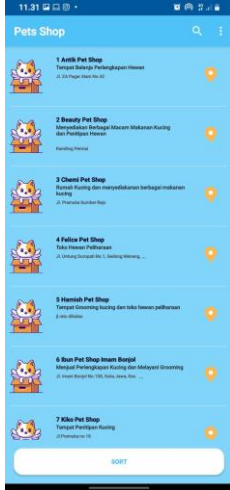



Tabel 2. Hasil Uji *Interface*.

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Tampilan <i>Interface Splash Screen</i>			
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>splash screen</i> pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>splash screen</i> pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>splash screen</i> pada perangkat ini.
Tampilan <i>Interface Sign Admin</i>			


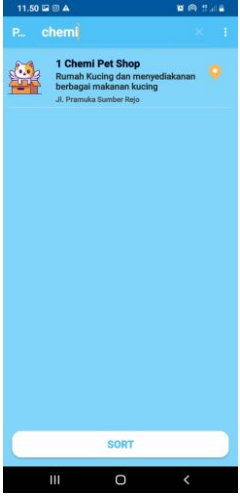

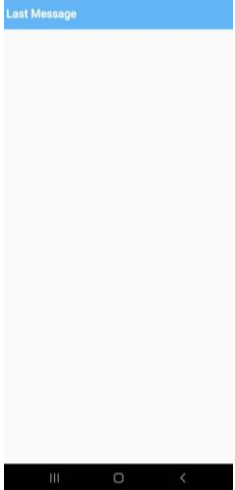
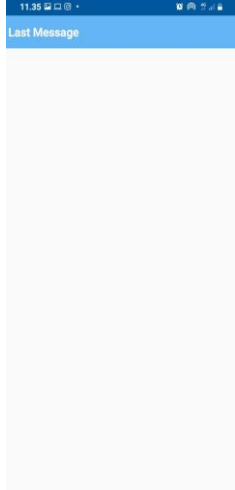
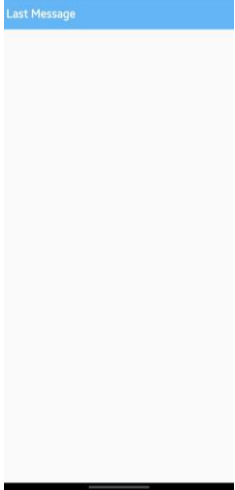
Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sign admin</i> pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sign admin</i> pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sign admin</i> pada perangkat ini.
Tampilan <i>Interface Register</i>			
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga pengguna dapat mendaftar pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga pengguna dapat mendaftar pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga pengguna dapat mendaftar pada perangkat ini.
Tampilan <i>Interface Daftar Pet Shop Admin</i>			

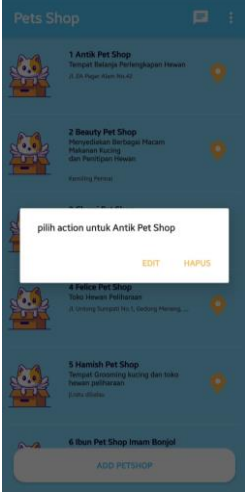
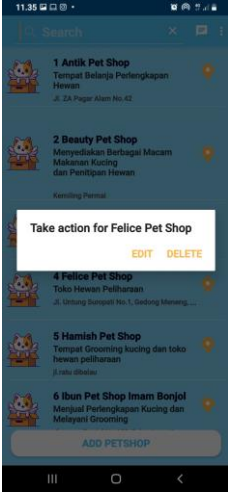
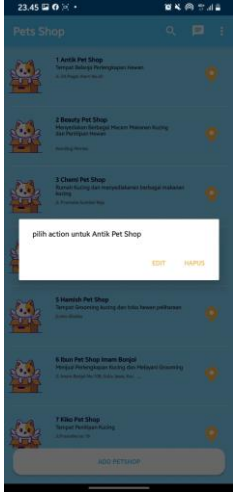

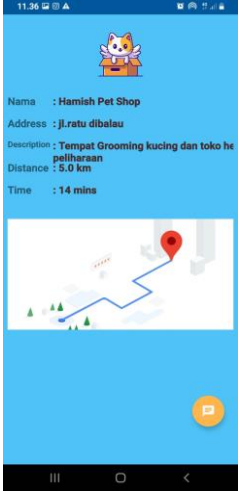
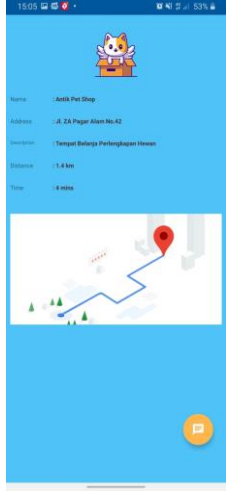
Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka menu untuk melihat <i>Pet Shop</i> yang tersedia	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka menu untuk melihat <i>Pet Shop</i> yang tersedia	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka menu untuk melihat <i>Pet Shop</i> yang tersedia
Tampilan <i>Interface</i> Halaman Daftar <i>Pet Shop User</i>			
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman daftar nama-nama <i>Pet Shop</i> .	Aplikasi dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman daftar nama-nama <i>Pet Shop</i> .	Aplikasi dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman daftar nama-nama <i>Pet Shop</i> .
Tampilan <i>Interface</i> Halaman Tambah <i>Pet Shop</i>			

Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka halaman tambah <i>Pet Shop</i> .	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka halaman tambah <i>Pet Shop</i> .	Aplikasi dapat berfungsi dengan baik sehingga admin dapat membuka halaman tambah <i>Pet Shop</i> .
Tampilan <i>Interface</i> Halaman Pencarian <i>Pet Shop User</i>			
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman pencarian nama <i>Pet Shop</i> pada <i>user</i> .	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman pencarian nama <i>Pet Shop</i> pada <i>user</i> .	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman pencarian nama <i>Pet Shop</i> pada <i>user</i> .
Tampilan <i>Interface</i> Halaman <i>Chat</i>			


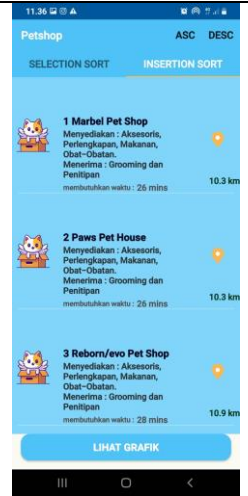

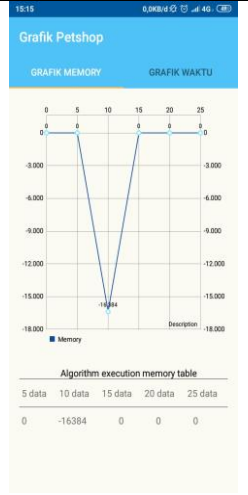
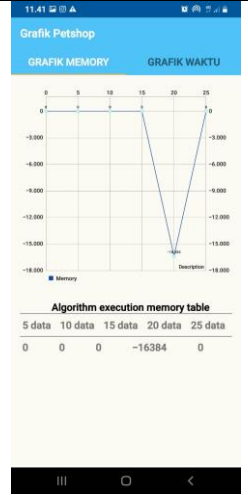
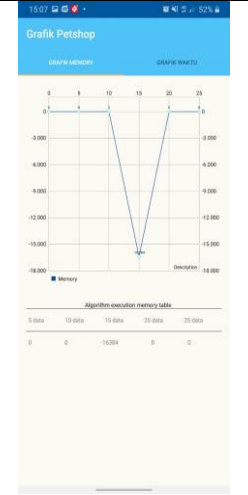
Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman <i>chat</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman <i>chat</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga <i>user</i> dapat membuka halaman <i>chat</i> pada perangkat ini.
Tampilan <i>Interface</i> Halaman Opsi			
Keterangan	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman opsi pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman opsi pada perangkat ini.	Aplikasi dapat berfungsi dengan baik sehingga dapat membuka halaman opsi pada perangkat ini.
Tampilan <i>Interface</i> Halaman Deskripsi <i>Pet Shop</i> dan <i>Map View</i>			

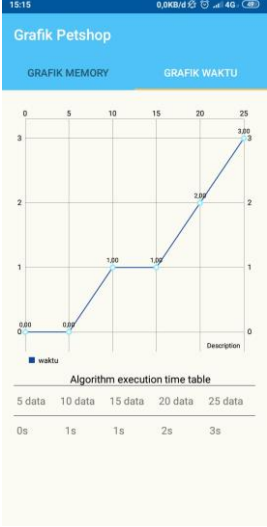
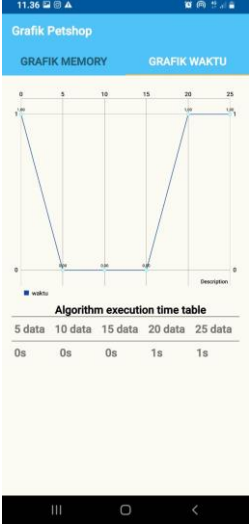
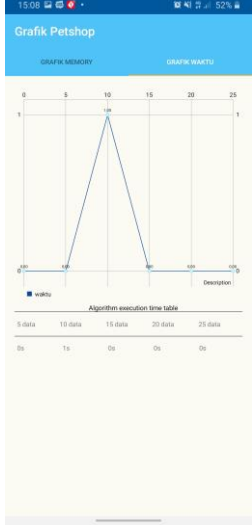
Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	Device 1	Device 2	Device 3
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman deskripsi <i>Pet Shop</i> di perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman deskripsi <i>Pet Shop</i> di perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman deskripsi <i>Pet Shop</i> di perangkat ini.
Tampilan <i>Interface</i> Halaman Rute <i>Pet Shop</i>			
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>maps view</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>maps view</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>maps view</i> pada perangkat ini.
Tampilan <i>Interface</i> Halaman <i>Sorting Selection Sort</i>			

Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi		
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Selection Sort</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Selection Sort</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Selection Sort</i> pada perangkat ini.
Tampilan Halaman <i>Interface Sorting Insertion Sort</i>			
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Insertion Sort</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Insertion Sort</i> pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka halaman <i>sorting Insertion Sort</i> pada perangkat ini.
Tampilan Halaman <i>Interface Grafik Memori Selection Sort</i>			

Tabel 2. Hasil Uji *Interface* (Lanjutan).

Proses	Hasil <i>screenshot</i> pengujian aplikasi																																
	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>																														
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik memori pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik memori pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik memori pada perangkat ini.																														
Tampilan Halaman <i>Interface</i> Grafik Waktu <i>Selection Sort</i>	 <p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>5 data</th> <th>10 data</th> <th>15 data</th> <th>20 data</th> <th>25 data</th> </tr> </thead> <tbody> <tr> <td>0s</td> <td>1s</td> <td>1s</td> <td>2s</td> <td>3s</td> </tr> </tbody> </table>	5 data	10 data	15 data	20 data	25 data	0s	1s	1s	2s	3s	 <p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>5 data</th> <th>10 data</th> <th>15 data</th> <th>20 data</th> <th>25 data</th> </tr> </thead> <tbody> <tr> <td>0s</td> <td>0s</td> <td>0s</td> <td>1s</td> <td>1s</td> </tr> </tbody> </table>	5 data	10 data	15 data	20 data	25 data	0s	0s	0s	1s	1s	 <p>Algorithm execution time table</p> <table border="1"> <thead> <tr> <th>5 data</th> <th>10 data</th> <th>15 data</th> <th>20 data</th> <th>25 data</th> </tr> </thead> <tbody> <tr> <td>0s</td> <td>1s</td> <td>0s</td> <td>0s</td> <td>0s</td> </tr> </tbody> </table>	5 data	10 data	15 data	20 data	25 data	0s	1s	0s	0s	0s
5 data	10 data	15 data	20 data	25 data																													
0s	1s	1s	2s	3s																													
5 data	10 data	15 data	20 data	25 data																													
0s	0s	0s	1s	1s																													
5 data	10 data	15 data	20 data	25 data																													
0s	1s	0s	0s	0s																													
Keterangan	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik waktu pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik waktu pada perangkat ini.	Aplikasi dan tombol dapat berfungsi dengan baik sehingga dapat membuka grafik waktu pada perangkat ini.																														

PSEUDOCODE ALGORITMA SELECTION SORT DAN INSERTION SORT

ALGORITMA SELECTION SORT

```
package com.apps.petsshop.fragment
import android.util.Log
object Selection {
    fun selection_sort(sampleArray:IntArray): IntArray {
        var n=sampleArray.size
        var temp:Int
        for(i in 0 until n){
            var indexOfMin = i
            for(j in n-1 downTo i){
                if(sampleArray[j]< sampleArray[indexOfMin])
                    indexOfMin=j
            }
            if(i!=indexOfMin){
                temp = sampleArray[i]
                sampleArray[i]= sampleArray[indexOfMin]
                sampleArray[indexOfMin]=temp
            }
        }
        return sampleArray
    }
    fun selection_sortDes(sampleArray:IntArray): IntArray {
        var n=sampleArray.size
        var temp:Int
        for(i in 0 until n){
            var indexOfMax = i
            for(j in n-1 downTo i){
                if(sampleArray[j] > sampleArray[indexOfMax])
                    indexOfMax=j
            }
            if(i!=indexOfMax){
                temp = sampleArray[i]
                sampleArray[i]= sampleArray[indexOfMax]
                sampleArray[indexOfMax]=temp
            }
        }
        return sampleArray
    }
    fun selectionMemory(sampleArray:IntArray):Long{
        var freemem = Runtime.getRuntime().freeMemory()
        Log.d("pretttfree", freemem.toString())
        var n=sampleArray.size
        var temp:Int
        for(i in 0 until n){
            var indexOfMin = i
            for(j in n-1 downTo i){
```



```

        if(sampleArray[j]< sampleArray[indexOfMin])
            indexOfMin=j
    }
    if(i!=indexOfMin){
        temp = sampleArray[i]
        sampleArray[i]= sampleArray[indexOfMin]
        sampleArray[indexOfMin]=temp
    }
}
var total = Runtime.getRuntime().totalMemory()
Log.d("pretttttotal", freemem.toString())
return total- freemem
}
fun selectionTime(sampleArray:IntArray):Long{
    var startTime = System.currentTimeMillis()
    var n=sampleArray.size
    var temp:Int
    for(i in 0 until n){
        var indexOfMin = i
        for(j in n-1 downTo i){
            if(sampleArray[j]< sampleArray[indexOfMin])
                indexOfMin=j
        }
        if(i!=indexOfMin){
            temp = sampleArray[i]
            sampleArray[i]= sampleArray[indexOfMin]
            sampleArray[indexOfMin]=temp
        }
    }
    var endTime = System.currentTimeMillis()
    return endTime - startTime
}
}
fun main(arg: Array<String>) {
    println("Before Sort")
    var A = intArrayOf(1, 7, 3, 9, 4)
    var B = intArrayOf(1, 7, 3, 9, 4)
    for (i in A) print(i)
    Selection.selection_sort(A)
    println("")
    println("Sorted array is : ")
    for (i in A) print(i)
    Selection.selection_sortDes(B)
    println("")
    println("Sorted array is : ")
    for (x in B) print(x)
}

```

ALGORITMA *INSERTION SORT*

```
package com.apps.petsshop.fragment
import android.util.Log
import com.apps.petsshop.fragment.Insertion.insertionsort
import com.apps.petsshop.fragment.Insertion.insertionsortDesc
object Insertion {
    fun insertionsort(items: MutableList<Int>):List<Int>{
        if (items.isEmpty() || items.size<2){
            return items
        }
        for (count in 1 until items.count()){
            // println(items)
            val item = items[count]
            var i = count
            while (i>0 && item < items[i - 1]){
                items[i] = items[i - 1]
                i -= 1
            }
            items[i] = item
        }
        sortJarak.addAll(items)
        return items
    }
    fun insertionsortDesc(items: MutableList<Int>):List<Int>{
        if (items.isEmpty() || items.size<2){
            return items
        }
        for (count in 1 until items.count()){
            // println(items)
            val item = items[count]
            var i = count
            while (i>0 && item > items[i - 1]){
                items[i] = items[i - 1]
                i -= 1
            }
            items[i] = item
        }
        sortJarak.addAll(items)
        return items
    }
}

fun insertionMemory(items:MutableList<Int>):Long{
    var freemem = Runtime.getRuntime().freeMemory()
    Log.d("pretttfree", freemem.toString())
    for (count in 1 until items.count()){
        println(items)
        val item = items[count]
        var i = count
```

```

        while (i>0 && item < items[i - 1]){
            items[i] = items[i - 1]
            i -= 1
        }
        items[i] = item
    }
    var total = Runtime.getRuntime().totalMemory()
    Log.d("pretttttotal", freemem.toString())
    return total- freemem
}

fun insertionTime(items: MutableList<Int>):Long{
    var startTime = System.currentTimeMillis()
    for (count in 1 until items.count()){
        // println(items)
        val item = items[count]
        var i = count
        while (i>0 && item < items[i - 1]){
            items[i] = items[i - 1]
            i -= 1
        }
        items[i] = item
    }
    var endTime = System.currentTimeMillis()
    return endTime - startTime
}

fun main(arg: Array<String>) {
    val names = mutableListOf(8, 3, 2, 7, 4)
    val desc = mutableListOf(8, 3, 2, 7, 4)
    println(names)
    var ordered = insertionsort(names)
    println(ordered)
    var orderedesc = insertionsortDesc(desc)
    println(orderedesc)
}

```