

BAB V HASIL DAN PEMBAHASAN

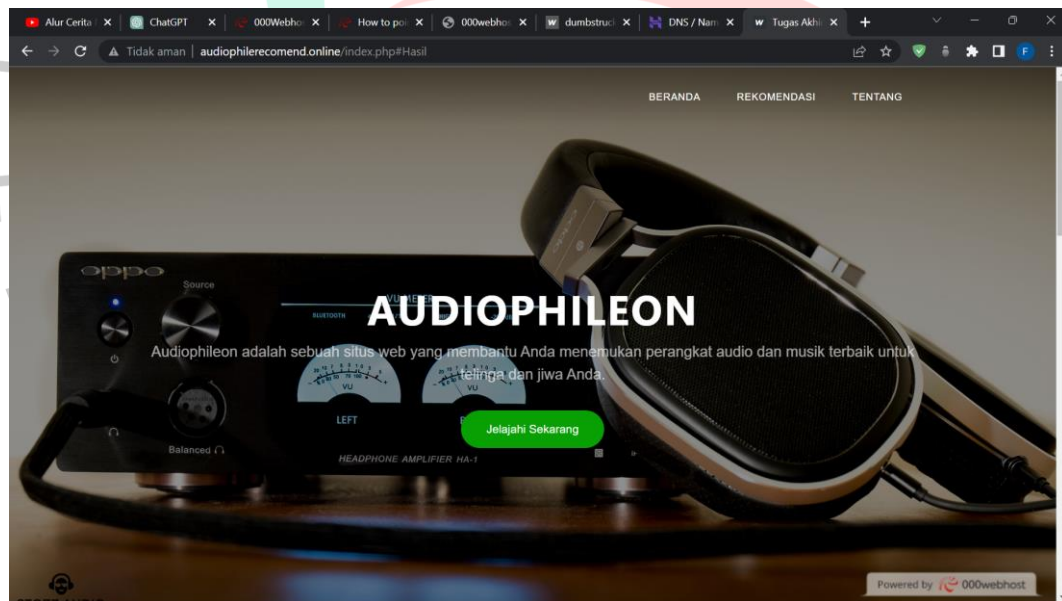
Setelah tahap perancangan, penelitian ini dikembangkan untuk melihat hasil yang diperoleh berjalan sesuai dengan rencana. Berikut adalah hasil dan pembahasan dari penelitian ini.

5.1 Hasil

Hasil yang diperoleh dari penelitian ini berupa tampilan dari aplikasi, data perhitungan algoritma SAW, dan hasil penilaian SAW.

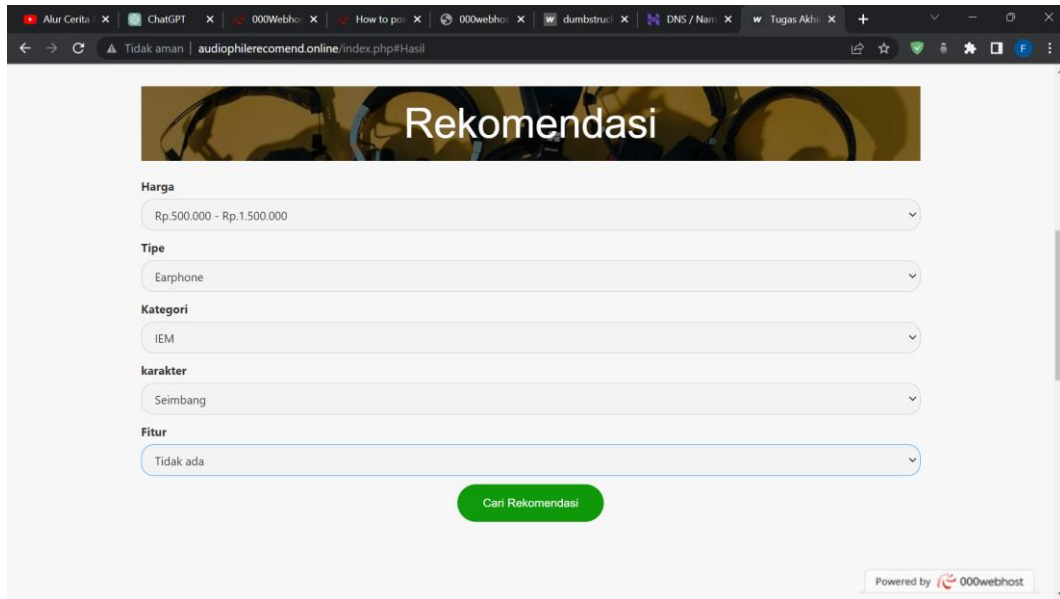
5.1.1 Hasil dari Perancangan Tampilan

Perancangan aplikasi rekomendasi perangkat audio telah berhasil dirancang. Berikut hasil perancangannya.



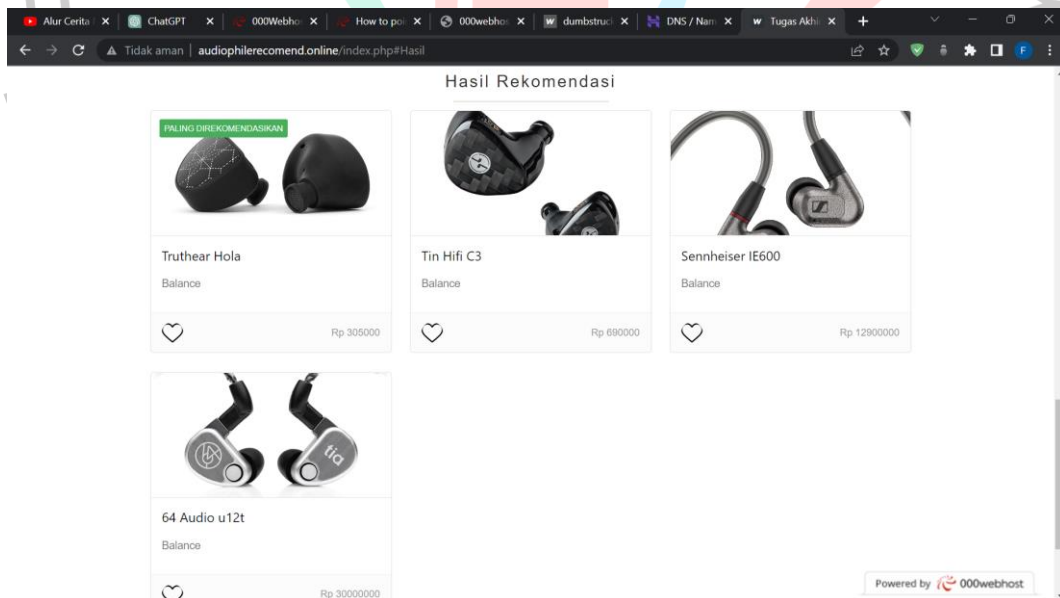
Gambar 5.1 Halaman Beranda

Gambar di atas merupakan halaman beranda dari aplikasi untuk masuk ke menu rekomendasi pengguna dapat memilih pada bagian *navbar* atau bisa *menscroll* tampilan kebawah pada bagian beranda.



Gambar 5.2 Halaman Rekomendasi

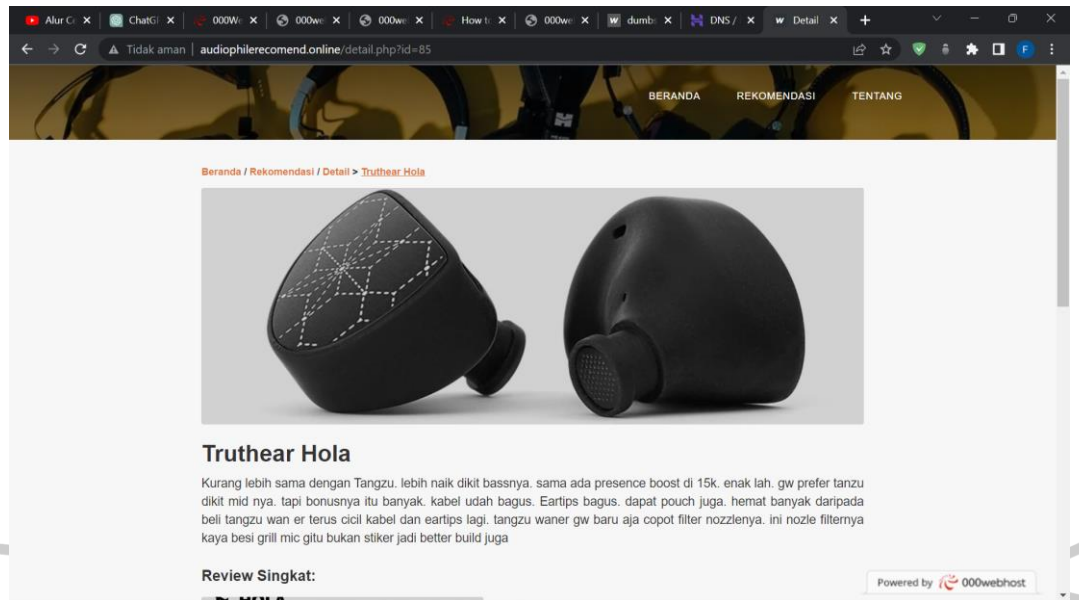
Gambar di atas merupakan tampilan pencarian rekomendasi pada aplikasi untuk mencari hasilnya pengguna dapat memilih menu yang sudah disiapkan untuk menggunakan aplikasi rekomendasi ini.



Gambar 5.3 Halaman Hasil Rekomendasi

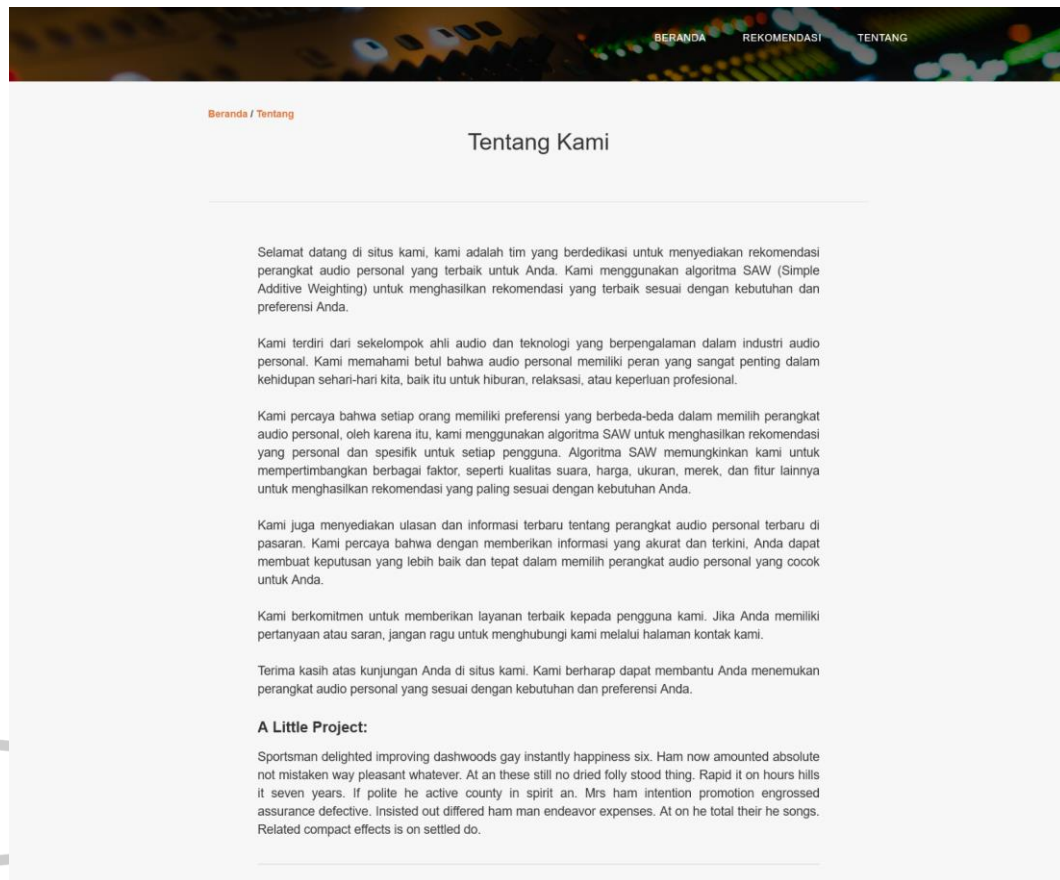
Gambar di atas merupakan halaman hasil rekomendasi dari pencarian rekomendasi. Halaman tersebut digunakan untuk menampilkan hasil dari

rekomendasi perangkat audio yang telah dicari sebelumnya, hasil tersebut berupa *card*, dan *card* tersebut dapat digunakan sebagai tombol navigasi ke halaman detail untuk melihat detail dari perangkat yang telah dipilih.



Gambar 5.4 Halaman Detail

Gambar di atas merupakan halaman detail dari perangkat audio yang sebelumnya dipilih pada bagian hasil rekomendasi atau dari halaman trending, pada halaman ini pengguna dapat melihat deskripsi dari produk yang dipilihnya, dan juga dapat mengunjungi situs toko *online* yang menjual perangkat audio yang dicari.



Gambar 5.5 Halaman Tentang

Gambar di atas merupakan halaman Tentang yang menjelaskan tentang dan maksud dari pembuatan aplikasi rekomendasi ini.

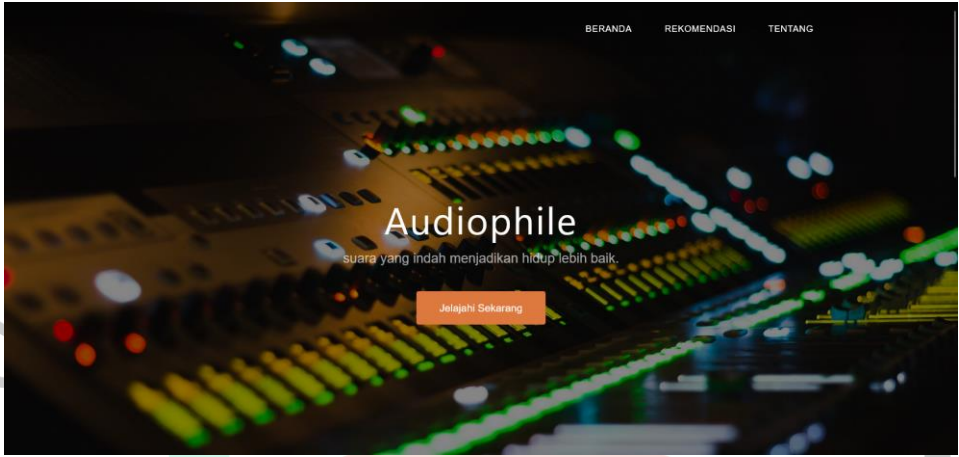
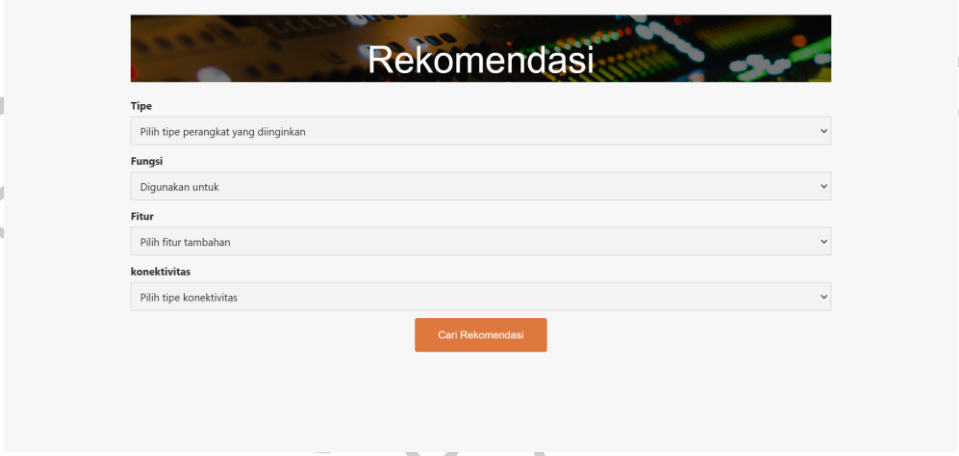
5.2 Pembahasan

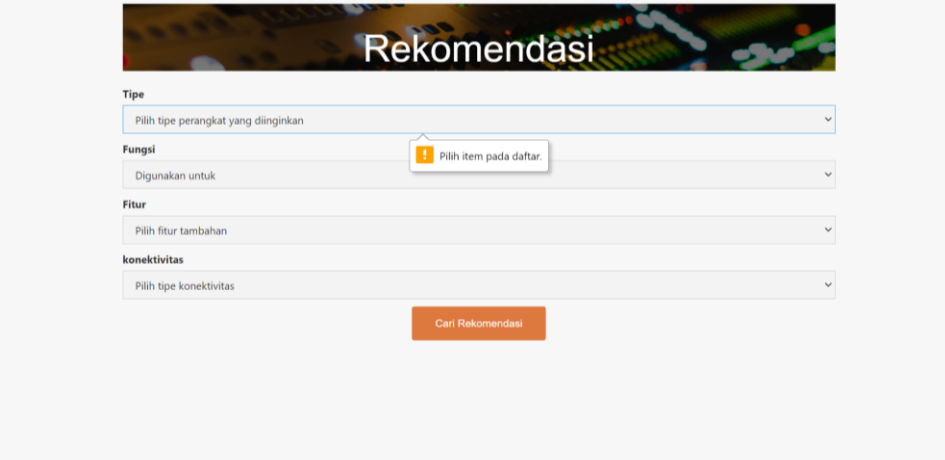
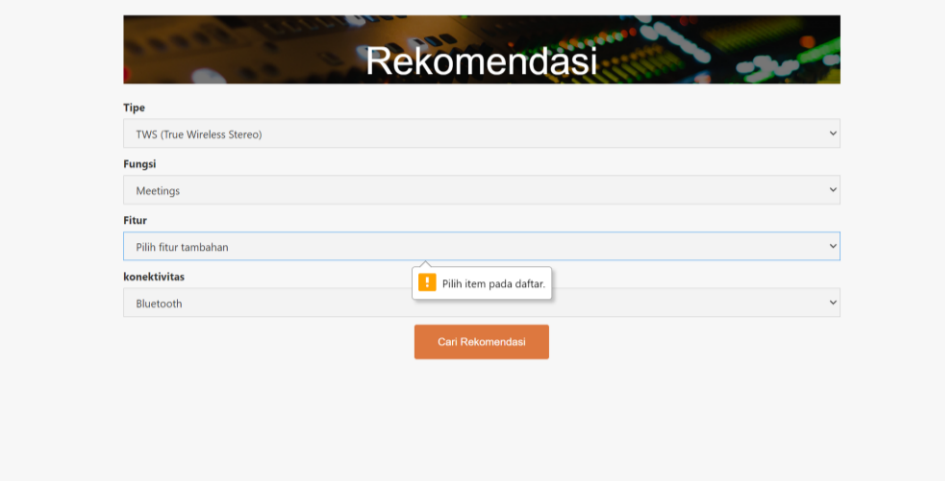
Pembahasan merupakan hasil dari pengujian sistem yang dilakukan untuk mengetahui kinerja atau fungsi dari aplikasi rekomendasi perangkat audio.

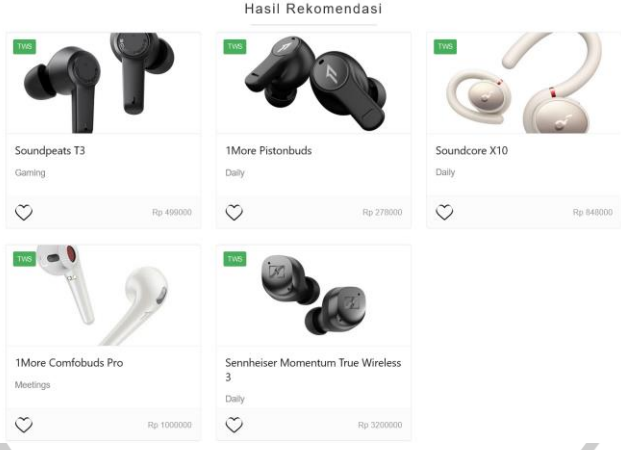
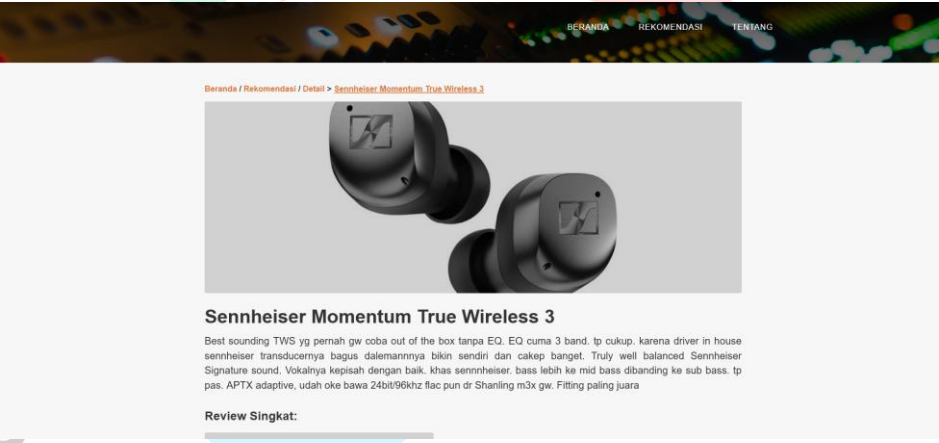
5.2.1 Pengujian Black Box

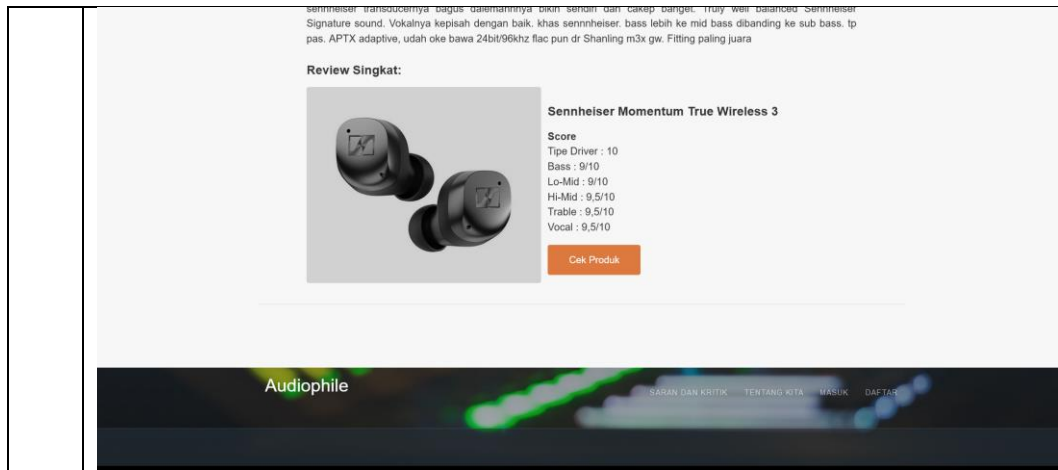
Metode *black box* merupakan sistem pengujian ini dilakukan untuk semua fitur aplikasi, sehingga penguji mengetahui apakah *output* yang diperoleh berkorelasi dengan *input* yang diberikan. Berikut adalah tabel skenario, *input*, dan hasil pengujian *blackbox*.

Tabel 5.1 Pengujian Black Box

No	Skenario Pengujian	Hasil yang diharapkan
1	Pengguna berhasil masuk ke halaman utama	Masuk ke halaman utama
	Hasil pengujian	
		
Kesimpulan: Pengguna berhasil masuk ke halaman utama		
2	Pengguna memilih menu Rekomendasi.	Mengalihkan ke halaman Rekomendasi.
	Hasil pengujian	
		
Kesimpulan: Berhasil mengalihkan ke halaman rekomendasi.		
3	Pengguna menekan tombol cari rekomendasi tanpa memilih pilihan yang ada.	Muncul notifikasi pilih item pada daftar
	Hasil pengujian	

		
	<p>Kesimpulan: Berhasil memunculkan notifikasi dan tidak dapat melakukan pencarian.</p>	
4	<p>Pengguna menekan tombol cari rekomendasi dengan data yang belum lengkap.</p>	<p>Muncul notifikasi pilih item pada daftar pada bagian yang belum diisi</p>
	<p>Hasil pengujian</p> 	
	<p>Kesimpulan: Berhasil memunculkan notifikasi dan tidak dapat melakukan pencarian.</p>	
5	<p>Aplikasi selesai memproses hasil pencarian rekomendasi.</p>	<p>Mengalihkan ke halaman hasil rekomendasi.</p>

		
	<p>Kesimpulan = Berhasil mengalihkan halaman rekomendasi menuju halaman hasil rekomendasi</p>	
6	<p>Pengguna mengklik gambar pada hasil rekomendasi</p>	<p>Mengalihkan ke halaman detail</p>
		
	<p>Kesimpulan: Berhasil menampilkan halaman detail dari perangkat yang dipilih</p>	
7	<p>Pengguna mengklik tombol “Cek Produk” pada halaman detail</p>	<p>Pengguna akan diarahkan pada <i>e-commerce</i> yang tersedia</p>

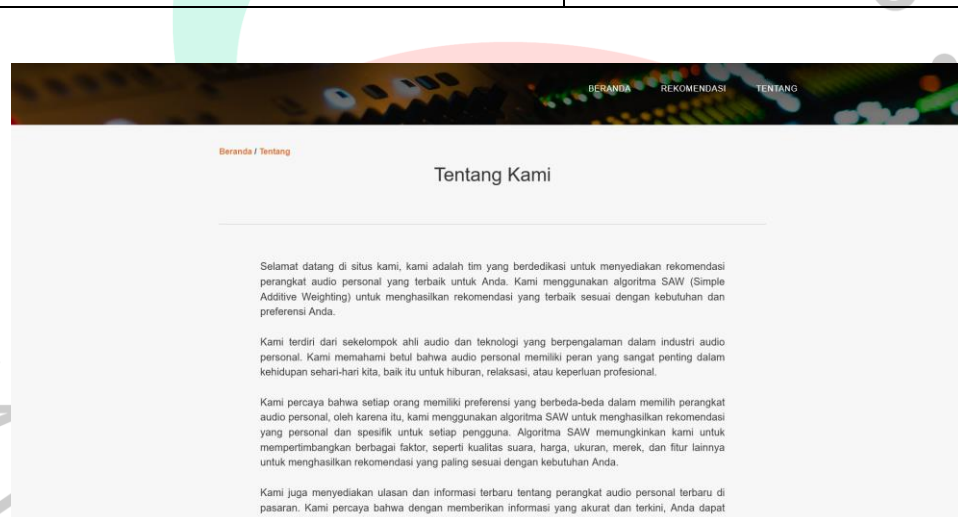


Kesimpulan: Diarahkan menuju *e-commerce*.

8

Pengguna memilih *Tentang* pada bagian *navbar*

Pengguna masuk ke halaman *Tentang*.



Kesimpulan: Berhasil masuk ke halaman *Tentang*.

5.2.2 Pengujian White Box

Pengujian *white box* diperlukan untuk merancang aplikasi, karena dengan pengujian ini dapat memastikan aplikasi ini berjalan dengan lancar sesuai dengan skenario yang diinginkan. Berikut adalah pengujian menggunakan metode *white box*.

Tabel 5.2 Pengujian White Box

No	Hasil yang Diharapkan	Source Code	Hasil Pengujian
----	-----------------------	-------------	-----------------

1	Variabel \$bobot menyimpan bobot atau nilai penting dari setiap kriteria rekomendasi.	<pre>\$bobot = ["harga" => 0.35, "karakter" => 0.39, "fitur" => 0.26,];</pre>	Variabel \$bobot terisi dengan bobot yang telah ditentukan.
2	Variabel \$max menyimpan nilai maksimum untuk setiap kriteria rekomendasi.	<pre>\$max = ["harga" => 30000000, "n_Suara" => 9, "n_Fitur" => 10,];</pre>	Variabel \$max terisi dengan nilai maksimum dari masing-masing atribut.
3	Iterasi pertama menggunakan foreach untuk mengupdate nilai maksimum dalam \$max dengan membandingkannya dengan nilai atribut dari setiap item dalam \$data.	<pre>foreach (\$data as \$item) { \$max["n_harga"] = max(\$max["harga"], \$item["harga"]); \$max["n_Suara"] = max(\$max["n_Suara"], \$item["n_Suara"]); \$max["n_Fitur"] = max(\$max["n_Fitur"], \$item["n_Fitur"]); }</pre>	Setiap data di dalam \$data diiterasi untuk menentukan nilai maksimum dari setiap atribut dengan mengupdate nilai dalam \$max.
4	<p>Iterasi kedua menggunakan foreach dengan referensi &\$item untuk menghitung nilai normalisasi dari setiap atribut:</p> <p>\$item["normal_harga"] dihitung menggunakan rumus $(\\$item["harga"] - 58900) / (58900 - \\$max["harga"])$.</p> <p>\$item["normal_karakter"] dihitung dengan membagi nilai atribut \$item["n_Suara"] dengan \$max["n_Suara"].</p> <p>\$item["normal_tipe"] dihitung berdasarkan kondisi jika \$item["tipe"] sama dengan \$tipe maka nilainya 1, jika tidak sama maka nilainya 0.</p> <p>\$item["normal_fitur"] dihitung dengan membagi nilai atribut \$item["n_Fitur"] dengan \$max["n_Fitur"].</p>	<pre>foreach (\$data as &\$item) { \$item["normal_harga"] = (\$item["harga"] - 58900) / (58900 - \$max["harga"]); \$item["normal_karakter"] = \$item["n_Suara"] / \$max["n_Suara"]; \$item["normal_tipe"] = \$item["tipe"] == \$tipe ? 1 : 0; \$item["normal_fitur"] = \$item["n_Fitur"] / \$max["n_Fitur"]; }</pre>	Setiap data di dalam \$data diiterasi untuk menghitung nilai normalisasi dari masing-masing atribut berdasarkan nilai maksimum dalam \$max.
5	Iterasi ketiga menggunakan foreach	<pre>foreach (\$data as &\$item) { \$item["nilai"] =</pre>	Setiap data di dalam \$data

	<p>dengan referensi &\$item untuk menghitung nilai akhir (\$item["nilai"]) berdasarkan bobot yang telah ditentukan dan nilai normalisasi dari atribut-atribut:</p> <p>\$item["nilai"] dihitung menggunakan rumus: (\$bobot["harga"] * \$item["normal_harga"]) + (\$bobot["karakter"] * \$item["normal_karakter"]) + (\$bobot["fitur"] * \$item["normal_fitur"]) + \$item["normal_tipe"]. Setiap nilai akhir (\$item["nilai"]) disimpan dalam array \$SESSION['hasil'][].</p>	<pre>(\$bobot["harga"] * \$item["normal_harga"]) + (\$bobot["karakter"] * \$item["normal_karakter"]) + (\$bobot["fitur"] * \$item["normal_fitur"]) + \$item["normal_tipe"]; \$SESSION['hasil'][] = \$item["nilai"]; }</pre>	<p>diiterasi untuk menghitung nilai akhir berdasarkan bobot yang telah ditentukan dan nilai normalisasi dari atribut-atribut.</p>
6	<p>Hasil pengurutan data dalam \$data berdasarkan nilai akhir (\$item["nilai"]) secara menurun menggunakan fungsi usort.</p>	<pre>usort(\$data, function (\$a, \$b) { return \$b["nilai"] <=> \$a["nilai"]; });</pre>	<p>Hasil pengujian berupa pengurutan data dalam \$data berdasarkan nilai akhir secara menurun</p>
7	<p>Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Harga" dari form dan disimpan dalam variabel \$harga.</p>	<pre>\$harga = \$_POST['Harga'];</pre>	<p>Variabel \$harga berisi nilai yang dipilih oleh pengguna pada elemen dengan id "Harga" dari form.</p>
8	<p>Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Tipe" dari form dan disimpan dalam variabel \$tipe.</p>	<pre>\$tipe = \$_POST['Tipe'];</pre>	<p>Variabel \$tipe berisi nilai yang dipilih oleh pengguna pada elemen dengan id "Tipe" dari form.</p>
9	<p>Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Kategori2" dari form dan disimpan dalam variabel \$kategori2.</p>	<pre>\$kategori2 = \$_POST['Kategori2'];</pre>	<p>Variabel \$kategori2 berisi nilai yang dipilih oleh pengguna pada elemen dengan id</p>

			"Kategori2" dari form.
10	Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Kategori1" dari form dan disimpan dalam variabel \$kategori1.	\$kategori1 = \$_POST['Kategori1'];	Variabel \$kategori1 berisi nilai yang dipilih oleh pengguna pada elemen dengan id "Kategori1" dari form.
11	Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Karakter" dari form dan disimpan dalam variabel \$karakter.	\$karakter = \$_POST['Karakter'];	Variabel \$karakter berisi nilai yang dipilih oleh pengguna pada elemen dengan id "Karakter" dari form.
12	Mengambil nilai yang dipilih oleh pengguna pada elemen dengan id "Fitur" dari form dan disimpan dalam variabel \$fitur.	\$fitur = \$_POST['Fitur'];	Variabel \$fitur berisi nilai yang dipilih oleh pengguna pada elemen dengan id "Fitur" dari form.
13	Memeriksa apakah sesi dengan nama "hasil" sudah diatur sebelumnya. Jika sudah diatur, melakukan pengolahan tambahan sesuai dengan logika yang ada di dalam blok if.	if (isset(\$_SESSION['hasil']))	Mengecek apakah sesi dengan nama "hasil" sudah diatur sebelumnya dan melakukan pengolahan tambahan sesuai dengan logika yang ada di dalam blok if.
14	Menampilkan tombol submit dengan teks "Cari Rekomendasi" pada halaman form.	<input type="submit" value="Cari Rekomendasi" data-wait="Please wait..." class="submit-button-2 w-button" />	Menampilkan tombol submit dengan teks "Cari Rekomendasi" pada halaman form.