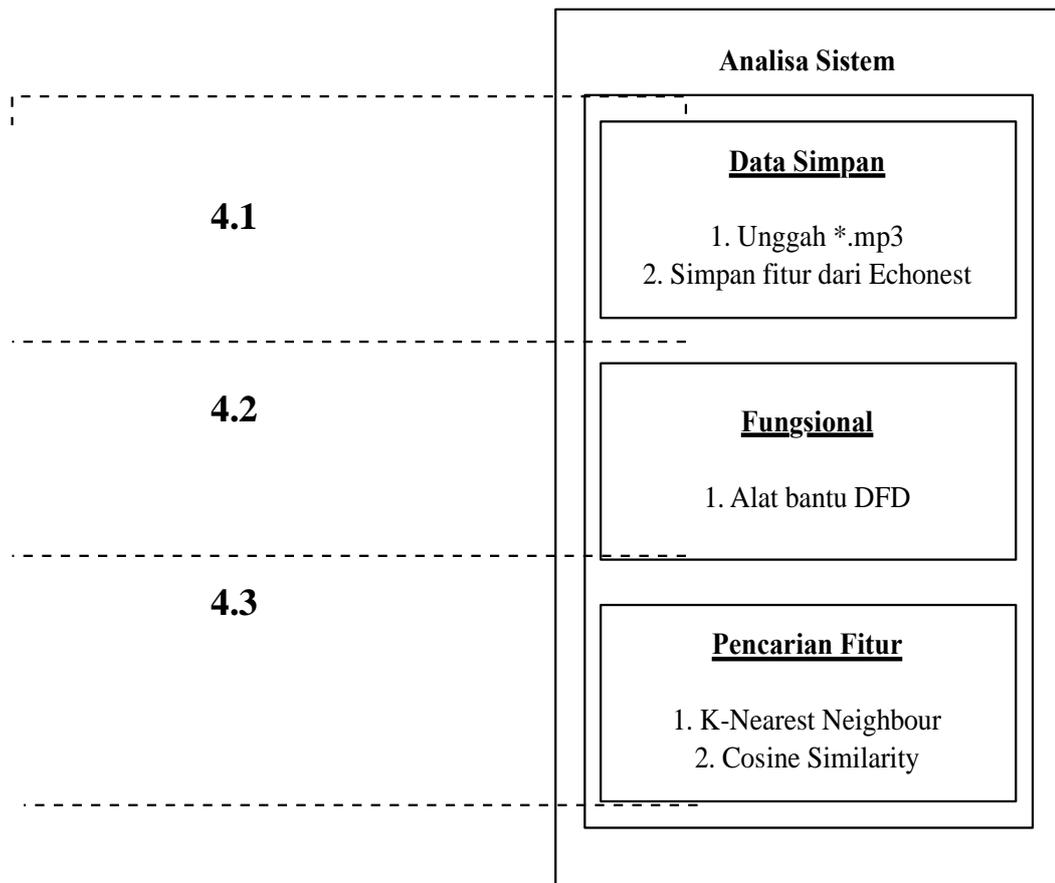


BAB IV

ANALISA DAN PERANCANGAN

Analisa dan perancangan yang akan dilakukan adalah menganalisa mengenai tahapan sistem bekerja serta menguji tingkat akurasi dari metode tersebut dalam proses pencarian musik. Bab ini akan fokus kepada tahapan pencarian musik serta rancangan tampilan sistem dan bagaimana proses *output* dihasilkan.

Sebelum dimulai tahapan pencarian musik dengan metode K-Nearest Neighbor dan Cosine Similarity, terdapat langkah-langkah dalam proses tersebut. Sub bab berikut akan membahas mengenai proses tersebut. Berikut adalah gambar *flowchart* dari tahapan pencarian musik.



Gambar 4.1 Flowchart penelitian

4.1 Analisa Sistem Data Simpan

Data simpan yang ada didalam sistem *Music Information Retrieval* ini merupakan hasil dari unggahan ke website Echonest kemudian mendapatkan nilai-nilai dari fitur yang akan disimpan menggunakan *database* MySQL dan menggunakan satu tabel yaitu tabel *musicdataset*. Tabel ini memiliki 9 buah *field*, yaitu *songID* sebagai *primary key* dengan tipe data *float*, *artistname* dengan tipe data *varchar(20)*, *title* sebagai judul lagu memiliki tipe data *varchar(20)*, *key* memiliki tipe data *float*, *mode* memiliki tipe data *float*, *loudness* memiliki tipe data *float*, *energy* memiliki tipe data *float*, *tempo* memiliki tipe data *float*, dan terakhir *genre* memiliki tipe data *varchar(20)*.

Tabel 4.1 Field *musicdataset*

Nama Field	Tipe Data
songID	Float
artistname	varchar
title	varchar
key	float
mode	float
loudness	float
energy	float
tempo	float
genre	varchar

Data yang dibutuhkan dalam penelitian ini adalah 280 musik yang dijadikan korpus dalam bentuk *.mp3 yang di susun dengan bit rate 64 kbit, kemudian dari 280 diambil 30 musik dengan metode sampel dijadikan bit rate 32 kbit ini dapat dilihat dari table dibawah ini :

Tabel 4.2 Data Musik 64 kbit

Artis	Judul	K	M	L	E	T	G
Fatin	Aku Memilih Setia	7	1	-8.211	0.335712	134.956	pop
Inul Daratista	Masa Lalu	2	0	-4.46	0.9325	91.387	Dangdut
Imelda	Kereta Malam	0	1	-5.626	0.924832	97.49	Dangdut

Peterpan	Kisah Cinta Ku	0	1	-7.795	0.494306	131.959	Pop
D'Massive	Jangan Menyerah	0	1	-9.442	0.543326	96.999	Pop

Table diatas selengkapnya dapat dilihat di lampiran A

Keterangan :

K = Key

M= Mode

L = Loudness

E = Energi

T = Tempo

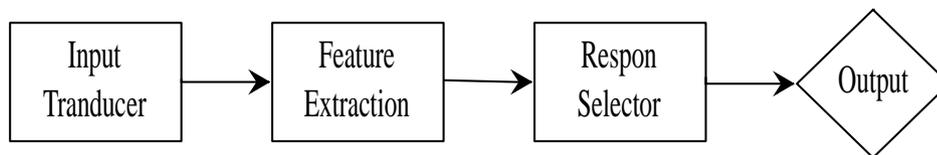
G = Genre

4.2 Analisa Fungsional

Sistem yang akan dibangun adalah sistem untuk mengidentifikasi suatu file bertipe MP3 dengan nama sistem adalah *Music Information Retrieval* Menggunakan API Echonest. Sistem pencarian informasi ini akan berbasis web yang nantinya *user* bisa mengunggah *file* mp3 dan sistem akan memberikan informasi tentang file tersebut. Jika data tidak ada, maka sistem akan menampilkan rekomendasi musik yang memiliki fitur yang mirip dengan *file* mp3 yang diunggah. Sistem ini akan digambarkan dengan beberapa perancangan seperti gambaran umum sistem, *data flow diagram*, *data flow diagram level 1*, deskripsi data, arsitektur sistem, dan desain antarmuka.

4.2.1 Gambaran Umum Sistem

Struktur sistem dari *Music Information Retrieval* ini adalah sebagai berikut:



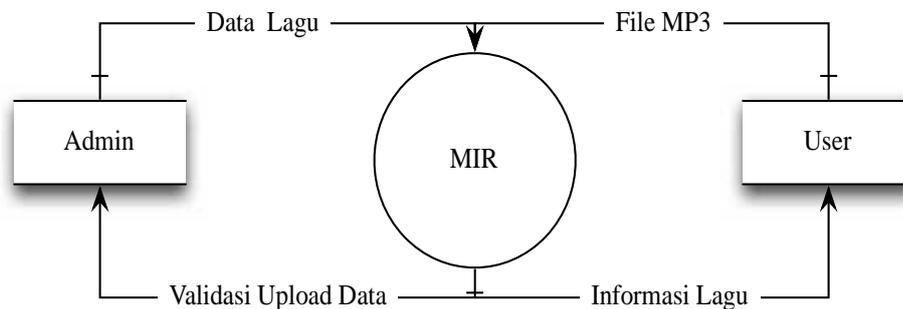
Gambar 4.2 Struktur sistem

Pada gambar 4.2 *input transducer*, proses yang dilakukan adalah pengunggahan *file* *.mp3 oleh pengguna sistem. Kemudian proses selanjutnya

adalah *Feature Extraction* yang dilakukan di *website Echonest* untuk menghasilkan fitur-fitur musik yang ada pada suatu musik. *Respon Selector* merupakan algoritma yang digunakan dalam identifikasi lagu. Dalam penelitian ini algoritma yang digunakan adalah algoritma *K-Nearest Neighbor* dan *Cosine Similarity*. Dimana perhitungan jarak *Euclidean* pada algoritma *K-Nearest Neighbor* diganti dengan perhitungan nilai *cosine similarity* dengan tujuan untuk menghindari adanya persamaan nilai jarak *Euclidean* dari data uji ke data sampel. *Output* yang dihasilkan dari sistem adalah berupa teks berisi judul, nama artis, dan informasi-informasi tentang lagu tersebut.

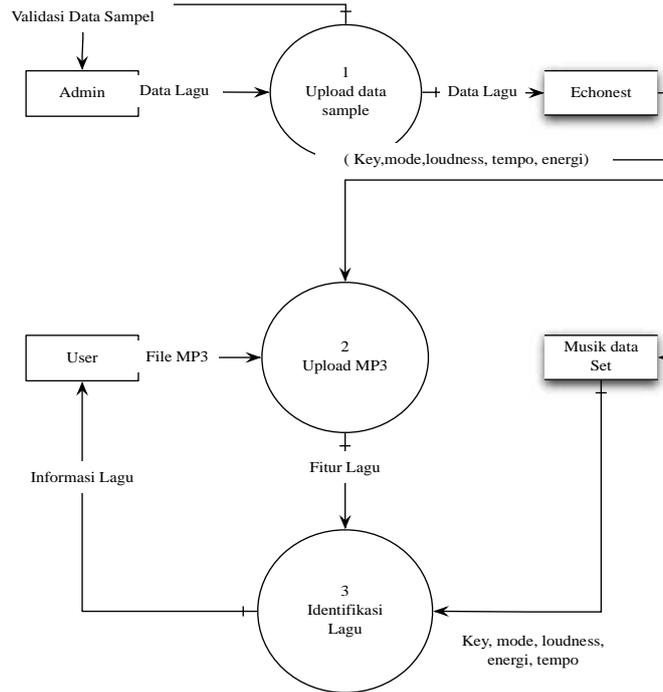
4.2.2 Data Flow Diagram (DFD)

Untuk menggambarkan alur data dalam proses-proses yang terjadi dalam sistem *Music Information Retrieval* digunakan DFD. Diagram DFD yang digunakan adalah *context diagram*, dan DFD.



Gambar 4.3 Context Diagram MIR

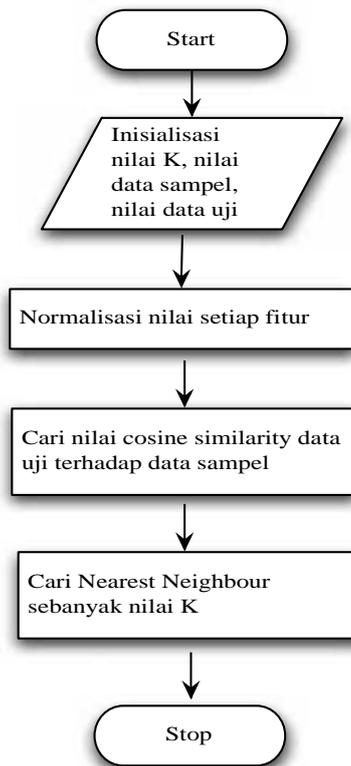
Pada gambar 4.3 *Context Diagram MIR* diatas menggambarkan *user* saat memasukan data dan keluaran yang didapatkan oleh *user*. *User* memberi masukan berupa *file* MP3 dan mendapatkan keluaran berupa judul lagu dari *file* yang diunggah dan rekomendasi lagu yang memiliki fitur yang mirip dengan *file* yang diunggah.



Gambar 4.4 Data flow diagram Level 0

Pada gambar 4.4 DFD level 0, *user* memberikan masukan berupa file MP3 ke *website* Echonest. Echonest memberikan hasil ekstraksi berupa fitur lagu dari *file* yang diupload dan akan dijadikan sebagai data uji pada proses 3. Data sampel diambil dari Echonest dan disimpan dalam *music data sampel*. Selanjutnya proses 3 dilakukan identifikasi lagu. Hasil dari proses 3 berupa informasi lagu yang diterima oleh user.

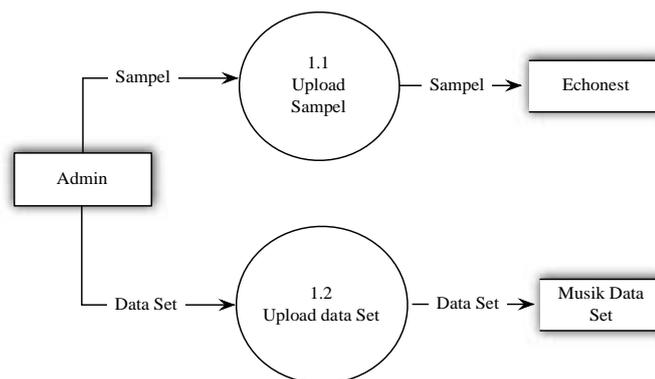
Pada proses 3, DFD digambarkan pada gambar 4.4 Dimana fitur lagu yang diunggah akan digunakan sebagai data uji dan data dari *music data sampel* akan dijadikan data sampel. Setelah dilakukan proses 3.1, maka akan didapatkan nilai *cosine similarity* yang akan digunakan pada proses 3.2. Kemudian rekomendasi lagu tersebut akan dicek untuk menentukan apakah lagu yang dicari ada atau tidak. Jika ada, maka sistem akan memberikan informasi lagu tersebut dan jika tidak ada, maka akan diberikan rekomendasi lagu yang memiliki fitur yang mirip. Untuk *flowchart* dari algoritma K-Nearest Neighbor yang digunakan dapat dilihat pada gambar 4.5.



Gambar 4.5 Flowchart algoritma K-Nearest Neighbor

4.2.3 Data Flow Diagram Level 1

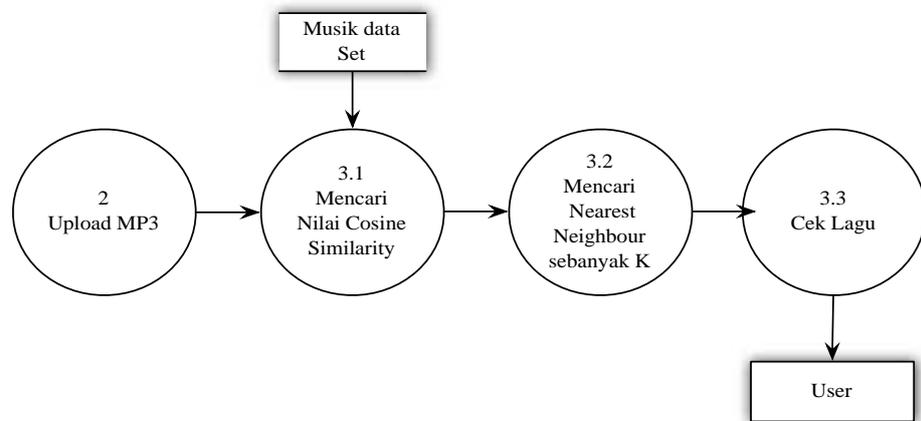
a. Proses 1 (upload data sampel)



Gambar 4.6 DFD Level 1

Pada proses upload data sampel digambarkan dengan jelas pada gambar 4.6 dimana admin bertugas untuk mengunggah sample ke echonest dan mengunggah kembali hasil dari echonest ke musik data set didalam database.

b. Proses 3 (Identifikasi Lagu)

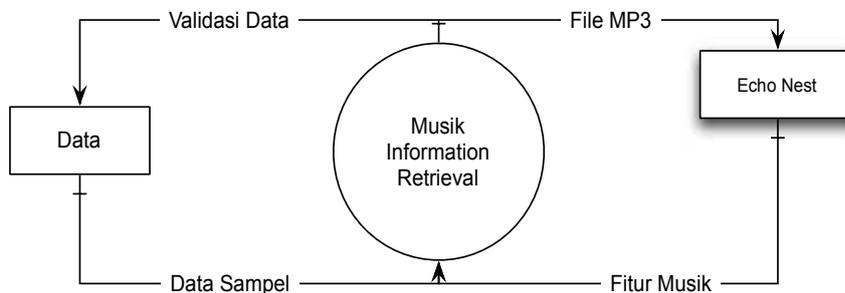


Gambar 4.7 DFD Level 1 Proses 3

Pada proses identifikasi lagu seperti yang terlihat pada gambar 4.7, bahwa musik yang masuk didalam data set akan mencari nilai Cosine Similarity dan dilanjutkan dengan mencari nilai K-Nearest Neighbor barulah mengecek lagu dan lagu yang terpilih akan ditampilkan pada user

4.2.4 Arsitektur Sistem

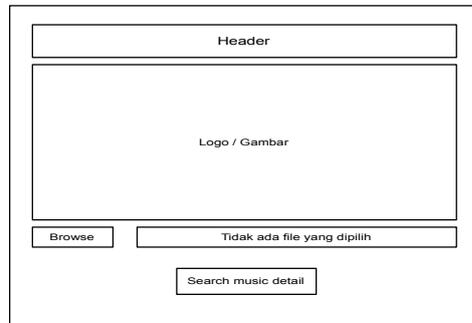
Arsitektur sistem yang dibangun pada MIR (*music information retrieval*) menggunakan teknologi seperti yang terlihat pada gambar 4.8. Saat user *upload file* menggunakan *cURL PHP*. Data uji merupakan hasil dari ekstraksi Echonest. Identifikasi musik menggunakan teknologi PHP.



Gambar 4.8 Arsitektur sistem MIR

4.2.5 Desain Antarmuka

Desain antarmuka untuk sistem MIR ini memiliki desain antarmuka berupa halaman web.



Gambar 4.9 Halaman Pencarian MIR

Halaman awal dari sistem MIR berisi *form HTML* yang digunakan untuk mengupload MP3.

4.3 Pencarian Fitur

Pencarian fitur data sampel dari MIR ini dilakukan dengan cara mengunggah lagu dengan *bit rate* 64 kbit satu per satu ke *website* Echonest untuk mendapatkan fitur-fitur yang nantinya akan digunakan untuk data sampel pada sistem MIR itu sendiri. Tahapan untuk pengambilan data sampel adalah sebagai berikut.

4.3.1. Inisialisasi URL Link Echonest

URL ini adalah url Echonest dimana lagu yang diunggah akan diekstrak dan diambil fitur-fitur musiknya. Penggalan kode dari inisialisasi URL adalah sebagai berikut.

```
$url=  
"http://developer.echonest.com/api/v4/track/upload?api_key=VEWHBITMWZ  
MSBMO85&format=json&filetype=mp3";
```

4.3.2 Inisialisasi Lokasi *File* MP3

Inisialisasi ini bertujuan untuk mendapatkan lokasi file MP3 yang akan diunggah. Lokasi ini bisa berupa lokasi pada *local drive* maupun *link website*. Penggalan kode dapat dilihat sebagai berikut.

```
$post_data = array('file_contents'=>'@'.$fMP3);
```

4.3.3 Eksekusi Sintaks Curl PHP

Eksekusi sintaks untuk memulai mengunggah *file* MP3. Penggalan kode dapat dilihat sebagai berikut.

```
$ch = curl_init();  
curl_setopt($ch, CURLOPT_TIMEOUT, 40000);  
curl_setopt($ch, CURLOPT_HTTPHEADER, Array("Content  
Type:application/octet-stream"));  
curl_setopt($ch, CURLOPT_URL, $url);  
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
curl_setopt($ch, CURLOPT_VERBOSE, 1);  
$data = curl_exec($ch);
```

4.3.4 Pengambilan ID Upload Musik

ID lagu diambil untuk mencari nilai fitur musik dari setiap lagu. Penggalan kode dapat dilihat sebagai berikut.

```
$id="";  
for ($x=0;$x<10000;$x++)  
{  
if($data[$x]==""&&$data[$x+1]=='i'&&$data[$x+2]=='  
d'&&$data[$x+3]==""&&$data[$x+4]=='.'&&$data[$x+5]=='  
'&&$data[$x+6]=="")  
{  
$id=$data[$x+7].$data[$x+8].$data[$x+9].$data[$x+10]  
. $data[$x+11].$data[$x+12].$data[$x+13].$data[$x+14].$da  
ta[$x+15].$data[$x+16].$data[$x+17].$data[$x+18].$data[$x  
+19].$data[$x+20].$data[$x+21].$data[$x+22].$data[$x+23]. $data[$x+24];  
}}  
}}
```

4.3.5 Pengambilan fitur-fitur musik pada lagu yang diunggah.

Penggalan kode dapat dilihat sebagai berikut.

```
$xmlfile=
"http://developer.echonest.com/api/v4/track/profile?api_
key=VEWHBITMWZMSBMO85&format=xml&id=".$id."&bucket=audio_
summary";
$data2 = simplexml_load_file($xmlfile);
$key = $data2->track->audio_summary->key;
$mode = $data2->track->audio_summary->mode;
$loudness = $data2->track->audio_summary->loudness;
$energy = $data2->track->audio_summary->energy;
$tempo = $data2->track->audio_summary->tempo;
```

Fitur lagu yang digunakan adalah *key*, *mode*, *loudness*, *energy*, dan tempo. Data sampel yang digunakan sebanyak 280 lagu dimana 30 lagu akan dipilih secara acak untuk dijadikan data uji pada sistem *Music Information Retrieval*.

4.3.6 Penggunaan Algoritma KNNCS Pada Proses Identifikasi Musik

Mencari nilai *Cosine Similarity* dari setiap data sampel yang telah diupload. Nilai *Cosine Similarity* dari setiap data sampel merepresentasikan tingkat kemiripan dari data sampel ke data uji. Nilai 1 merupakan nilai maksimum dari nilai *Cosine Similarity* dan nilai 0 adalah nilai terkecil.

A. Pencarian Nilai *Cosine Similarity*

Setiap data uji memiliki nilai *cosine similarity* sebanyak data sampel. Nilai *cosine similarity* inilah yang digunakan dalam mencari tingkat kemiripan antara data uji dengan data sampel. Penggalan kode dapat dilihat sebagai berikut

```
for($no=1;$no<=60;$no++){
$dotProduct = ($nkey[$no]*$nfkey + $nmode[$no]*$nfmode +
$nloudness[$no]*$nfloudness + $nenergy[$no]*$nfenergy +
$ntempo[$no]*$nftempo );
$absolute = sqrt($nkey[$no]*$nkey[$no] + $nmode[$no]*$nmode[$no]
+$nloudness[$no]*$nloudness[$no] + $nenergy[$no]*$nenergy[$no] +
$ntempo[$no]*$ntempo[$no]) * sqrt($nfkey*$nfkey + $nfmode*$nfmode +
$nfloudness*$nfloudness + $nfenergy*$nfenergy + $nftempo*$nftempo);
$coSim[$no] = $dotProduct / $absolute; }
```

B. Pengurutan nilai *Cosine Similarity*

Nilai *cosine similarity* yang didapatkan diurutkan dari nilai terbesar hingga terkecil. Penggalan kode dapat dilihat sebagai berikut :

```
for($x=1;$x<=60;$x++){
for($y=$x+1;$y<=60;$y++){
if($coSim[$x]<$coSim[$y]){
$tmpCos = $coSim[$x];
$tmpID = $songID[$x];
$coSim[$x] = $coSim[$y];
$songID[$x]= $songID[$y];
$coSim[$y] = $tmpCos;
$songID[$y]= $tmpID; }
```

Setelah nilai *cosine similarity* diurutkan, maka akan ditentukan data mana yang relevan terhadap data uji. Banyaknya data yang relevan dibatasi sebesar nilai K yang digunakan dalam pengujian. Dalam pengujian sistem ini, nilai K yang digunakan, yaitu K= 5, 7, 10. Nilai K tersebut juga untuk menentukan banyaknya rekomendasi lagu yang akan ditampilkan oleh sistem diurutkan dari data sampel yang memiliki nilai *Cosine Similarity* terbesar hingga terkecil. Rekomendasi yang akan ditampilkan juga dipengaruhi besarnya nilai *threshold*. Nilai *threshold* yang digunakan di sistem ini sendiri menggunakan nilai 0.999, 0.998, 0.997 . Nilai *threshold* ini juga akan digunakan saat pengujian sistem nantinya.