

## BAB V HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan hasil dari pengembangan aplikasi sistem pendukung keputusan untuk evaluasi nilai saham menggunakan *Analytical Hierarchy Process* (AHP).

### 5.1 Hasil

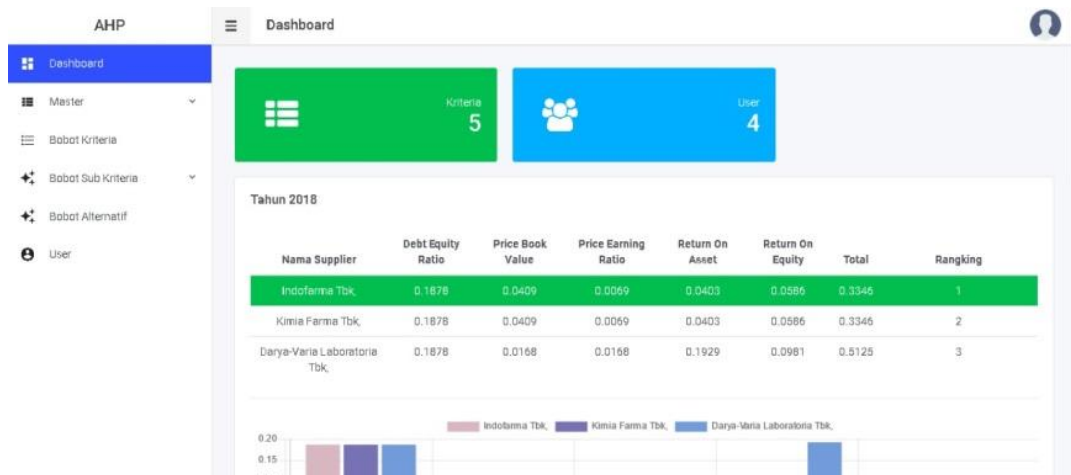
*Website* AHP untuk evaluasi nilai saham telah berhasil dikembangkan, dengan menggunakan data sekunder yang didapatkan dari Bursa Efek Indonesia (BEI). Sistem dapat mengolah data dan mendapatkan peringkat nilai saham berdasarkan perhitungan bobot prioritas dan nilai prioritas terhadap kriteria. Selain itu, *website* ini memiliki beberapa fitur untuk memudahkan penggunaanya dalam mengelola data kriteria, subkriteria, bobot kriteria, bobot subkriteria dan bobot alternatif sesuai kebutuhan.

#### 5.1.1 Hasil Perancangan Tampilan

Antarmuka *website* aplikasi AHP dirancang dengan memperhatikan prinsip-prinsip kegunaan dan pengalaman pengguna. Tujuan utama adalah memastikan bahwa setiap elemen antarmuka menciptakan lingkungan yang ramah pengguna (*user-friendly*). Tampilan yang intuitif dan mudah digunakan akan mempermudah pengguna dalam mengoperasikan aplikasi, sehingga proses evaluasi kinerja dapat dilakukan dengan lebih efektif.

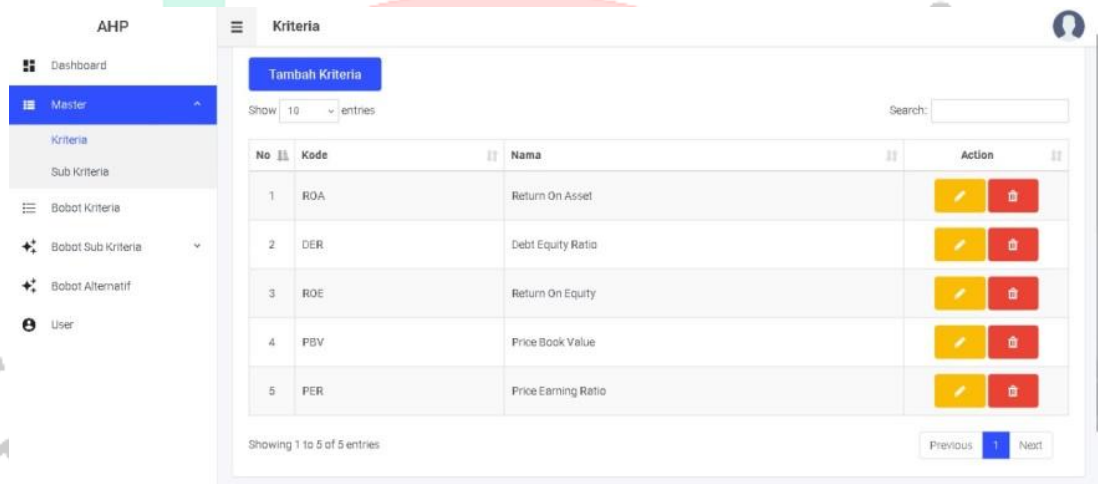
Hasil dari tampilan antarmuka pada *website* aplikasi AHP dibedakan sesuai jenis peran pengguna masing-masing, di antaranya sebagai berikut.

##### 5.1.1.1 Hasil Perancangan Tampilan Peran Admin



Gambar 5. 1 Tampilan Menu Dashboard

Gambar 5.1 tampilan halaman dashboard peran admin menampilkan informasi terdapat jumlah user, jumlah kriteria, jumlah subkriteria, data perusahaan teratas dalam bentuk diagram.



Gambar 5. 2 Tampilan Menu Kriteria

Gambar 5.2 tampilan menu kriteria peran admin menampilkan daftar kriteria dalam bentuk tabel dan memiliki fungsi untuk menambahkan data kriteria baru, melihat detail, mengubah, dan menghapus data kriteria. Selain itu, terdapat fungsi pencarian untuk mencari data kriteria yang tersimpan di dalam sistem.

No	Kriteria	Kode	Nama	Action
1	ROA - Return On Asset	> 6%	Diatas 6%	[Edit] [Delete]
2	ROA - Return On Asset	4% - 6%	diantara 4 hingga 6 persen	[Edit] [Delete]
3	ROA - Return On Asset	< 4%	Dibawah 4 persen	[Edit] [Delete]
4	DER - Debt Equity Ratio	< 80%	Dibawah 80%	[Edit] [Delete]
5	DER - Debt Equity Ratio	80% - 100%	Diantara 80 hingga 100 persen	[Edit] [Delete]
6	DER - Debt Equity Ratio	> 100%	Diatas 100%	[Edit] [Delete]
7	ROE - Return On Equity	< 1%	Dibawah 1% persen	[Edit] [Delete]

Gambar 5. 3 Tampilan Menu SubKriteria

Gambar 5.3 tampilan menu subkriteria peran admin menampilkan daftar subkriteria dalam bentuk tabel dan memiliki fungsi untuk menambahkan data subkriteria baru, melihat detail, mengubah, dan menghapus data subkriteria. Selain itu, terdapat fungsi pencarian untuk mencari data subkriteria yang tersimpan di dalam sistem.

**Matriks Perbandingan Kriteria**

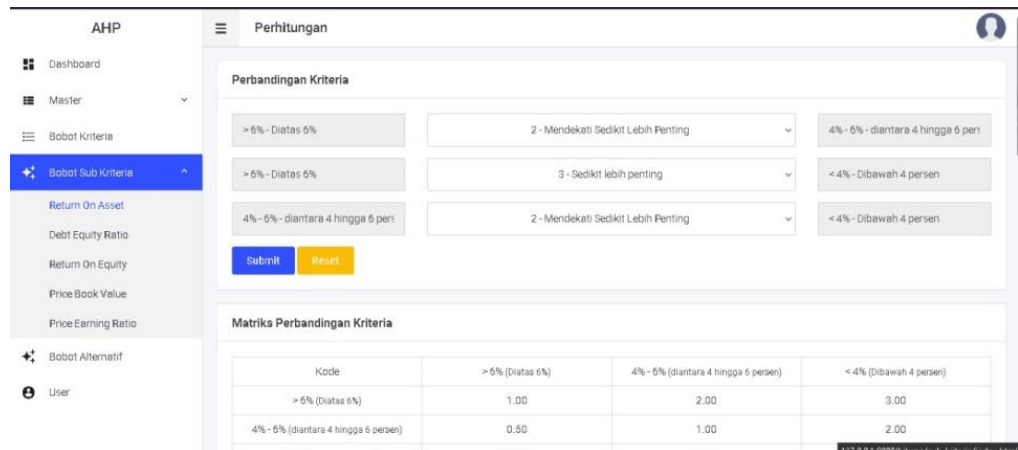
Kode	ROA (Return On Asset)	DER (Debt Equity Ratio)	ROE (Return On Equity)	PBV (Price Book Value)	PER (Price Earning Ratio)
ROA (Return On Asset)	1.00	1.00	3.00	5.00	5.00
DER (Debt Equity Ratio)	1.00	1.00	3.00	5.00	5.00
ROE (Return On Equity)	0.33	0.33	1.00	3.00	3.00
PBV (Price Book Value)	0.20	0.20	0.33	1.00	1.00
PER (Price Earning Ratio)	0.20	0.20	0.33	1.00	1.00
Total	2.73	2.73	7.67	15.00	15.00

**Matriks Nilai Kriteria**

Kode	ROA (Return On Asset)	DER (Debt Equity Ratio)	ROE (Return On Equity)	PBV (Price Book Value)	PER (Price Earning Ratio)	Jumlah	Prioritas	Eigen Value
ROA (Return On Asset)	0.37	0.37	0.20	0.20	0.33	1.70	0.16	0.08

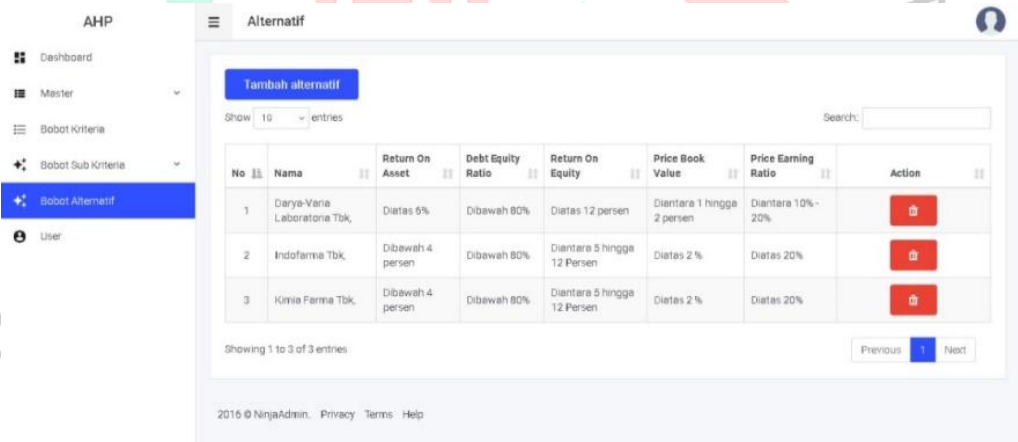
Gambar 5. 4 Tampilan Menu Bobot Kriteria

Gambar 5.4 tampilan menu bobot kriteria peran admin menampilkan matriks perbandingan kriteria dalam bentuk tabel dan dibawahnya terdapat matriks nilai kriteria dalam bentuk tabel.



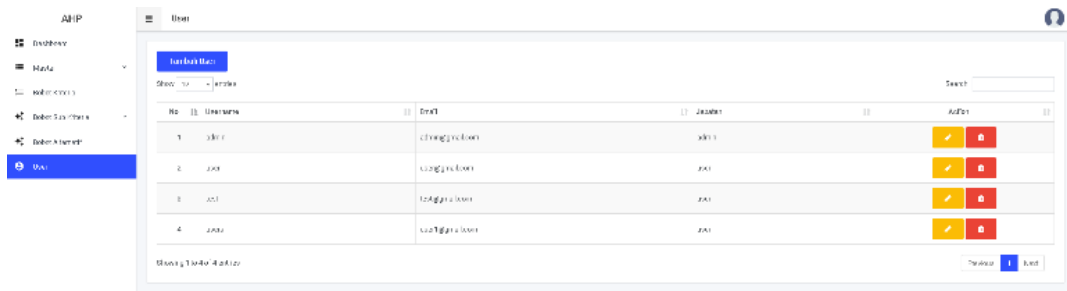
Gambar 5. 5 Tampilan Menu Bobot Subkriteria

Gambar 5.5 tampilan menu bobot subkriteria menampilkan kolom perbandingan kriteria dimana admin dapat menginput nilai perbandingan yang di inginkan di bawahnya juga menampilkan matriks perbandingan kriteria dalam bentuk tabel



Gambar 5. 6 Tampilan Halaman Bobot Alternatif

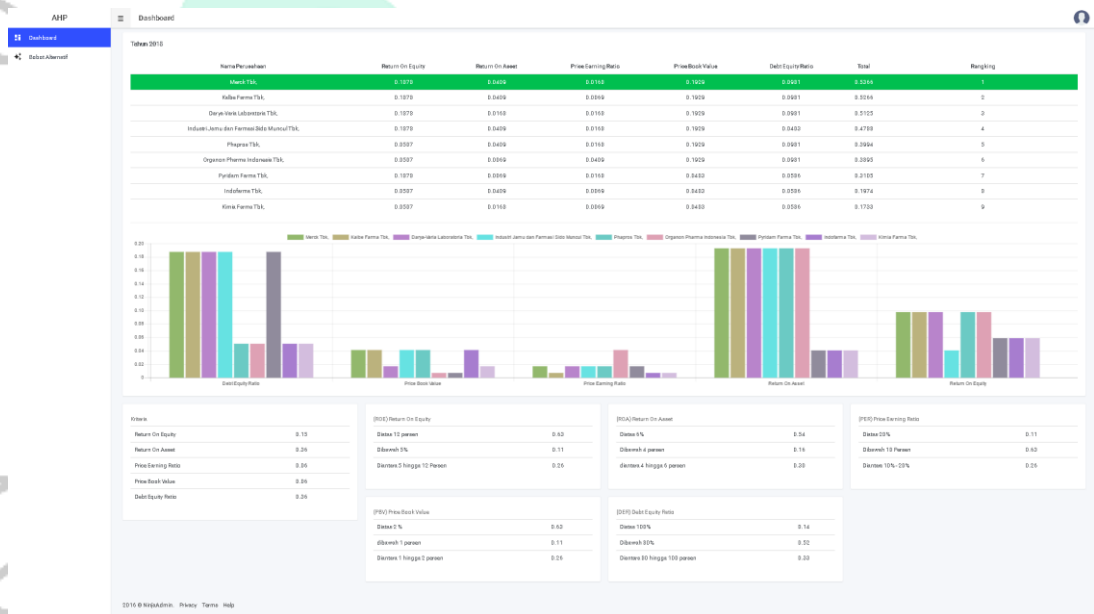
Gambar 5.6 tampilan halaman bobot alternatif menampilkan bobot alternatif yang tersimpan di dalam database dalam bentuk tabel, admin dapat menambah dan menghapus alternatif yang terdapat di dalam database.



Gambar 5. 7 Tampilan Menu User

Gambar 5.7 tampilan menu *user* menampilkan data *user* yang tersimpan ke dalam database, dimana *admin* dapat melihat detail dari user yang diinginkan.

### 5.1.1.2 Hasil Perancangan Tampilan Peran User



Gambar 5. 8 Tampilan Menu Dashboard

Gambar 5.8 tampilan menu dashboard peran user menampilkan informasi peringkat nilai saham tertinggi per tahun dalam bentuk bagan dan tabel. Selain itu, terdapat peringkat nilai saham berdasarkan tahun dalam bentuk bagan yang menampilkan 3 peringkat teratas dan dalam bentuk tabel yang menampilkan seluruh daftar peringkat nilai saham.

### 5.1.2 Hasil Pengujian

Setelah pengembangan *website* aplikasi AHP selesai, diperlukan pengujian ulang untuk memastikan bahwa fungsionalitas dan logika internal

aplikasi berfungsi dengan baik dan sesuai dengan kebutuhan serta spesifikasi yang telah ditentukan.

### 5.1.1.3 Pengujian Kotak Putih

Selain melakukan pengujian kotak hitam, pengujian kotak Putih juga merupakan langkah penting untuk menjamin bahwa aplikasi berfungsi sesuai dengan persyaratan yang telah ditetapkan. Proses pengujian melibatkan evaluasi menyeluruh terhadap aplikasi dengan memusatkan perhatian pada skenario yang telah direncanakan sebelumnya.

Tabel 5. 1 Hasil Pengujian Kotak Hitam (Black Box Testing) Admin

No	Algoritma	Perancangan Kode
1	Perbandingan Kriteria	<pre> // Mencari Nilai Bobot \$count = count(\$data['kriteria']); \$nilai_ri = \$data_ri[\$count];  \$totalJumlah = 0; \$totalPrioritas = 0; \$totalEigen = 0;  \$totalNilai = []; \$array_totalNilai = []; foreach (\$data['kriteria'] as \$kriteria1) {     \$totalNilai[\$kriteria1['kode']] = 0;     foreach (\$data['kriteria'] as \$kriteria2) {         foreach (\$data['bobot'] as \$b) {             if (                 \$kriteria2['kode'] . \$kriteria1['kode'] ==                 \$b['kriteria_id'] . \$b['kriteria_id2']                    \$kriteria1['kode'] . \$kriteria2['kode'] ==                 \$b['kriteria_id2'] . \$b['kriteria_id']             ) {                 \$totalNilai[\$kriteria1['kode']] += \$b['nilai'];             }         }     } }  \$kk = [     'kode' =&gt; \$kriteria1['kode'],     'nilai' =&gt; \$totalNilai[\$kriteria1['kode']] ];  array_push(\$array_totalNilai,\$kk);  //Hasil total nilai Kriteria::where('kode',\$kriteria1['kode']-&gt;update([     'total_nilai' =&gt; \$totalNilai[\$kriteria1['kode']] ])); }  // mencari nilai total matriks nilai kriteria tanpa jumlah, prioritas, dan eigen \$totalNilais = []; </pre>

		<pre> \$rray_totalNilais = []; foreach (\$data['kriteria'] as \$kriteria1) {     \$totalNilais[\$kriteria1['kode']] = 0;     foreach (\$data['kriteria'] as \$kriteria2) {         foreach (\$data['bobot'] as \$b) {             if (                 \$kriteria2['kode'] . \$kriteria1['kode'] ==                 \$b['kriteria_id'] . \$b['kriteria_id2']                    \$kriteria1['kode'] . \$kriteria2['kode'] ==                 \$b['kriteria_id2'] . \$b['kriteria_id']             ) {                 \$o = \$b['nilai'] / \$totalNilai[\$kriteria1['kode']];                 \$totalNilais[\$kriteria1['kode']] += \$o;             }         }     }     \$kk = [         'kode' =&gt; \$kriteria1['kode'],         'nilai' =&gt; \$totalNilais[\$kriteria1['kode']]     ];     array_push(\$rray_totalNilais,\$kk); }  // mencari nilai total jumlah, prioritas, dan eigen \$totalNilaiJumlah = []; \$rray_totalNilaiJumlah = []; foreach (\$data['kriteria'] as \$kriteria1) {     \$totalNilaiJumlah[\$kriteria1['kode']] = 0;     foreach (\$data['bobot'] as \$b) {         if (\$kriteria1['kode'] == \$b['kriteria_id']) {             \$o = \$b['nilai'] / \$totalNilai[\$b['kriteria_id2']];             \$totalNilaiJumlah[\$kriteria1['kode']] += \$o;         }     }     \$kk = [         'kode' =&gt; \$kriteria1['kode'],         'nilai' =&gt; \$totalNilaiJumlah[\$kriteria1['kode']]     ];     array_push(\$rray_totalNilaiJumlah,\$kk);      \$kriteria::where('kode',\$kriteria1['kode'])-&gt;update([         'jumlah' =&gt; \$totalNilaiJumlah[\$kriteria1['kode']],         'prioritas' =&gt; \$totalNilaiJumlah[\$kriteria1['kode']] / count(\$data['kriteria']),         'eigen_value' =&gt; (\$totalNilaiJumlah[\$kriteria1['kode']] / count(\$data['kriteria'])) * \$totalNilai[\$kriteria1['kode']]     ]); }  foreach (\$data['kriteria'] as \$kriteria1){     \$totalJumlah += \$totalNilaiJumlah[\$kriteria1['kode']];     \$totalPrioritas += \$totalNilaiJumlah[\$kriteria1['kode']] / count(\$data['kriteria']);     \$totalEigen += (\$totalNilaiJumlah[\$kriteria1['kode']] / count(\$data['kriteria'])) *     \$totalNilai[\$kriteria1['kode']]; } DB::table('hasil_bobot_kriteria')-&gt;truncate(); DB::table('hasil_bobot_kriteria')-&gt;insert([     'ci' =&gt; (\$totalEigen - count(\$data['kriteria'])) / (count(\$data['kriteria']) - 1),     'ri' =&gt; \$nilai_ri,     'cr' =&gt; (\$totalEigen - count(\$data['kriteria'])) / (count(\$data['kriteria']) - 1) / \$nilai_ri, ]); </pre>
--	--	--

```

// return [
//   'ci' => ($totalEigen - count($data['kriteria'])) /
(count($data['kriteria']) - 1),
//   'ri' => $nilai_ri,
//   'cr' => ($totalEigen - count($data['kriteria'])) /
(count($data['kriteria']) - 1) / $nilai_ri,
// ];

return redirect()->back()->with('success','Berhasil tambah data!');
} catch (\Exception $e) {
return redirect()->back()->with('error','Terjadi kesalahan saat membuat
data: '. $e->getMessage());
}
}

public function reset(){
try {
KriteriaBobot::truncate();
return redirect()->back()->with('success','Berhasil reset data!');
} catch (\Exception $e) {
return redirect()->back()->with('error','Terjadi kesalahan saat reset
data: '. $e->getMessage());
}
}

public function indexSub($id){
$krit = SubKriteria::where('kriteria_id',$id)->get();

$bobot = SubKriteriaBobot::where('kriteria_ids',$id)->get();
$bobot->map(function($x){
$x['rel_1'] = SubKriteria::where(['kode' => $x['kriteria_id']])->first();

$x['rel_2'] = SubKriteria::where(['kode' => $x['kriteria_id2']])->first();
$x->kodes = $x['kriteria_id'].'_'.$x['kriteria_id2'].'_'.$x['nilai'];
return $x;
});

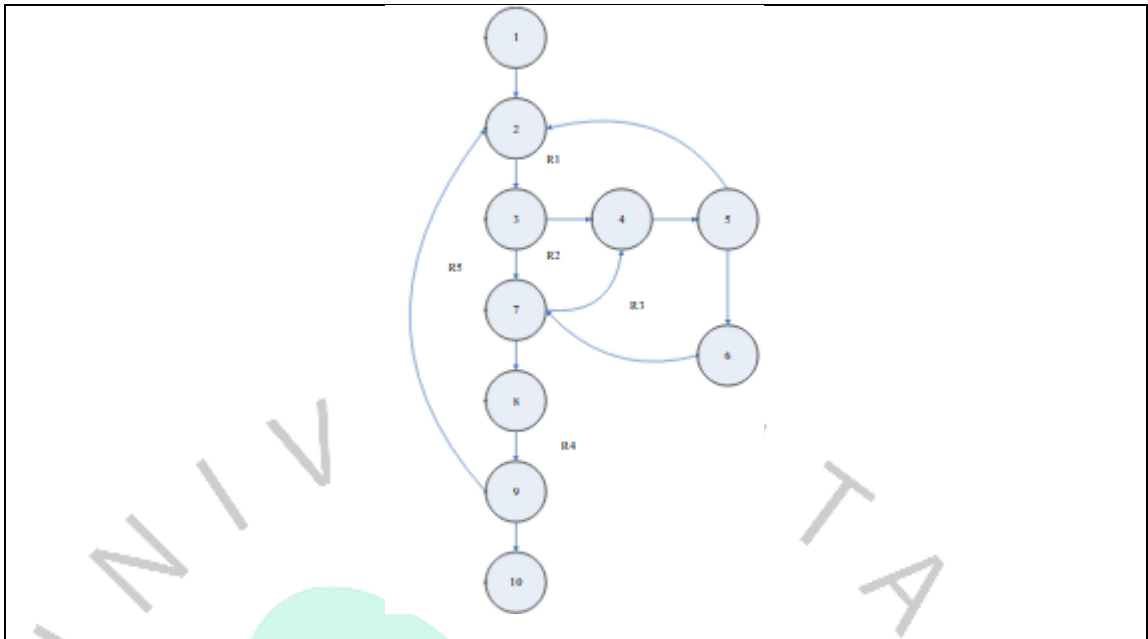
$data = [
'title' => "Perhitungan",
'kriteria' => $krit,
'bobot' => $bobot
];

return view('hitung.index',compact('data'));
}

```

Flowgraph





Hasil

AHP

Perhitungan

Matriks Perbandingan Kriteria

Kode	ROA (Netun On Asset)	DER (Net Equity Ratio)	ROE (Netun On Equity)	PBV (Net Book Value)	PER (Net Equity Ratio)
ROA (Netun On Asset)	1.00	1.00	3.00	5.00	5.00
DER (Net Equity Ratio)	1.00	1.00	3.00	5.00	5.00
ROE (Netun On Equity)	0.33	0.33	1.00	3.00	3.00
PBV (Net Book Value)	0.20	0.20	0.33	1.00	1.00
PER (Net Equity Ratio)	0.20	0.20	0.33	1.00	1.00
Total	2.73	2.73	7.67	15.00	15.00

Matriks Nilai Kriteria

Kode	ROA (Netun On Asset)	DER (Net Equity Ratio)	ROE (Netun On Equity)	PBV (Net Book Value)	PER (Net Equity Ratio)	Jumlah	Prioritas	Eigen Value
ROA (Netun On Asset)	0.37	0.37	0.39	0.23	0.33	1.79	0.26	0.98
DER (Net Equity Ratio)	0.37	0.37	0.39	0.23	0.33	1.79	0.26	0.98
ROE (Netun On Equity)	0.12	0.12	0.13	0.20	0.20	0.77	0.15	1.19
PBV (Net Book Value)	0.07	0.07	0.04	0.07	0.07	0.32	0.06	0.97
PER (Net Equity Ratio)	0.07	0.07	0.04	0.07	0.07	0.32	0.06	0.97
Total	1.00	1.00	1.00	1.00	1.00	5.00	1.00	5.08

Matriks Random Consistency Index

Matrix Size	1	2	3	4	5	6	7	8	9	10
Random Consistency Index (RI)	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Matriks Nilai Kriteria

CI (Consistency Index)	0.02
RI (Ratio Index)	1.12
CR (Consistency Ratio)	0.02

Konsisten

Pembahasan

Sistem berhasil melakukan perhitungan antar kriteria. Sistem melakukan kalkulasi perbandingan berpasangan antar kriteria total kolom, normalisasi matriks, bobot prioritas dan *consistency measure*, dan *consistency ratio*.

2	Perbandingan Sub Kriteria	<pre> try {   \$oke = [];   foreach(\$request['kriteria'] as \$key =&gt; \$value){      \$cc = [       'rel_1' =&gt; \$request['rel_1'][\$key],       'rel_2' =&gt; \$request['rel_2'][\$key],       'nilai' =&gt; \$value     ];   } </pre>
---	---------------------------	--

		<pre> array_push(\$oke, \$cc); }  foreach (\$oke as \$item) {      // Cek jika pasangan rel_2 dan rel_1 belum ada     \$isReverseExist = false;     foreach (\$oke as \$existing) {         if (\$existing['rel_1'] === \$item['rel_2'] &amp;&amp; \$existing['rel_2'] === \$item['rel_1']) {             \$isReverseExist = true;             break;         }     }      // Jika pasangan rel_2 dan rel_1 belum ada, tambahkan ke array baru     if (!\$isReverseExist) {         \$reversePair = [             'rel_1' =&gt; \$item['rel_2'],             'rel_2' =&gt; \$item['rel_1'],             'nilai' =&gt; 1 / \$item['nilai']         ];         array_push(\$oke, \$reversePair);     } }  // usort(\$oke, function(\$a, \$b) { //     return strcmp(\$a['rel_1'], \$b['rel_1']); // });  SubKriteriaBobot::where('kriteria_ids',\$request-&gt;ids)-&gt;delete();  foreach(\$oke as \$value){     SubKriteriaBobot::create([         'kriteria_ids' =&gt; \$request-&gt;ids,         'kriteria_id' =&gt; \$value['rel_1'],         'kriteria_id2' =&gt; \$value['rel_2'],         'nilai' =&gt; \$value['nilai'],     ]); }  \$krit = SubKriteria::where('kriteria_id',\$request-&gt;ids)-&gt; orderby('kode',"ASC")-&gt;get();  \$bobot = SubKriteriaBobot::where('kriteria_ids',\$request-&gt;ids)-&gt;get(); \$bobot-&gt;map(function(\$x){     \$x['rel_1'] = SubKriteria::where(['kode' =&gt; \$x['kriteria_id']])-&gt; first();      \$x['rel_2'] = SubKriteria::where(['kode' =&gt; \$x['kriteria_id2']])-&gt; first();     \$x-&gt;kodes = \$x['kriteria_id'].'_'.\$x['kriteria_id2'].'_'.\$x['nilai'];     return \$x; });  \$data = [     'title' =&gt; "Perhitungan",     'kriteria' =&gt; \$krit,     'bobot' =&gt; \$bobot ];  \$data_ri = [     '1' =&gt; 0.0,     '2' =&gt; 0.0,     '3' =&gt; 0.58,     '4' =&gt; 0.9, </pre>
--	--	--

		<pre> '5' =&gt; 1.12, '6' =&gt; 1.24, '7' =&gt; 1.32, '8' =&gt; 1.41, '9' =&gt; 1.45, '10' =&gt; 1.49, ];  \$count = count(\$data['kriteria']); \$nilai_ri = \$data_ri[\$count];  \$totalJumlah = 0; \$totalPrioritas = 0; \$totalEigen = 0;  \$totalNilai = []; \$array_totalNilai = []; foreach (\$data['kriteria'] as \$kriteria1) {     \$totalNilai[\$kriteria1['kode']] = 0;     foreach (\$data['kriteria'] as \$kriteria2) {         foreach (\$data['bobot'] as \$b) {             if (                 \$kriteria2['kode'] . \$kriteria1['kode'] ==                 \$b['kriteria_id'] . \$b['kriteria_id2']                    \$kriteria1['kode'] . \$kriteria2['kode'] ==                 \$b['kriteria_id2'] . \$b['kriteria_id']             ) {                 \$totalNilai[\$kriteria1['kode']] += \$b['nilai'];             }         }     } }  \$skk = [     'kode' =&gt; \$kriteria1['kode'],     'nilai' =&gt; \$totalNilai[\$kriteria1['kode']] ]; array_push(\$array_totalNilai,\$skk); SubKriteria::where('kode',\$kriteria1['kode'])-&gt;update([     'total_nilai' =&gt; \$totalNilai[\$kriteria1['kode']] ]); }  \$totalNilais = []; \$array_totalNilais = []; foreach (\$data['kriteria'] as \$kriteria1) {     \$totalNilais[\$kriteria1['kode']] = 0;     foreach (\$data['kriteria'] as \$kriteria2) {         foreach (\$data['bobot'] as \$b) {             if (                 \$kriteria2['kode'] . \$kriteria1['kode'] ==                 \$b['kriteria_id'] . \$b['kriteria_id2']                    \$kriteria1['kode'] . \$kriteria2['kode'] ==                 \$b['kriteria_id2'] . \$b['kriteria_id']             ) {                 \$o = \$b['nilai'] / \$totalNilai[\$kriteria1['kode']];                 \$totalNilais[\$kriteria1['kode']] += \$o;             }         }     } }  \$skk = [     'kode' =&gt; \$kriteria1['kode'],     'nilai' =&gt; \$totalNilais[\$kriteria1['kode']] ]; array_push(\$array_totalNilais,\$skk); } </pre>
--	--	--

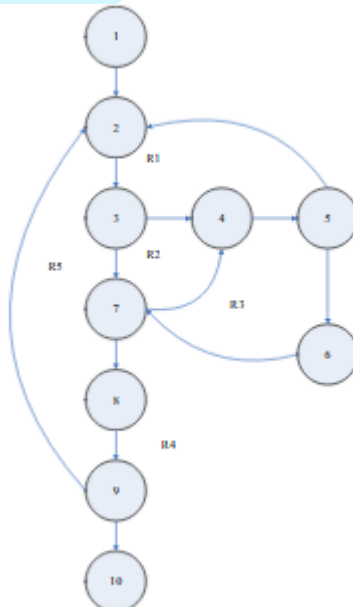
```

$totalNilaiJumlah = [];
$array_totalNilaiJumlah = [];
foreach ($data['kriteria'] as $kriteria1) {
    $totalNilaiJumlah[$kriteria1['kode']] = 0;
    foreach ($data['bobot'] as $b) {
        if ($kriteria1['kode'] == $b['kriteria_id']) {
            $so = $b['nilai'] / $totalNilai[$b['kriteria_id2']];
            $totalNilaiJumlah[$kriteria1['kode']] += $so;
        }
    }
    $skk = [
        'kode' => $kriteria1['kode'],
        'nilai' => $totalNilaiJumlah[$kriteria1['kode']]
    ];
    array_push($array_totalNilaiJumlah,$skk);
    SubKriteria::where('kode',$kriteria1['kode'])->update([
        'jumlah' => $totalNilaiJumlah[$kriteria1['kode']],
        'prioritas' => $totalNilaiJumlah[$kriteria1['kode']] /
        count($data['kriteria']),
        'eigen_value' => ($totalNilaiJumlah[$kriteria1['kode']] /
        count($data['kriteria'])) * $totalNilai[$kriteria1['kode']]
    ]);
}

foreach ($data['kriteria'] as $kriteria1){
    $totalJumlah += $totalNilaiJumlah[$kriteria1['kode']];
    $totalPrioritas += $totalNilaiJumlah[$kriteria1['kode']] /
    count($data['kriteria']);
    $totalEigen += ($totalNilaiJumlah[$kriteria1['kode']] /
    count($data['kriteria'])) *
    $totalNilai[$kriteria1['kode']];
}
DB::table('hasil_bobot_sub_kriteria')->truncate();
DB::table('hasil_bobot_sub_kriteria')->insert([
    'sub_kriteria_id' => $request->ids,
    'ci' => ($totalEigen - count($data['kriteria'])) /
    (count($data['kriteria']) - 1),
    'ri' => $nilai_ri,
    'cr' => ($totalEigen - count($data['kriteria'])) /
    (count($data['kriteria']) - 1) / $nilai_ri,

```

Flowgraph



## Hasil

AHP

Perhitungan

Kode	> 5% (Dinas 1)	4% - 5% (Dinas 4 hingga 6 persen)	< 4% (Dibawah 4 persen)
> 5% (Dinas 1)	1.00	2.00	3.00
4% - 5% (Dinas 4 hingga 6 persen)	0.50	1.00	2.00
< 4% (Dibawah 4 persen)	0.33	0.50	1.00
Total	1.83	3.50	6.00

Matriks Nilai Kriteria

Kode	> 5% (Dinas 1)	4% - 5% (Dinas 4 hingga 6 persen)	< 4% (Dibawah 4 persen)	Jumlah	Prioritas	Eigen Value
> 5% (Dinas 1)	0.55	0.57	0.50	1.62	0.54	0.99
4% - 5% (Dinas 4 hingga 6 persen)	0.27	0.29	0.33	0.89	0.30	1.04
< 4% (Dibawah 4 persen)	0.18	0.14	0.17	0.49	0.16	0.99
Total	1.00	1.00	1.00	3.00	1.00	3.01

Matriks Random Consistency Index

Matrix Size	1	2	3	4	5	6	7	8	9	10
Random Consistency Index (RI)	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Matriks Nilai Kriteria

CI (Consistency Index)	0.01
RI (Ratio Index)	0.58
CR (Consistency Ratio)	0.01

Konsisten

### Pembahasan

Sistem berhasil melakukan perhitungan antar subkriteria. Sistem melakukan kalkulasi perbandingan berpasangan antar kriteria total kolom, normalisasi matriks, bobot prioritas dan *consistency measure*, dan *consistency ratio*.

2

Perankingan

```
class DashboardController extends Controller
{
    public function index(){
        $user = User::count();
        $years = DB::table('alternatif')
        ->select('tahun')
        ->groupBy('tahun')
        ->pluck('tahun');
        // return $years;

        $loops = [];
        foreach($years as $value){
            $tahun = $value;
            $datas = Alternatif::where('tahun',$value)->get();
            $datas->map(function($x){
                $explode = explode('_', $x->sub_kriteria_ids);
                $x['sub_krits'] = SubKriteria::whereIn('id',$explode)->get();

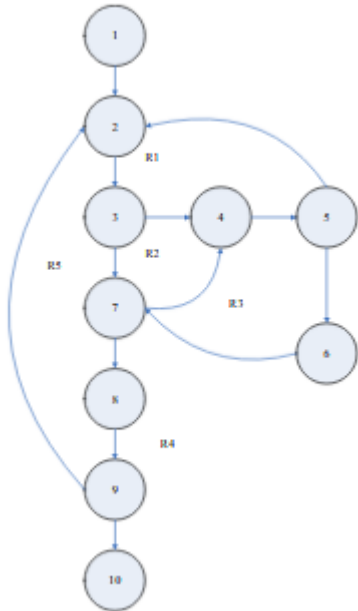
                $x['sub_krits']->map(function($y) use ($x){
                    $y['kritis'] = Kriteria::where('id',$y->kriteria_id)->first();
                    $x['total_rangking'] += ($y['kritis']['prioritas'] * $y['prioritas']);
                    return $y;
                });
                $x['explode'] = $explode;
                return $x;
            });
            $datas = $datas->sortByDESC('total_rangking')->values();
            // Assign rankings based on sorted order
            $datas->each(function($x, $index) {
                $x->number_rangking = $index + 1;
            });
        }
    }
}
```

```

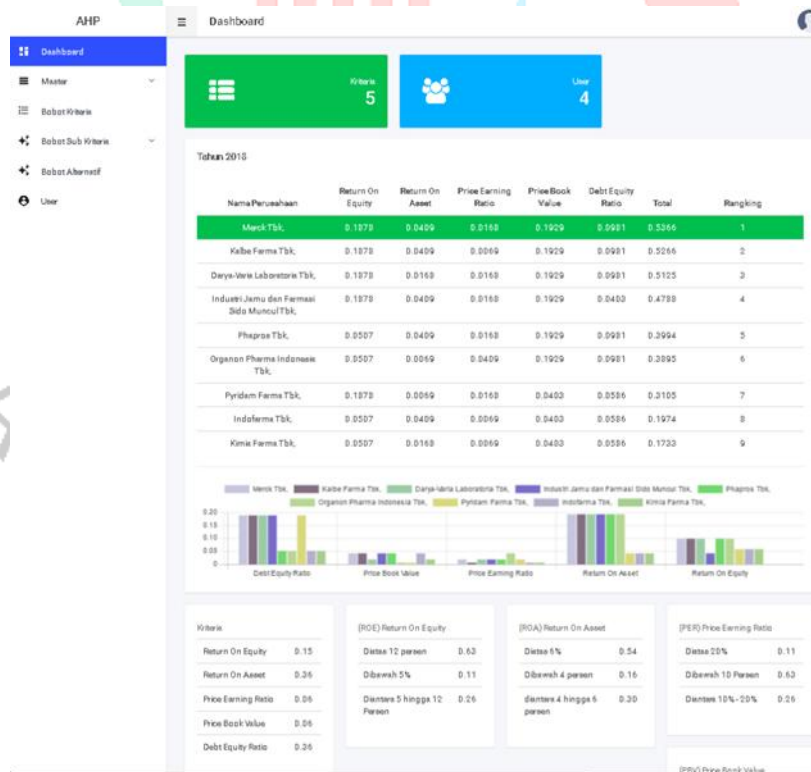
$x = [
'tahun' => $value,
'data' => $datas
];
array_push($loops, $x);

```

Flowgraph



Hasil



Pembahasan

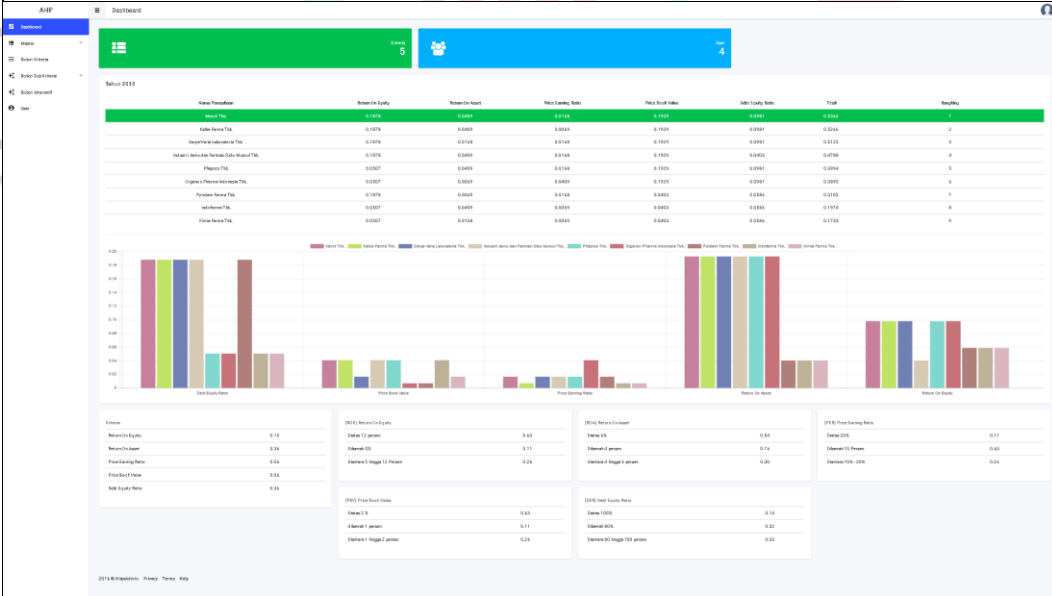
Sistem berhasil melakukan perhitungan perankingan. Sistem melakukan kalkulasi bobot prioritas kriteria, rata-rata nilai, dan ranking Perusahaan.

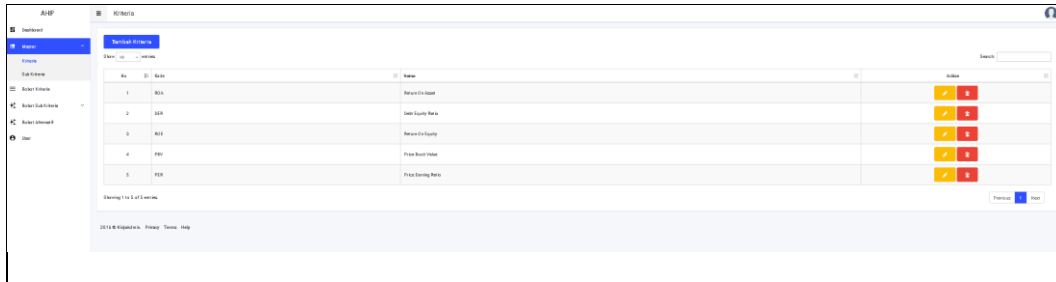
### 5.1.1.4 Pengujian Kotak Putih

Pengujian kotak putih memastikan bahwa implementasi kode dan logika dalam suatu aplikasi berjalan sesuai dengan perancangan yang dibuat.

Selain melakukan pengujian kotak putih, pengujian kotak hitam juga merupakan langkah penting untuk menjamin bahwa aplikasi berfungsi sesuai dengan persyaratan yang telah ditetapkan. Proses pengujian melibatkan evaluasi menyeluruh terhadap aplikasi dengan memusatkan perhatian pada skenario yang telah direncanakan sebelumnya.

Tabel 5. 2 Hasil Pengujian Kotak Putih (Black Box Testing) Admin

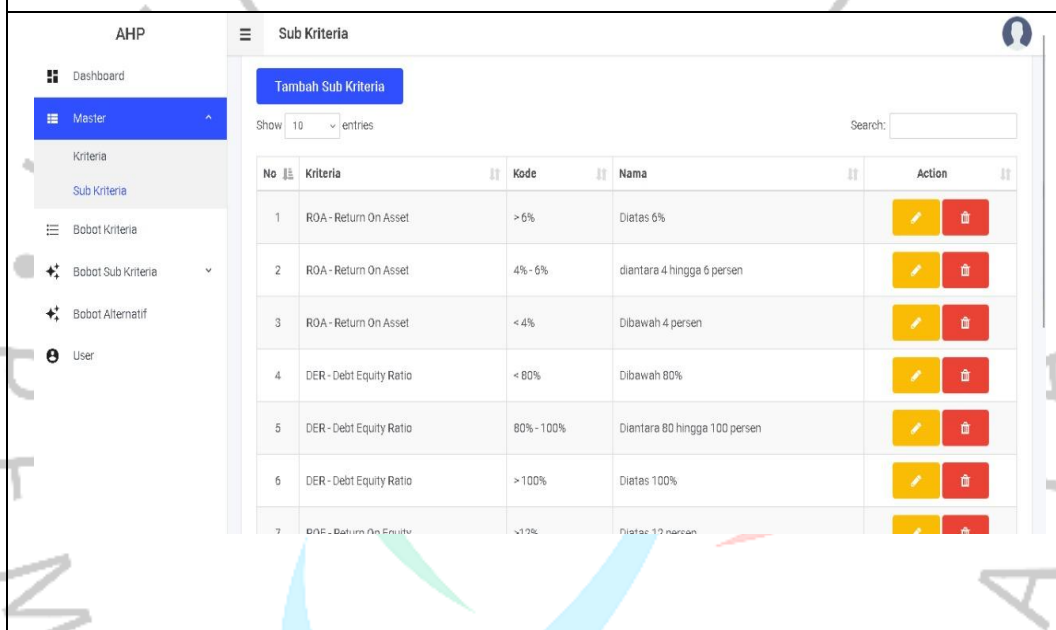
No.	Skenario Pengujian	Hasil yang Diharapkan
1	Admin mengakses <i>website</i> aplikasi AHP	Sistem menampilkan menu halaman dashboard.
<p>Hasil</p>  <p>The screenshot shows the AHP Admin Dashboard. At the top, there's a navigation menu with options like 'Home', 'Menu', 'Admin', 'Kategori', 'Kategori Baru', 'Kategori Hapus', and 'User'. The main content area features a table with columns for 'Nama Transaksi', 'Rekanan Ekuitas', 'Rekanan Kas', 'Piutang Usaha', 'Piutang Usaha', 'Saldo Utang Usaha', 'Total', and 'Rekapitulasi'. Below the table is a bar chart comparing various financial metrics. At the bottom, there are several summary tables for different categories like 'Rekanan Ekuitas', 'Rekanan Kas', 'Piutang Usaha', 'Saldo Utang Usaha', 'Rekanan Ekuitas', 'Rekanan Kas', 'Piutang Usaha', and 'Saldo Utang Usaha'.</p>		
Kesimpulan: menu dashboard berhasil diakses.		
2	Admin mengakses menu kriteria	Menampilkan kriteria yang tersedia beserta fitur Tambah, ubah, dan hapus data kriteria.
Hasil		



Kesimpulan: sistem berhasil menampilkan menu kriteria

3	Admin mengakses menu subkriteria	Menampilkan subkriteria yang tersedia beserta fitur tambah, melihat detail, ubah, dan hapus data subkriteria.
---	----------------------------------	---

Hasil



Kesimpulan: sistem berhasil menampilkan menu subkriteria.

4	Admin mengakses menu bobot kriteria.	Menampilkan matriks nilai kriteria dalam bentuk tabel.
---	--------------------------------------	--

Hasil



Kode	ROA (Return On Asset)	DER (Debt Equity Ratio)	ROE (Return On Equity)	PBV (Price Book Value)	PER (Price Earning Ratio)	Jumlah	Prioritas	Eigen Value
ROA (Return On Asset)	0.37	0.37	0.39	0.33	0.33	1.79	0.36	0.98
DER (Debt Equity Ratio)	0.37	0.37	0.39	0.33	0.33	1.79	0.36	0.98
ROE (Return On Equity)	0.12	0.12	0.13	0.20	0.20	0.77	0.15	1.19
PBV (Price Book Value)	0.07	0.07	0.04	0.07	0.07	0.32	0.06	0.97
PER (Price Earning Ratio)	0.07	0.07	0.04	0.07	0.07	0.32	0.06	0.97
Total	1.00	1.00	1.00	1.00	1.00	5.00	1.00	5.08

Kesimpulan: sistem berhasil menampilkan menu bobot kriteria.

5	Admin mengakses menu bobot subkriteria.	Menampilkan kolom input nilai perbandingan kriteria dan hasil perhitungan perbandingannya.
---	---	--

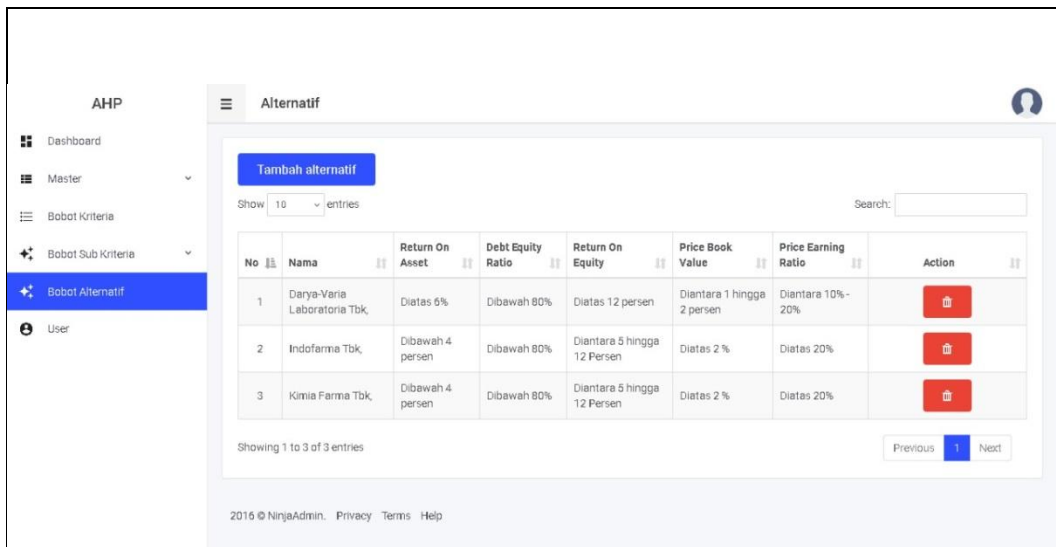
Hasil

Kode	> 6% (Diatas 6%)	4% - 6% (diantara 4 hingga 6 persen)	< 4% (Dibawah 4 persen)
> 6% (Diatas 6%)	1.00	2.00	3.00
4% - 6% (diantara 4 hingga 6 persen)	0.50	1.00	2.00

Kesimpulan: sistem berhasil menampilkan menu bobot subkriteria.

6	Admin mengakses menu bobot alternatif.	Menampilkan daftar alternatif dalam bentuk tabel, beserta fitur untuk tambah dan hapus alternatif.
---	--	--

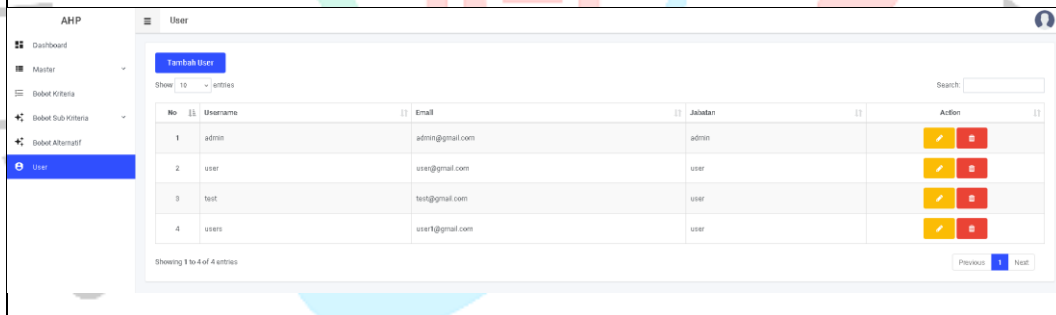
Hasil



Kesimpulan: sistem berhasil menampilkan menu bobot alternatif

7	Admin mengakses menu user	Menampilkan daftar user beserta fitur untuk melihat detail user
---	---------------------------	---

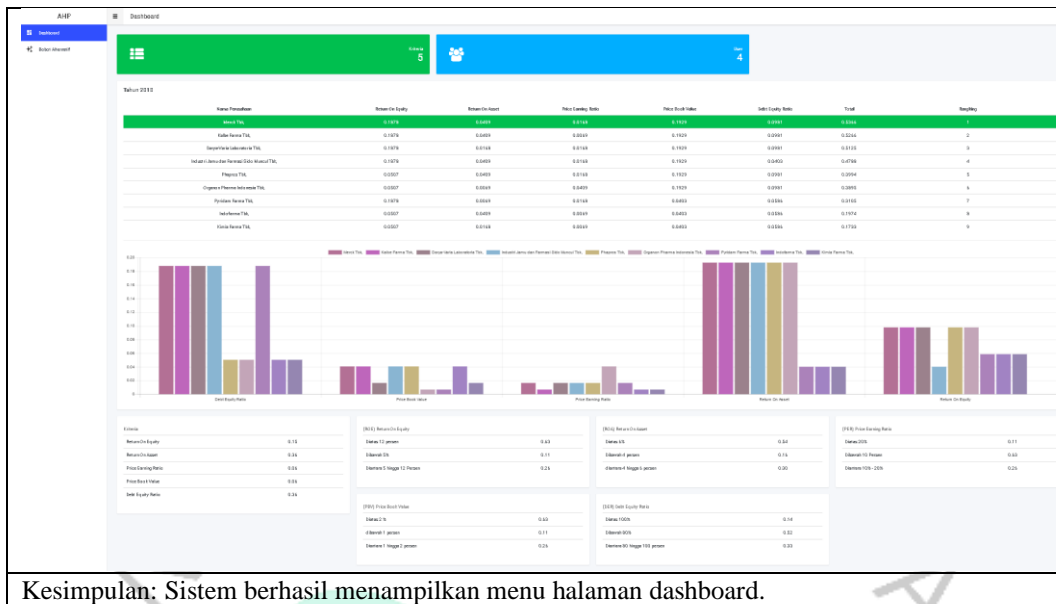
Hasil



Kesimpulan: sistem berhasil menampilkan menu user.

Tabel 5.3 Hasil Pengujian Kotak Hitam (Black Box Testing) User

No.	Skenario Pengujian	Hasil yang Diharapkan
1	User mengakses website aplikasi AHP	Sistem menampilkan menu halaman dashboard.
Hasil		



Kesimpulan: Sistem berhasil menampilkan menu halaman dashboard.

## 5.2 Pembahasan

Di bawah ini adalah group Saham di dalamnya terdapat 9 Perusahaan yang dinilai menggunakan penilaian dan perhitungan *Analytical Hierarchy Process (AHP)* berdasarkan kriteria yang telah ditentukan.

Tabel 5. 4 Bobot Prioritas Kriteria

Hasil Perhitungan					
	PER	PBV	ROE	DER	ROA
DVLA	0,016836	0,016836	0,0980846	0,1878	0,192913
INAF	0,006861	0,040934	0,0403427	0,050669	0,058623
KAEF	0,006861	0,016836	0,0403427	0,050669	0,058623
KLBF	0,006861	0,040934	0,0980846	0,1878	0,192913
MERK	0,016836	0,040934	0,0980846	0,1878	0,192913
PEHA	0,016836	0,040934	0,0980846	0,050669	0,192913
PYFA	0,016836	0,006861	0,0403427	0,1878	0,058623
SCPI	0,040934	0,006861	0,0980846	0,050669	0,192913
SIDO	0,016836	0,040934	0,0000021	0,1878	0,192913

Tabel 5. 5 Rangking Saham

Kode Perusahaan	Total	Ranking
MERK	0,5366	1
KLBF	0,5266	2

DVLA	0,5125	3
SIDO	0,4385	4
PEHA	0,3994	5
SCPI	0,3895	6
PYFA	0,3105	7
INAF	0,1974	8
KAEF	0,1733	9

