APPENDIX

PROCEDURE HITUNG PERAMALAN

```
1. <?php
2.
3. namespace App\Http\Controllers;
4.
5. use Illuminate\Http\Request;
6. use App\Imports\DatasetImport;
7. use App\Model\Datasets;
8. use App\Model\Result;
9.
10.  class DashboardController extends Controller
11.  {
12.      private float $alpha = 0.1;
13.
14.      public function index() {
15.                      $datasets  =  Datasets::orderBy('tahun',  'asc')-
   >orderBy('bulan', 'asc')->get();
16.
17.          return view('my-dashboard', [
18.              'datasets' => $datasets,
19.              'result' => null
20.          ]);
21.      }
22.
23.      public function import(Request $request) {
24.          try {
25.              if ($request->hasFile('file_excel')){
26.                  Datasets::query()->truncate();
27.
28.                  $path = $request->file('file_excel')->getRealPath();
29.                          \Excel::import(new  DatasetImport,  $request-
   >file_excel);
30.              }
31.
32.                      return  redirect()->back()->with('successMessage',
   'aDataset berhasil di import');
33.
34.          } catch(\Exception $e) {
35.                      return  redirect()->back()->with('errorMessage',  $e-
   >getMessage());
36.          }
37.      }
38.
39.      public function hitung(Request $request) {
40.          if ($request->filled('alpha')) {
41.              $this->alpha = $request->get('alpha', 0.1);
42.          }
```

```
43.
44.                     $datasets   =   Datasets::orderBy('tahun',   'asc')-
   >orderBy('bulan', 'asc')->get();
45.
46.          Result::query()->truncate();
47.
48.          $counter = 0;
49.          $recordTerakhir = null;
50.          $hasil = [];
51.
52.          foreach($datasets as $dataset) {
53.              $hasil[$counter]['bulan']      = $dataset->bulan;
54.              $hasil[$counter]['nama_bulan'] = $dataset->nama_bulan;
55.              $hasil[$counter]['tahun']      = $dataset->tahun;
56.              $hasil[$counter]['jumlah']     = $dataset->jumlah;
57.
58.              if ($counter == 0) {
59.                  $hasil[$counter]['st1'] = $dataset->jumlah;
60.                  $hasil[$counter]['st2'] = $dataset->jumlah;
61.                                      $hasil[$counter]['at']    =  $this-
   >hitungAt($hasil[$counter]['st2'], $hasil[$counter]['st1']);
62.                                      $hasil[$counter]['bt']    =  $this-
   >hitungBt($hasil[$counter]['st2'], $hasil[$counter]['st1']);
63.                   $hasil[$counter]['yt']  = $dataset->jumlah;
64.                                  $hasil[$counter]['mape']  = 0;
65.
66.              } else {
67.                                      $hasil[$counter]['st1']  =  $this-
   >hitungSt1($hasil[$counter-1]['jumlah'], $hasil[$counter-1]['st1']);
68.                                      $hasil[$counter]['st2']  =  $this-
   >hitungSt2($hasil[$counter]['st1'], $hasil[$counter-1]['st2']);
69.                                      $hasil[$counter]['at']    =  $this-
   >hitungAt($hasil[$counter-1]['st2'], $hasil[$counter-1]['st1']);
70.                                      $hasil[$counter]['bt']    =  $this-
   >hitungBt($hasil[$counter-1]['st2'], $hasil[$counter-1]['st1']);
71.                                      $hasil[$counter]['yt']    =  $this-
   >hitungYt($hasil[$counter]['at'], $hasil[$counter]['bt']);
72.                                   $hasil[$counter]['mape']    =  $this-
   >hitungMape($hasil[$counter]['yt'], $hasil[$counter]['jumlah']);
73.              }
74.
75.              $hasil[$counter]['created_at'] = now();
76.              $hasil[$counter]['updated_at'] = now();
77.                          $hasil[$counter]['prediksi']        =    $this-
   >hitungPrediksi($hasil[$counter]['at'], $hasil[$counter]['bt']);
78.              $counter++;
79.          }
80.
81.          Result::insert($hasil);
82.
83.          $result = Result::all();
```

```
 84.
 85.          return view('my-dashboard', [
 86.              'datasets' => $datasets,
 87.              'result' => $result
 88.          ]);
 89.      }
40.
41.
42.     public function hitungSt1($jumlah, $st1) {
43.         return ($this->alpha * $jumlah) + ((1-$this->alpha) * $st1);
44.     }
45.
46.     public function hitungSt2($st1, $st2) {
47.         return ($this->alpha * $st1) + ((1-$this->alpha) * $st2);
48.     }
49.
50.     public function hitungAt($st2, $st1) {
51.         return round(2 * $st2 - $st1,0);
52.     }
53.
54.     public function hitungBt($st2, $st1) {
55.   return  abs  (round(($this->alpha  /  (1-$this->alpha))  *  ($st2  -
   $st1),0));
56.     }
57.
58.     public function hitungYt($at, $bt) {
59.         return ($at + ($bt * 1));
60.     }
61.          public function hitungMape($jumlah, $yt) {
62.         return abs(( $jumlah - $yt ) / $jumlah)) *100 ;
63.     }
64.   public function hitungPrediksi($at, $bt) {
65.         return abs( ($at + ($bt  *2) )) ;
66.     }
```

PLAGIARISM CHECK.ORG

7.2% PLAGIARISM APPROXIMATELY

Report #13398525

CHAPTER 1 INTRODUCTION 1.1 Background In today's era, the use of technology is needed to support efficiency especially with a pandemic like this. With current conditions, the number of passengers has decreased drastically. Of course, the amount of expenditure will be greater than the amount of income. To solve this problem, a forecasting system is needed that can make predictions in the future related to the number of passengers. In this project, a forecasting system will be made using the Double exponential smoothing algorithm. in its application this program will perform repeated calculations with alpha parameters 0.1 to 0.9 and then a comparison will be made to determine the results with the smallest error percentage. The data to be used is flight history data for the last three years. This algorithm will calculate the predicted number of passenger departures and then a comparison will be made against the actual data. To determine the accuracy of this method, it takes a calculation method called mean absolute percentage