

LAMPIRAN

Lampiran 1 Proses Pengambilan Data

pengambilan Januari 2018

tgl	18	32	40	24	mc	keterangan
30/12/2017	100	100				alat dromo
	100	100				alat dromo
	101	100				alat dromo
	200					alat dromo
	2					alat dromo
4/1/2018	45	100				alat dromo
	1	17				alat dromo
	1	17				alat dromo
	303					alat dromo
	203					alat dromo
5/1/2018	505	101				alat dromo
	2					alat dromo
	2					alat dromo
	404					alat dromo
	100	100	80	10		alat dromo
6/1/2018	1010					alat dromo
	700					alat dromo
	2					alat dromo
8/1/2018	20					alat dromo
	200					alat dromo
	2					alat dromo
9/1/2018	1816					alat dromo
	50					alat dromo
	101	101	101			alat dromo
	15					alat dromo
	1					alat dromo
10/1/2018	202					alat dromo
	4					alat dromo
	101					alat dromo
	25					alat dromo
	4					alat dromo
11/1/2018	202					alat dromo
	100					alat dromo
	100					alat dromo
	4					alat dromo
	505					alat dromo
12/1/2018	1010					alat dromo
	204					alat dromo
	4					alat dromo
	2					alat dromo
13/1/2018	1010					alat dromo
	100					alat dromo
	1					alat dromo
14/1/2018	505					alat dromo
	101					alat dromo
	202					alat dromo
	202					alat dromo
16/1/2018	505					alat dromo
	101					alat dromo
	202					alat dromo
	303					alat dromo
	2					alat dromo

Januari 2018

Hasil Produksi

tgl	prod	18	32	40	24	botol	c.baocor	mc	rsk
25/12/2017	60	100	540						
26/12/2017	84		619						8
27/12/2017	84	100	778					1	40
28/12/2017	120		1.072	101				2	
29/12/2017	96		837	100					10
30/12/2017	120	116	1.001	100					
31/12/2017	60	200	486					4	
1/1/2018	60		465	102				1	20
2/1/2018	60	46	441	100				1	5
3/1/2018	60	55	565						4
4/1/2018	96		545	68	227			27	
5/1/2018	96		944					2	
6/1/2018	120		1.184					7	
7/1/2018	120		1.183					10	
8/1/2018	120	100	958	132				9	
9/1/2018	120	140	979	100				21	
10/1/2018	120	119	1.113					25	
11/1/2018	120		1.178					31	
12/1/2018	120		946	197				10	
13/1/2018	120	200	1.061					6	160
14/1/2018	120		191	797				15	
15/1/2018	120	954	255	314				9	
16/1/2018	120	145	1.094					11	
17/1/2018	120		1.191					16	80
18/1/2018	120		1.190					17	80
19/1/2018	120		1.182					25	120
20/1/2018	120		1.184					24	80
21/1/2018	60		592					18	
22/1/2018	60		597					15	8
23/1/2018	60		593					12	10
24/1/2018	60		597					6	
25/1/2018	60		596						
26/1/2018	60		596						
27/1/2018	60		596						
28/1/2018	60		596						
29/1/2018	60		596						
30/1/2018	60		596						
31/1/2018	60		596						
31/1/2018	3,096	2,275	26,379	2,111	727	-	337	613	

cup cup

Lampiran 2 Dataset Penjualan 2017-2021

No	Bulan	Tahun	Isi(Perdus)		
			18	32	40
1	Januari	2017	611	13304	916
2	Februari	2017	355	9944	665
3	Maret	2017	424	21360	1311
4	April	2017	885	22175	1311
5	Mei	2017	2640	37894	1928
6	Juni	2017	959	32713	3196
7	Juli	2017	1037	15338	1683
8	Agustus	2017	-	-	-
9	September	2017	933	7459	702
10	Oktober	2017	408	7847	469
11	November	2017	497	7718	480
12	Desember	2017	1037	16407	1192
13	Januari	2018	1728	19350	1647
14	Februari	2018	-	-	-
15	Maret	2018	1480	40306	1103
16	April	2018	1847	44046	1049
17	Mei	2018	3708	48561	2824
18	Juni	2018	2295	28023	3618
19	Juli	2018	227	7603	547
20	Agustus	2018	823	6500	716
21	September	2018	997	10993	734
22	Oktober	2018	742	10307	353
23	November	2018	1021	15000	588
24	Desember	2018	1344	18529	805
25	Januari	2019	1051	13375	666
26	Februari	2019	2160	28109	1430
27	Maret	2019	855	38381	761

28	April	2019	2473	46240	1631
29	Mei	2019	2548	48624	2281
30	Juni	2019	1433	22958	1272
31	Juli	2019	1656	18726	618
32	Agustus	2019	728	10503	563
33	September	2019	1329	10936	435
34	Oktober	2019	848	10956	495
35	November	2019	1027	14582	731
36	Desember	2019	1740	20008	857
37	Januari	2020	4343	28769	2015
38	Februari	2020	2216	40211	683
39	Maret	2020	1942	37009	1531
40	April	2020	1949	25823	683
41	Mei	2020	-	-	-
42	Juni	2020	728	5598	650
43	Juli	2020	549	3170	463
44	Agustus	2020	309	2587	520
45	September	2020	85	6125	188
46	Oktober	2020	989	7180	214
47	November	2020	898	7012	320
48	Desember	2020	210	6089	278
49	Januari	2021	540	10203	402
50	Februari	2021	760	12650	520
51	Maret	2021	1070	7890	698
52	April	2021	1533	10203	750
53	Mei	2021	1329	9438	442

Lampiran 3 Dataset Persediaan Stok 2017-2021

No	Bulan	Tahun	Isi(Perdus)		
			18	32	40
1	Januari	2017	890	14273	1072
2	Februari	2017	200	11730	697
3	Maret	2017	411	21319	727
4	April	2017	1048	23002	2727
5	Mei	2017	2332	35200	1925
6	Juni	2017	601	31218	2959
7	Juli	2017	1309	23241	2158
8	Agustus	2017	-	-	-
9	September	2017	1018	6999	789
10	Oktober	2017	258	6392	213
11	November	2017	744	7706	537
12	Desember	2017	955	18129	1386
13	Januari	2018	2275	26379	2111
14	Februari	2018	-	-	-
15	Maret	2018	998	38991	636
16	April	2018	1686	44055	1091
17	Mei	2018	3693	44317	3121
18	Juni	2018	2038	23988	2534
19	Juli	2018	400	3545	596
20	Agustus	2018	1015	6512	414
21	September	2018	811	9023	560
22	Oktober	2018	1071	15802	817
23	November	2018	846	11258	456
24	Desember	2018	1147	17158	906
25	Januari	2019	700	14622	563
26	Februari	2019	2638	33855	1020
27	Maret	2019	238	39330	785
28	April	2019	2269	40778	1105

29	Mei	2019	2073	46010	1747
30	Juni	2019	-	-	-
31	Juli	2019	2609	15519	666
32	Agustus	2019	871	10299	562
33	September	2019	1319	11586	259
34	Oktober	2019	961	12247	597
35	November	2019	784	12669	581
36	Desember	2019	1659	20422	685
37	Januari	2020	4080	5547	30785
38	Februari	2020	2027	34095	805
39	Maret	2020	2362	56575	1304
40	April	2020	974	25120	897
41	Mei	2020	-	-	-
42	Juni	2020	1303	8090	3546
43	Juli	2020	1669	5880	3123
44	Agustus	2020	210	4120	2109
45	September	2020	1290	3520	2960
46	Oktober	2020	3128	4098	2730
47	November	2020	209	10810	1847
48	Desember	2020	962	9226	1203
49	Januari	2021	1403	15981	2539
50	Februari	2021	1050	10879	3098
51	Maret	2021	950	5379	3340
52	April	2021	1253	7273	2509
53	Mei	2021	792	9792	1180

Lampiran 4 Source Code Implementasi

a. Data Admin

```
public function aksi_tambah()
{
    $nama=$this->input->post('nama');
    $telepon=$this->input->post('telepon');
    $username=$this->input->post('username');
    $password=$this->input->post('password');

    $data_insert = array(
        'nama' => $nama,
        'telepon' => $telepon,
        'username' => $username,
        'password' => $password,
        'level' => 2
    );
    $this->model_data >insert($data_insert,'login');
    redirect('superadmin/c_data_admin');
}

public function update_admin()
{
    $id_login = $this->input->get('id_login');
    $nama=$this->input->post('nama');
    $telepon=$this->input->post('telepon');
    $username=$this->input->post('username');
    $password=$this->input->post('password');

    $where = array(
        'id_login' => $id_login
    );
    $data_insert = array(
        'nama' => $nama,
        'telepon' => $telepon,
        'username' => $username,
        'password' => $password,
        'level' => 2
    );
    $this->model_data>edit_data($where, $data_insert,'login');

    redirect('superadmin/c_data_admin');
}

public function aksi_hapus()
{
    $id_login = $this->input->get('id_login');
    $where = array(
        'id_login' => $id_login
    );
    $this->model_data->delete_data($where, 'login');

    redirect('superadmin/c_data_admin');
}
```

b. Data Kategori

```
public function aksi_tambah()
{
    $nama_kategori=$this->input->post('nama_kategori');

    $data_insert = array(
        'nama_kategori' => $nama_kategori
    );
}

public function update_kategori()
{
    $id_kategori = $this->input->get('id_kategori');
    $nama=$this->input->post('nama');

    $where = array(
        'id_kategori' => $id_kategori
    );
    $data_insert = array(
        'nama' => $nama
    );
    $this->model_data->
>edit_data($where, $data_insert,'kategori');

    redirect('superadmin/c_kategori');
}

public function aksi_hapus()
{
    $id_penjualan = $this->input->get('id_penjualan');
    $where = array(
        'id_penjualan' => $id_penjualan
    );
    $this->model_data->delete_data($where, 'data_penjualan');

    redirect('admin/c_penjualan/index');
}
```

c. Data Penjualan

```
public function aksi_tambah()
{
    $list_id_kategori=$this->input->post('list_id_kategori');
    $list_bulan=$this->input->post('list_bulan');
    $tahun=$this->input->post('tahun');
    $jumlah_penjualan=$this->input->post('jumlah_penjualan');
```

```

        $data_insert = array(
            'id_kategori' => $list_id_kategori,
            'bulan' => $list_bulan,
            'tahun' => $tahun,
            'jumlah_penjualan' => $jumlah_penjualan
        );
        $this->model_data->insert($data_insert, 'data_penjualan');

        redirect('admin/c_penjualan');
    }

    public function update_penjualan()
    {
        $id_penjualan = $this->input->get('id_penjualan');
        $list_id_kategori=$this->input->post('list_id_kategori');
        $list_bulan=$this->input->post('list_bulan');
        $tahun=$this->input->post('tahun');
        $jumlah_penjualan=$this->input->post('jumlah_penjualan');
        $where = array(
            'id_penjualan' => $id_penjualan
        );
        $data_insert = array(
            'id_kategori' => $list_id_kategori,
            'bulan' => $list_bulan,
            'tahun' => $tahun,
            'jumlah_penjualan' => $jumlah_penjualan
        );
        $this->model_data->
>edit_data($where, $data_insert, 'data_penjualan');
        redirect('admin/c_penjualan/index');
    }

    public function aksi_hapus()
    {
        $id_penjualan = $this->input->get('id_penjualan');
        $where = array(
            'id_penjualan' => $id_penjualan
        );
        $this->model_data->delete_data($where, 'data_penjualan');

        redirect('admin/c_penjualan/index');
    }
}

```


d. Data Persediaan

```
public function aksi_tambah()
{
    $list_id_kategori=$this->input->post('list_id_kategori');
    $list_bulan=$this->input->post('list_bulan');
    $tahun=$this->input->post('tahun');
    $jumlah_persediaan=$this->input-
>post('jumlah_persediaan');

    $data_insert = array(
        'id_kategori' => $list_id_kategori,
        'bulan' => $list_bulan,
        'tahun' => $tahun,
        'jumlah_persediaan' => $jumlah_persediaan
    );
    $this->model_data-
>insert($data_insert,'data_persediaan');

    redirect('admin/c_persediaan');
}

public function update_persediaan()
{
    $id_persediaan = $this->input->get('id_persediaan');
    $list_id_kategori=$this->input->post('list_id_kategori');
    $list_bulan=$this->input->post('list_bulan');
    $tahun=$this->input->post('tahun');
    $jumlah_persediaan=$this->input-
>post('jumlah_persediaan');
    $where = array(
        'id_persediaan' => $id_persediaan );
    $data_insert = array(
        'id_kategori' => $list_id_kategori,
        'bulan' => $list_bulan,
        'tahun' => $tahun,
        'jumlah_persediaan' => $jumlah_persediaan);
    $this->model_data-
>edit_data($where, $data_insert,'data_persediaan');
    redirect('admin/c_persediaan/index');
}

public function aksi_hapus()
{
    $id_persediaan = $this->input->get('id_persediaan');
    $where = array(
        'id_persediaan' => $id_persediaan
    );
    $this->model_data-
>delete_data($where, 'data_persediaan');

    redirect('admin/c_persediaan/index');
}
```

e. Proses Prediksi Metode Extreme Learning Machine

```
// Mencari nilai min-max setiap fitur
private function max($array)
{
    $n      = count($array);
    sort($array);
    $max    = $array[0];
    for ($i = 1; $i < $n; $i++) {
        if ($max < $array[$i]) {
            $max = $array[$i];
        }
    }
    return $max;
}

private function min($array)
{
    $array = array_unique($array);
    sort($array); //mengurutkan nilai array
    $n      = count($array);
    $min    = $array[1];
    for ($i = 1; $i < $n; $i++) {
        if ($min > $array[$i]) {
            $min = $array[$i];
        }
    }
    return $min;
}

public function min_max() {
    $array = $this->dataset();
    foreach ($array as $a) {
        $x1[] = $a['x1'];
        $x2[] = $a['x2'];
        $x3[] = $a['x3'];
        $T[]  = $a['T'];
    }
    $min_x1 = $this->min($x1);
    $max_x1 = $this->max($x1);

    $min_x2 = $this->min($x2);
    $max_x2 = $this->max($x2);

    // Proses normalisasi data
    public function normalisasi() {
        $array = $this->dataset();
        foreach ($array as $a) {
            $x1[] = $a['x1'];
            $x2[] = $a['x2'];
            $x3[] = $a['x3'];
            $T[]  = $a['T'];
        }

        $min_x1 = $this->min($x1);
        $max_x1 = $this->max($x1);

        $min_x2 = $this->min($x2);
        $max_x2 = $this->max($x2);
    }
}
```

```

$min_x3 = $this->min($x3);
$max_x3 = $this->max($x3);

$min_T = $this->min($T);
$max_T = $this->max($T);

$jumlah_array = count($x1);
for ($i=0; $i < $jumlah_array; $i++) {
    $data[] =
        array(
            'x1'=> ($x1[$i]-$min_x1)/($max_x1-$min_x1),
            'x2'=> ($x2[$i]-$min_x2)/($max_x2-$min_x2),
            'x3'=> ($x3[$i]-$min_x3)/($max_x3-$min_x3),
            'T'=> ($T[$i]-$min_T)/($max_T-$min_T)
        );
    }

    return $data;
}

// INISIALISASI INPUT WEIGHT DAN BIAS
public function weight() {
    $w = array(
        array(
            'w1' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1),
            'w2' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1),
            'w3' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1)
        ),
        array(
            'w1' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1),
            'w2' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1),
            'w3' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1)
        )
    );
    return $w;
}

public function bias() {
    $b = array(
        array(
            'b1' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1),
            'b2' => mt_rand(-1*10000, 1*10000)/10000 * (1-(-1)) + (-1)
        )
    );
    return $b;
}

```

```

// PROSES DATA TRAINING
public function keluaran_hidden_layer() {
    $bias    = $this->bias();
    foreach ($bias as $b) {
        $b1    = $b['b1'];
        $b2    = $b['b2'];
    }

    $weight = $this->weight();
    foreach ($weight as $we) {
        $w1[]  = $we['w1'];
        $w2[]  = $we['w2'];
        $w3[]  = $we['w3'];
    }
    $normalisasi = $this->normalisasi();
    foreach ($normalisasi as $nm) {
        $x1[]  = $nm['x1'];
        $x2[]  = $nm['x2'];
        $x3[]  = $nm['x3'];
        $T[]   = $nm['T'];
    }
    $jml_data_norm    = count($normalisasi);
    $persentase_train = $jml_data_norm * 80/100;
    $jml_train        = round($persentase_train, 0);
    for ($i=0; $i < $jml_train; $i++) {
        $data[] =
            array(
                '1'=>    (($w1[0]      *      $x1[$i])+($w2[0]      *
                $x2[$i])+($w3[0] * $x3[$i]))+$b1, //(w1 * X1) + (w1
                * X1) + (w1 * X1) + bias
                '2'=>    (($w1[1]      *      $x1[$i])+($w2[1]      *
                $x2[$i])+($w3[1] * $x3[$i]))+$b2
            );
    }
    return $data;
}

public function khl_fa() {
    $khl = $this->keluaran_hidden_layer();
    foreach ($khl as $hl) {
        $khl1[]  = $hl['1'];
        $khl2[]  = $hl['2'];
    }

    $normalisasi = $this->normalisasi();
    foreach ($normalisasi as $nm) {
        $x1[]  = $nm['x1'];
        $x2[]  = $nm['x2'];
        $x3[]  = $nm['x3'];
        $T[]   = $nm['T'];
    }
}

```

```

    $jml_data_norm = count
    $persentase_train = $jml_data_norm * 80/100;
    $jml_train = round($persentase_train, 0);
    for ($i=0; $i < $jml_train; $i++) {
        $data[] =
            array(
                '1'=> 1 / (1 + exp(- $khl1[$i])),
                '2'=> 1 / (1 + exp(- $khl2[$i]))
            );
    }
    return $data;
}

public function khl_fa_transpose() {
    $khl_fa = $this->khl_fa();
    foreach ($khl_fa as $kf) {
        $khl_fa1[] = $kf['1'];
        $khl_fa2[] = $kf['2'];
    }
    $normalisasi = $this->normalisasi();
    $jml_data_norm = count($normalisasi);
    $persentase_train = $jml_data_norm * 80/100;
    $jml_train = round($persentase_train, 0);

    for ($i=0; $i < $jml_train; $i++) {
        $data[] = array(
            array(
                '1'=> $khl_fa1[$i]
            ),
            array(
                '2'=> $khl_fa2[$i]
            )
        );
    }
    return $data;
}

public function khl_fa_perkalian() {
    $khl = $this->khl_fa();
    foreach ($khl as $hl) {
        $khl1[] = $hl['1'];
        $khl2[] = $hl['2'];
    }

    $khlt = $this->khl_fa_transpose();
    foreach ($khlt as $hlt) {
        $khlt1[] = $hlt[0]['1'];
    }
    foreach ($khlt as $hlt) {
        $khlt2[] = $hlt[1]['2'];
    }
    $jml_khlt = count($khlt);
}

```

```

for ($i=0; $i < $jml_khlt; $i++) {
    $hasil[] = array(
        array(
            '1' => $khl1[$i]*$khlt1[$i],
            '2' => $khl1[$i]*$khlt2[$i]
        ),
        array(
            '1' => $khl2[$i]*$khlt1[$i],
            '2' => $khl2[$i]*$khlt2[$i]
        )
    );
}
foreach ($hasil as $hasil1) {
    $h1_1 += $hasil1[0]['1'];
    $h1_2 += $hasil1[0]['2'];
    $h2_1 += $hasil1[1]['1'];
    $h2_2 += $hasil1[1]['2'];
}
$data[] = array(
    'h1_1' => $h1_1, //C87
    'h1_2' => $h1_2, //D87
    'h2_1' => $h2_1, //C88
    'h2_2' => $h2_2 //D88
);
return $data;
}

public function invers() {
    $khlfa_perkalian = $this->khl_fa_perkalian();
    foreach ($khlfa_perkalian as $matrix) {
        $C87 = $matrix['h1_1'];
        $D87 = $matrix['h1_2'];
        $C88 = $matrix['h2_1'];
        $D88 = $matrix['h2_2'];
    }

    $invers = 1 / $C87*$D88-$C88*$D87;
    return $invers;
}
$invers;
}

public function layers_matrix() {
    $khlfa_perkalian = $this->khl_fa_perkalian();
    foreach ($khlfa_perkalian as $matrix) {
        $C87 = $matrix['h1_1'];
        $D87 = $matrix['h1_2'];
        $C88 = $matrix['h2_1'];
        $D88 = $matrix['h2_2'];
    }
    $invers = $this->invers();
}

```

```

    $data[] = array(
        'layer11' => $invers * $D88,
        'layer12' => $invers * -$D87,
        'layer21' => $invers * -$C88,
        'layer22' => $invers * $C87
    );

    return $data;
}

public function mpg_invers() {
    $khlt = $this->khl_fa_transpose();
    $jml_khlt = count($khlt);
    foreach ($khlt as $hlt) {
        $khlt1[] = $hlt[0]['1'];
    }
    foreach ($khlt as $hlt) {
        $khlt2[] = $hlt[1]['2'];
    }
    $layer_matrix = $this->layers_matrix();
    foreach ($layer_matrix as $lm) {
        $k87 = $lm['layer11'];
        $l87 = $lm['layer12'];
        $k88 = $lm['layer21'];
        $l88 = $lm['layer22'];
    }
    $normalisasi = $this->normalisasi();
    $jml_data_norm = count($normalisasi);
    $persentase_train = $jml_data_norm * 80/100;
    $jml_train = round($persentase_train, 0);
    for ($i=0; $i < $jml_train; $i++) {
        $data[] = array(
            '1' => ($khlt1[$i] * $k87) + ($l87 * $khlt2[$i]),
            '2' => ($k88 * $khlt1[$i]) + ($l88 * $khlt2[$i])
        );
    }
    return $data;
}

public function output_weight() {
    $normalisasi = $this->normalisasi();
    foreach($normalisasi as $n) {
        $t[] = $n['T'];
    }
    $mpg_invers = $this->mpg_invers();
    foreach($mpg_invers as $mpg) {
        $data1[] = $mpg['1'];
        $data2[] = $mpg['2'];
    }
    $ow1 = 0;
    $ow2 = 0;
    $jml_mpg = count($mpg_invers);
}

```

```

        for ($i=0; $i < $jml_mpg; $i++) {
            $ow1 += ($data1[$i] * $t[$i]);
            $ow2 += ($data2[$i] * $t[$i]);
        }
        $data[] = array(
            '1' => $ow1,
            '2' => $ow2
        );
        return $data;
    }

// PROSES DATA TESTING
public function testing_khl() {
    $bias = $this->bias(); //mengambil data Bias
    foreach ($bias as $b) {
        $b1 = $b['b1'];
        $b2 = $b['b2'];
    }
    $weight = $this->weight();
    foreach ($weight as $we) {
        $w1[] = $we['w1'];
        $w2[] = $we['w2'];
        $w3[] = $we['w3'];
    }

    $normalisasi = $this->data_testing();
    foreach ($normalisasi as $nm) {
        $x1[] = $nm['x1'];
        $x2[] = $nm['x2'];
        $x3[] = $nm['x3'];
        $T[] = $nm['T'];
    }
    $jml_data_norm = count($normalisasi);
    $persentase_test = round($jml_data_norm, 0) * 20/100;
    $urutan_test = $jml_data_norm - $persentase_test
    for ($i=0; $i < $jml_data_norm; $i++) {
        $data[] =
            array(
                '1'=> (($w1[0] * $x1[$i])+$w2[0] *
                    $x2[$i])+$w3[0] * $x3[$i))+$b1, //(w1 * X1)
                    + (w1 * X1) + (w1 * X1) + bias
                '2'=> (($w1[1] * $x1[$i])+$w2[1] *
                    $x2[$i])+$w3[1] * $x3[$i))+$b2
            );
    }
    return $data;
}

public function testing_khl_fa() {
    $khl = $this->testing_khl();
    foreach ($khl as $hl) {
        $khl1[] = $hl['1'];
        $khl2[] = $hl['2'];
    }
}

```



```

urutan_test = count($khl);
for ($i=0; $i < $urutan_test; $i++) {
    $data[] =
        array(
            '1' => 1 / (1 + exp(- $khl1[$i])),
            '2' => 1 / (1 + exp(- $khl2[$i]))
        );
    return $data;
}

public function testing_output_layer() {
    $khl_fa = $this->testing_khl_fa();
    foreach ($khl_fa as $khlfa) {
        $khlfa1[] = $khlfa['1'];
        $khlfa2[] = $khlfa['2'];
    }
    $ow = $this->output_weight();
    foreach($ow as $o) {
        $ow1 = $o['1'];
        $ow2 = $o['2'];
    }
    $jml_ow = count($khl_fa);
    $y1 = 0;
    $y2 = 0;
    $j = $jml_ow - 1;
    $i = $jml_ow - 2;

    $y1 += $khlfa1[$i]*$ow1;
    $y2 += $khlfa2[$i]*$ow1;
    $y1 += $khlfa1[$j]*$ow2;
    $y2 += $khlfa2[$j]*$ow2;

    $data[] = array(
        '1' => $y1,
        '2' => $y2
    );
    return $data;
}

//PERHITUNGAN NILAI MSE
public function mse() {
    $normalisasi = $this->normalisasi();
    foreach($normalisasi as $nm) {
        $x1[] = $nm['x1'];
        $x2[] = $nm['x2'];
        $x3[] = $nm['x3'];
        $T[] = $nm['T'];
    }

    $ol = $this->testing_output_layer();
    foreach ($ol as $o) {
        $o1 = $o['1'];
        $o2 = $o['2'];
    }
}

```

```

$normalisasi = $this->normalisasi();
$jml_data_norm = count($normalisasi);
$persentase_train = $jml_data_norm * 20/100;
$urut = $jml_data_norm - round($persentase_train, 0);
for ($i=$urut; $i < $jml_data_norm; $i++) {
    $T_test[] = $T[$i];
}
$jmt = count($T_test) - 1;
$j = 1;
$h = 0;
$data = 0;
for ($i=0; $i < $jmt; $i++) {
    $data += pow(($o1-$T_test[$h]), 2) + pow(($o2
    $T_test[$j]), 2);
    $h+2;
    $j+2;
}
return $data;
}

// PROSES DENORMALISASI DATA
public function denormalisasi_data() {
    $array = $this->dataset();
    foreach ($array as $a) {
        $x1[] = $a['x1'];
        $x2[] = $a['x2'];
        $x3[] = $a['x3'];
        $T[] = $a['T'];
    }
    $min_x3 = $this->min($x3);
    $min_T = $this->min($T);
    $max_T = $this->max($T);

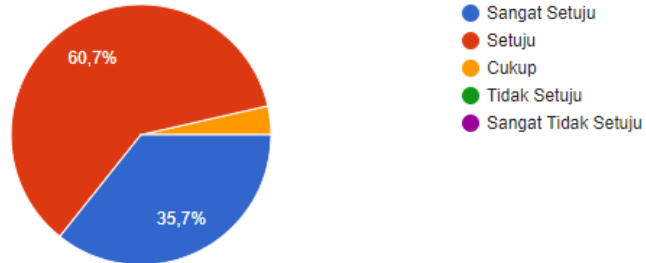
    $t_max = $max_T;
    $t_min = $min_T;
    $min = $min_x3;
    $o1 = $this->testing_output_layer();
    foreach ($o1 as $o) {
        $o1 = $o['1'];
        $o2 = $o['2'];
    }
    $denormalisasi[] = array(
        '1' => $o1*($t_max-$t_min)+$min,
        '2' => $o2*($t_max-$t_min)+$min
    );
    return $denormalisasi;
}

```

Lampiran 5 Hasil Kuisisioner Pengujian user

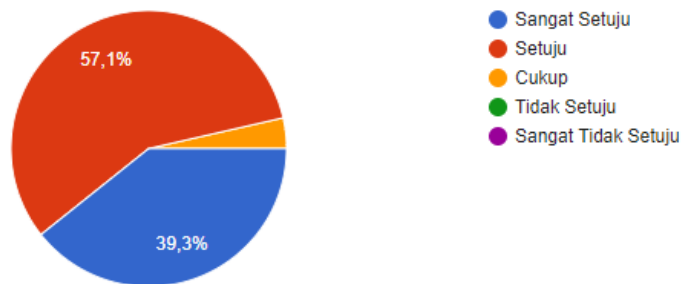
Menurut anda apakah tampilan website ini menarik ?

28 jawaban



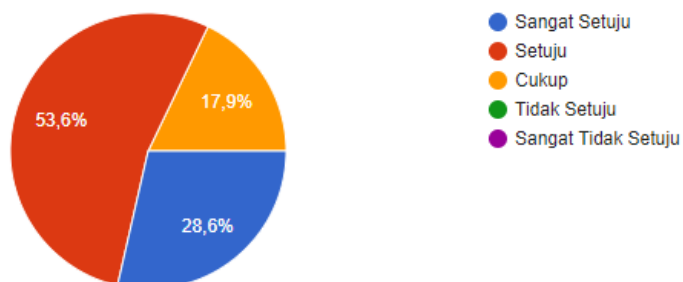
Apakah fitur pada sistem mudah dipahami ?

28 jawaban



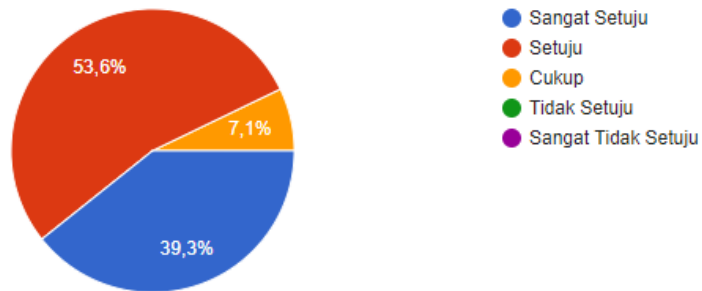
Apakah fitur yang terdapat pada sistem ini sudah lengkap?

28 jawaban



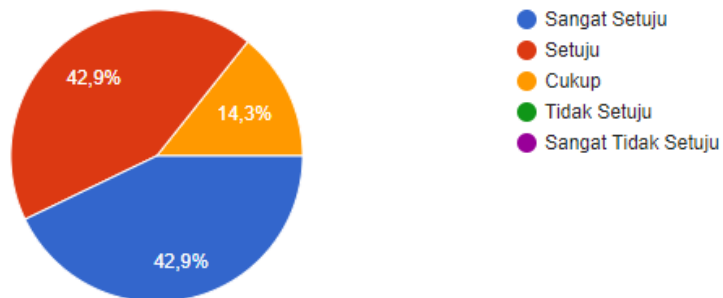
Apakah menu yang berada pada sistem ini mudah dipahami?

28 jawaban



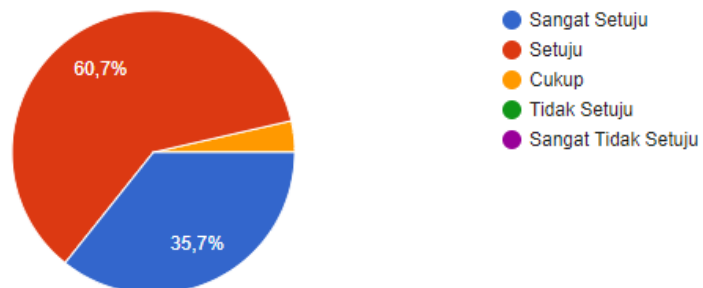
Apakah menu yang berada pada sistem ini sudah berjalan dengan fungsinya?

28 jawaban



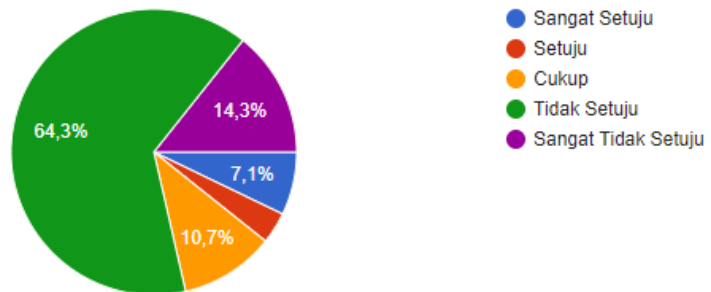
Apakah proses pencarian data yang diinginkan langsung ditampilkan?

28 jawaban



Apakah anda menemukan error pada saat melakukan klik pada website?

28 jawaban



Apakah sistem ini sudah membantu pengguna dalam memprediksi persediaan stok pada bulan berikutnya ?

28 jawaban

