

## DAFTAR PUSTAKA

- Academy of Managed Care Pharmacy (AMCP), The Reality of Drug Shortages. 2010.
- Amalia Febriani, Rizki, 2013. *Pengertian, Cara Pengumpulan, dan Jenis-Jenis Data dan Sample*.  
<https://rizkiamaliafebriani.wordpress.com/2013/04/19/pengertian-cara-pengumpulan-dan-jenis-jenis-data-dan-sample/>
- Arsyil Hendra Saputra. 2012. Analisis Data Runtun Waktu Dengan Metode Adaptive Neuro-Fuzzy Inference System (ANFIS). Semarang. Universitas Diponegoro.
- Aryanto, R. 2012. Perbaikan sistem persediaan bahan baku produk d-box pada PT. Homa Sejahtera. Skripsi. Universitas Binus
- Bagus Fatkhurrozi, M. Aziz Muslim, dan Didik R. Santoso, 2012. Penggunaan Artificial Neuro Fuzzy Inference Sistem (ANFIS) dalam Penentuan Status Aktivitas Gunung Merapi. Jurnal EECCIS Vol. 6, No. 2.
- Desokey Nader Eslam, Badr Amr, Hegazy Fatah Abdel, 2017. *Enhancing Stock Prediction Clustering Using K-Means with Genetic Algorithm*. ICENCO.
- Gaspersz, V. 2001. Production Planning and Inventory Control. Jakarta: Gramedia Pustaka Utama.

- Heizer, J. & Rander, B. 2005. Operations Management: Manajemen Operasi. Jakarta: Salemba Empat.
- Herjanto, E. 2006. Manajemen Operasi Edisi Ketiga. Jakarta: Grasindo.
- Jang, JSR., CT Sun, dan E Mizutani. 1997. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. London: Prentice-Hall, Inc.
- Jumingan. 2009. Studi Kelayakan Bisnis – Teori dan Pembuatan Proposal Kelayakan. Jakarta: Bumi Aksara
- Kusumadewi, S & Hartati S. 2006. Neuro Fuzzy: Integrasi Sistem Fuzzy & Jaringan Syaraf. Yogyakarta: Graha Ilmu.
- Lewis, C. D. 1982. Industrial and Business Forecasting Methods. London: Butterworths.
- Mariz Erlan Putra, Sobri Muhammad, Putra Ade, 2017. Sistem Informasi Prediksi Stok Barang Menggunakan Metode Holt Winters Pada Cv Budi Jaya Mandiri. Jurnal Ilmiah Fak. Ilkom Vol.1 No.1.
- Nurmawan. 2016. Prediksi produk karet menggunakan adaptive neuro fuzzy inference system (ANFIS). Medan. Universitas Sumatera Utara
- Palit, A.K. & Popovic, D. 2005. Computational Intelligence in Time Series Forecasting: Theory and Engineering Application. London: Springer.
- Peraturan Menteri Kesehatan No. 58 tahun 2014 tentang Standar Pelayanan Farmasidi Rumah Sakit.
- Peraturan Pemerintah No. 51 tahun 2009 tentang Pekerjaan Kefarmasian.

- Pritalia G L, 2018. Penerapan Algoritma C4.5 untuk Penentuan Ketersediaan Barang *E-commerce*. Indonesian Journal of Information Systems (IJIS) Vol. 1, No. 1.
- Renie Cuyuno Mellen, Widodo J.Pudjirahardjo, 2013. Faktor Penyebab Dan Kerugian Akibat Stockout Dan Stagnant Obat Di Unit Logistik Rsu Haji Surabaya. Jurnal Administrasi Kesehatan Indonesia Volume 1 Nomor 1.
- Srinivasan N, Lakshmi C, 2017. *Stock Prediction and Analysis Using Intermittent Training Data with Artificial Neural Networks*. ICII ECS.
- Suciati, Susi dan Adisasmito, Wiku BB, Analisis Perencanaan Obat Berdasarkan ABC Indeks Kritis di Instalasi Farmasi. Jurnal Manajemen Pelayanan Kesehatan. 2006.
- Veronika Afni, 2013. Analysis Data Time Series. Available: <https://veronikaafri.wordpress.com/2013/11/22/analysis-data-time-series/#:~:text=Data%20berkala%20adalah%20data%20yang,d ata%20atau%20disingkat%20time%20series>.
- Warsito, B dan Ispriyanti D. 2009. Uji Linearitas Data Time Series dengan RESET Test. Jurnal Matematika dan Komputer 7(3):36-44.
- Waqar Muhammad, et al., 2017. Prediction of Stock Market by Principal Component Analysis. CIS.

# LAMPIRAN

## BARIS KODE PROGRAM

### Baris Kode Program Menampilkan List Obat dan Alat Kesehatan

#### View

```
<?= $this->extend('layout/template'); ?>

<?= $this->section('content'); ?>

<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <h1 class="h3 mb-2 text-gray-800">Data Obat dan Alat Kesehatan</h1>

    <!-- DataTales Example -->
    <div class="card shadow mb-4">
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered" id="dataTable" width="100%"
cellspacing="0">
                    <thead>
                        <tr>
                            <th>Nama</th>
                            <th>Jenis</th>
                            <th>Penyedia</th>
                            <th>Detail</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php foreach ($obat as $o) : ?>
                            <tr>
                                <td><?= $o['OBAT']; ?></td>
                                <td><?= $o['NAMA_KATEGORI']; ?></td>
                                <td><?= $o['NAMA_SUPPLIER']; ?></td>
                                <td align="center"><a href="/obat/<?=
$o['ID_OBAT']; ?>" class="btn btn-success">Detail</a></td>
                            </tr>
                        <?php endforeach; ?>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
```

```

        </div>
    </div>

</div>
<!-- /.container-fluid -->

<?= $this->endSection(); ?>

```

## Controller

```

<?php

namespace App\Controllers;

use App\Models\ModelObat;

class Obat extends BaseController
{
    protected $ModelObat;
    public function __construct()
    {
        $this->ModelObat = new ModelObat();
    }

    public function index()
    {
        $data = [
            'title'      => 'Daftar Obat dan Alat Kesehatan',
            'obat'       => $this->ModelObat->getObat()
        ];
        return view('obat', $data);
    }

    public function detail($slug)
    {
        $data = [
            'title'      => 'Detail Transaksi',
            'transaksi'  => $this->ModelObat->getObat($slug)
        ];
        return view('detail', $data);
    }
}

```

## Model

```
<?php

namespace App\Models;

use CodeIgniter\Model;

class ModelObat extends Model
{
    protected $table = "tb_obat";
    protected $useTimestamps = true;

    public function getObat($obat = false)
    {
        if ($obat == false) {
            return $this->db->table('tb_obat')
                ->join('kategori', 'kategori.ID=tb_obat.KATEGORI')
                ->join('tb_supplier', 'tb_supplier.ID=tb_obat.PENYEDIA')
                ->join('obatkeluar', 'obatkeluar.NAMA=tb_obat.OBAT')
                ->join('tb_transaksifull',
'tb_transaksifull.NAMA_OBAT=tb_obat.OBAT')
                ->groupBy('obatkeluar.NAMA')
                ->get()->getResultArray();
        }
        return $this->db->table('tb_obat')
            ->join('satuan', 'satuan.ID=tb_obat.SATUAN')
            ->join('tb_transaksifull',
'tb_transaksifull.NAMA_OBAT=tb_obat.OBAT')
            ->groupBy('tb_transaksifull.TR_KE')
            ->where(['ID_OBAT' => $obat])->get()->getResultArray();
    }
}
```

## Baris Kode Program Menampilkan Detail Prediksi

```
<?= $this->extend('layout/template'); ?>

<?= $this->section('content'); ?>
<?php
ini_set('max_execution_time', 0);
ini_set('memory_limit', '2048M');
```

```

?>

<!-- Begin Page Content -->
<div class="container-fluid">

    <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>

    <!-- Page Heading -->
    <h1 class="h3 mb-2 text-gray-800"><?= $title; ?></h1>

    <!-- Area Chart -->

    <!-- Proses ANFIS -->
    <?php

        //fuzzy c-means
        $jumlah_cluster = 2;
        $bobot = 2;
        $epsilon = 0.01;
        $iterasi = 1000;
        $sum1 = 0;
        $sum2 = 0;
        $sum3 = 0;
        $sum4 = 0;
        $sumku1 = 0;
        $sumku2 = 0;
        $sumku3 = 0;
        $sumku4 = 0;
        $i = 0;
        $jumlah_data_n = 0;
        $jumlah_data_t = 0;
        $ac1 = array();
        $ac2 = array();
        $sumc1k = array();
        $sumc2k = array();
        $sumc1kx1 = array();
        $sumc1kx2 = array();
        $sumc1kx3 = array();
        $sumc2kx1 = array();
        $sumc2kx2 = array();
        $sumc2kx3 = array();
        $x = $transaksi;
        $a = 0;
        $b = 1;
        $c = 2;

```



```

while ($c < count($transaksi)) {
    if ($i % 2 == 0) {
        $c1 = 0.4;
        $c2 = 0.6;
        $c1k = pow($c1, 2);
        $c2k = pow($c2, 2);
        $i++;
    } else {
        $c1 = 0.6;
        $c2 = 0.4;
        $c1k = pow($c1, 2);
        $c2k = pow($c2, 2);
        $i++;
    }
    $c1kx1 = $c1k * $x[$a]['JUMLAH'];
    $c1kx2 = $c1k * $x[$b]['JUMLAH'];
    $c1kx3 = $c1k * $x[$c]['JUMLAH'];
    $c2kx1 = $c2k * $x[$a]['JUMLAH'];
    $c2kx2 = $c2k * $x[$b]['JUMLAH'];
    $c2kx3 = $c2k * $x[$c]['JUMLAH'];
    $ac1[] = $c1;
    $ac2[] = $c2;
    $sumc1k[] = $c1k;
    $sumc2k[] = $c2k;
    $sumc1kx1[] = $c1kx1;
    $sumc1kx2[] = $c1kx2;
    $sumc1kx3[] = $c1kx3;
    $sumc2kx1[] = $c2kx1;
    $sumc2kx2[] = $c2kx2;
    $sumc2kx3[] = $c2kx3;
    $a++;
    $b++;
    $c++;
}
$pusatclu1 = array_sum($sumc1kx1) / array_sum($sumc1k);
$pusatclu2 = array_sum($sumc1kx2) / array_sum($sumc1k);
$pusatclu3 = array_sum($sumc1kx3) / array_sum($sumc1k);
$pusatclu4 = array_sum($sumc2kx1) / array_sum($sumc2k);
$pusatclu5 = array_sum($sumc2kx2) / array_sum($sumc2k);
$pusatclu6 = array_sum($sumc2kx3) / array_sum($sumc2k);
$axv1 = array();
$axv2 = array();
$al1 = array();
$al2 = array();
$asum1 = array();

```

```

$alt1 = array();
$alt2 = array();
$atotallt = array();
$j = 0;
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    $xv1 = (($x[$a]['JUMLAH'] - $pusatclu1) * ($x[$a]['JUMLAH'] -
$pusatclu1)) + (($x[$b]['JUMLAH'] - $pusatclu2) * ($x[$b]['JUMLAH'] -
$pusatclu2)) + (($x[$c]['JUMLAH'] - $pusatclu3) * ($x[$c]['JUMLAH'] -
$pusatclu3));
    $xv2 = (($x[$a]['JUMLAH'] - $pusatclu4) * ($x[$a]['JUMLAH'] -
$pusatclu4)) + (($x[$b]['JUMLAH'] - $pusatclu5) * ($x[$b]['JUMLAH'] -
$pusatclu5)) + (($x[$c]['JUMLAH'] - $pusatclu6) * ($x[$c]['JUMLAH'] -
$pusatclu6));
    $axv1[] = $xv1;
    $axv2[] = $xv2;
    if ($j % 2 == 0) {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $a11[] = $l1;
        $a12[] = $l2;
        $sum1 = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $totallt = $lt1 + $lt2;
        $a11[] = $l1;
        $a12[] = $l2;
        $asuml[] = $sum1;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    } else {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $sum1 = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $totallt = $lt1 + $lt2;
        $a11[] = $l1;
        $a12[] = $l2;
        $asuml[] = $sum1;
        $alt1[] = $lt1;
    }
}

```

```

        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    }
    $a++;
    $b++;
    $c++;
}
$fungsi_objectivea = array_sum($asuml);
$selisih = abs($fungsi_objectivea - 0);
$ulang = 1;

while ($selisih > $epsilon && $ulang < $iterasi) {
    unset($ac1);
    unset($ac2);
    $i = 0;
    $a = 0;
    $b = 1;
    $c = 2;
    while ($c < count($transaksi)) {
        if ($i % 2 == 0) {
            $c1 = $alt1[$i] / $atotallt[$i];
            $c2 = $alt2[$i] / $atotallt[$i];
            $c1k = pow($c1, 2);
            $c2k = pow($c2, 2);
            $i++;
        } else {
            $c1 = $alt1[$i] / $atotallt[$i];
            $c2 = $alt2[$i] / $atotallt[$i];
            $c1k = pow($c1, 2);
            $c2k = pow($c2, 2);
            $i++;
        }
        $c1kx1 = $c1k * $x[$a]['JUMLAH'];
        $c1kx2 = $c1k * $x[$b]['JUMLAH'];
        $c1kx3 = $c1k * $x[$c]['JUMLAH'];
        $c2kx1 = $c2k * $x[$a]['JUMLAH'];
        $c2kx2 = $c2k * $x[$b]['JUMLAH'];
        $c2kx3 = $c2k * $x[$c]['JUMLAH'];
        $ac1[] = $c1;
        $ac2[] = $c2;
        $sumc1k[] = $c1k;
        $sumc2k[] = $c2k;
        $sumc1kx1[] = $c1kx1;
        $sumc1kx2[] = $c1kx2;
    }
}

```

```

        $sumc1kx3[] = $c1kx3;
        $sumc2kx1[] = $c2kx1;
        $sumc2kx2[] = $c2kx2;
        $sumc2kx3[] = $c2kx3;
        $a++;
        $b++;
        $c++;
    }
    $pusatclu1 = array_sum($sumc1kx1) / array_sum($sumc1k);
    $pusatclu2 = array_sum($sumc1kx2) / array_sum($sumc1k);
    $pusatclu3 = array_sum($sumc1kx3) / array_sum($sumc1k);
    $pusatclu4 = array_sum($sumc2kx1) / array_sum($sumc2k);
    $pusatclu5 = array_sum($sumc2kx2) / array_sum($sumc2k);
    $pusatclu6 = array_sum($sumc2kx3) / array_sum($sumc2k);
    $axv1 = array();
    $axv2 = array();
    $al1 = array();
    $al2 = array();
    $asum1 = array();
    $alt1 = array();
    $alt2 = array();
    $atotallt = array();
    $j = 0;
    $a = 0;
    $b = 1;
    $c = 2;
    while ($c < count($transaksi)) {
        $xv1 = (($x[$a]['JUMLAH'] - $pusatclu1) * ($x[$a]['JUMLAH'] -
        $pusatclu1)) + (($x[$b]['JUMLAH'] - $pusatclu2) * ($x[$b]['JUMLAH'] -
        $pusatclu2)) + (($x[$c]['JUMLAH'] - $pusatclu3) * ($x[$c]['JUMLAH'] -
        $pusatclu3));
        $xv2 = (($x[$a]['JUMLAH'] - $pusatclu4) * ($x[$a]['JUMLAH'] -
        $pusatclu4)) + (($x[$b]['JUMLAH'] - $pusatclu5) * ($x[$b]['JUMLAH'] -
        $pusatclu5)) + (($x[$c]['JUMLAH'] - $pusatclu6) * ($x[$c]['JUMLAH'] -
        $pusatclu6));
        $axv1[] = $xv1;
        $axv2[] = $xv2;
        if ($j % 2 == 0) {
            $l1 = $xv1 * $sumc1k[$j];
            $l2 = $xv2 * $sumc2k[$j];
            $al1[] = $l1;
            $al2[] = $l2;
            $sum1 = $l1 + $l2;
            $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
            $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        }
    }

```

```

        $totallt = $lt1 + $lt2;
        $al1[] = $l1;
        $al2[] = $l2;
        $asum1[] = $sum1;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    } else {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $sum1 = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $totallt = $lt1 + $lt2;
        $al1[] = $l1;
        $al2[] = $l2;
        $asum1[] = $sum1;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    }
    $a++;
    $b++;
    $c++;
}
$temp = $fungsi_objectivea;
$fungsi_objectiveb = array_sum($asum1);
$selisih = abs($fungsi_objectiveb - $temp);
$fungsi_objectivea = $fungsi_objectiveb;
$ulang++;
}

```

//Lapisan 1 Generalized Bell

//Pencarian Mean dan Standar Deviasi

```

$sum1 = 0;
$sum2 = 0;
$sum3 = 0;
$sum4 = 0;
$sumku1 = 0;
$sumku2 = 0;
$sumku3 = 0;
$sumku4 = 0;

```

```

$jumlah_data_n = 0;
$jumlah_data_t = 0;
$i = 0;
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    if ($ac1[$i] < $ac2[$i]) {
        $clu = 'turun';
        $sum1 = $sum1 + $x[$a]['JUMLAH'];
        $sum2 = $sum2 + $x[$b]['JUMLAH'];
        $sumku1 = $sumku1 + pow($x[$a]['JUMLAH'], 2);
        $sumku2 = $sumku2 + pow($x[$b]['JUMLAH'], 2);
        $jumlah_data_t = $jumlah_data_t + 1;
        $i++;
    } else {
        $clu = 'naik';
        $sum3 = $sum3 + $x[$a]['JUMLAH'];
        $sum4 = $sum4 + $x[$b]['JUMLAH'];
        $sumku3 = $sumku3 + pow($x[$a]['JUMLAH'], 2);
        $sumku4 = $sumku4 + pow($x[$b]['JUMLAH'], 2);
        $jumlah_data_n = $jumlah_data_n + 1;
        $i++;
    }
    $a++;
    $b++;
    $c++;
}

$mean1 = $sum1 / $jumlah_data_t;
$mean2 = $sum2 / $jumlah_data_t;
$mean3 = $sum3 / $jumlah_data_n;
$mean4 = $sum4 / $jumlah_data_n;

$devi3 = sqrt(($sumku3 - (pow($sum3, 2) / $jumlah_data_n)) / ($jumlah_data_n
- 1));
$devi4 = sqrt(($sumku4 - (pow($sum4, 2) / $jumlah_data_n)) / ($jumlah_data_n
- 1));
$devi1 = sqrt(($sumku1 - (pow($sum1, 2) / $jumlah_data_t)) / ($jumlah_data_t
- 1));
$devi2 = sqrt(($sumku2 - (pow($sum2, 2) / $jumlah_data_t)) / ($jumlah_data_t
- 1));

//Derajat Keanggotaan Generalized Bell
$data_ke = 0;

```

```

$simwb1 = array();
$simwb2 = array();
$simwb1x1 = array();
$simwb2x1 = array();
$simwb1x2 = array();
$simwb2x2 = array();
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    $data_ke++;
    $ma1 = 1 / (1 + (pow(((($x[$a]['JUMLAH'] - $mean3) / $devi3), 2))));
    $ma2 = 1 / (1 + (pow(((($x[$a]['JUMLAH'] - $mean4) / $devi4), 2))));
    $ma3 = 1 / (1 + (pow(((($x[$b]['JUMLAH'] - $mean1) / $devi1), 2))));
    $ma4 = 1 / (1 + (pow(((($x[$b]['JUMLAH'] - $mean2) / $devi2), 2))));
    //Lapisan 2
    $w1 = $ma1 * $ma3;
    $w2 = $ma2 * $ma4;
    //Lapisan 3
    $wb1 = $w1 / ($w1 + $w2);
    $wb2 = $w2 / ($w1 + $w2);
    //Lapisan 4
    $wb1x1 = $wb1 * $x[$a]['JUMLAH'];
    $wb1x2 = $wb1 * $x[$b]['JUMLAH'];
    $wb2x1 = $wb2 * $x[$a]['JUMLAH'];
    $wb2x2 = $wb2 * $x[$b]['JUMLAH'];
    $simwb1[] = $wb1;
    $simwb2[] = $wb2;
    $simwb1x1[] = $wb1x1;
    $simwb2x1[] = $wb2x1;
    $simwb1x2[] = $wb1x2;
    $simwb2x2[] = $wb2x2;
    $a++;
    $b++;
    $c++;
}

```

```
//Estimasi dengan LSE Rekursiif
```

```
// Fungsi perkalian matriks
```

```
function perkalian_matriks2($matriks_a, $matriks_b)
{
    $hasil = array();
    for ($i = 0; $i < sizeof($matriks_a); $i++) {
        for ($j = 0; $j < sizeof($matriks_b[0]); $j++) {

```

```

        $temp = 0;
        for ($k = 0; $k < sizeof($matriks_b); $k++) {
            $temp += $matriks_a[$i][$k] * $matriks_b[$k][$j];
        }
        $hasil[$i][$j] = $temp;
    }
}
return $hasil;
}

// Matriks A
$a = array();
$u = 0;
for ($i = 2; $i < sizeof($transaksi); $i++) {
    $a[] = array($simwb1x1[$u], $simwb1x2[$u], $simwb1[$u], $simwb2x1[$u],
$simwb2x2[$u], $simwb2[$u]);
    $u++;
}

// Matriks AT
$b = array();
$b[] = $simwb1x1;
$b[] = $simwb1x2;
$b[] = $simwb1;
$b[] = $simwb2x1;
$b[] = $simwb2x2;
$b[] = $simwb2;

//Matriks Target
$c = array();
for ($i = 2; $i < sizeof($transaksi); $i++) {
    $c[] = array($x[$i]['JUMLAH']);
}

// Kalikan
$hasil1 = perkalian_matriks2($b, $a);

// Inverts Matriks
function invert($A, $debug = FALSE)
{
    // @todo check rows = columns
    $n = count($A);

    // get and append identity matrix
    $I = identity_matrix($n);

```



```

for ($i = 0; $i < $n; ++$i) {
    $A[$i] = array_merge($A[$i], $I[$i]);
}

if ($debug) {
    echo "\nStarting matrix: ";
    print_matrix($A);
}

// forward run
for ($j = 0; $j < $n - 1; ++$j) {
    // for all remaining rows (diagonally)
    for ($i = $j + 1; $i < $n; ++$i) {
        // if the value is not already 0
        if ($A[$i][$j] !== 0) {
            // adjust scale to pivot row
            // subtract pivot row from current
            $scalar = $A[$j][$j] / $A[$i][$j];
            for ($jj = $j; $jj < $n * 2; ++$jj) {
                $A[$i][$jj] *= $scalar;
                $A[$i][$jj] -= $A[$j][$jj];
            }
        }
    }
    if ($debug) {
        echo "\nForward iteration $j: ";
        print_matrix($A);
    }
}

// reverse run
for ($j = $n - 1; $j > 0; --$j) {
    for ($i = $j - 1; $i >= 0; --$i) {
        if ($A[$i][$j] !== 0) {
            $scalar = $A[$j][$j] / $A[$i][$j];
            for ($jj = $i; $jj < $n * 2; ++$jj) {
                $A[$i][$jj] *= $scalar;
                $A[$i][$jj] -= $A[$j][$jj];
            }
        }
    }
    if ($debug) {
        echo "\nReverse iteration $j: ";
        print_matrix($A);
    }
}

```

```

}

// last run to make all diagonal 1s
// @note this can be done in last iteration (i.e. reverse run) too!
for ($j = 0; $j < $n; ++$j) {
    if ($A[$j][$j] != 1) {
        $scalar = 1 / $A[$j][$j];
        for ($jj = $j; $jj < $n * 2; ++$jj) {
            $A[$j][$jj] *= $scalar;
        }
    }
    if ($debug) {
        echo "\n1-out iteration $j: ";
        print_matrix($A);
    }
}

// take out the matrix inverse to return
$Inv = array();
for ($i = 0; $i < $n; ++$i) {
    $Inv[$i] = array_slice($A[$i], $n);
}

return $Inv;
}

function print_matrix($A, $decimals = 3)
{
    echo "<br><table border='1' cellspacing='0' cellpadding='5'>";
    foreach ($A as $row) {
        echo "<tr>";
        foreach ($row as $i) {
            echo "<td>" . sprintf("%01.{ $decimals}f", round($i, $decimals))
. "</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}

function identity_matrix($n)
{
    $I = array();
    for ($i = 0; $i < $n; ++$i) {
        for ($j = 0; $j < $n; ++$j) {

```

```

        $I[$i][$j] = ($i == $j) ? 1 : 0;
    }
}
return $I;
}

$A = $hasil1;

$B = invert($A);

$hasil2 = perkalian_matriks2($b, $c);

$hasil3 = perkalian_matriks2($B, $hasil2);

$p1 = $hasil3[0][0];
$q1 = $hasil3[1][0];
$r1 = $hasil3[2][0];
$p2 = $hasil3[3][0];
$q2 = $hasil3[4][0];
$r2 = $hasil3[5][0];

//Lapisan 5

$data_ke = 0;
$summape = array();
$summse = array();
$apre = array();
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    $data_ke++;
    $ma1 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean3) / $devi3), 2)));
    $ma2 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean4) / $devi4), 2)));
    $ma3 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean1) / $devi1), 2)));
    $ma4 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean2) / $devi2), 2)));
    //Lapisan 2
    $w1 = $ma1 * $ma3;
    $w2 = $ma2 * $ma4;
    //Lapisan 3
    $wb1 = $w1 / ($w1 + $w2);
    $wb2 = $w2 / ($w1 + $w2);
    //Lapisan 4
    $wb1x1 = $wb1 * x[$a]['JUMLAH'];
    $wb1x2 = $wb1 * x[$b]['JUMLAH'];
}

```

```

    $wb2x1 = $wb2 * $x[$a]['JUMLAH'];
    $wb2x2 = $wb2 * $x[$b]['JUMLAH'];
    $pre = ($wb1x1 * $p1) + ($wb1x2 * $q1) + ($wb1 * $r1) + ($wb2x1 * $p2) +
($wb2x2 * $q2) + ($wb2 * $r2);
    $error = $x[$c]['JUMLAH'] - $pre;
    $summape[] = abs($error) / $x[$c]['JUMLAH'];
    //$summse[] = pow($error, 2);
    $apre[] = $pre;
    $a++;
    $b++;
    $c++;
}
?>

<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary"><?=$x[0]['NAMA_OBAT']; ?></h6>
        <h6 class="m-0 font-weight-bold text-primary">Chart Perbandingan Data Asli dan Data Prediksi</h6>
    </div>
    <div class="card-body">
        <div class="chart-area">
            <canvas id="myChart2"></canvas>
            <script>
                // Set new default font family and font color to mimic
Bootstrap's default styling
                Chart.defaults.global.defaultFontFamily = 'Nunito', '-apple-system,system-ui,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,sans-serif';
                Chart.defaults.global.defaultFontColor = '#858796';

                function number_format(number, decimals, dec_point,
thousands_sep) {
                    // *      example: number_format(1234.56, 2, ',', ' ');
                    // *      return: '1 234,56'
                    number = (number + '').replace(',', '').replace(' ',
''');
                    var n = !isFinite(+number) ? 0 : +number,
                        prec = !isFinite(+decimals) ? 0 :
Math.abs(decimals),
                        sep = (typeof thousands_sep === 'undefined') ? ',' :
thousands_sep,
                        dec = (typeof dec_point === 'undefined') ? '.' :
dec_point,

```

```

        s = '',
        toFixedFix = function(n, prec) {
            var k = Math.pow(10, prec);
            return '' + Math.round(n * k) / k;
        };
        // Fix for IE parseFloat(0.55).toFixed(0) = 0;
        s = (prec ? toFixedFix(n, prec) : '' +
Math.round(n)).split('.');
        if (s[0].length > 3) {
            s[0] = s[0].replace(/\B(?=(?:\d{3})+(?! \d))/g, sep);
        }
        if ((s[1] || '').length < prec) {
            s[1] = s[1] || '';
            s[1] += new Array(prec - s[1].length + 1).join('0');
        }
        return s.join(dec);
    }

// Area Chart Example
var ctx = document.getElementById("myChart2");
var myLineChart = new Chart(ctx, {
    type: 'line',
    data: {
        labels: [
            <?php
            $p = 0;
            if (count($transaksi) > 0) {
                for ($u = 2; $u < count($transaksi); $u++) {
                    $p++;
                    echo "" . $p . ", ";
                }
            }
            ?>
        ],
        datasets: [{
            label: "Transaksi",
            lineTension: 0.3,
            backgroundColor: "rgba(78, 115, 223, 0.05)",
            borderColor: "rgba(78, 115, 223, 1)",
            pointRadius: 3,
            pointBackgroundColor: "rgba(78, 115, 223, 1)",
            pointBorderColor: "rgba(78, 115, 223, 1)",
            pointHoverRadius: 3,
            pointHoverBackgroundColor: "rgba(78, 115, 223,
1)",

```

```

        pointHoverBorderColor: "rgba(78, 115, 223, 1)",
        pointHitRadius: 10,
        pointBorderWidth: 2,
        data: [
            <?php
            if (count($transaksi) > 0) {
                for ($u = 2; $u < count($transaksi);
                    $u++) {
                        echo "' . $x[$u]['JUMLAH'] . ','";
                    }
                }
            ?>
        ],
    }, {
        label: "Prediksi",
        lineTension: 0.3,
        backgroundColor: "rgba(78, 115, 223, 0.05)",
        borderColor: "orange",
        pointRadius: 3,
        pointBackgroundColor: "orange",
        pointBorderColor: "orange",
        pointHoverRadius: 3,
        pointHoverBackgroundColor: "orange",
        pointHoverBorderColor: "orange",
        pointHitRadius: 10,
        pointBorderWidth: 2,
        data: [
            <?php
            if (count($transaksi) > 0) {
                $q = 0;
                for ($u = 2; $u < count($transaksi);
                    $u++) {
                        echo "' . round($apre[$q], 3) .
                            ','";
                        $q++;
                    }
                }
            ?>
        ],
    }
    ]],
},
options: {
    maintainAspectRatio: false,
    layout: {
        padding: {

```

```

        left: 10,
        right: 25,
        top: 25,
        bottom: 0
    }
},
scales: {
    xAxes: [{
        time: {
            unit: 'date'
        },
        gridLines: {
            display: false,
            drawBorder: false
        },
        ticks: {
            maxTicksLimit: 20
        }
    }],
    yAxes: [{
        ticks: {
            maxTicksLimit: 10,
            padding: 10,
            // Include a dollar sign in the ticks
            callback: function(value, index, values)
                return number_format(value);
        }
    },
    gridLines: {
        color: "rgb(234, 236, 244)",
        zeroLineColor: "rgb(234, 236, 244)",
        drawBorder: false,
        borderDash: [2],
        zeroLineBorderDash: [2]
    }
    }],
},
legend: {
    display: false
},
tooltips: {
    backgroundColor: "rgb(255,255,255)",
    bodyFontColor: "#858796",
    titleMarginBottom: 10,

```

```

        titleFontColor: '#6e707e',
        titleFontSize: 14,
        borderColor: '#dddfeb',
        borderWidth: 1,
        xPadding: 15,
        yPadding: 15,
        displayColors: false,
        intersect: false,
        mode: 'index',
        caretPadding: 10,
        callbacks: {
            label: function(tooltipItem, chart) {
                var datasetLabel =
chart.datasets[tooltipItem.datasetIndex].label || '';
                return datasetLabel + ': ' +
number_format(tooltipItem.yLabel);
            }
        }
    });
</script>
</div>
<?php
//prediksi bulan selanjutnya
$ma1 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean3) / $devi3), 2)));
$ma2 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean4) / $devi4), 2)));
$ma3 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean1) / $devi1), 2)));
$ma4 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean2) / $devi2), 2)));
//Lapisan 2
$w1 = $ma1 * $ma3;
$w2 = $ma2 * $ma4;
//Lapisan 3
$wb1 = $w1 / ($w1 + $w2);
$wb2 = $w2 / ($w1 + $w2);
//Lapisan 4
$wb1x1 = $wb1 * x[$a]['JUMLAH'];
$wb1x2 = $wb1 * x[$b]['JUMLAH'];
$wb2x1 = $wb2 * x[$a]['JUMLAH'];
$wb2x2 = $wb2 * x[$b]['JUMLAH'];
$pre = ($wb1x1 * $p1) + ($wb1x2 * $q1) + ($wb1 * $r1) + ($wb2x1 *
$p2) + ($wb2x2 * $q2) + ($wb2 * $r2);
?>
MAPE : <?= round((array_sum($summape) / $data_ke) * 100, 3); ?>%

```



```

        <br>Prediksi Bulan Selanjutnya : <?= round($pre, 0); ?> <?=
    $x[$b]['DESKRIPSI']; ?>
    </div>
</div>

<!-- /.container-fluid -->

<?= $this->endSection(); ?>

```

## Baris Kode Program Menampilkan Contoh Perhitungan

```

<?= $this->extend('layout/template'); ?>

<?= $this->section('content'); ?>
<?php
ini_set('max_execution_time', 0);
ini_set('memory_limit', '2048M');
?>
<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <h1 class="h3 mb-2 text-gray-800">Contoh Perhitungan</h1>

    <!-- DataTales Example -->
    <div class="card shadow mb-4">
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered" width="100%"
cellspacing="0">
                    <thead>
                        <tr>
                            <th>X1</th>
                            <th>X2</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php
//fuzzy c-means
$jumlah_cluster = 2;
$bobot = 2;
$epsilon = 0.01;
$iterasi = 1000;

```

```

$sum1 = 0;
$sum2 = 0;
$sum3 = 0;
$sum4 = 0;
$sumku1 = 0;
$sumku2 = 0;
$sumku3 = 0;
$sumku4 = 0;
$i = 0;
$jumlah_data_n = 0;
$jumlah_data_t = 0;
$ac1 = array();
$ac2 = array();
$sumc1k = array();
$sumc2k = array();
$sumc1kx1 = array();
$sumc1kx2 = array();
$sumc1kx3 = array();
$sumc2kx1 = array();
$sumc2kx2 = array();
$sumc2kx3 = array();
$x = $transaksi;
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    if ($i % 2 == 0) {
        $c1 = 0.4;
        $c2 = 0.6;
        $c1k = pow($c1, 2);
        $c2k = pow($c2, 2);
        $i++;
    } else {
        $c1 = 0.6;
        $c2 = 0.4;
        $c1k = pow($c1, 2);
        $c2k = pow($c2, 2);
        $i++;
    }
    $c1kx1 = $c1k * $x[$a]['JUMLAH'];
    $c1kx2 = $c1k * $x[$b]['JUMLAH'];
    $c1kx3 = $c1k * $x[$c]['JUMLAH'];
    $c2kx1 = $c2k * $x[$a]['JUMLAH'];
    $c2kx2 = $c2k * $x[$b]['JUMLAH'];
    $c2kx3 = $c2k * $x[$c]['JUMLAH'];
}

```

```

        $ac1[] = $c1;
        $ac2[] = $c2;
        $sumc1k[] = $c1k;
        $sumc2k[] = $c2k;
        $sumc1kx1[] = $c1kx1;
        $sumc1kx2[] = $c1kx2;
        $sumc1kx3[] = $c1kx3;
        $sumc2kx1[] = $c2kx1;
        $sumc2kx2[] = $c2kx2;
        $sumc2kx3[] = $c2kx3;
        $a++;
        $b++;
        $c++;
    }
    $pusatclu1 = array_sum($sumc1kx1) / array_sum($sumc1k);
    $pusatclu2 = array_sum($sumc1kx2) / array_sum($sumc1k);
    $pusatclu3 = array_sum($sumc1kx3) / array_sum($sumc1k);
    $pusatclu4 = array_sum($sumc2kx1) / array_sum($sumc2k);
    $pusatclu5 = array_sum($sumc2kx2) / array_sum($sumc2k);
    $pusatclu6 = array_sum($sumc2kx3) / array_sum($sumc2k);
    $axv1 = array();
    $axv2 = array();
    $al1 = array();
    $al2 = array();
    $asuml = array();
    $alt1 = array();
    $alt2 = array();
    $atotallt = array();
    $j = 0;
    $a = 0;
    $b = 1;
    $c = 2;
    while ($c < count($transaksi)) {
        $xv1 = (($x[$a]['JUMLAH'] - $pusatclu1) *
($x[$a]['JUMLAH'] - $pusatclu1)) + (($x[$b]['JUMLAH'] - $pusatclu2) *
($x[$b]['JUMLAH'] - $pusatclu2)) + (($x[$c]['JUMLAH'] - $pusatclu3) *
($x[$c]['JUMLAH'] - $pusatclu3));
        $xv2 = (($x[$a]['JUMLAH'] - $pusatclu4) *
($x[$a]['JUMLAH'] - $pusatclu4)) + (($x[$b]['JUMLAH'] - $pusatclu5) *
($x[$b]['JUMLAH'] - $pusatclu5)) + (($x[$c]['JUMLAH'] - $pusatclu6) *
($x[$c]['JUMLAH'] - $pusatclu6));
        $axv1[] = $xv1;
        $axv2[] = $xv2;
        if ($j % 2 == 0) {
            $l1 = $xv1 * $sumc1k[$j];

```

```

        $l2 = $xv2 * $sumc2k[$j];
        $al1[] = $l1;
        $al2[] = $l2;
        $suml = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $totallt = $lt1 + $lt2;
        $al1[] = $l1;
        $al2[] = $l2;
        $asuml[] = $suml;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    } else {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $suml = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $totallt = $lt1 + $lt2;
        $al1[] = $l1;
        $al2[] = $l2;
        $asuml[] = $suml;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    }
}
$a++;
$b++;
$c++;
}
$fungsi_objectivea = array_sum($asuml);
$selisih = abs($fungsi_objectivea - 0);
$ulang = 1;

while ($selisih > $epsilon && $ulang < $iterasi) {
    unset($ac1);
    unset($ac2);
    $i = 0;
    $a = 0;
    $b = 1;
    $c = 2;
    while ($c < count($transaksi)) {

```

```

if ($i % 2 == 0) {
    $c1 = $alt1[$i] / $atotallt[$i];
    $c2 = $alt2[$i] / $atotallt[$i];
    $c1k = pow($c1, 2);
    $c2k = pow($c2, 2);
    $i++;
} else {
    $c1 = $alt1[$i] / $atotallt[$i];
    $c2 = $alt2[$i] / $atotallt[$i];
    $c1k = pow($c1, 2);
    $c2k = pow($c2, 2);
    $i++;
}
}
$c1kx1 = $c1k * $x[$a]['JUMLAH'];
$c1kx2 = $c1k * $x[$b]['JUMLAH'];
$c1kx3 = $c1k * $x[$c]['JUMLAH'];
$c2kx1 = $c2k * $x[$a]['JUMLAH'];
$c2kx2 = $c2k * $x[$b]['JUMLAH'];
$c2kx3 = $c2k * $x[$c]['JUMLAH'];
$ac1[] = $c1;
$ac2[] = $c2;
$sumc1k[] = $c1k;
$sumc2k[] = $c2k;
$sumc1kx1[] = $c1kx1;
$sumc1kx2[] = $c1kx2;
$sumc1kx3[] = $c1kx3;
$sumc2kx1[] = $c2kx1;
$sumc2kx2[] = $c2kx2;
$sumc2kx3[] = $c2kx3;
$a++;
$b++;
$c++;
}
$pusatclu1 = array_sum($sumc1kx1) /
array_sum($sumc1k);
$pusatclu2 = array_sum($sumc1kx2) /
array_sum($sumc1k);
$pusatclu3 = array_sum($sumc1kx3) /
array_sum($sumc1k);
$pusatclu4 = array_sum($sumc2kx1) /
array_sum($sumc2k);
$pusatclu5 = array_sum($sumc2kx2) /
array_sum($sumc2k);
$pusatclu6 = array_sum($sumc2kx3) /
array_sum($sumc2k);

```

```

$axv1 = array();
$axv2 = array();
$a11 = array();
$a12 = array();
$asum1 = array();
$alt1 = array();
$alt2 = array();
$atotal1t = array();
$j = 0;
$a = 0;
$b = 1;
$c = 2;
while ($c < count($transaksi)) {
    $xv1 = (($x[$a]['JUMLAH'] - $pusatclu1) *
($x[$a]['JUMLAH'] - $pusatclu1)) + (($x[$b]['JUMLAH'] - $pusatclu2) *
($x[$b]['JUMLAH'] - $pusatclu2)) + (($x[$c]['JUMLAH'] - $pusatclu3) *
($x[$c]['JUMLAH'] - $pusatclu3));
    $xv2 = (($x[$a]['JUMLAH'] - $pusatclu4) *
($x[$a]['JUMLAH'] - $pusatclu4)) + (($x[$b]['JUMLAH'] - $pusatclu5) *
($x[$b]['JUMLAH'] - $pusatclu5)) + (($x[$c]['JUMLAH'] - $pusatclu6) *
($x[$c]['JUMLAH'] - $pusatclu6));
    $axv1[] = $xv1;
    $axv2[] = $xv2;
    if ($j % 2 == 0) {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $a11[] = $l1;
        $a12[] = $l2;
        $sum1 = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
        $total1t = $lt1 + $lt2;
        $a11[] = $l1;
        $a12[] = $l2;
        $asum1[] = $sum1;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotal1t[] = $total1t;
        $j++;
    } else {
        $l1 = $xv1 * $sumc1k[$j];
        $l2 = $xv2 * $sumc2k[$j];
        $sum1 = $l1 + $l2;
        $lt1 = pow($axv1[$j], -1 / ($bobot - 1));
        $lt2 = pow($axv2[$j], -1 / ($bobot - 1));
    }
}

```

```

        $totallt = $lt1 + $lt2;
        $al1[] = $l1;
        $al2[] = $l2;
        $asuml[] = $sum1;
        $alt1[] = $lt1;
        $alt2[] = $lt2;
        $atotallt[] = $totallt;
        $j++;
    }
    $a++;
    $b++;
    $c++;
}
$temp = $fungsi_objectivea;
$fungsi_objectiveb = array_sum($asuml);
$selisih = abs($fungsi_objectiveb - $temp);
$fungsi_objectivea = $fungsi_objectiveb;
$ulang++;
}
?>
<?php
$k = 0;
for ($c = 2; $c < count($transaksi); $c++) {
?>
    <tr>
        <td><?= $ac1[$k]; ?></td>
        <td><?= $ac2[$k]; ?></td>
    </tr>
<?php
    $k++;
}
?>
</tbody>
</table>
Selisih Fungsi Objective: <?= $selisih; ?><br><br>
Hasil Pembagian Cluster dengan Fuzzy C-Means
<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>X1</th>
            <th>X2</th>
            <th>X Target</th>
            <th>Cluster</th>
        </tr>

```

```

</thead>
<tbody>
  <?php
    $sum1 = 0;
    $sum2 = 0;
    $sum3 = 0;
    $sum4 = 0;
    $sumku1 = 0;
    $sumku2 = 0;
    $sumku3 = 0;
    $sumku4 = 0;
    $jumlah_data_n = 0;
    $jumlah_data_t = 0;
    $i = 0;
    $a = 0;
    $b = 1;
    $c = 2;
    while ($c < count($transaksi)) {
      if ($ac1[$i] < $ac2[$i]) {
        $clu = 'turun';
        $sum1 = $sum1 + $x[$a]['JUMLAH'];
        $sum2 = $sum2 + $x[$b]['JUMLAH'];
        $sumku1 = $sumku1 + pow($x[$a]['JUMLAH'], 2);
        $sumku2 = $sumku2 + pow($x[$b]['JUMLAH'], 2);
        $jumlah_data_t = $jumlah_data_t + 1;
        $i++;
      } else {
        $clu = 'naik';
        $sum3 = $sum3 + $x[$a]['JUMLAH'];
        $sum4 = $sum4 + $x[$b]['JUMLAH'];
        $sumku3 = $sumku3 + pow($x[$a]['JUMLAH'], 2);
        $sumku4 = $sumku4 + pow($x[$b]['JUMLAH'], 2);
        $jumlah_data_n = $jumlah_data_n + 1;
        $i++;
      }
    }
  >
  <tr>
    <td><?php echo $x[$a]['JUMLAH']; ?></td>
    <td><?php echo $x[$b]['JUMLAH']; ?></td>
    <td><?php echo $x[$c]['JUMLAH']; ?></td>
    <td><?php echo $clu; ?></td>
  </tr>
  <?php
    $a++;
    $b++;

```



```

        $c++;
    } ?>
</tbody>
</table>
<p>Mean dan Standar Deviasi</p>
<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>Ket</th>
            <th colspan="2">Naik</th>
            <th colspan="2">Turun</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>-</th>
            <th>x1</th>
            <th>x2</th>
            <th>x1</th>
            <th>x2</th>
        </tr>
        <tr>
            <th>Mean</th>
            <?php
                $mean1 = $sum1 / $jumlah_data_t;
                $mean2 = $sum2 / $jumlah_data_t;
                $mean3 = $sum3 / $jumlah_data_n;
                $mean4 = $sum4 / $jumlah_data_n;
            ?>
            <td><?= round($mean3, 3); ?></td>
            <td><?= round($mean4, 3); ?></td>
            <td><?= round($mean2, 3); ?></td>
            <td><?= round($mean1, 3); ?></td>
        </tr>
        <tr>
            <?php
                $devi3 = sqrt(($sumku3 - (pow($sum3, 2) /
$jumlah_data_n)) / ($jumlah_data_n - 1));
                $devi4 = sqrt(($sumku4 - (pow($sum4, 2) /
$jumlah_data_n)) / ($jumlah_data_n - 1));
                $devi1 = sqrt(($sumku1 - (pow($sum1, 2) /
$jumlah_data_t)) / ($jumlah_data_t - 1));
                $devi2 = sqrt(($sumku2 - (pow($sum2, 2) /
$jumlah_data_t)) / ($jumlah_data_t - 1));

```

```

        ?>
        <th>Standar Deviasi</th>
        <td><?= round($devi3, 3); ?></td>
        <td><?= round($devi4, 3); ?></td>
        <td><?= round($devi1, 3); ?></td>
        <td><?= round($devi2, 3); ?></td>
    </tr>
</tbody>
</table>
<p>Lapisan 1 : Fuzzifikasi menggunakan derajat keanggotaan
Generalized Bell</p>
<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>Data Ke</th>
            <th colspan="4">Output Lapisan 1</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>-</th>
            <th>x1</th>
            <th>x2</th>
            <th>x1</th>
            <th>x2</th>
        </tr>
        <?php
        $data_ke = 0;
        $a = 0;
        $b = 1;
        $c = 2;
        while ($c < count($transaksi)) {
            $data_ke++;
            $ma1 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean3) /
$devi3), 2)));
            $ma2 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean4) /
$devi4), 2)));
            $ma3 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean1) /
$devi1), 2)));
            $ma4 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean2) /
$devi2), 2)));
        ?>
        <tr>
            <th><?= $data_ke; ?></th>

```

```

        <td><?= round($ma1, 3); ?></td>
        <td><?= round($ma2, 3); ?></td>
        <td><?= round($ma3, 3); ?></td>
        <td><?= round($ma4, 3); ?></td>
    </tr>
</tbody>
</table>
<?php
    $a++;
    $b++;
    $c++;
} ?>
</table>
<p>Lapisan 2 : Menentukan neuron tetap dengan perhitungan Firing
Strength</p>
<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>Data Ke</th>
            <th colspan="2">Output Lapisan 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>-</th>
            <th>w1</th>
            <th>w2</th>
        </tr>
        <?php
            $data_ke = 0;
            $a = 0;
            $b = 1;
            $c = 2;
            while ($c < count($transaksi)) {
                $data_ke++;
                $ma1 = 1 / (1 + (pow(((($x[$a]['JUMLAH'] - $mean3) /
$devi3), 2))));
                $ma2 = 1 / (1 + (pow(((($x[$a]['JUMLAH'] - $mean4) /
$devi4), 2))));
                $ma3 = 1 / (1 + (pow(((($x[$b]['JUMLAH'] - $mean1) /
$devi1), 2))));
                $ma4 = 1 / (1 + (pow(((($x[$b]['JUMLAH'] - $mean2) /
$devi2), 2))));
                $w1 = $ma1 * $ma3;
                $w2 = $ma2 * $ma4;
            }
        </?php
    </tbody>
</table>

```

```

?>
    <tr>
        <th><?= $data_ke; ?></th>
        <td><?= round($w1, 3); ?></td>
        <td><?= round($w2, 3); ?></td>
    </tr>
<?php
    $a++;
    $b++;
    $c++;
} ?>
</tbody>
</table>
<p>Lapisan 3 : Menentukan neuron tetap dari rasio Firing
Strength ke-i terhadap jumlah dari keseluruhan Firing Strength (Normalized
Firing Strength)</p>
<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>Data Ke</th>
            <th colspan="2">Output Lapisan 3</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>-</th>
            <th>w1</th>
            <th>w2</th>
        </tr>
        <?php
        $data_ke = 0;
        $a = 0;
        $b = 1;
        $c = 2;
        while ($c < count($transaksi)) {
            $data_ke++;
            $ma1 = 1 / (1 + (pow((((x[$a]['JUMLAH'] - $mean3) /
$devi3), 2))));
            $ma2 = 1 / (1 + (pow((((x[$a]['JUMLAH'] - $mean4) /
$devi4), 2))));
            $ma3 = 1 / (1 + (pow((((x[$b]['JUMLAH'] - $mean1) /
$devi1), 2))));
            $ma4 = 1 / (1 + (pow((((x[$b]['JUMLAH'] - $mean2) /
$devi2), 2))));

```

```

        $w1 = $ma1 * $ma3;
        $w2 = $ma2 * $ma4;
        $wb1 = $w1 / ($w1 + $w2);
        $wb2 = $w2 / ($w1 + $w2);
    ?>
    <tr>
        <th><?= $data_ke; ?></th>
        <td><?= round($wb1, 3); ?></td>
        <td><?= round($wb2, 3); ?></td>
    </tr>
<?php
    $a++;
    $b++;
    $c++;
} ?>
</tbody>
</table>
<p>Lapisan 4 : Menentukan neuron yang adaktif terhadap

```

output</p>

```

<table class="table table-bordered" width="100%"
cellspacing="0">

```

```

    <thead>
        <tr>
            <th>Data Ke</th>
            <th>w1*x1</th>
            <th>w1*x2</th>
            <th>w1</th>
            <th>w2*x1</th>
            <th>w2*x2</th>
            <th>w2</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $data_ke = 0;
            $simwb1 = array();
            $simwb2 = array();
            $simwb1x1 = array();
            $simwb2x1 = array();
            $simwb1x2 = array();
            $simwb2x2 = array();
            $a = 0;
            $b = 1;
            $c = 2;
            while ($c < count($transaksi)) {

```

```

    $data_ke++;
    $ma1 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean3) /
$devi3), 2)));
    $ma2 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean4) /
$devi4), 2)));
    $ma3 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean1) /
$devi1), 2)));
    $ma4 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean2) /
$devi2), 2)));

    $w1 = $ma1 * $ma3;
    $w2 = $ma2 * $ma4;
    $wb1 = $w1 / ($w1 + $w2);
    $wb2 = $w2 / ($w1 + $w2);
    $wb1x1 = $wb1 * x[$a]['JUMLAH'];
    $wb1x2 = $wb1 * x[$b]['JUMLAH'];
    $wb2x1 = $wb2 * x[$a]['JUMLAH'];
    $wb2x2 = $wb2 * x[$b]['JUMLAH'];
    $simwb1[] = $wb1;
    $simwb2[] = $wb2;
    $simwb1x1[] = $wb1x1;
    $simwb2x1[] = $wb2x1;
    $simwb1x2[] = $wb1x2;
    $simwb2x2[] = $wb2x2;
?>
<tr>
    <th><?=$data_ke; ?></th>
    <td><?= round($wb1x1, 3); ?></td>
    <td><?= round($wb1x2, 3); ?></td>
    <td><?= round($wb1, 3); ?></td>
    <td><?= round($wb2x1, 3); ?></td>
    <td><?= round($wb2x2, 3); ?></td>
    <td><?= round($wb2, 3); ?></td>
</tr>
<?php
    $a++;
    $b++;
    $c++;
} ?>
</tbody>
</table>
<p>Matrik</p>
<?php
// Fungsi perkalian matriks
function perkalian_matriks2($matriks_a, $matriks_b)
{

```

```

        $hasil = array();
        for ($i = 0; $i < sizeof($matriks_a); $i++) {
            for ($j = 0; $j < sizeof($matriks_b[0]); $j++) {
                $temp = 0;
                for ($k = 0; $k < sizeof($matriks_b); $k++) {
                    $temp += $matriks_a[$i][$k] *
$matriks_b[$k][$j];
                }
                $hasil[$i][$j] = $temp;
            }
        }
        return $hasil;
    }

//-----
-----

// Contoh penggunaan :
// Matriks A
$a = array();
$u = 0;
for ($i = 2; $i < sizeof($transaksi); $i++) {
    $a[] = array($simwb1x1[$u], $simwb1x2[$u], $simwb1[$u],
$simwb2x1[$u], $simwb2x2[$u], $simwb2[$u]);
    $u++;
}

// Matriks AT
$b = array();
$b[] = $simwb1x1;
$b[] = $simwb1x2;
$b[] = $simwb1;
$b[] = $simwb2x1;
$b[] = $simwb2x2;
$b[] = $simwb2;

//Matriks Target
$c = array();
for ($i = 2; $i < sizeof($transaksi); $i++) {
    $c[] = array($x[$i]['JUMLAH']);
}

// Kalikan
$hasil1 = perkalian_matriks2($b, $a);
echo "<p>(A<sup>T</sup> x A)</p><table border='1'
cellspacing='0' cellpadding='5'>";

```

```

for ($i = 0; $i < sizeof($hasil1); $i++) {
    echo "<tr>";
    for ($j = 0; $j < sizeof($hasil1[$i]); $j++) {
        echo "<td>" . round($hasil1[$i][$j], 3) . "</td>";
    }
    echo "</tr>";
}
echo "</table>";

// Inverts Matriks
function invert($A, $debug = FALSE)
{
    /// @todo check rows = columns

    $n = count($A);

    // get and append identity matrix
    $I = identity_matrix($n);
    for ($i = 0; $i < $n; ++$i) {
        $A[$i] = array_merge($A[$i], $I[$i]);
    }

    if ($debug) {
        echo "\nStarting matrix: ";
        print_matrix($A);
    }

    // forward run
    for ($j = 0; $j < $n - 1; ++$j) {
        // for all remaining rows (diagonally)
        for ($i = $j + 1; $i < $n; ++$i) {
            // if the value is not already 0
            if ($A[$i][$j] !== 0) {
                // adjust scale to pivot row
                // subtract pivot row from current
                $scalar = $A[$j][$j] / $A[$i][$j];
                for ($jj = $j; $jj < $n * 2; ++$jj) {
                    $A[$i][$jj] *= $scalar;
                    $A[$i][$jj] -= $A[$j][$jj];
                }
            }
        }
    }
    if ($debug) {
        echo "\nForward iteration $j: ";
        print_matrix($A);
    }
}

```



```

    }
}

// reverse run
for ($j = $n - 1; $j > 0; --$j) {
    for ($i = $j - 1; $i >= 0; --$i) {
        if ($A[$i][$j] != 0) {
            $scalar = $A[$j][$j] / $A[$i][$j];
            for ($jj = $i; $jj < $n * 2; ++$jj) {
                $A[$i][$jj] *= $scalar;
                $A[$i][$jj] -= $A[$j][$jj];
            }
        }
    }
    if ($debug) {
        echo "\nReverse iteration $j: ";
        print_matrix($A);
    }
}

// last run to make all diagonal 1s
/// @note this can be done in last iteration (i.e. reverse
run) too!

for ($j = 0; $j < $n; ++$j) {
    if ($A[$j][$j] != 1) {
        $scalar = 1 / $A[$j][$j];
        for ($jj = $j; $jj < $n * 2; ++$jj) {
            $A[$j][$jj] *= $scalar;
        }
    }
    if ($debug) {
        echo "\n1-out iteration $j: ";
        print_matrix($A);
    }
}

// take out the matrix inverse to return
$Inv = array();
for ($i = 0; $i < $n; ++$i) {
    $Inv[$i] = array_slice($A[$i], $n);
}

return $Inv;
}

```

```

/**
 * Prints matrix
 *
 * @param array $A matrix
 * @param integer $decimals number of decimals
 */
function print_matrix($A, $decimals = 3)
{
    echo "<table border='1' cellspacing='0' cellpadding='5'>";
    foreach ($A as $row) {
        echo "<tr>";
        foreach ($row as $i) {
            echo "<td>" . sprintf("%01.{ $decimals}f", round($i,
$decimals)) . "</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}

/**
 * Produces an identity matrix of given size
 *
 * @param integer $n size of identity matrix
 *
 * @return array identity matrix
 */
function identity_matrix($n)
{
    $I = array();
    for ($i = 0; $i < $n; ++$i) {
        for ($j = 0; $j < $n; ++$j) {
            $I[$i][$j] = ($i == $j) ? 1 : 0;
        }
    }
    return $I;
}

$A = $hasil1;

// echo "<br>Matrix:";
// print_matrix($A);
// echo "\n";

$B = invert($A);

```

```

echo "<br><p>(A<sup>T</sup> x A)<sup>-1</sup></p>";
print_matrix($B);
echo "\n\n";

$hasil2 = perkalian_matriks2($b, $c);
echo "<br><p>(A<sup>T</sup> x Target)</p><table border='1'
cellspacing='0' cellpadding='5'>";
for ($i = 0; $i < sizeof($hasil2); $i++) {
    echo "<tr>";
    for ($j = 0; $j < sizeof($hasil2[$i]); $j++) {
        echo "<td>" . round($hasil2[$i][$j], 3) . "</td>";
    }
    echo "</tr>";
}
echo "</table>";

$hasil3 = perkalian_matriks2($B, $hasil2);
echo "<br><p>#952 = (A<sup>T</sup> x A)<sup>-1</sup> x (A x
Target)</p><table border='1' cellspacing='0' cellpadding='5'>";
for ($i = 0; $i < sizeof($hasil3); $i++) {
    echo "<tr>";
    for ($j = 0; $j < sizeof($hasil3[$i]); $j++) {
        echo "<td>" . round($hasil3[$i][$j], 3) . "</td>";
    }
    echo "</tr>";
}
echo "</table>";

$p1 = $hasil3[0][0];
$q1 = $hasil3[1][0];
$r1 = $hasil3[2][0];
$p2 = $hasil3[3][0];
$q2 = $hasil3[4][0];
$r2 = $hasil3[5][0];

?>
<br>
<?> "<b>p1: " . round($p1, 3) . ", q1: " . round($q1, 3) . ",
r1: " . round($r1, 3) . ", p2: " . round($p2, 3) . ", q2: " . round($q2, 3) . ",
r2: " . round($r2, 3) . "</b>"; ?>
<br><br>
<p>Lapisan 5 : Neuron tunggal dan keluaran untuk hasil
prediksi</p>

```

```

<table class="table table-bordered" width="100%"
cellspacing="0">
    <thead>
        <tr>
            <th>Data Ke</th>
            <th>Target</th>
            <th>Prediksi</th>
            <th>Error</th>
        </tr>
    </thead>
    <tbody>
        <?php
        $data_ke = 0;
        $summape = array();
        $summse = array();
        $apre = array();
        $a = 0;
        $b = 1;
        $c = 2;
        while ($c < count($transaksi)) {
            $data_ke++;
            $epoch = 10;
            $eror = 10;
            $ma1 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean3) /
$devi3), 2))));
            $ma2 = 1 / (1 + (pow(((x[$a]['JUMLAH'] - $mean4) /
$devi4), 2))));
            $ma3 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean1) /
$devi1), 2))));
            $ma4 = 1 / (1 + (pow(((x[$b]['JUMLAH'] - $mean2) /
$devi2), 2))));

            $w1 = $ma1 * $ma3;
            $w2 = $ma2 * $ma4;
            $wb1 = $w1 / ($w1 + $w2);
            $wb2 = $w2 / ($w1 + $w2);
            $wb1x1 = $wb1 * x[$a]['JUMLAH'];
            $wb1x2 = $wb1 * x[$b]['JUMLAH'];
            $wb2x1 = $wb2 * x[$a]['JUMLAH'];
            $wb2x2 = $wb2 * x[$b]['JUMLAH'];
            $pre = ($wb1x1 * $p1) + ($wb1x2 * $q1) + ($wb1 *
$r1) + ($wb2x1 * $p2) + ($wb2x2 * $q2) + ($wb2 * $r2);
            $error = x[$c]['JUMLAH'] - $pre;
            // if ($error < $eror) {
            //     $error = -2 * $error;
            //     $pre = x[$c]['JUMLAH'] - $error;

```

```

        // }
        $summape[] = abs($error / $x[$c]['JUMLAH']) * 100;
        $summse[] = $error;
        $apre[] = $pre;
    ?>
    <tr>
        <th><?= $data_ke; ?></th>
        <td><?= $x[$c]['JUMLAH'] ?></td>
        <td><?= round($pre, 3); ?></td>
        <td><?= round($error, 3); ?></td>
    </tr>
    <?php
        $a++;
        $b++;
        $c++;
    } ?>
</tbody>
</table>
MAPE : <?= round(array_sum($summape) / $data_ke, 3); ?>%<br>
Rata-rata Error : <?= round(array_sum($summse) / $data_ke, 3);
?><br>

<!-- hasil : <p id="hasil">
    <?php
        $q = 0;
        $a = 0;
        $b = 1;
        $c = 2;
        while ($c < count($transaksi)) {
            echo round($apre[$q], 3);
            $q++;
            $a++;
            $b++;
            $c++;
        }
    ?></p>
    <button id="proses" href="" class="btn btn-
success">Proses</button> -->
    </div>
</div>
</div>

<!-- /.container-fluid -->

<?= $this->endSection(); ?>

```