NIM: 231232028
Nama: Falmesino Abdul Hamid

Tugas Pertemuan 14
Struktur Data dan Algoritma (SDA)

URL Source Code: https://github.com/falmesino/sda-praktikum-14

```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Logistic Regresssion dengan Gradient Descent Regularized (L2)
   *
   * Dataset diasumsikan berupa array asosiatif:
   * - X: matriks fitur (m x n)
   * - y: label (0 atau 1)
   */

  function sigmoid($z) {
    return 1 / (1 + exp(-$z));
  }

  function logisticRegressionGD($X, $y, $learning_rate = 0.01, $iterations = 1000, $lambda = 0.1) {
    $m = count($y);     // Jumlah data
    $n = count($X[0]);  // Jumlah fitur
    $theta = array_fill(0, $n, 0.0); // Inisialisasi parameter model

    // Gradient descent
    for ($iter = 0; $iter < $iterations; $iter++) {
      $gradients = array_fill(0, $n, 0.0);

      // Hitung gradient untuk setiap data
      for ($i = 0; $i < $m; $i++) {
        $z = 0;
        for ($j = 0; $j < $n; $j++) {
          $z += $theta[$j] * $X[$i][$j];
        }
        $h = sigmoid($z);
        $error = $h - $y[$i];
        for ($j = 0; $j < $n; $j++) {
          $gradients[$j] += $error * $X[$i][$j];
        }
      }

      // Update parameter dengan regulisasi L2 (kecuali theta[0] bias)
      for ($j = 0; $j < $n; $j++) {
        $reg = ($j == 0) ? 0 : ($lambda / $m) * $theta[$j];
        $theta[$j] -= $learning_rate * (($gradients[$j] / $m) + $reg);
      }
    }

    return $theta;
  }

  // Contoh data (m = 4, n = 3)
  $X = [
    [1, 2.5, 3.1],  // Baris pertama data (termasuk fitur bias jika diperlukan)
    [1, 3.0, 3.8],
    [1, 2.8, 3.0],
    [1, 3.2, 3.9]
  ];
  $y = [0, 1, 0, 1];
  $theta = logisticRegressionGD($X, $y, 0.05, 20000, 0.05);

  echo "Parameter model (theta):\n";
  print_r($theta);

  /**
   * Analisis
   * - Setiap iterasi memproses m data dengan fitur -> O(m * n) per iterasi.
   * - Total iterasi T menghasilkan kompleksitas waktu O(T * m * n).
   * - Ruang O(n) untuk theta dan gradien.
   */

?>
```

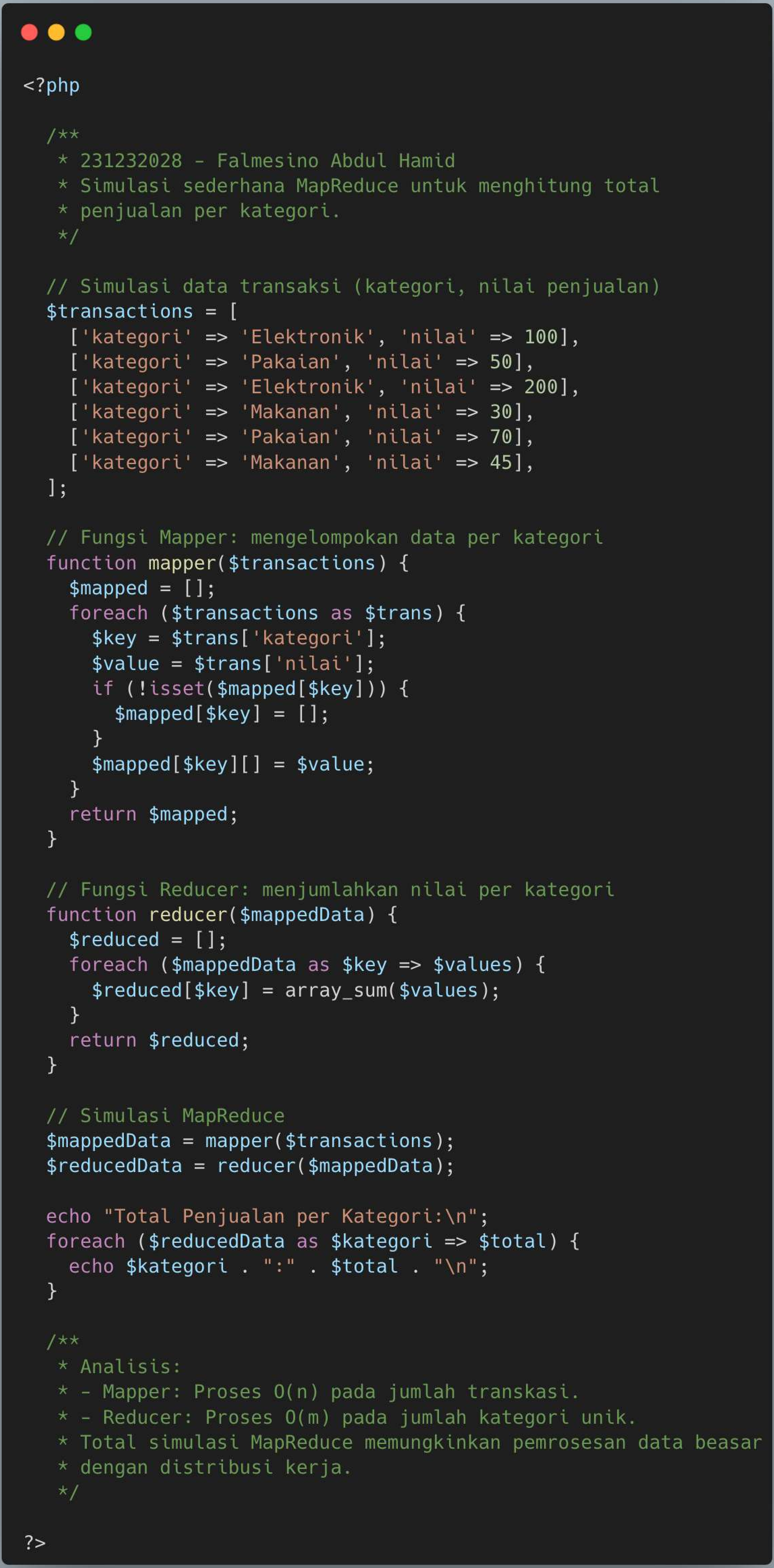
```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Logistic Regresssion dengan Gradient Descent Regularized (L2)
   *
   * Dataset diasumsikan berupa array asosiatif:
   * - X: matriks fitur (m x n)
   * - y: label (0 atau 1)
   */

  function sigmoid($z) {
    return 1 / (1 + exp(-$z));
  }

  function logisticRegressionGD($X, $y, $learning_rate = 0.01, $iterations = 1000, $lambda = 0.1) {
    $m = count($y);     // Jumlah data
    $n = count($X[0]);  // Jumlah fitur
    $theta = array_fill(0, $n, 0.0); // Inisialisasi parameter model

    // Gradient descent
    for ($iter = 0; $iter < $iterations; $iter++) {
      $gradients = array_fill(0, $n, 0.0);

      // Hitung gradient untuk setiap data
      for ($i = 0; $i < $m; $i++) {
        $z = 0;
        for ($j = 0; $j < $n; $j++) {
          $z += $theta[$j] * $X[$i][$j];
        }
        $h = sigmoid($z);
        $error = $h - $y[$i];
        for ($j = 0; $j < $n; $j++) {
          $gradients[$j] += $error * $X[$i][$j];
        }
      }

      // Update parameter dengan regulisasi L2 (kecuali theta[0] bias)
      for ($j = 0; $j < $n; $j++) {
        $reg = ($j == 0) ? 0 : ($lambda / $m) * $theta[$j];
        $theta[$j] -= $learning_rate * (($gradients[$j] / $m) + $reg);
      }
    }

    return $theta;
  }

  // Contoh data (m = 4, n = 3)
  $X = [
    [1, 2.5, 3.1],  // Baris pertama data (termasuk fitur bias jika diperlukan)
```

```
➜  sda-praktikum-14 git:(main) ls -la
total 40
drwxr-xr-x   8 falmesino  staff   256 Feb  8 02:36 .
drwxr-xr-x  38 falmesino  staff  1216 Feb  7 22:45 ..
drwxr-xr-x  12 falmesino  staff   384 Feb  8 03:07 .git
-rw-r--r--   1 falmesino  staff  1832 Feb  8 00:38 1.php
-rw-r--r--   1 falmesino  staff  3728 Feb  8 02:13 2.php
-rw-r--r--   1 falmesino  staff  1557 Feb  8 02:20 3.php
-rw-r--r--   1 falmesino  staff  2728 Feb  8 03:07 4.php
-rw-r--r--   1 falmesino  staff  1811 Feb  8 02:45 5.php
➜  sda-praktikum-14 git:(main) valet php 1.php
Parameter model (theta):
Array
(
    [0] => -16.062525491614
    [1] => 1.4083091811625
    [2] => 3.497920854293
)
➜  sda-praktikum-14 git:(main)
```

```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Contoh Implementasi A* Sederhana
   */

  class Node {
    public $x;
    public $y;
    public $g; // cost from start
    public $h; // heuristic cost to goal
    public $f; // total cost = g + h
    public $parent;

    public function __construct($x, $y, $g = 0, $h = 0, $parent = null) {
      $this->x = $x;
      $this->y = $y;
      $this->g = $g;
      $this->h = $h;
      $this->f = $g + $h;
      $this->parent = $parent;
    }
  }

  // Fungsi heuristic (menggunakan jarak Manhattan)
  function heuristic($node, $goal) {
    return abs($node->x - $goal->x) + abs($node->y - $goal->y);
  }

  // Fungsi untuk mendapatkan tetangga (4 arah)
  function getNeighbors($node, $grid) {
    $neighbors = [];
    $directions = [[0,1], [1,0], [0,-1], [-1,0]];
    foreach ($directions as $d) {
      $nx = $node->x + $d[0];
      $ny = $node->y + $d[1];

      if (isset($grid[$ny][$nx]) && $grid[$ny][$nx] == 0) { // 0 = jalur bebas
        $neighbors[] = new Node($nx, $ny);
      }
    }
    return $neighbors;
  }

  // Implementasi A*
  function aStar($grid, $start, $goal) {
    $openSet = [];
    $closedSet = [];
    $startNode = new Node($start[0], $start[1]);
    $startNode->h = heuristic($startNode, new Node($goal[0], $goal[1]));
    $startNode->f = $startNode->g + $startNode->h;
    $openSet[] = $startNode;

    while(!empty($openSet)) {
      // Pilih node dengan nilai f terkecil
      usort($openSet, function($a, $b) {
        return $a->f <=> $b->f;
      });
      $current = array_shift($openSet);

      // Jika mencapai tujuan, rekonstruksi jalur
      if ($current->x == $goal[0] && $current->y == $goal[1]) {
        $path = [];
        while ($current != null) {
          $path[] = [$current->x, $current->y];
          $current = $current->parent;
        }
        return array_reverse($path);
      }

      $closedSet[] = $current;
      foreach(getNeighbors($current, $grid) as $neighbor) {
        // Jika sudah ada di closedSet, lewati
        $skip = false;
        foreach($closedSet as $closed) {
          if ($closed->x == $neighbor->x && $closed->y == $neighbor->y) {
            $skip = true;
            break;
          }
        }

        if ($skip) continue;

        $tentative_g = $current->g + 1; // Asumsi jarak antar tetangga = 1

        // Jika neighbor sudah ada di openSet dengan nilai g lebih kecil, lewati
        $inOpen = false;

        foreach ($openSet as $node) {
          if ($node->x == $neighbor->x && $node->y == $neighbor->y) {
            $inOpen = true;
            if ($tentative_g >= $node->g) {
              $skip = true;
            }
            break;
          }
        }
        if ($skip) continue;

        $neighbor->g = $tentative_g;
        $neighbor->h = heuristic($neighbor, new Node($goal[0], $goal[1]));
        $neighbor->f = $neighbor->g + $neighbor->h;
        $neighbor->parent = $current;
        $openSet[] = $neighbor;
      }
    }
    return null; // Tidak ditemukan jalur
  }

  // Contoh grid (0 = bebas, 1 = halangan)
  $grid = [
    [0, 0, 0, 0, 0],
    [0, 1, 1, 1, 0],
    [0, 0, 0, 1, 0],
    [0, 1, 0, 0, 0],
    [0, 0, 0, 1, 0]
  ];
  $start = [0, 0];
  $goal = [4, 4];

  $path = aStar($grid, $start, $goal);

  if ($path) {
    echo "Jalur ditemukan:\n";
    foreach ($path as $p) {
      echo "(" . $p[0] . ", " . $p[1] . ") ";
    }
    echo "\n";
  } else {
    echo "Jalur tidak ditemukan.\n";
  }

  /**
   * Analisis
   * - Kompleksitas waktu bergantung pada jumlah node dan heuristik, secara umum
   *   O(n log n) dengan penggunaan priority queue (di sini disort setiap iterasi).
   * - Kompleksitas ruang O(n) untuk openSet dan closedSet.
   */

?>
```

```php
<?php

  // Implementasi A*
  function aStar($grid, $start, $goal) {
    $openSet = [];
    $closedSet = [];
    $startNode = new Node($start[0], $start[1]);
    $startNode->h = heuristic($startNode, new Node($goal[0], $goal[1]));
    $startNode->f = $startNode->g + $startNode->h;
    $openSet[] = $startNode;

    while(!empty($openSet)) {
      // Pilih node dengan nilai f terkecil
      usort($openSet, function($a, $b) {
        return $a->f <=> $b->f;
      });
      $current = array_shift($openSet);

      // Jika mencapai tujuan, rekonstruksi jalur
      if ($current->x == $goal[0] && $current->y == $goal[1]) {
        $path = [];
        while ($current != null) {
          $path[] = [$current->x, $current->y];
          $current = $current->parent;
        }
        return array_reverse($path);
      }

      $closedSet[] = $current;
      foreach(getNeighbors($current, $grid) as $neighbor) {
        // Jika sudah ada di closedSet, lewati
        $skip = false;
        foreach($closedSet as $closed) {
          if ($closed->x == $neighbor->x && $closed->y == $neighbor->y) {
            $skip = true;
            break;
          }
        }

        if ($skip) continue;

        $tentative_g = $current->g + 1; // Asumsi jarak antar tetangga = 1

        // Jika neighbor sudah ada di openSet dengan nilai g lebih kecil, lewati
        $inOpen = false;

        foreach ($openSet as $node) {
          if ($node->x == $neighbor->x && $node->y == $neighbor->y) {
            $inOpen = true;
            if ($tentative_g >= $node->g) {
              $skip = true;
```

```
● → sda-praktikum-14 git:(main) ls -la
total 40
drwxr-xr-x   8 falmesino  staff   256 Feb  8 02:36 .
drwxr-xr-x  38 falmesino  staff  1216 Feb  7 22:45 ..
drwxr-xr-x  12 falmesino  staff   384 Feb  8 03:07 .git
-rw-r--r--   1 falmesino  staff  1832 Feb  8 00:38 1.php
-rw-r--r--   1 falmesino  staff  3728 Feb  8 02:13 2.php
-rw-r--r--   1 falmesino  staff  1557 Feb  8 02:20 3.php
-rw-r--r--   1 falmesino  staff  2728 Feb  8 03:07 4.php
-rw-r--r--   1 falmesino  staff  1811 Feb  8 02:45 5.php
● → sda-praktikum-14 git:(main) valet php 2.php
Jalur ditemukan:
(0, 0) (0, 1) (0, 2) (1, 2) (2, 2) (2, 3) (3, 3) (4, 3) (4, 4)
○ → sda-praktikum-14 git:(main) 
```



```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Simulasi sederhana MapReduce untuk menghitung total
   * penjualan per kategori.
   */

  // Simulasi data transaksi (kategori, nilai penjualan)
  $transactions = [
    ['kategori' => 'Elektronik', 'nilai' => 100],
    ['kategori' => 'Pakaian', 'nilai' => 50],
    ['kategori' => 'Elektronik', 'nilai' => 200],
    ['kategori' => 'Makanan', 'nilai' => 30],
    ['kategori' => 'Pakaian', 'nilai' => 70],
    ['kategori' => 'Makanan', 'nilai' => 45],
  ];

  // Fungsi Mapper: mengelompokan data per kategori
  function mapper($transactions) {
    $mapped = [];
    foreach ($transactions as $trans) {
      $key = $trans['kategori'];
      $value = $trans['nilai'];
      if (!isset($mapped[$key])) {
        $mapped[$key] = [];
      }
      $mapped[$key][] = $value;
    }
    return $mapped;
  }

  // Fungsi Reducer: menjumlahkan nilai per kategori
  function reducer($mappedData) {
    $reduced = [];
    foreach ($mappedData as $key => $values) {
      $reduced[$key] = array_sum($values);
    }
    return $reduced;
  }

  // Simulasi MapReduce
  $mappedData = mapper($transactions);
  $reducedData = reducer($mappedData);

  echo "Total Penjualan per Kategori:\n";
  foreach ($reducedData as $kategori => $total) {
    echo $kategori . ":" . $total . "\n";
  }

  /**
   * Analisis:
   * - Mapper: Proses O(n) pada jumlah transkasi.
   * - Reducer: Proses O(m) pada jumlah kategori unik.
   * Total simulasi MapReduce memungkinkan pemrosesan data beasar
   * dengan distribusi kerja.
   */

?>
```

sda-praktikum-14
EXPLORER
SDA-PRAKTIKUM-14
1.php
2.php
3.php
4.php
5.php
OUTLINE
PORTS
GITLENS
TERMINAL
zsh
main
Launchpad
Falmesino Abdul Hamid (35 minutes ago)
Ln 19, Col 18
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier

```php
<?php
/**
 * Simulasi sederhana MapReduce untuk menghitung total
 * penjualan per kategori.
 */

// Simulasi data transaksi (kategori, nilai penjualan)
$transactions = [
  ['kategori' => 'Elektronik', 'nilai' => 100],
  ['kategori' => 'Pakaian', 'nilai' => 50],
  ['kategori' => 'Elektronik', 'nilai' => 200],
  ['kategori' => 'Makanan', 'nilai' => 30],
  ['kategori' => 'Pakaian', 'nilai' => 70],
  ['kategori' => 'Makanan', 'nilai' => 45],
];

// Fungsi Mapper: mengelompokan data per kategori
function mapper($transactions) {
  $mapped = [];
  foreach ($transactions as $trans) {
    $key = $trans['kategori'];
    $value = $trans['nilai'];
    if (!isset($mapped[$key])) {
      $mapped[$key] = [];
    }
    $mapped[$key][] = $value;
  }
  return $mapped;
}

// Fungsi Reducer: menjumlahkan nilai per kategori
function reducer($mappedData) {
  $reduced = [];
  foreach ($mappedData as $key => $values) {
    $reduced[$key] = array_sum($values);
  }
  return $reduced;
}

// Simulasi MapReduce
$mappedData = mapper($transactions);
$reducedData = reducer($mappedData);

echo "Total Penjualan per Kategori:\n";
foreach ($reducedData as $kategori => $total) {
  echo $kategori . ":" . $total . "\n";
}

/**
 * Analisis:
```

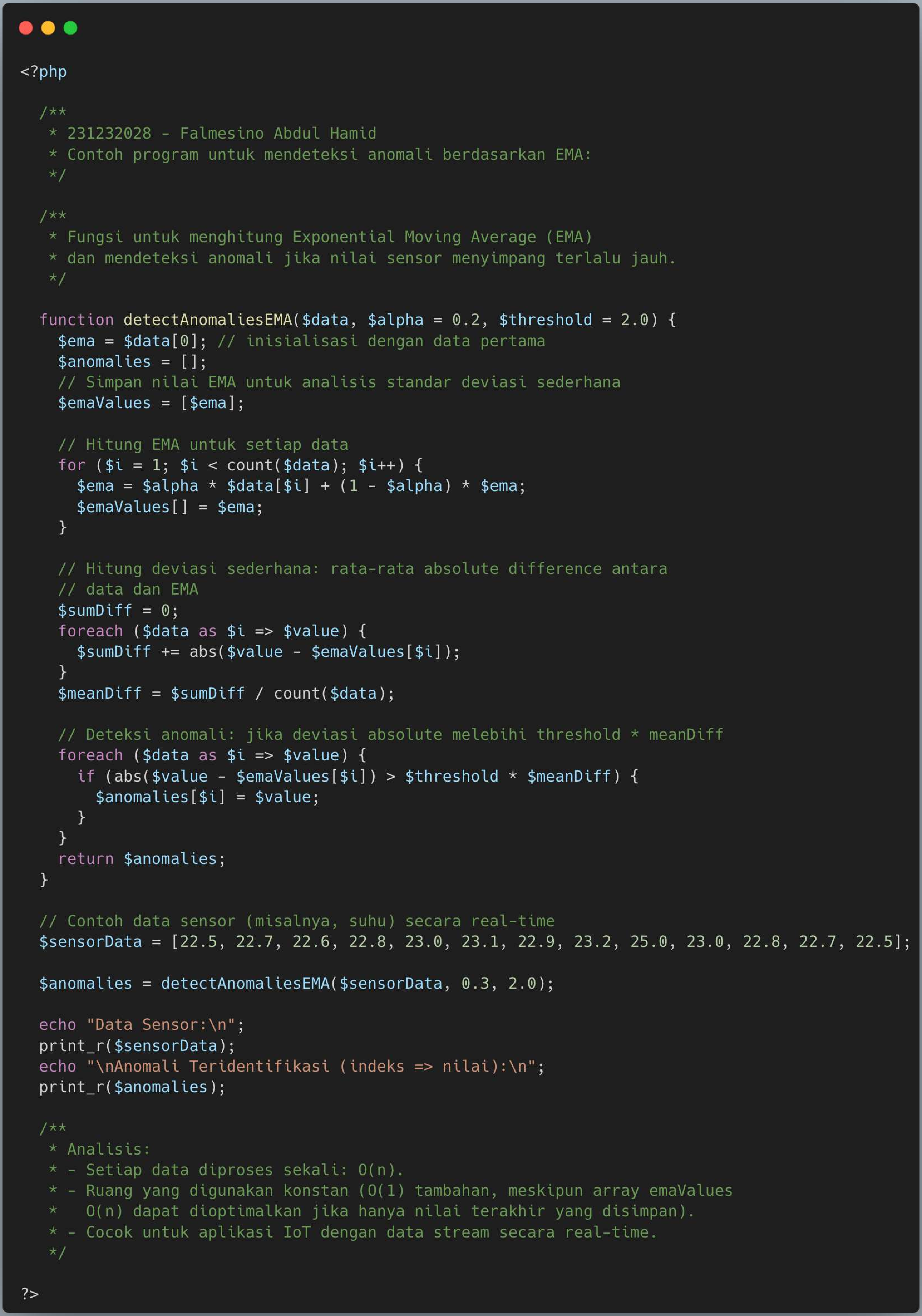
```
➜  sda-praktikum-14 git:(main) ls -la
total 40
drwxr-xr-x   8 falmesino  staff   256 Feb  8 02:36 .
drwxr-xr-x  38 falmesino  staff  1216 Feb  7 22:45 ..
drwxr-xr-x  12 falmesino  staff   384 Feb  8 03:07 .git
-rw-r--r--   1 falmesino  staff  1832 Feb  8 00:38 1.php
-rw-r--r--   1 falmesino  staff  3728 Feb  8 02:13 2.php
-rw-r--r--   1 falmesino  staff  1557 Feb  8 02:20 3.php
-rw-r--r--   1 falmesino  staff  2728 Feb  8 03:07 4.php
-rw-r--r--   1 falmesino  staff  1811 Feb  8 02:45 5.php
➜  sda-praktikum-14 git:(main) valet php 3.php
Total Penjualan per Kategori:
Elektronik:300
Pakaian:120
Makanan:75
➜  sda-praktikum-14 git:(main)
```

```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Contoh sederhana Apriori untuk frequent 1-itemset dan 2-itemset
   */

   // Data transaksi: setiap transaksi merupakan array item
   $transactions = [
    ['susu', 'roti', 'keju'],
    ['roti', 'keju'],
    ['susu', 'roti'],
    ['roti', 'keju'],
    ['susu', 'keju']
   ];
   $minSupport = 0.6;
   $totalTransactions = count($transactions);

   // Fungsi untuk menghitung support suatu itemset
   function countSupport($transactions, $itemset) {
    $count = 0;
    foreach ($transactions as $trans) {
      if (count(array_intersect($trans, $itemset)) == count($itemset)) {
        $count++;
      }
    }
    return $count / count($transactions);
   }

   // Mendapatkan frequent 1-itemset
   function apriori1itemset($transactions, $minSupport) {
    $itemCounts = [];
    foreach ($transactions as $trans) {
      foreach ($trans as $item) {
        if (!isset($itemCounts[$item])) {
          $itemCounts[$item] = 0;
        }
        $itemCounts[$item]++;
      }
    }
    $frequent1 = [];

    foreach ($itemCounts as $item => $count) {
      $support = $count / count($transactions);
      if ($support >= $minSupport) {
        $frequent1[$item] = $support; // Gunakan string key
      }
    }
    return $frequent1;
   }

   // Mendapatkan kandidat 2-itemset dari frequent 1-itemset
   function generateCandidates($frequent1) {
    $items = array_keys($frequent1); // Ambil item dari frequent1
    $candidates = [];
    $n = count($items);
    for ($i = 0; $i < $n; $i++) {
      for ($j = $i + 1; $j < $n; $j++) {
        $candidate = [$items[$i], $items[$j]];
        sort($candidate); // Urutkan untuk konsistensi
        $candidates[] = $candidate;
      }
    }
    return $candidates;
   }

   // Evaluasi frequent 2-itemset
   function apriori2Itemset($transactions, $candidates, $minSupport) {
    $frequent2 = [];
    foreach ($candidates as $candidate) {
      $support = countSupport($transactions, $candidate);
      if ($support >= $minSupport) {
        $frequent2[implode(',', $candidate)] = $support; // Assign ke frequent2
      }
    }
    return $frequent2;
   }

   $frequent1 = apriori1itemset($transactions, $minSupport);
   $candidates2 = generateCandidates($frequent1);
   $frequent2 = apriori2Itemset($transactions, $candidates2, $minSupport);

   echo "Frequent 1-itemsets:\n";
   print_r($frequent1);
   echo "Frequent 2-itemsets:\n";
   print_r($frequent2);

   /**
    * Analisis:
    * - Proses frequent 1-itemset: O(n * k) dengan n transaksi dan k item
    *   rata-rata per transaksi.
    * - Proses kandidate 2-itemset: O(m^2) dengan m item unik.
    * - Support dihitung untuk tiap kandidat dengan proses iteratif.
    */
?>
```

sda-praktikum-14
EXPLORER
SDA-PRAKTIKUM-14
1.php
2.php
3.php
4.php
5.php
OUTLINE
PORTS
GITLENS
TERMINAL
zsh
main
Launchpad
You, 4 minutes ago
Falmesino Abdul Hamid (4 minutes ago)
Ln 71, Col 13
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier

```php
<?php
  /**
  function generateCandidates($frequent1) {
   }
   return $candidates;
  }

  // Evaluasi frequent 2-itemset
  function apriori2Itemset($transactions, $candidates, $minSupport) {
   $frequent2 = [];
   foreach ($candidates as $candidate) {
     $support = countSupport($transactions, $candidate);      You, 4 minutes ago • Uncommitted changes
     if ($support >= $minSupport) {
       $frequent2[implode(',', $candidate)] = $support; // Assign ke frequent2
     }
   }
   return $frequent2;
  }

  $frequent1 = apriori1itemset($transactions, $minSupport);
  $candidates2 = generateCandidates($frequent1);
  $frequent2 = apriori2Itemset($transactions, $candidates2, $minSupport);

  echo "Frequent 1-itemsets:\n";
  print_r($frequent1);
  echo "Frequent 2-itemsets:\n";
  print_r($frequent2);

  /**
   * Analisis:
   * - Proses frequent 1-itemset: O(n * k) dengan n transaksi dan k item
   *   rata-rata per transaksi.
   * - Proses kandidate 2-itemset: O(m^2) dengan m item unik.
   * - Support dihitung untuk tiap kandidat dengan proses iteratif.
   */
?>
```

```
➜  sda-praktikum-14 git:(main) ls -la
total 40
drwxr-xr-x   8 falmesino  staff   256 Feb  8 02:36 .
drwxr-xr-x  38 falmesino  staff  1216 Feb  7 22:45 ..
drwxr-xr-x  12 falmesino  staff   384 Feb  8 03:07 .git
-rw-r--r--   1 falmesino  staff  1832 Feb  8 00:38 1.php
-rw-r--r--   1 falmesino  staff  3728 Feb  8 02:13 2.php
-rw-r--r--   1 falmesino  staff  1557 Feb  8 02:20 3.php
-rw-r--r--   1 falmesino  staff  2728 Feb  8 03:07 4.php
-rw-r--r--   1 falmesino  staff  1811 Feb  8 02:45 5.php
➜  sda-praktikum-14 git:(main) valet php 4.php
Frequent 1-itemsets:
Array
(
    [susu] => 0.6
    [roti] => 0.8
    [keju] => 0.8
)
Frequent 2-itemsets:
Array
(
    [keju,roti] => 0.6
)
➜  sda-praktikum-14 git:(main)
```



```php
<?php

  /**
   * 231232028 - Falmesino Abdul Hamid
   * Contoh program untuk mendeteksi anomali berdasarkan EMA:
   */

  /**
   * Fungsi untuk menghitung Exponential Moving Average (EMA)
   * dan mendeteksi anomali jika nilai sensor menyimpang terlalu jauh.
   */

  function detectAnomaliesEMA($data, $alpha = 0.2, $threshold = 2.0) {
    $ema = $data[0]; // inisialisasi dengan data pertama
    $anomalies = [];
    // Simpan nilai EMA untuk analisis standar deviasi sederhana
    $emaValues = [$ema];

    // Hitung EMA untuk setiap data
    for ($i = 1; $i < count($data); $i++) {
      $ema = $alpha * $data[$i] + (1 - $alpha) * $ema;
      $emaValues[] = $ema;
    }

    // Hitung deviasi sederhana: rata-rata absolute difference antara
    // data dan EMA
    $sumDiff = 0;
    foreach ($data as $i => $value) {
      $sumDiff += abs($value - $emaValues[$i]);
    }
    $meanDiff = $sumDiff / count($data);

    // Deteksi anomali: jika deviasi absolute melebihi threshold * meanDiff
    foreach ($data as $i => $value) {
      if (abs($value - $emaValues[$i]) > $threshold * $meanDiff) {
        $anomalies[$i] = $value;
      }
    }
    return $anomalies;
  }

  // Contoh data sensor (misalnya, suhu) secara real-time
  $sensorData = [22.5, 22.7, 22.6, 22.8, 23.0, 23.1, 22.9, 23.2, 25.0, 23.0, 22.8, 22.7, 22.5];

  $anomalies = detectAnomaliesEMA($sensorData, 0.3, 2.0);

  echo "Data Sensor:\n";
  print_r($sensorData);
  echo "\nAnomali Teridentifikasi (indeks => nilai):\n";
  print_r($anomalies);

  /**
   * Analisis:
   * - Setiap data diproses sekali: O(n).
   * - Ruang yang digunakan konstan (O(1) tambahan, meskipun array emaValues
   *   O(n) dapat dioptimalkan jika hanya nilai terakhir yang disimpan).
   * - Cocok untuk aplikasi IoT dengan data stream secara real-time.
   */

?>
```

sda-praktikum-14
EXPLORER
SDA-PRAKTIKUM-14
1.php
2.php
3.php
4.php
5.php
OUTLINE
PORTS
GITLENS
TERMINAL
zsh
main
Launchpad
You, 27 minutes ago
Falmesino Abdul Hamid (26 minutes ago)
Ln 34, Col 21
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier

```php
<?php
  function detectAnomaliesEMA($data, $alpha = 0.2, $threshold = 2.0) {

    // Hitung deviasi sederhana: rata-rata absolute difference antara
    // data dan EMA
    $sumDiff = 0;
    foreach ($data as $i => $value) {
      $sumDiff += abs($value - $emaValues[$i]);
    }
    $meanDiff = $sumDiff / count($data);

    // Deteksi anomali: jika deviasi absolute melebihi threshold * meanDiff
    foreach ($data as $i => $value) {      You, 27 minutes ago • Uncommitted changes
      if (abs($value - $emaValues[$i]) > $threshold * $meanDiff) {
        $anomalies[$i] = $value;
      }
    }
    return $anomalies;
  }

  // Contoh data sensor (misalnya, suhu) secara real-time
  $sensorData = [22.5, 22.7, 22.6, 22.8, 23.0, 23.1, 22.9, 23.2, 25.0, 23.0, 22.8, 22.7, 22.5];

  $anomalies = detectAnomaliesEMA($sensorData, 0.3, 2.0);

  echo "Data Sensor:\n";
  print_r($sensorData);
  echo "\nAnomali Teridentifikasi (indeks => nilai):\n";
  print_r($anomalies);

  /**
   * Analisis:
   * - Setiap data diproses sekali: O(n).
   * - Ruang yang digunakan konstan (O(1) tambahan, meskipun array emaValues
   *   O(n) dapat dioptimalkan jika hanya nilai terakhir yang disimpan).
   * - Cocok untuk aplikasi IoT dengan data stream secara real-time.
   */

?>
```

```
→ sda-praktikum-14 git:(main) ls -la
total 40
drwxr-xr-x   8 falmesino  staff   256 Feb  8 02:36 .
drwxr-xr-x  38 falmesino  staff  1216 Feb  7 22:45 ..
drwxr-xr-x  12 falmesino  staff   384 Feb  8 03:07 .git
-rw-r--r--   1 falmesino  staff  1832 Feb  8 00:38 1.php
-rw-r--r--   1 falmesino  staff  3728 Feb  8 02:13 2.php
-rw-r--r--   1 falmesino  staff  1557 Feb  8 02:20 3.php
-rw-r--r--   1 falmesino  staff  2728 Feb  8 03:07 4.php
-rw-r--r--   1 falmesino  staff  1811 Feb  8 02:45 5.php
→ sda-praktikum-14 git:(main) valet php 5.php
Data Sensor:
Array
(
    [0] => 22.5
    [1] => 22.7
    [2] => 22.6
    [3] => 22.8
    [4] => 23
    [5] => 23.1
    [6] => 22.9
    [7] => 23.2
    [8] => 25
    [9] => 23
    [10] => 22.8
    [11] => 22.7
    [12] => 22.5
)

Anomali Teridentifikasi (indeks => nilai):
Array
(
    [8] => 25
)
→ sda-praktikum-14 git:(main) 
```