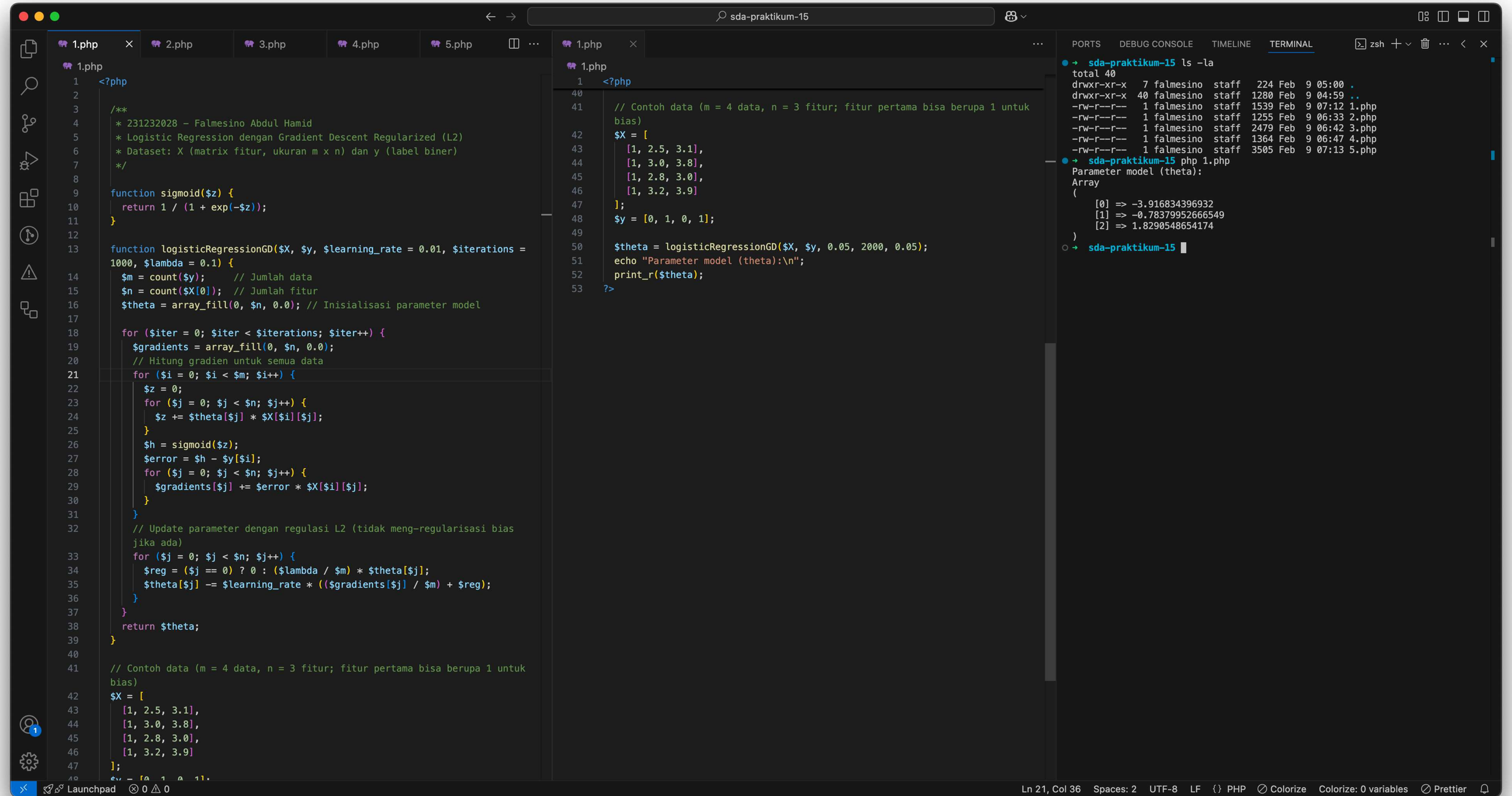
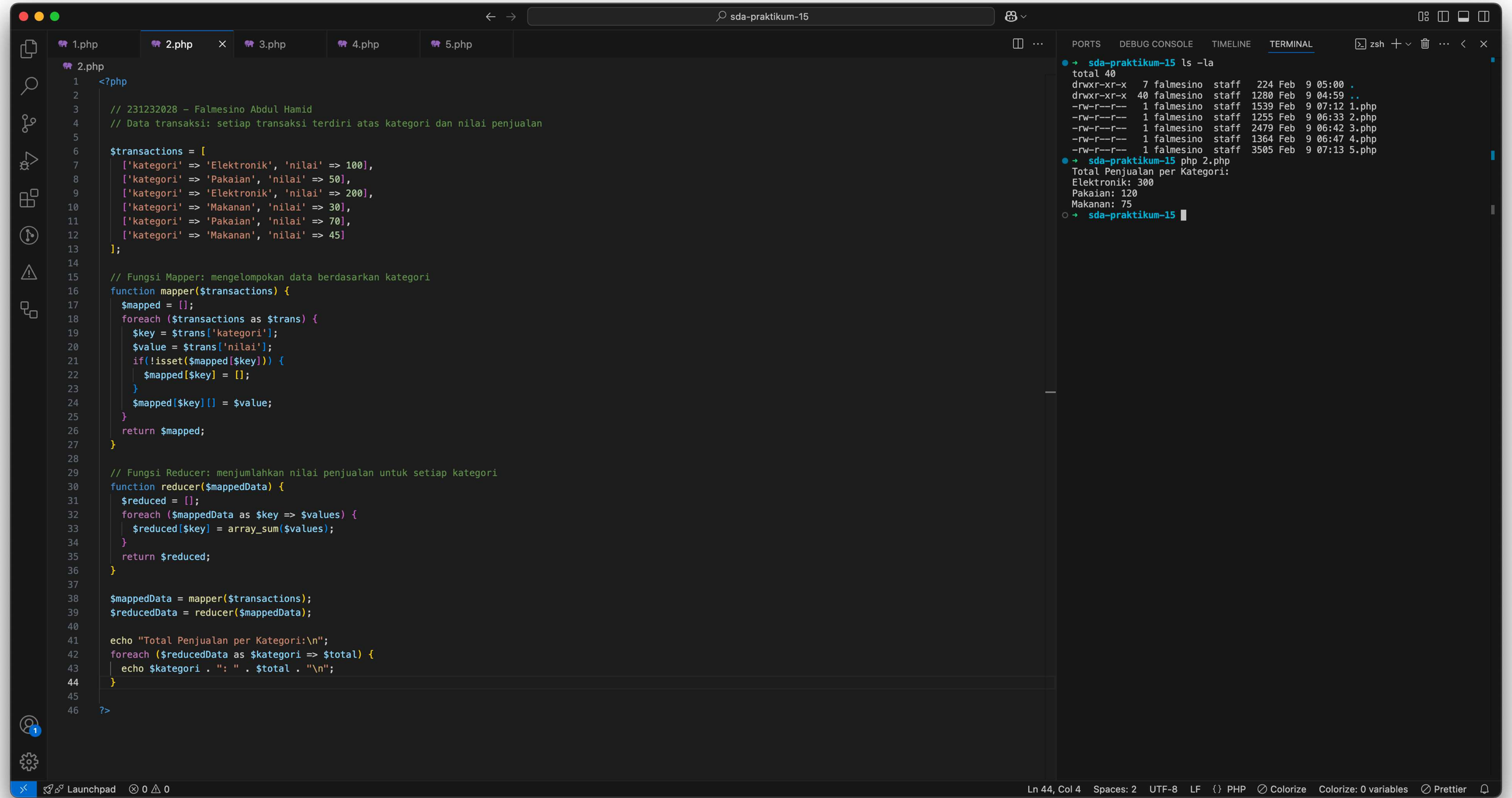
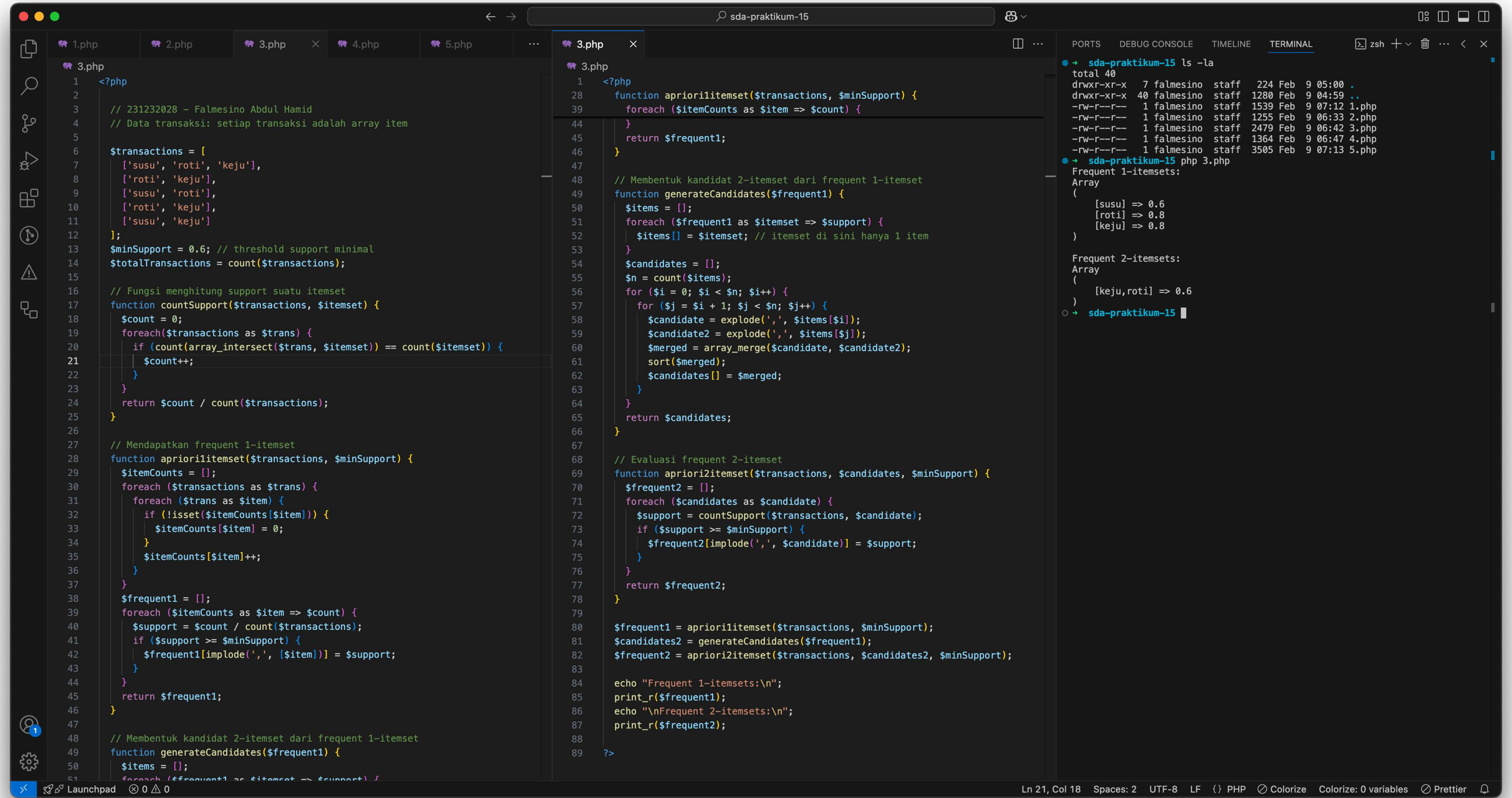
NIM: 231232028
Nama: Falmesino Abdul Hamid

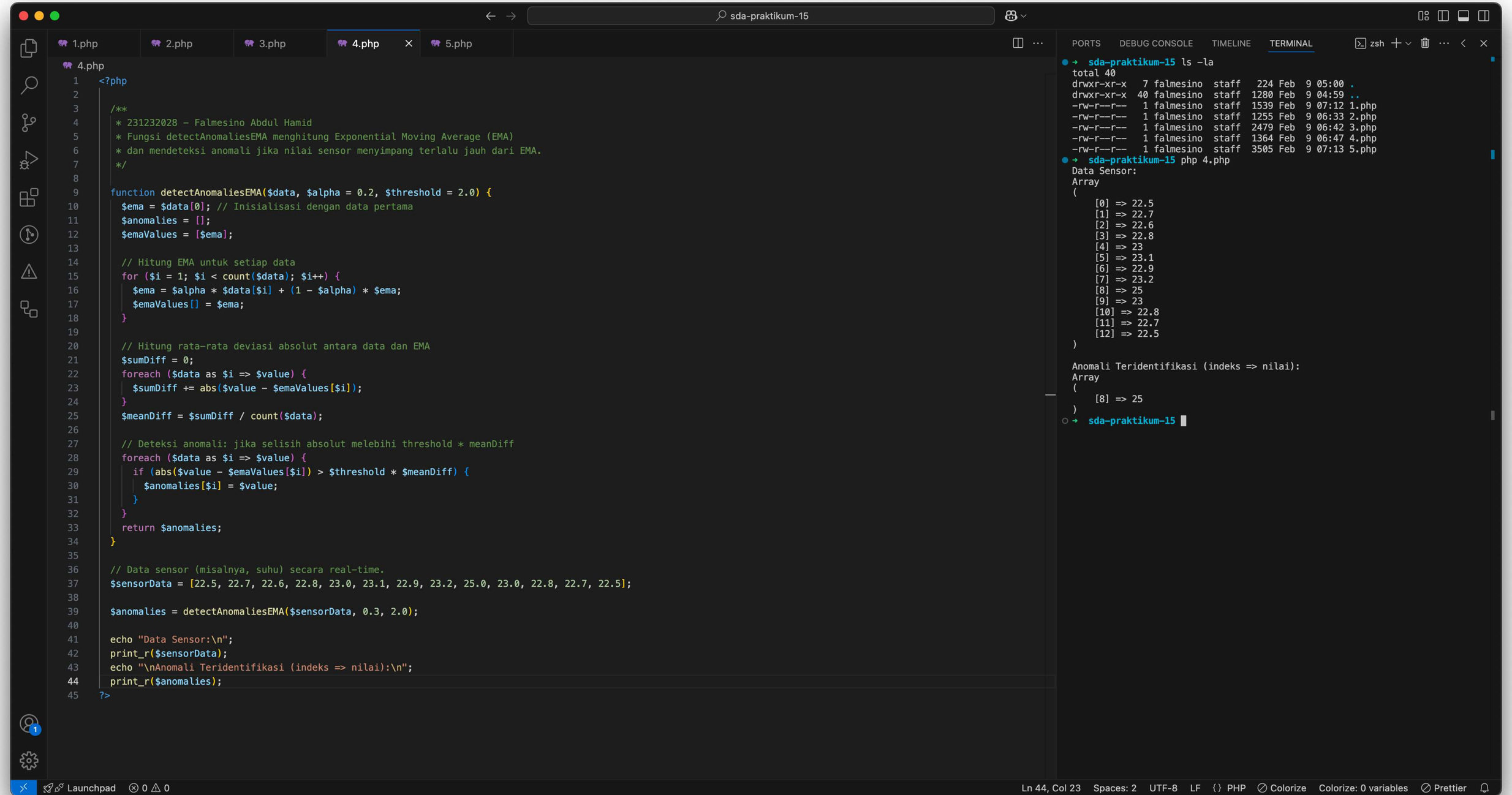
Tugas Pertemuan 15
Struktur Data dan Algoritma (SDA)

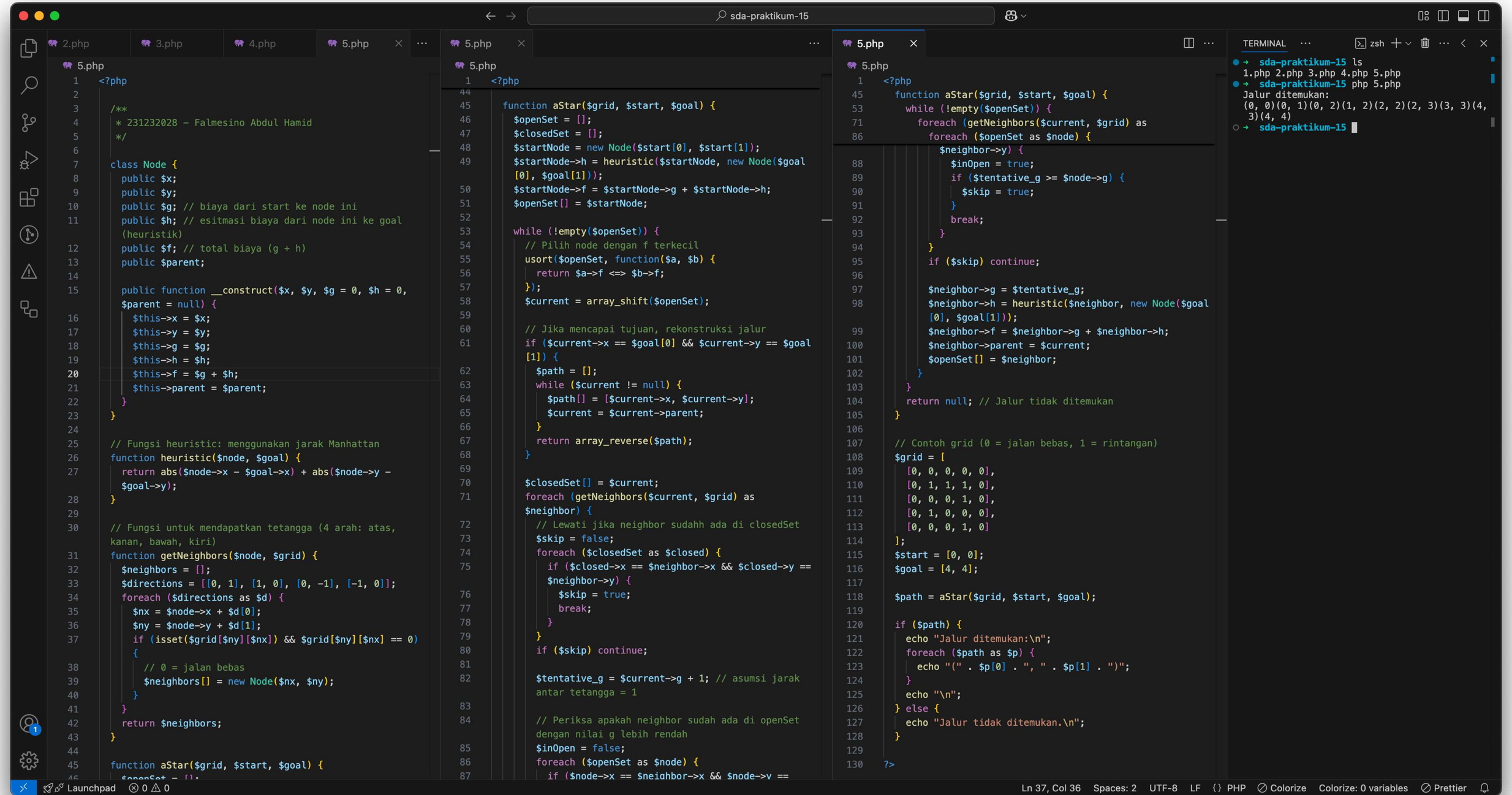
URL Source Code: https://github.com/falmesino/sda-praktikum-15


sda-praktikum-15
1.php
2.php
3.php
4.php
5.php
<?php
/**
* 231232028 - Falmesino Abdul Hamid
* Logistic Regression dengan Gradient Descent Regularized (L2)
* Dataset: X (matrix fitur, ukuran m x n) dan y (label biner)
*/
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, $n, 0.0); // Inisialisasi parameter model
for ($iter = 0; $iter < $iterations; $iter++) {
$gradients = array_fill(0, $n, 0.0);
// Hitung gradien untuk semua data
for ($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][$j];
}
}
// Update parameter dengan regulasi L2 (tidak meng-regularisasi bias jika ada)
for ($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 data, n = 3 fitur; fitur pertama bisa berupa 1 untuk bias)
$X = [
[1, 2.5, 3.1],
[1, 3.0, 3.8],
[1, 2.8, 3.0],
[1, 3.2, 3.9]
];
$y = [0, 1, 0, 1];
$theta = logisticRegressionGD($X, $y, 0.05, 2000, 0.05);
echo "Parameter model (theta):\n";
print_r($theta);
?>
PORTS
DEBUG CONSOLE
TIMELINE
TERMINAL
zsh
sda-praktikum-15 ls -la
total 40
drwxr-xr-x 7 falmesino staff 224 Feb 9 05:00 .
drwxr-xr-x 40 falmesino staff 1280 Feb 9 04:59 ..
-rw-r--r-- 1 falmesino staff 1539 Feb 9 07:12 1.php
-rw-r--r-- 1 falmesino staff 1255 Feb 9 06:33 2.php
-rw-r--r-- 1 falmesino staff 2479 Feb 9 06:42 3.php
-rw-r--r-- 1 falmesino staff 1364 Feb 9 06:47 4.php
-rw-r--r-- 1 falmesino staff 3505 Feb 9 07:13 5.php
sda-praktikum-15 php 1.php
Parameter model (theta):
Array
(
[0] => -3.916834396932
[1] => -0.78379952666549
[2] => 1.8290548654174
)
sda-praktikum-15
Launchpad
Ln 21, Col 36
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier


sda-praktikum-15
1.php
2.php
3.php
4.php
5.php
2.php
<?php
// 231232028 – Falmesino Abdul Hamid
// Data transaksi: setiap transaksi terdiri atas kategori dan nilai penjualan
$transactions = [
['kategori' => 'Elektronik', 'nilai' => 100],
['kategori' => 'Pakaian', 'nilai' => 50],
['kategori' => 'Elektronik', 'nilai' => 200],
['kategori' => 'Makanan', 'nilai' => 30],
['kategori' => 'Pakaian', 'nilai' => 70],
['kategori' => 'Makanan', 'nilai' => 45]
];
// Fungsi Mapper: mengelompokan data berdasarkan kategori
function mapper($transactions) {
$mapped = [];
foreach ($transactions as $trans) {
$key = $trans['kategori'];
$value = $trans['nilai'];
if(!isset($mapped[$key])) {
$mapped[$key] = [];
}
$mapped[$key][] = $value;
}
return $mapped;
}
// Fungsi Reducer: menjumlahkan nilai penjualan untuk setiap kategori
function reducer($mappedData) {
$reduced = [];
foreach ($mappedData as $key => $values) {
$reduced[$key] = array_sum($values);
}
return $reduced;
}
$mappedData = mapper($transactions);
$reducedData = reducer($mappedData);
echo "Total Penjualan per Kategori:\n";
foreach ($reducedData as $kategori => $total) {
echo $kategori . ": " . $total . "\n";
}
?>
PORTS
DEBUG CONSOLE
TIMELINE
TERMINAL
zsh
→ sda-praktikum-15 ls -la
total 40
drwxr-xr-x 7 falmesino staff 224 Feb 9 05:00 .
drwxr-xr-x 40 falmesino staff 1280 Feb 9 04:59 ..
-rw-r--r-- 1 falmesino staff 1539 Feb 9 07:12 1.php
-rw-r--r-- 1 falmesino staff 1255 Feb 9 06:33 2.php
-rw-r--r-- 1 falmesino staff 2479 Feb 9 06:42 3.php
-rw-r--r-- 1 falmesino staff 1364 Feb 9 06:47 4.php
-rw-r--r-- 1 falmesino staff 3505 Feb 9 07:13 5.php
→ sda-praktikum-15 php 2.php
Total Penjualan per Kategori:
Elektronik: 300
Pakaian: 120
Makanan: 75
→ sda-praktikum-15
Launchpad
0 0
Ln 44, Col 4
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier


sda-praktikum-15
1.php
2.php
3.php
4.php
5.php
3.php
<?php
// 231232028 - Falmesino Abdul Hamid
// Data transaksi: setiap transaksi adalah array item
$transactions = [
['susu', 'roti', 'keju'],
['roti', 'keju'],
['susu', 'roti'],
['roti', 'keju'],
['susu', 'keju']
];
$minSupport = 0.6; // threshold support minimal
$totalTransactions = count($transactions);
// Fungsi 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) {
$frequent1[implode(',', [$item])] = $support;
}
}
return $frequent1;
}
// Membentuk kandidat 2-itemset dari frequent 1-itemset
function generateCandidates($frequent1) {
$items = [];
3.php
<?php
function apriori1itemset($transactions, $minSupport) {
foreach ($itemCounts as $item => $count) {
}
return $frequent1;
}
// Membentuk kandidat 2-itemset dari frequent 1-itemset
function generateCandidates($frequent1) {
$items = [];
foreach ($frequent1 as $itemset => $support) {
$items[] = $itemset; // itemset di sini hanya 1 item
}
$candidates = [];
$n = count($items);
for ($i = 0; $i < $n; $i++) {
for ($j = $i + 1; $j < $n; $j++) {
$candidate = explode(',', $items[$i]);
$candidate2 = explode(',', $items[$j]);
$merged = array_merge($candidate, $candidate2);
sort($merged);
$candidates[] = $merged;
}
}
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;
}
}
return $frequent2;
}
$frequent1 = apriori1itemset($transactions, $minSupport);
$candidates2 = generateCandidates($frequent1);
$frequent2 = apriori2itemset($transactions, $candidates2, $minSupport);
echo "Frequent 1-itemsets:\n";
print_r($frequent1);
echo "\nFrequent 2-itemsets:\n";
print_r($frequent2);
?>
PORTS
DEBUG CONSOLE
TIMELINE
TERMINAL
zsh
sda-praktikum-15 ls -la
total 40
drwxr-xr-x 7 falmesino staff 224 Feb 9 05:00 .
drwxr-xr-x 40 falmesino staff 1280 Feb 9 04:59 ..
-rw-r--r-- 1 falmesino staff 1539 Feb 9 07:12 1.php
-rw-r--r-- 1 falmesino staff 1255 Feb 9 06:33 2.php
-rw-r--r-- 1 falmesino staff 2479 Feb 9 06:42 3.php
-rw-r--r-- 1 falmesino staff 1364 Feb 9 06:47 4.php
-rw-r--r-- 1 falmesino staff 3505 Feb 9 07:13 5.php
sda-praktikum-15 php 3.php
Frequent 1-itemsets:
Array
(
[susu] => 0.6
[roti] => 0.8
[keju] => 0.8
)
Frequent 2-itemsets:
Array
(
[keju,roti] => 0.6
)
sda-praktikum-15
Launchpad
Ln 21, Col 18
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier


sda-praktikum-15
1.php
2.php
3.php
4.php
5.php
4.php
<?php
/**
* 231232028 - Falmesino Abdul Hamid
* Fungsi detectAnomaliesEMA menghitung Exponential Moving Average (EMA)
* dan mendeteksi anomali jika nilai sensor menyimpang terlalu jauh dari EMA.
*/
function detectAnomaliesEMA($data, $alpha = 0.2, $threshold = 2.0) {
$ema = $data[0]; // Inisialisasi dengan data pertama
$anomalies = [];
$emaValues = [$ema];
// Hitung EMA untuk setiap data
for ($i = 1; $i < count($data); $i++) {
$ema = $alpha * $data[$i] + (1 - $alpha) * $ema;
$emaValues[] = $ema;
}
// Hitung rata-rata deviasi absolut antara data dan EMA
$sumDiff = 0;
foreach ($data as $i => $value) {
$sumDiff += abs($value - $emaValues[$i]);
}
$meanDiff = $sumDiff / count($data);
// Deteksi anomali: jika selisih absolut melebihi threshold * meanDiff
foreach ($data as $i => $value) {
if (abs($value - $emaValues[$i]) > $threshold * $meanDiff) {
$anomalies[$i] = $value;
}
}
return $anomalies;
}
// 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);
?>
PORTS
DEBUG CONSOLE
TIMELINE
TERMINAL
zsh
➜ sda-praktikum-15 ls -la
total 40
drwxr-xr-x 7 falmesino staff 224 Feb 9 05:00 .
drwxr-xr-x 40 falmesino staff 1280 Feb 9 04:59 ..
-rw-r--r-- 1 falmesino staff 1539 Feb 9 07:12 1.php
-rw-r--r-- 1 falmesino staff 1255 Feb 9 06:33 2.php
-rw-r--r-- 1 falmesino staff 2479 Feb 9 06:42 3.php
-rw-r--r-- 1 falmesino staff 1364 Feb 9 06:47 4.php
-rw-r--r-- 1 falmesino staff 3505 Feb 9 07:13 5.php
➜ sda-praktikum-15 php 4.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-15
Launchpad
0 0
Ln 44, Col 23
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier


sda-praktikum-15
2.php
3.php
4.php
5.php
5.php
<?php
/**
* 231232028 - Falmesino Abdul Hamid
*/
class Node {
public $x;
public $y;
public $g; // biaya dari start ke node ini
public $h; // esitmasi biaya dari node ini ke goal (heuristik)
public $f; // total biaya (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 - $goal->y);
}
// Fungsi untuk mendapatkan tetangga (4 arah: atas, kanan, bawah, kiri)
function getNeighbors($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 = jalan bebas
$neighbors[] = new Node($nx, $ny);
}
}
return $neighbors;
}
function aStar($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 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) {
// Lewati jika neighbor sudahh ada di closedSet
$skip = false;
foreach ($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
// Periksa apakah neighbor sudah ada di openSet dengan nilai g lebih rendah
$inOpen = false;
foreach ($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; // Jalur tidak ditemukan
}
// Contoh grid (0 = jalan bebas, 1 = rintangan)
$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";
}
?>
TERMINAL
zsh
sda-praktikum-15 ls
1.php 2.php 3.php 4.php 5.php
sda-praktikum-15 php 5.php
Jalur ditemukan:
(0, 0)(0, 1)(0, 2)(1, 2)(2, 2)(2, 3)(3, 3)(4, 3)(4, 4)
sda-praktikum-15
Launchpad
Ln 37, Col 36
Spaces: 2
UTF-8
LF
PHP
Colorize
Colorize: 0 variables
Prettier