

LAPORAN TUGAS 3
DATA MINING
“Euclidean dan Manhattan Distance”



Oleh :

NAMA : Nazar Aulia

NIM : 2101301098

Dosen Pengampu :

Jaka Permadi S.Si., M.Cs

PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI TANAH LAUT
2022

KATA PENGANTAR

Puji dan syukur kita haturkan kepada Allah SWT karena berkat rahmat dan hidayahnya penyusun dapat menyelesaikan Laporan Praktikum Data Mining. Tujuan penulisan laporan ini adalah untuk memenuhi tugas dari Bapak **Jaka Permadi, S.Si., M.Cs** pada mata kuliah Data Mining untuk menambah wawasan tentang *Euclidean* dan *Manhattan Distance*. Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat kekurangan baik pada ketepatan penjawaban tugas, teknis penulisan, maupun materi. Untuk kritik dan saran yang membangun dari semua pihak sangat diharapkan demi penyempurnaan pembuatan laporan.

Pelaihari, 16 Oktober 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB I	1
PENDAHULUAN	1
1. 1 Latar Belakang	1
1. 2 Rumusan Masalah	1
1. 3 Tujuan.....	1
1. 4 Manfaat.....	2
BAB II.....	3
TINJAUAN PUSTAKA	3
2.1 Pengertian <i>Euclidean</i> dan <i>Manhattan Distance</i>	3
2.1.1 Pengertian <i>Euclidean Distance</i>	3
2.1.2 Pengertian <i>Manhattan Distance</i>	3
2.2 Perbedaan <i>Euclidean</i> dengan <i>Manhattan Distance</i>	3
BAB III	5
PRAKTIKUM	5
3.1 Menentukan Titik Setiap Bangunan dari Google Maps	5
3.2 Membuat Database	6
3.3 Membangun Web PHP	7
BAB IV	11
PENUTUP.....	11
4.1 Kesimpulan.....	11
4.2 Saran	11
DAFTAR PUSTAKA	12

BAB I

PENDAHULUAN

1. 1 Latar Belakang

Pada saat ingin menentukan rute perjalanan, kita pasti akan mengambil rute terdekat agar dapat menghemat pengeluaran. Ketika kita mencari suatu bangunan di dalam Google Maps, kita akan mencari bangunan terdekat dengan fasilitas yang sama daripada yang lebih jauh.

Nah, di dalam Maps, kita nantinya disuguhkan rute perjalanan terpendek, yang mana rute terpendek itu kita sebut sebagai *manhattan distance*, karena setiap jalan itu tidak mungkin akan lurus saja. Lalu apa yang akan kita sebut sebagai *euclidean distance*? Untuk itu, mari kita sama-sama bahas di dalam praktikum ini.

Pada desa Tegal Sari di kecamatan Satui, terdapat 6 buah bangunan yang terdaftar di dalam Google Maps. Dengan berdasarkan letak latitude dan longitude nya, bangunan apa yang paling dekat dengan bangunan kantor desa tegal sari, untuk itu, saya membuat program yang dapat menghitung jarak terdekat dengan menarik garis *euclidean* pada setiap titik yang terhubung dengan kantor desa tegal sari.

1. 2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka masalah dapat dirumuskan sebagai berikut.

1. Apa itu *euclidean* dan *manhattan distance*?
2. Bagaimana cara membuat aplikasi penentuan jarak terdekat di desa Tegal Sari, kec. Satui?
3. Bagaimana kode di dalam program yang memuat aplikasi penentuan jarak terdekat tersebut?

1. 3 Tujuan

Berdasarkan rumusan masalah di atas, dapat dinyatakan tujuan pembuatan sebagai berikut.

1. Mengetahui tentang *euclidean* dan *manhattan distance*.

2. Mempraktikkan pembuatan program aplikasi penentuan jarak terdekat di desa Tegal Sari, kec. Satui.
3. Mengetahui sintaks kode program yang memuat aplikasi penentuan jarak terdekat.

1. 4 Manfaat

Dengan membuat program program yang memuat penentuan jarak terdekat ini, diharapkan mahasiswa dapat mengetahui lebih banyak tentang pengaplikasian *eulidean* dan *manhattan distance* di dalam kehidupan.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian *Euclidean* dan *Manhattan Distance*

2.1.1 Pengertian *Euclidean Distance*

Euclidean distance adalah perhitungan untuk mengukur jarak dua titik dalam *euclidean space* yang mempelajari hubungan antara sudut dan jarak. *Euclidean space* diperkenalkan oleh Euclid, seorang matematikawan dari Yunani sekitar tahun 300 sebelum era umum untuk mempelajari hubungan antara sudut dan jarak. *Euclidean* ini berkaitan dengan Teorema Pythagoras dan biasanya diterapkan pada 1, 2 dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

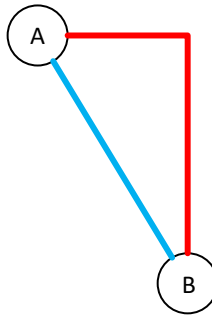
Jarak *euclidean* biasa diterapkan untuk membantu proses klasifikasi pada *data mining*. Contoh algoritma yang menggunakan jarak ini adalah k-nearest neighbor. Jarak *euclidean* juga bisa diterapkan untuk algoritma *image retrieval* (temu kembali citra).

2.1.2 Pengertian *Manhattan Distance*

Manhattan distance atau jarak *Manhattan* sering juga disebut *Taxicab distance* atau *City Block distance*. *Manhattan distance* adalah metrik ukur yang umumnya digunakan untuk menghitung jarak antara dua titik data dalam jalur seperti *grid*.

2.2 Perbedaan *Euclidean* dengan *Manhattan Distance*

Dalam mengukur kedekatan data, seringkali menggunakan *euclidean distance*. Apa beda *euclidean distance* dengan *manhattan distance*? Perhatikan ilustrasi berikut.



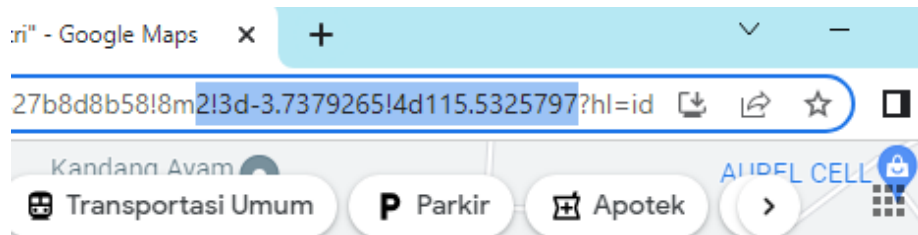
Garis berwarna biru dari A ke B, memiliki panjang yang lebih kecil jika dibandingkan dengan garis berwarna merah. Mungkin kita masih ingat dengan rumus pythagoras, $c^2 = a^2 + b^2$. Jika diperhatikan, panjang c (sisi miring segitiga) pastilah lebih pendek daripada panjang a yang ditambahkan dengan panjang b . Dengan kata lain, jarak dari A ke B pada ilustrasi di atas akan lebih pendek jika kita tarik garis lurus dari A ke B (garis biru), daripada menggunakan garis merah. Panjang garis merah di atas diukur dengan menggunakan *manhattan distance*, sementara garis biru menggunakan *euclidean distance*.

BAB III

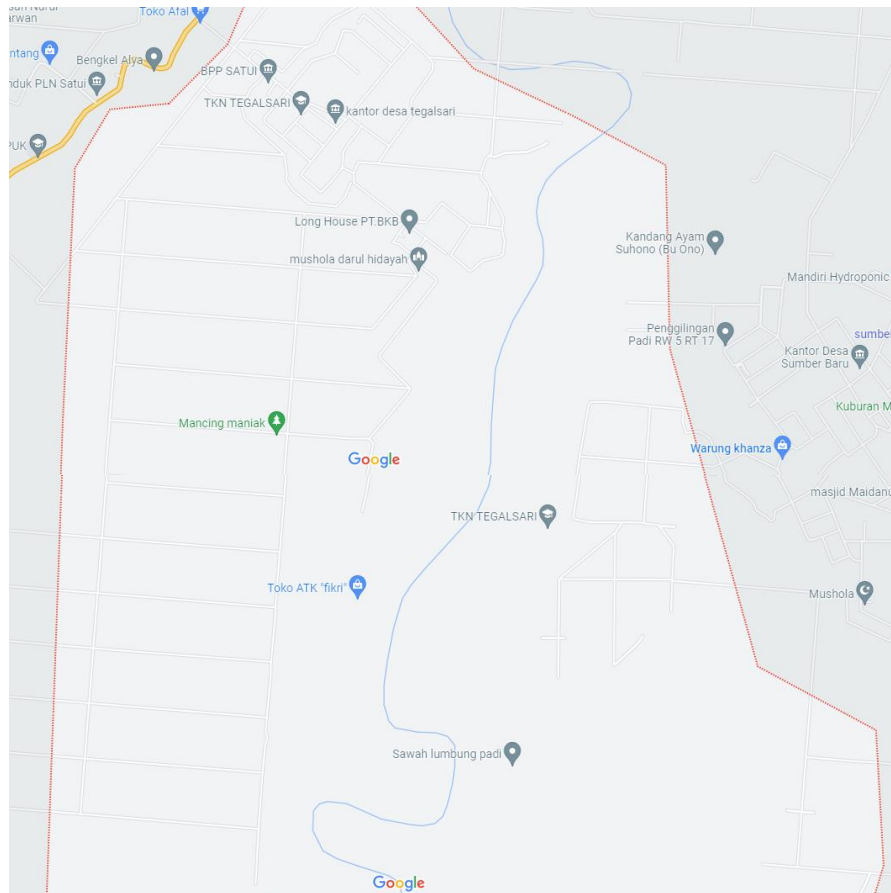
PRAKTIKUM

3.1 Menentukan Titik Setiap Bangunan dari Google Maps

Untuk mengetahui titik koordinat suatu bangunan, kita bisa melihat di link webnya, titik itu biasanya berada pada paling ujung bagian link, dengan tanda koma yang diganti dengan !4d.



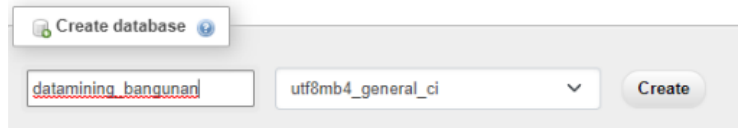
Bangunan-bangunan yang ada di dalam Maps desa Tegalsari adalah sebagai berikut.



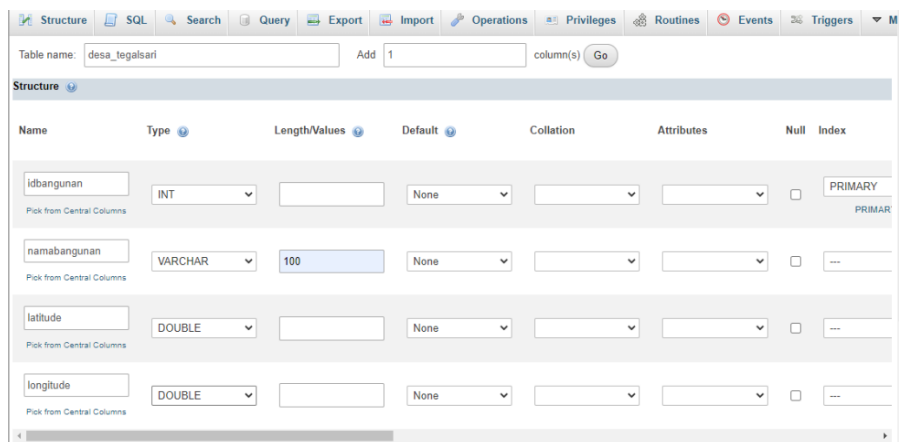
3.2 Membuat Database

Buka PHPMYAdmin, buat database baru dengan nama datamining_bangunan.

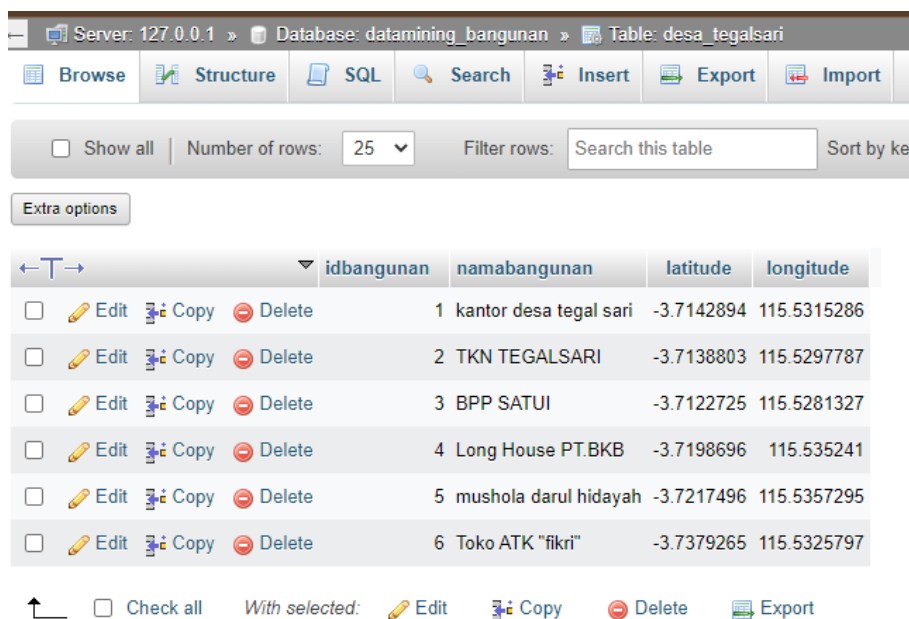
Databases



Kemudian buat tabel baru dengan nama desa_tegalsari yang atributnya ada idbangunan, namabangunan, latitude, dan longitude yang tipe datanya bisa dilihat seperti gambar berikut.



Setelah itu, kita masukkan data setiap bangunan desa Tegal Sari yang sebelumnya kita dapat dari Google Maps.



	idbangunan	namabangunan	latitude	longitude
<input type="checkbox"/> Edit Copy Delete	1	kantor desa tegal sari	-3.7142894	115.5315286
<input type="checkbox"/> Edit Copy Delete	2	TKN TEGALSARI	-3.7138803	115.5297787
<input type="checkbox"/> Edit Copy Delete	3	BPP SATUI	-3.7122725	115.5281327
<input type="checkbox"/> Edit Copy Delete	4	Long House PT.BKB	-3.7198696	115.535241
<input type="checkbox"/> Edit Copy Delete	5	mushola darul hidayah	-3.7217496	115.5357295
<input type="checkbox"/> Edit Copy Delete	6	Toko ATK "fikri"	-3.7379265	115.5325797

3.3 Membangun Web PHP

Dalam membangun sebuah web, kita sebelumnya harus membuat folder baru di dalam `C:→xampp→htdocs` dengan nama folder `datamining-bangunan`. Kemudian buka Visual Studio Code, tambahkan file `index.php` dan `koneksi.php` di dalamnya

Pada `koneksi.php`, file ini digunakan untuk menghubungkan ke database `datamining_bangunan`.

```
index.php \  koneksi.php CA:\datamining_penduduk_apk  koneksi.php X
koneksi.php > ...
1  <?php
2  $conn=mysqli_connect('localhost','root','','datamining_bangunan');
3  ?>
4
```

Kemudian pada file `index.php` akan membangun tampilan dengan dokumen html. Seperti pada tabel di database sebelumnya, kita ambil dulu data dari tabel tersebut pada `index.php`.

```
index.php \ X  index.php CA:\datamining_penduduk_apk  koneksi.php
index.php > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>bangunan terdekat</title>
9      <h2>
10         <center>APLIKASI PENENTU JARAK DEKAT</center>
11     </h2>
12 </head>
13
14 <body>
15
16     <?php
17
18     include "koneksi.php";
19
20     $i = 0;
21     $query = mysqli_query($conn, "SELECT * FROM desa_tegalsari ORDER BY idbangunan");
22     $jumdata = mysqli_num_rows($query);
23     while ($data = mysqli_fetch_array($query)) {
24         $bangunan[$i] = $data['namabangunan'];
25         $lat[$i] = $data['latitude'];
26         $long[$i] = $data['longitude'];
27         $i++;
28     }
```

Setelah diambil datanya, saya akan menampilkan bangunan kantor desa tegal sari beserta latitude dan longitude nya yang ada pada indeks ke-0. Untuk menampilkan, kodenya adalah sebagai berikut.

```
29
30     echo "Nama Bangunan : $bangunan[0]";
31     echo "<br>($lat[0], $long[0])";
32
```

APLIKASI PENENTU JARAK DEKAT

Nama Bangunan : kantor desa tegal sari
(-3.7142894, 115.5315286)

Sebelum kita memasuki algoritma untuk normalisasi, kita ambil dahulu data minimum dan maksimum dari latitude dan longitude dari query berikut.

```
index.php > html
32
33 $query = mysqli_query($conn, "SELECT MIN(latitude) as latmin, MAX(latitude) as latmax, MIN(longitude) as longmin, MAX
(longitude) as longmax FROM `desa_tegalsari`");
34 $data = mysqli_fetch_array($query);
35 $latmin = $data['latmin'];
36 $latmax = $data['latmax'];
37 $longmin = $data['longmin'];
38 $longmax = $data['longmax'];
39
```

Setelah itu kita buat algoritma untuk mencari hasil normalisasi di setiap latitude dan longitude bangunan, yang mana untuk normalisasi latitude akan menggunakan variable `$latnorm[$i]`, variable ini akan menghasilkan pembagian dari rumus normalisasi *min-max feature scaling*. Sedangkan untuk normalisasi longitude, akan menggunakan variable `$longnorm[$i]` yang juga akan menghasilkan pembagian dari rumus normalisasi *min-max feature scaling*.

```
40 for ($i = 0; $i < $jumdata; $i++) {
41     $latnorm[$i] = ($lat[$i] - $latmin) / ($latmax - $latmin);
42     $longnorm[$i] = ($long[$i] - $longmin) / ($longmax - $longmin);
43 }
44
```

Kemudian barulah kita hitung jarak setiap data dengan bangunan kantor desa tegal sari. Karena kantor desa tegal sari merupakan data ke-0, maka kita lakukan perulangan dari indeks array ke-1 saja. Sehingga kantor desa tegal sari tidak dihitung jaraknya ke dirinya sendiri.

```
index.php > html
40
47
48 echo "<br>";
49 echo "<br>";
50 echo "Data jarak :";
51 echo "<br>";
52
53 //awal tabel
54 echo "<table border='1' style='border-collapse: collapse;'>";
55 echo "<th style='padding-left:10px; padding-right:10px;'> Nama Bangunan </th>";
56 echo "<th style='padding-left:10px; padding-right:10px;'> Latitude </th>";
57 echo "<th style='padding-left:10px; padding-right:10px;'> Longitude </th>";
58 echo "<th style='padding-left:10px; padding-right:10px;'> Latitude (normal) </th>";
59 echo "<th style='padding-left:10px; padding-right:10px;'> longitude (normal) </th>";
60 echo "<th style='padding-left:10px; padding-right:10px;'> Jarak </th>";
61
62 for ($i = 1; $i < $jumdata; $i++) {
63     $jarak[$i] = sqrt(pow($latnorm[$i] - $latnorm[0], 2) + pow($longnorm[$i] - $longnorm[0], 2));
64 }
```

Kemudian kita masukkan datanya ke dalam tabel.

```

65
66 //memasukkan data ke tabel berdasarkan baris(row)
67 echo "<tr>";
68 echo "<td style='padding-left=10px; padding-right=10px;'> $bangunan[$i] </td>";
69 echo "<td style='padding-left=10px; padding-right=10px;'> $lat[$i] </td>";
70 echo "<td style='padding-left=10px; padding-right=10px;'> $long[$i] </td>";
71 echo "<td style='padding-left=10px; padding-right=10px;'> $latnorm[$i] </td>";
72 echo "<td style='padding-left=10px; padding-right=10px;'> $longnorm[$i] </td>";
73 echo "<td style='padding-left=10px; padding-right=10px;'> $jarak[$i] </td>";
74 echo "</tr>";
75 }
76 echo "</table>";
77 //table selesai
78

```

Hasilnya adalah sebagai berikut.

APLIKASI PENENTU JARAK DEKAT

Nama Bangunan : kantor desa tegal sari
(-3.7142894, 115.5315286)

Data jarak :

Nama Bangunan	Latitude	Longitude	Latitude (normal)	longitude (normal)	Jarak
TKN TEGALSARI	-3.7138803	115.5297787	0.93732751227879	0.21667017691573	0.23089832479178
BPP SATUI	-3.7122725	115.5281327	1	0	0.45387811488868
Long House PT BKB	-3.7198696	115.535241	0.70386294534965	0.93569660909819	0.53490332252696
mushola darul hidayah	-3.7217496	115.5357295	0.63058002650659	1	0.62478399190562
Toko ATK "fikri"	-3.7379265	115.5325797	0	0.58537805391717	0.93171137996678

Setelah itu, kita cari jarak terdekat dengan kantor desa tegal sari tadi dengan menggunakan perulangan dari indeks array 1.

```

index.php > html
75 }
76 echo "</table>";
77 //table selesai
78
79 if ($jumdata > 0) {
80     for ($i = 1; $i < $jumdata; $i++) {
81         if ($i == 1) {
82             $jarakterdekat = $jarak[$i];
83             $bangunanterdekat = $bangunan[$i];
84         } else {
85             if ($jarak[$i] < $jarakterdekat) {
86                 $jarakterdekat = $jarak[$i];
87                 $bangunanterdekat = $bangunan[$i];
88             }
89         }
90     }
91 }
92
93 echo "<br>";
94 echo "<br>";
95 echo "Bangunan Terdekat dengan kantor desa tegal sari: $bangunanterdekat";
96 echo "<br>";
97 echo "Jarak : $jarakterdekat";
98
99
100 ?>
101
102 </body>
103
104 </html>

```

Hasilnya adalah sebagai berikut.

APLIKASI PENENTU JARAK DEKAT

Nama Bangunan : kantor desa tegal sari
(-3.7142894, 115.5315286)

Data jarak :

Nama Bangunan	Latitude	Longitude	Latitude (normal)	longitude (normal)	Jarak
TKN TEGALSARI	-3.7138803	115.5297787	0.93732751227879	0.21667017691573	0.23089832479178
BPP SATUI	-3.7122725	115.5281327	1	0	0.45387811488868
Long House PT.BKB	-3.7198696	115.535241	0.70386294534965	0.93569660909819	0.53490332252696
mushola darul hidayah	-3.7217496	115.5357295	0.63058002650659	1	0.62478399190562
Toko ATK "fikri"	-3.7379265	115.5325797	0	0.58537805391717	0.93171137996678

Bangunan Terdekat dengan kantor desa tegal sari: TKN TEGALSARI
Jarak : 0.23089832479178

BAB IV

PENUTUP

4.1 Kesimpulan

Dalam pengukuran kedekatan data, terdapat penarikan jarak terdekat yang dinamakan dengan *euclidean distance* dan *manhattan distance*. Keduanya merupakan suatu penarikan dari jarak terdekat, namun bedanya adalah kalau *euclidean* langsung melakukan penarikan garis lurus dari titik ke titik, sedangkan *manhattan* menentukan jarak terdekatnya dengan mencari jalur terpendek, walaupun tidak lurus.

Di dalam praktik yang telah dikerjakan, saya melakukan pengambilan data dari bangunan yang terdaftar di Maps desa Tegal Sari, kemudian menormalisasikan data koordinatnya, lalu mencari jarak terdekatnya dengan menggunakan rumus *pythagoras*.

Dengan mengambil bangunan kantor desa tegal sari sebagai objek, lalu menggunakan metode yang telah diajarkan, saya dapat menentukan jarak terdekat dari kantor desa tegal sari tersebut.

4.2 Saran

Demikian laporan yang telah dibuat, tentu masih banyak hal yang kurang, jika berkenan tolong beri komentar dan saran agar sekiranya saya dapat menjadi lebih baik lagi.

DAFTAR PUSTAKA

(Fauzan, 2019)

(Trivusi, 2022)