

LAMPIRAN

1. SOURCE CODE ALGORITMA

```
<?php

namespace App\Modules;

use App\Models\Gejala;
use App\Models\Penyakit;

class NaiveBayes
{
    public $listGejala = [];
    public $allGejala = [];
    public $dataPenyakit = [];
    public $nilaiGejala = [];
    public $nilaiPenyakit = [];
    public $nilaiPerkalian = [];
    public $nilaiHasil = [];

    public function __construct($listGejala = [], $allGejala)
    {
        $this->listGejala = $listGejala;
        $this->allGejala = $allGejala;
        $this->dataPenyakit = $this->getPenyakit();
        $this->nilaiGejala = $this->getNilaiGejala();
        $this->nilaiPenyakit = $this->getNilaiPenyakit();
        $this->nilaiPerkalian = $this->getNilaiPerkalian();
        $this->nilaiHasil = $this->getNilaiHasil();
    }

    public function getPenyakit()
    {
        $gejalaIds = [];
        for ($i = 0; $i < count($this->listGejala); $i++)
            array_push($gejalaIds, $this->listGejala[$i]->id);

        $penyakit = Penyakit::get();

        for ($i = 0; $i < count($penyakit); $i++) {
            $res = $penyakit[$i];
            $penyakit[$i]->gejalas = Gejala::whereIn('id',
                $gejalaIds)->with(['aturans' => function ($query) use ($res) {
```

```

        $query->where('penyakit_id', $res->id);
    ]])->get();
    }

    return $penyakit;
}

public function getNilaiGejala()
{
    $penyakit = $this->dataPenyakit;

    for ($i = 0; $i < count($penyakit); $i++) {
        $total = 0;
        foreach ($penyakit[$i]->gejalas as $item)
            $total = $total + (count($item->aturans) == 0 ? 0.1
: $item->aturans[0]->nilai);

        $penyakit[$i]->total_nilai_bobot_gejala = $total;
    }

    return $penyakit;
}

public function getNilaiPenyakit()
{
    $nilaiPenyakit = $this->nilaiGejala;

    for ($i = 0; $i < count($nilaiPenyakit); $i++)
        // $nilaiPenyakit[$i]->total_nilai_bobot_penyakit =
$nilaiPenyakit[$i]->total_nilai_bobot_gejala /
count($nilaiPenyakit);
        $nilaiPenyakit[$i]->total_nilai_bobot_penyakit =
$nilaiPenyakit[$i]->bobot;

    return $nilaiPenyakit;
}

public function getNilaiPerkalian()
{
    $nilaiPerkalian = $this->nilaiPenyakit;

    $probabilitasPerkalian = 0;

    for ($i = 0; $i < count($nilaiPerkalian); $i++) {

```

```

        // Hitung total nilai perkalian bobot gejala
        $totalNilaiPerkalianBobotGejala = 0;
        for ($index = 0; $index < count($nilaiPerkalian[$i]-
>gejalas); $index++) {
            $item = $nilaiPerkalian[$i]->gejalas[$index];
            $nilaiAturan = (count($item->aturans) == 0 ? 0.1 :
$item->aturans[0]->nilai);
            $totalNilaiPerkalianBobotGejala = $index == 0 ?
$nilaiAturan : $totalNilaiPerkalianBobotGejala * $nilaiAturan;
        }

        // Hitung total nilai perkalian
        $perkalian = $totalNilaiPerkalianBobotGejala *
$nilaiPerkalian[$i]->total_nilai_bobot_penyakit;

        // Hitung probabilitas perkalian
        $probabilitasPerkalian = $probabilitasPerkalian +
$perkalian;

        $nilaiPerkalian[$i]->total_nilai_perkalian_bobot_gejala
= $totalNilaiPerkalianBobotGejala;
        $nilaiPerkalian[$i]->nilai_perkalian = $perkalian;
    }

    for ($i = 0; $i < count($nilaiPerkalian); $i++)
        $nilaiPerkalian[$i]->probabilitas_perkalian =
$probabilitasPerkalian;

    return $nilaiPerkalian;
}

public function getNilaiHasil()
{
    $nilaiHasil = $this->nilaiPerkalian;

    // Hitung hasil
    for ($i = 0; $i < count($nilaiHasil); $i++)
        $nilaiHasil[$i]->nilai_hasil = $nilaiHasil[$i]-
>nilai_perkalian / $nilaiHasil[$i]->probabilitas_perkalian;

    return $nilaiHasil;
}
}

```