

Manual Book Framework BLT Menggunakan ci4

Disusun & dibuat oleh:

Syahdina Annisa (21220031)

SI-5L



Untuk memenuhi penilaian tugas UAS dalam bentuk manual book, project Framework. Yang kami hormati Bapak Rolly Yesputra, S.Kom, M.Kom. Manual book sendiri adalah buku panduan yang berisi seputar informasi yang ditujukan dan diberikan kepada si pembaca. Sebagai buku panduan dalam memberikan informasi mengenai data yang kami angkat mengenai BLT (Bantuan Langsung Tunai). Untuk memecahkan masalah pada proses penyaluran program BLT.

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatu, Puji syukur kami panjatkan kepada Allah SWT, karena atas berkat dan rahmatNya “Manual Book –Penerimaan BLT di Dadimulyo ” ini dapat terselesaikan. Manual Book ini kami buat untuk memberikan bantuan petunjuk secara teknis tentang cara mengoperasikan layanan atau aplikasi. Instruksi penggunaan disajikan dengan selugas mungkin sehingga dapat mudah dipahami oleh semua pihak.

Manual Book penerimaan BLT di Dadimulyo ini merupakan website yang di buat dengan menggunakan metode Naïve bayes. Tahapan analisa yang dipakai berdasarkan metode CRISP-DM Dengan tujuan untuk mengetahui bagaimana penerapan algoritma Naive bayes dalam mengklasifikasikan penerima Bantuan Langsung Tunai di desa Dadimulyo agar dapat hasil yang optimal.

Penyusunan Manual Book ini tentunya masih jauh dari kata sempurna, baik secara konteks maupun konten, untuk itu saya membuka diri untuk saran dan kritik demi perbaikan ke depan. Terima kasih saya ucapkan kepada semua pihak yang telah banyak memberikan kontribusi dalam penyusunan Manual Book ini, semoga Allah SWT membalas segala kebaikan semua pihak yang telah membantu. Semoga bermanfaat untuk semua.

Kisaran,20 Januari 2024

Hormat Kami

DAFTAR ISI

Table of Contents

DAFTAR ISI.....	2
BAB 1 PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan	4
BAB 2 ISI.....	5
2.1 Penjelasan BLT (Bantuan Langsung Tunai)	5
2.2 Metode yang digunakan	5
2.3 Tahapan Analisa.....	5
2.4 Cara login ke dashboard.....	5
Gambar 1.1	6
Gambar 1.2.....	7
Gambar 1.3.....	7
Gambar 1.4.....	8
2.5 Tampilan-tampilan coding	8
Gambar 2.1	8
.....	9
Gambar 2.2 (Training Model).....	9
.....	9
Gambar 2.3 (Testing Model).....	10
.....	10
Gambar 2.4 (Routes ->Admin)	10
Gambar 2.5 (Menampilkan coding halaman dashboard)	10
Gambar 2.6 (Tampilan coding db data testing).....	11
Gambar 2.7 (Tampilan coding db data training).....	11
Gambar 2.8 (Tampilan coding halaman dashboard).....	12
Gambar 2.9 (Tampilan coding halaman dashboard	12
Gambar 3.0 (Tampilan coding halaman data training)	12

Gambar 3.1 (Tampilan coding db data testing).....	13
Gambar 3.2 (Tampilan pada controller).....	13
Gambar 3.3 (Tampilan codingan content)	14
BAB 3 PENUTUP	15
3.1 Kesimpulan	15
3.1.1 Saran.....	15

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pengklasifikasian Naïve Bayes penerimaan di BLT di Dadimulyo, mengklasifikasikan calon penerima BLT. Untuk memecahkan suatu masalah pada proses penyaluran program bantuan langsung tunai. Belum dapat berjalan secara optimal & tepat sasaran.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka permasalahan yang akan dibahas dapat dirumuskan sebagai berikut :

- a) Apa pengertian dari BLT?
- b) Metode apa saja yang digunakan dalam menganalisis data di dalam website tersebut?
- c) Bagaimana langkah-langkah cara mengoperasikan website nya dengan benar
- d) Mekanisme bagaimana cara kerja website
- e) Tampilan-tampilan gambar & codingan website

1.3 Tujuan

Tujuan dari website ini dari topik yang kami bahas mengenai penerimaan BLT di desa Dadimulyo yang tidak lain adalah.

- 1 Untuk memecahkan suatu masalah pada proses penyaluran program bantuan langsung tunai.
- 2 Untuk mengetahui bagaimana penerapan algoritma Naive bayes dalam mengklasifikasikan penerima Bantuan Langsung Tunai di desa Dadimulyo agar mendapatkan hasil yang optimal.

BAB 2 ISI

2.1 Penjelasan BLT (Bantuan Langsung Tunai)

Bantuan Langsung Tunai (BLT): BLT seringkali merujuk kepada program bantuan tunai yang diselenggarakan oleh pemerintah atau organisasi non-pemerintah untuk membantu individu atau keluarga yang membutuhkan bantuan finansial langsung. Dalam bentuk tunai (uang).

2.2 Metode yang digunakan

Metode yang dipakai dalam website ini adalah Naïve Bayes. Metode Naive Bayes adalah salah satu algoritma klasifikasi yang didasarkan pada teorema Bayes. Algoritma ini dianggap "naive" karena mengasumsikan bahwa semua fitur yang digunakan untuk klasifikasi adalah independen satu sama lain, meskipun dalam situasi nyata, fitur-fitur tersebut mungkin tidak sepenuhnya independen.

2.3 Tahapan Analisa

Di dalam pengklasifikasian ini berdasarkan menggunakan tahapan analisa CRSIP-DM. CRISP-DM (Cross Industry Standard Process for Data Mining) adalah suatu metodologi umum yang digunakan dalam proses analisis data dan data mining. Metodologi ini terdiri dari sejumlah tahapan yang dapat membimbing praktisi dalam merancang dan melaksanakan proyek data mining. Berikut adalah tahapan-tahapan dalam metodologi CRISP-DM;

1. Understanding the Business Understanding
2. Data Understanding
3. Data Understanding
4. Modeling
5. Evaluation
6. Deployment
7. Business Understanding

2.4 Cara login ke dashboard

A.Login Dashboard

Cara masuk ke dashboard administrator website:

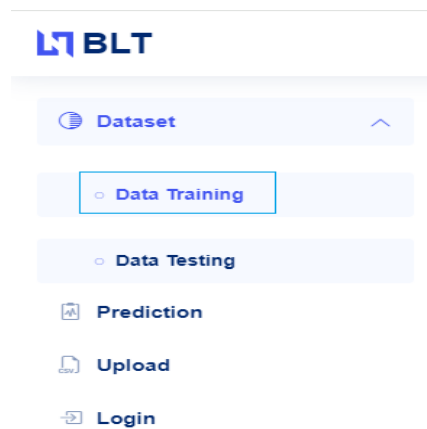
1. Aktifkan xampp, Ketik di terminal Php Spark Serve, maka anda akan di arahkan ke browser
2. Ketik url : localhost:8080/admin Akan tampil seperti berikut:



Gambar 1.1

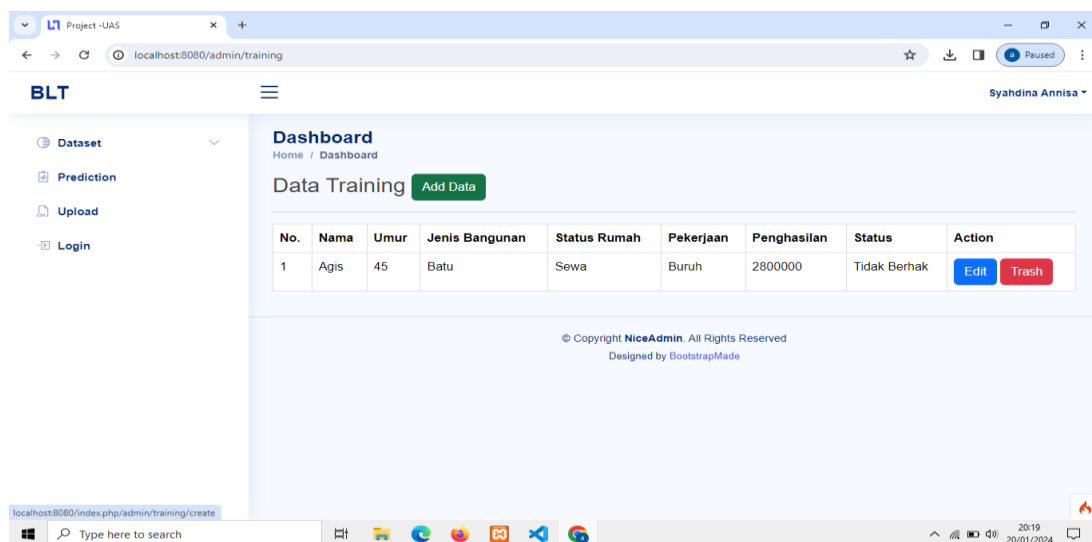
B. side menu

Halaman Ini Berisi Menu yang tersedia di dalam website. menu yang tersedia antara lain Dataset, Prediction, Upload, Login. Pilih/Klik Judul yang diinginkan.



Gambar 1.2

Ini adalah tampilan dari side menu data training, cara penggunaan nya adalah klik Add Data. Maka akan tampil halaman sebagai berikut



Gambar 1.3

Berikut tampilan dari isi field yang kosong sesuai kebutuhan

localhost:8080/index.php/admin/training/create

BLT

- Dataset
- Prediction
- Upload
- Login

Dashboard
Home / Dashboard

Nama

Umur

Jenis Bangunan

Status Rumah

Pekerjaan

Penghasilan

Status

Save

Gambar 1.4

2.5 Tampilan-tampilan coding

Berikut adalah bentuk folder dari data *training* dan data *testing*. Yang mana terdiri dari komponen CRUD.



Gambar 2.1

```

<?php

namespace App\Models;

use CodeIgniter\Model;

class TrainingModel extends Model
{
    protected $DBGroup          = 'default';
    protected $table            = 'trainings';
    protected $primaryKey       = 'training_id';
    protected $useAutoIncrement = true;
    protected $returnType       = 'object';
    protected $useSoftDeletes   = false;
    protected $protectFields    = true;
    protected $allowedFields    =

["training_id", 'nama', 'umur', 'jenis_bangunan', 'status_rumah', 'pekerjaan', 'penghasilan', 'status'];

    // Dates
    protected $useTimestamps = false;
    protected $dateFormat    = 'datetime';
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';

    // Validation
    protected $validationRules = [];
    protected $validationMessages = [];
    protected $skipValidation = false;
    protected $cleanValidationRules = true;

    // Callbacks
    protected $allowCallbacks = true;
    protected $beforeInsert   = [];
    protected $afterInsert    = [];
    protected $beforeUpdate   = [];
    protected $afterUpdate    = [];
    protected $beforeFind     = [];
    protected $afterFind      = [];
    protected $beforeDelete   = [];
    protected $afterDelete    = [];
}

```

Gambar 2.2 (Training Model)

```

<?php

namespace App\Models;

use CodeIgniter\Model;

class TestingModel extends Model
{
    protected $table            = 'testings';
    protected $primaryKey       = 'testing_id';
    protected $useAutoIncrement = true;
    protected $returnType       = 'object';
    protected $useSoftDeletes   = false;
    protected $protectFields    = true;
    protected $allowedFields    =

["testing_id", 'testingnama', 'testingumur', 'testingjenis_bangunan', 'testingstatus_rumah', 'testingpekerjaan', 'testingpenghasilan', 'testingstatus'];

    // Dates
    protected $useTimestamps = false;
    protected $dateFormat    = 'datetime';
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';

    // Validation
    protected $validationRules = [];
    protected $validationMessages = [];
    protected $skipValidation = false;
    protected $cleanValidationRules = true;

    // Callbacks
    protected $allowCallbacks = true;
    protected $beforeInsert   = [];
    protected $afterInsert    = [];
    protected $beforeUpdate   = [];
    protected $afterUpdate    = [];
    protected $beforeFind     = [];
    protected $afterFind      = [];
    protected $beforeDelete   = [];
    protected $afterDelete    = [];
}

```

Gambar 2.3 (Testing Model)

```
// Routes Admin
$routees->get('/admin','Admin::index');
```

Gambar 2.4 (Routes ->Admin)

```
// Menampilkan halaman dashboard
$routees->get('admin/dashboard', 'Dashboard::index');
```

Gambar 2.5 (Menampilkan coding halaman dashboard)

```
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
use CodeIgniter\Database\RawSql;
class Testing extends Migration
{
    public function up()
    {
        //Memuat struktur tabel untuk testing
        $this->forge->addField([
            'testing_id' => [
                'type' => 'INT',
                'constraint' => 11,
                'unsigned' => true,
                'auto_increment' => true,
            ],
            'testingnama' => [
                'type' => 'VARCHAR',
                'constraint' => '100',
            ],
            'testingumur' => [
                'type' => 'INT',
                'constraint' => 11,
            ],
            'testingjenis_bangunan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingstatus_rumah' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingpekerjaan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingpenghasilan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingstatus' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'updated_at' => [
                'type' => 'TIMESTAMP',
                'default' => new RawSql('CURRENT_TIMESTAMP'),
            ],
            'created_at' => [
                'type' => 'TIMESTAMP',
                'default' => new RawSql('CURRENT_TIMESTAMP'),
            ],
        ]);
        $this->forge->addKey('testing_id', true);
        $this->forge->createTable('testings');
    }
    public function down()
    {
        //
        $this->forge->dropTable('testings');
    }
}
```

Gambar 2.6 (Tampilan coding db data testing)

```
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
use CodeIgniter\Database\RawSql;

class Training extends Migration
{
    public function up()
    {
        //Memuat struktur tabel untuk training
        $this->forge->addField([
            'training_id' => [
                'type'           => 'INT',
                'constraint'     => 11,
                'unsigned'       => true,
                'auto_increment' => true,
            ],
            'nama' => [
                'type'           => 'VARCHAR',
                'constraint'     => '100',
            ],
            'umur' => [
                'type'           => 'INT',
                'constraint'     => 11,
            ],
            'jenis_bangunan' => [
                'type'           => 'VARCHAR',
                'constraint'     => '50',
            ],
            'status_rumah' => [
                'type'           => 'VARCHAR',
                'constraint'     => '50',
            ],
            'pekerjaan' => [
                'type'           => 'VARCHAR',
                'constraint'     => '50',
            ],
            'penghasilan' => [
                'type'           => 'VARCHAR',
                'constraint'     => '50',
            ],
            'status' => [
                'type'           => 'VARCHAR',
                'constraint'     => '50',
            ],
            'updated_at' => [
                'type'           => 'TIMESTAMP',
                'default'        => new RawSql('CURRENT_TIMESTAMP'),
            ],
            'created_at' => [
                'type'           => 'TIMESTAMP',
                'default'        => new RawSql('CURRENT_TIMESTAMP'),
            ],
        ]);
        $this->forge->addKey('training_id', true);
        $this->forge->createTable('trainings');
    }

    public function down()
    {
        //
        $this->forge->dropTable('trainings');
    }
}
```

Gambar 2.7 (Tampilan coding db data training)

```
<?php

namespace App\Controllers;

use App\Controllers\BaseController;
use CodeIgniter\HTTP\ResponseInterface;

class Dashboard extends BaseController
{
    public function index()
    {
        //tampilkan halaman dashboard
        return view('dashboard/index');
    }
}
```

Gambar 2.8 (Tampilan coding halaman dashboard)

```
<?php
namespace App\Controllers;
use App\Controllers\BaseController;
class Admin extends BaseController
{
    public function index()
    {
        //tampilkan halaman admin
        return view('admin/index');
    }
    public function training()
    {
        //tampilkan halaman training
        return view('admin/training');
    }
}
```

Gambar 2.9 (Tampilan coding halaman dashboard)

```
// Menampilkan Halaman Data training
$route->get('admin/training', 'Training::index');
$route->get('admin/training/create', 'Training::create', ['as' => 'training.create']);
$route->post('admin/training/store', 'Training::store');
$route->get('admin/training/edit/(:num)', 'Training::edit/$1');
$route->post('admin/training/update', 'Admin::updateTraining');
$route->post('admin/training/update/(:num)', 'Training::update/$1');
$route->get('admin/training/delete/(:num)', 'Training::delete/$1');
$route->get('admin/training/find/(:num)', 'Training::find/$1');
```

Gambar 3.0 (Tampilan coding halaman data training)

```

<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;
use CodeIgniter\Database\RawSql;

class Testing extends Migration
{
    public function up()
    {
        //Memuat struktur tabel untuk testing
        $this->forge->addField([
            'testing_id' => [
                'type' => 'INT',
                'constraint' => 11,
                'unsigned' => true,
                'auto_increment' => true,
            ],
            'testingnama' => [
                'type' => 'VARCHAR',
                'constraint' => '100',
            ],
            'testingumur' => [
                'type' => 'INT',
                'constraint' => 11,
            ],
            'testingjenis_bangunan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingstatus_rumah' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingpekerjaan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingpenghasilan' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'testingstatus' => [
                'type' => 'VARCHAR',
                'constraint' => '50',
            ],
            'updated_at' => [
                'type' => 'TIMESTAMP',
                'default' => new RawSql('CURRENT_TIMESTAMP'),
            ],
            'created_at' => [
                'type' => 'TIMESTAMP',
                'default' => new RawSql('CURRENT_TIMESTAMP'),
            ],
        ]);
        $this->forge->addKey('testing_id', true);
        $this->forge->createTable('testings');
    }

    public function down()
    {
        //
        $this->forge->dropTable('testings');
    }
}

```

Gambar 3.1 (Tampilan coding db data testing)

```

<?php
namespace App\Controllers;
use App\Controllers\BaseController;

class Admin extends BaseController
{
    public function index()
    {
        //tampilkan halaman admin
        return view('admin/index');
    }
    public function training()
    {
        //tampilkan halaman training
        return view('admin/training');
    }
}

```

Gambar 3.2 (Tampilan pada controller)

```

<?= $this->extend('layouts/admin') ?>

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

<!-- app/Views/dashboard/index.php -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>

    <!-- Include your CSS stylesheet here -->
    <link rel="stylesheet" href="/css/dashboard.css">

    <!-- Include any additional scripts or external script tags here -->
</head>

<body>
    <h3><strong>Klasifikasi Kelayakan Penerimaan BLT di Dadimulyo</strong></h3>
    <h5>Metode Naive-Bayes</h5>
    
    <small> Penerapan Naive Bayes dalam mengklasifikasi calon penerima Bantuan Langsung Tunai (BLT) di Desa Dadimulyo.
        dibuat untuk memecahkan masalah pada proses penyaluran program bantuan langsung tunai. Belum berjalan secara optimal dan tepat sasaran.
        Adapun tujuan dari pembuatan website ini adalah untuk mengetahui bagaimana penerapan Algoritma Naive Bayes dalam mengklasifikasi Penerima Bantuan Langsung Tunai di Desa Dadimulyo sehingga memperoleh hasil yang optimal.
        Metode yang digunakan dalam penelitian ini adalah metode klasifikasi Naive Bayes. Tahapan analisa data dilakukan berdasarkan metode CRISP-DM sedangkan pengujian algoritma dilakukan pada perangkat lunak python sebagai perbandingan antara hitungan manual dan hitungan perangkat lunak.
        Keuntungan penggunaan Naive Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian.
        Naive Bayes sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.
    </small>
    <small>
        Dalam teorema bayes, probabilitas atau peluang bersyarat dinyatakan sebagai:  $P(H|X) = \frac{P(X \cap H)}{P(X)}$  .
        Kemudian dapat disederhanakan menjadi rumus sebagai berikut:
        
$$P(H|X) = \frac{P(X|H) P(H)}{P(X)}$$

        Dimana: X : Data dengan class yang belum diketahui H : Hipotesis data X merupakan suatu class spesifik.
         $P(H|X)$  : Probabilitas hipotesis H berdasar kondisi X (posteriori probability)
         $P(H)$  : Probabilitas hipotesis H (prior probability)
         $P(X|H)$  : Probabilitas X berdasar kondisi pada hipotesis H
         $P(X)$  : Probabilitas dari X
    </small>
</body>
</html>

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

```

Gambar 3.3 (Tampilan codingan content)

BAB 3 PENUTUP

3.1 Kesimpulan

Dari data framework website ci4 ini dapat ditarik kesimpulannya bahwa website ini menampilkan langkah-langkah dalam mempermudah pengklasifikasian penyaluran BLT di masyarakat.

3.1.1 Saran

Semoga saya lebih bisa membuat dan mengembangkan website ini menjadi lebih baik lagi kedepannya. Dan saya ucapkan terima kasih kepada dosen pengampu Framework SI5L, Bapak Rolly Yesputra, S.Kom, M.Kom yang telah mengarahkan dan mengajarkan Framework kepada kami.

Wassalamualaikum Warahmatullahi Wabarakatu