

MATERI PHP: Instalasi dan Integrasi Frontend dengan Backend

A. Tujuan Praktikum

Materi ini bertujuan untuk **membuat aplikasi CRUD (Create, Read, Update, Delete)** sederhana dengan menggunakan **PHP Native** dan **MySQL** sebagai backend dan **HTML** untuk frontend.

B. Materi

1. Tools
2. Instalasi
3. Integrasi Frontend dengan Backend

C. Alat dan Bahan Praktikum

1. Laptop
2. Code Editor (VS Code)
3. Laragon
4. PhpMyAdmin/HeidiSQL

D. Metode Praktikum

1. Penayangan layar laptop dan memberikan panduan langkah demi langkah.
2. Penjelasan terinci setiap langkahnya kepada mahasiswa.
3. Mahasiswa beserta dosen, instruktur, asisten melakukan *coding* bersama untuk membuat program sederhana sesuai dengan materi.
4. Dosen, instruktur, asisten memberikan tugas diakhir sesi dengan studi kasus tertentu guna mereview materi yang telah disampaikan.

E. Penilaian

1. Penilaian tugas akan diberikan dari skala 1-100.
2. Tugas yang terlambat akan mengalami pengurangan nilai.

F. Persiapan Instalasi

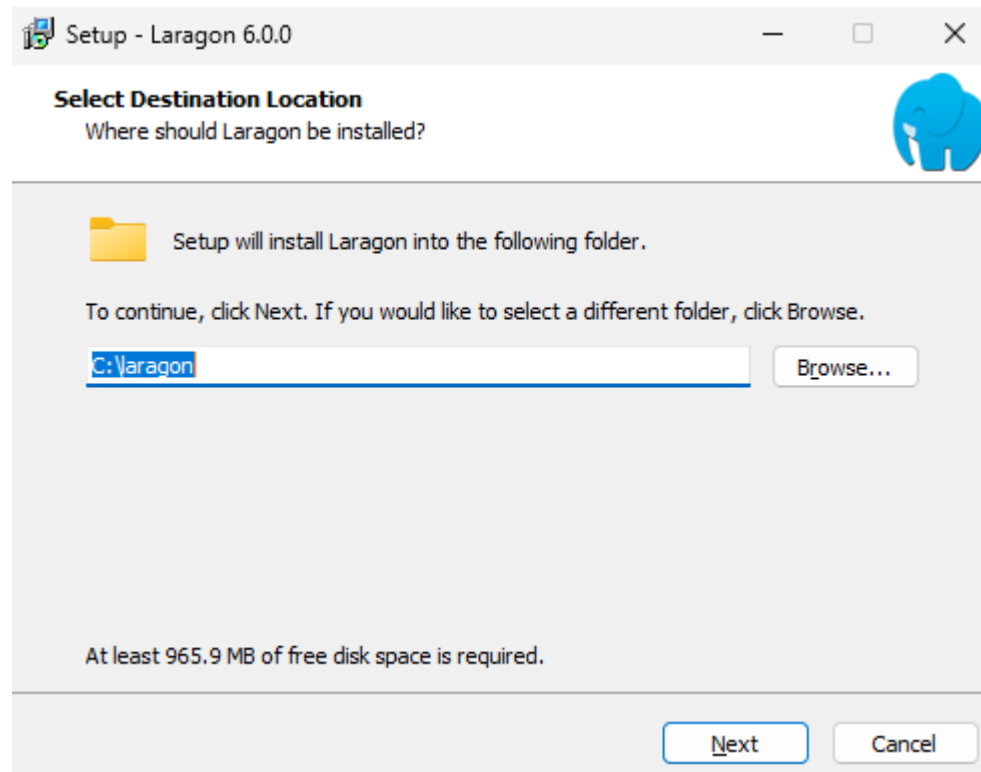
Sebelum mulai coding, pastikan Anda memiliki lingkungan pengembangan PHP dan MySQL yang siap digunakan. Anda dapat menggunakan XAMPP, Laragon, atau MAMP untuk mempermudah pengaturan server lokal.

Kita akan menggunakan Laragon dan berikut ini beberapa *file* yang perlu diunduh:

1. Laragon 6.0.0.exe - <https://github.com/leokhoa/laragon/releases/download/6.0.0/laragon-wamp.exe>
2. PHP 8.3.17.zip - <https://windows.php.net/download#php-8.3>, unduh versi **VS16 x64 Thread Safe**

G. Proses Instalasi

1. Laragon
 - a. Buka *file* laragon-wamp.exe



Gambar 1. 1 Setup laragon-wamp.exe

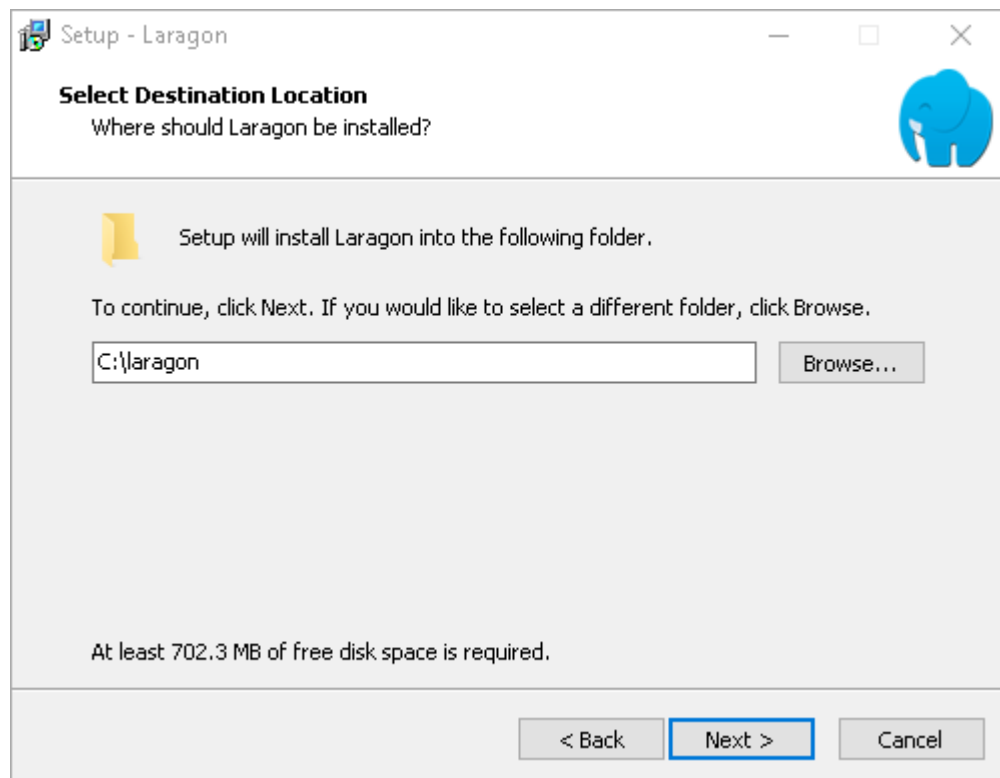
b. Pilih tempat instalasi

Catatan :

- Saya sarankan untuk *install* di disk yang menggunakan SSD
- Gunakan folder tanpa spasi
- Jika disk C sudah hampir penuh silahkan gunakan disk lain

Benar : C:\laragon atau D:\laragon

Salah : C:\Program File\laragon

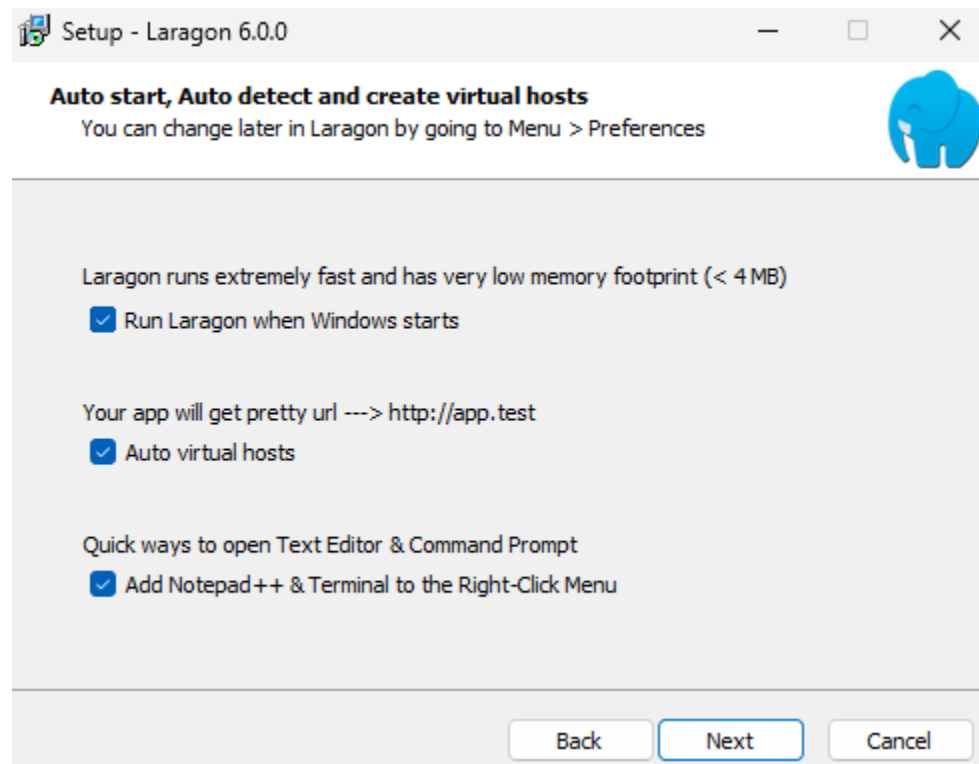


Gambar 1. 2 Setup tempat instalasi Laragon

c. Perhatikan yang di centang

Catatan :

- Saya sarankan untuk hilangkan centang pada “Run Laragon when Windows start” agar windows tidak terbebani saat booting, sehingga laragon akan aktif jika dibuka saja.



Gambar 1. 3 Setup opsional ceklis

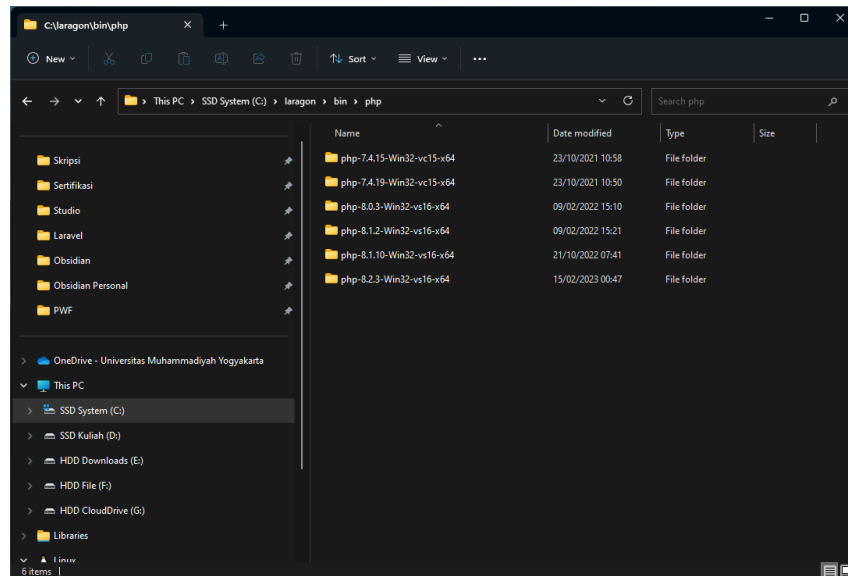
d. Buka Laragon



Gambar 1. 4 Hasil Laragon berhasil di install

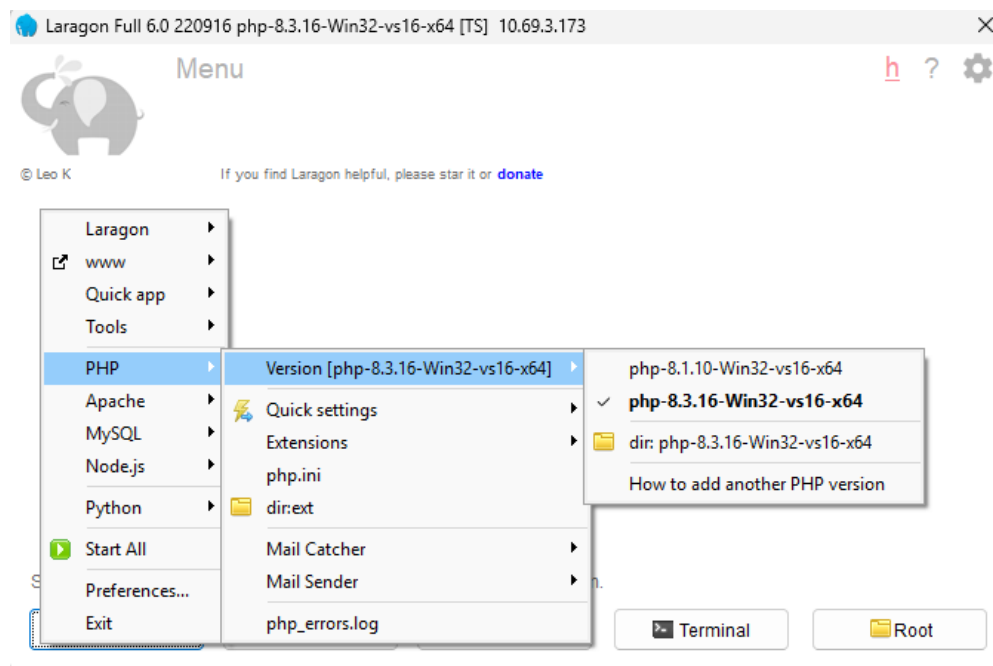
2. PHP

- Ekstrak file php.zip yang telah didownload dan tidak perlu mengubah nama hasil ekstraknya.
- Buka folder **C:\laragon\bin\php** dan pindahkan hasil ekstrak langkah sebelumnya



Gambar 1. 5 Tempat ekstraksi PHP

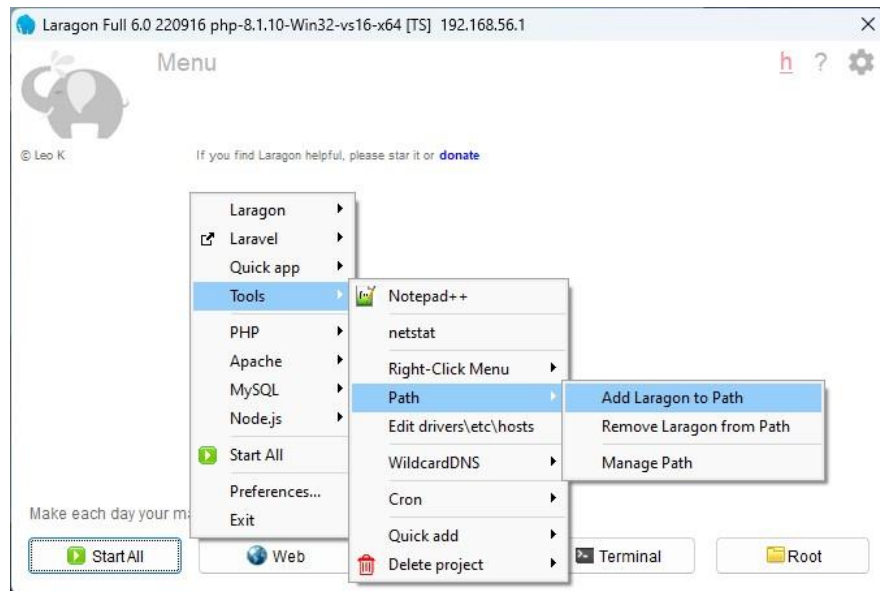
- Buka Laragon dan ganti versi PHP ke yang telah diunduh



Gambar 1. c Ubah versi PHP di Laragon

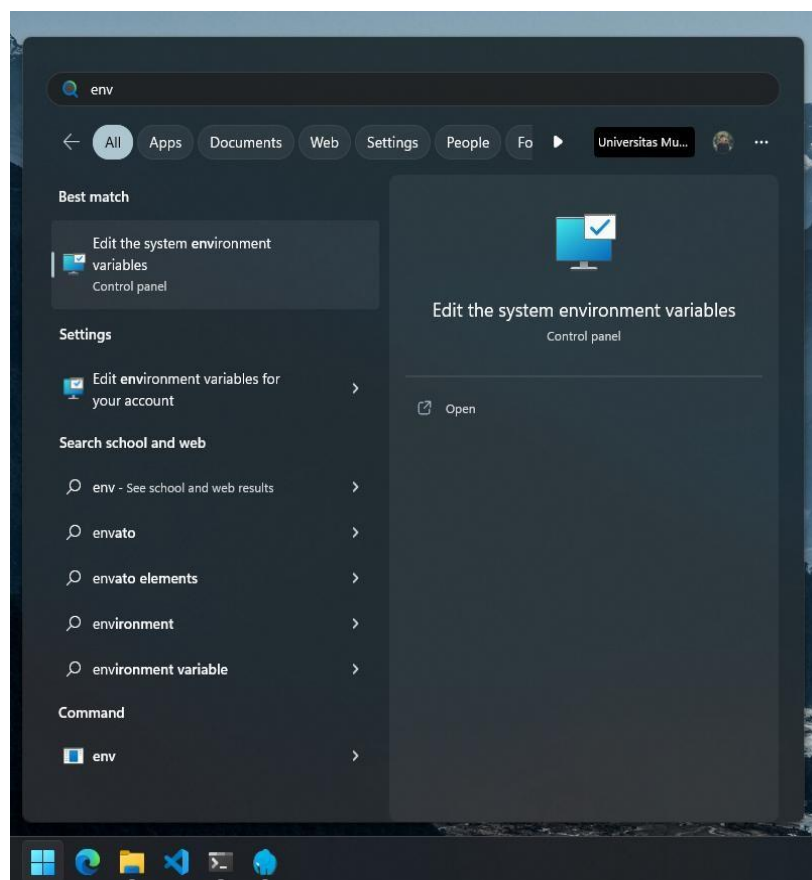
3. Penambahan Laragon path ke windows environment

a. Buka laragon



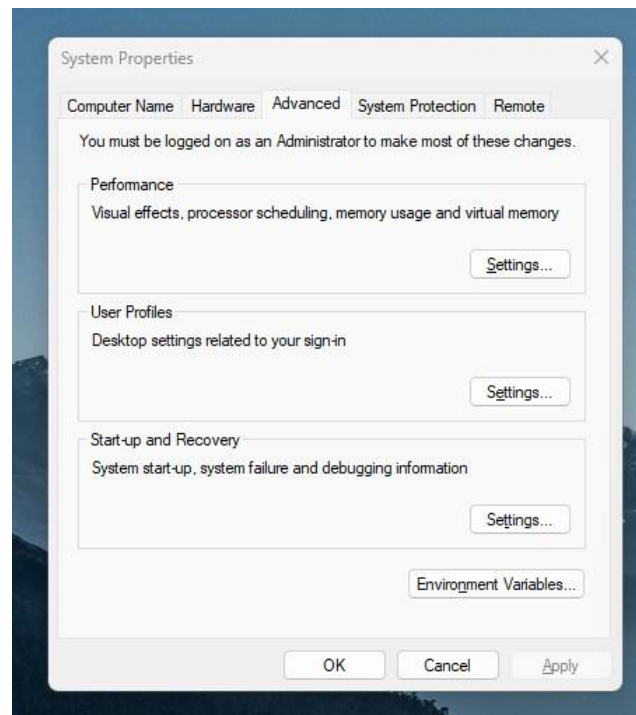
Gambar 1. 8 Penambahan env Windows melalui Laragon

b. Buka Windows Env dengan cara ketik “env” pada windows start



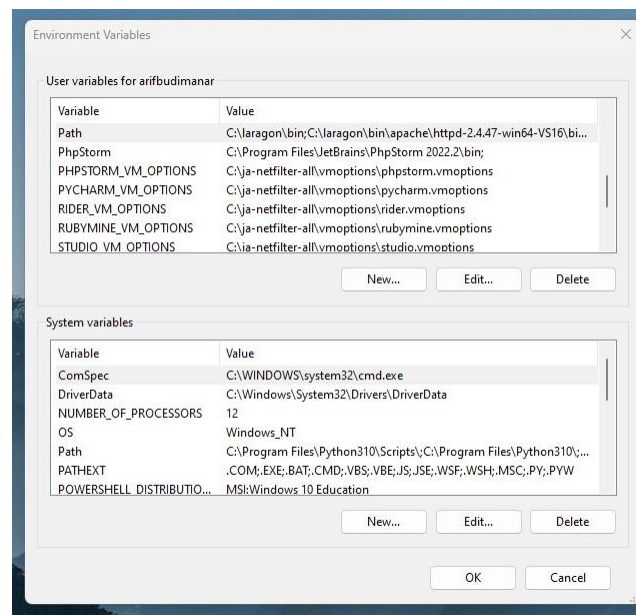
Gambar 1. 5 Windows start env

c. Klik “Environment Variables”



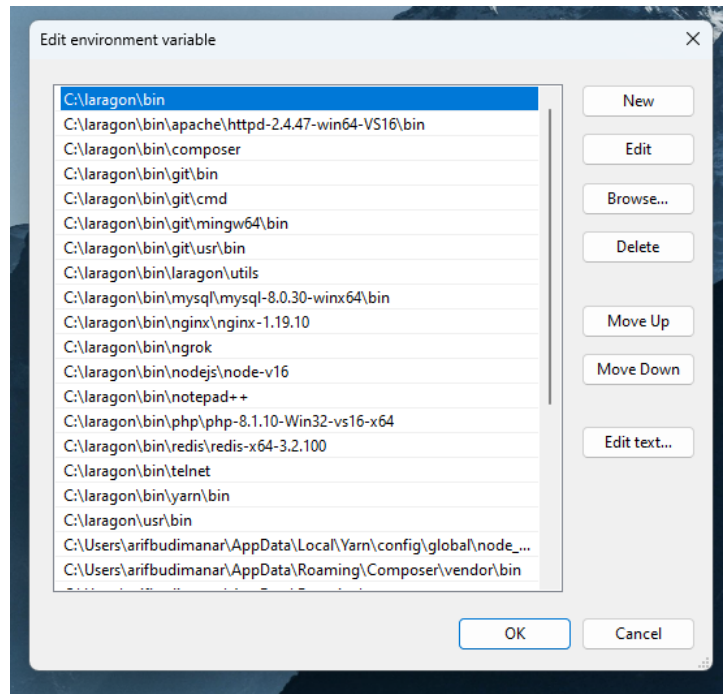
Gambar 1. 10 Windows System Properties

d. Pilih Variable “Path” pada User variables dan klik edit



Gambar 1. 11 Windows Environment Variables

- e. Pastikan directory laragon sudah masuk



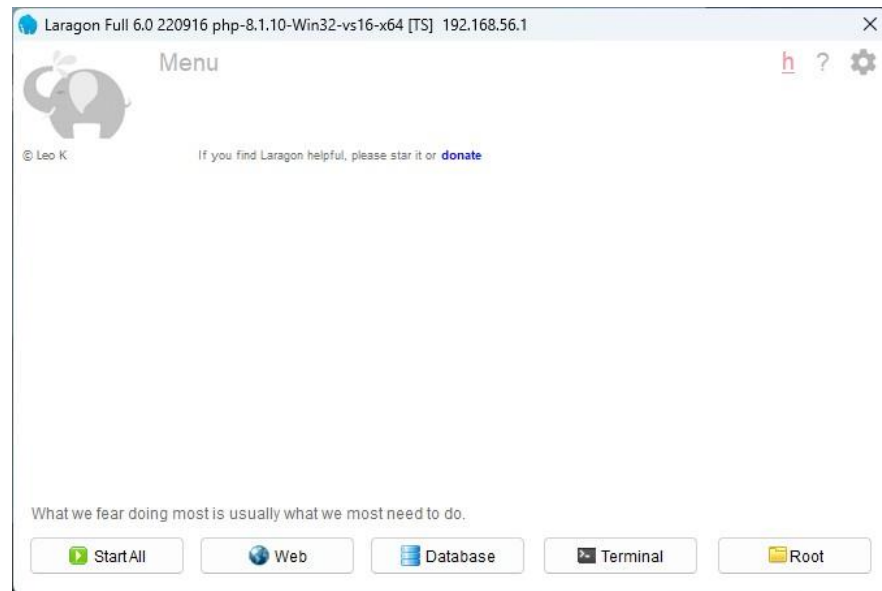
Gambar 1. 12 Windows edit Environment Variable

- f. Saya rekomendasikan untuk restart laptop

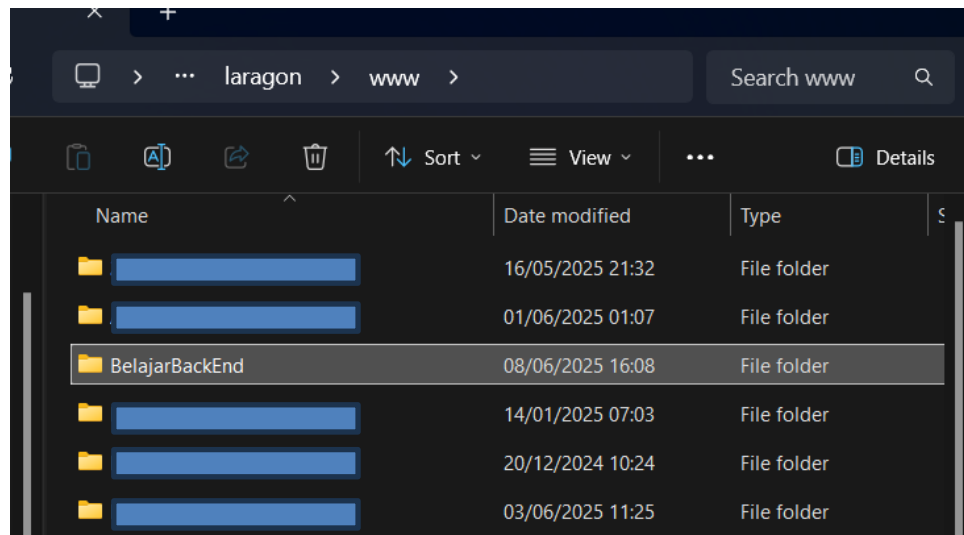
H. Persiapan Project Website

1. Copy folder project anda di Root directory Laragon

- a. Buka Laragon dan klik “Root”



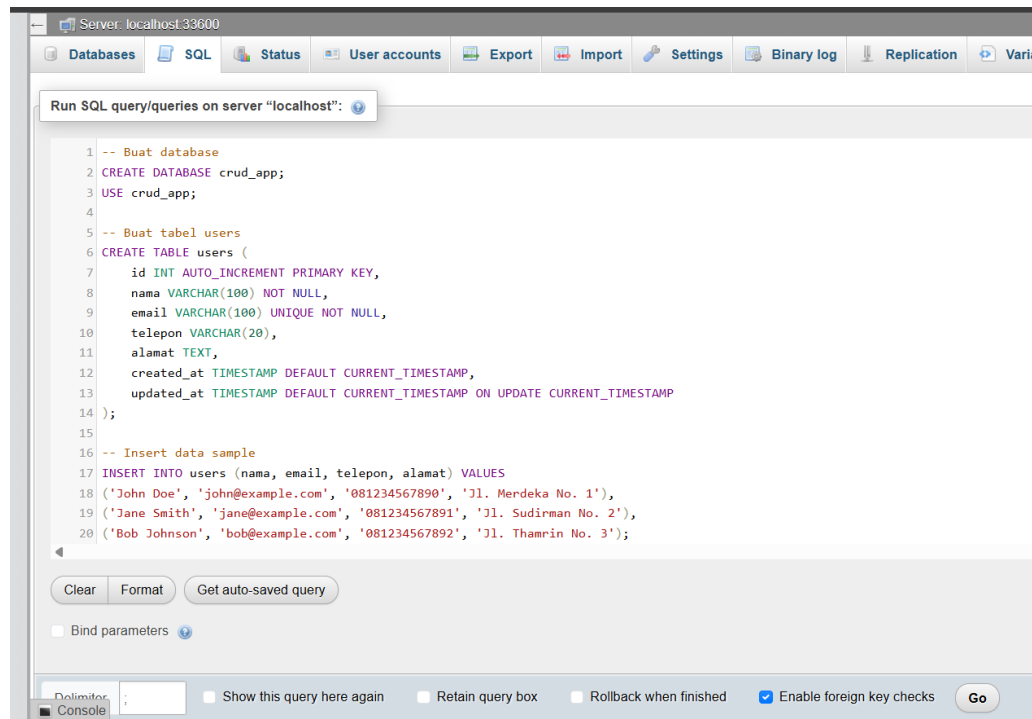
Gambar 1. 13 Menu Laragon



Gambar 1. 14 Pindahkan folder project ke laragon/www

2. Membuat Database dan Tabel MySQL

- a. Buka menu database dari laragon, jalankan query berikut di SQL kemudian klik Go:



Gambar 1. 1S Database Schema

3. Backend - Koneksi ke Database

Di dalam folder proyek anda, buat file config/database.php untuk mengatur koneksi ke database MySQL.

Kemudian tuliskan code berikut:

```

1  <?php
2  class Database {
3      private $host = '127.0.0.1';
4      private $port = '33600';
5      private $db_name = 'crud_app';
6      private $username = 'root';
7      private $password = '';
8      private $conn;
9
10     public function connect() {
11         $this->conn = null;
12
13         try {
14             $this->conn = new PDO(
15                 "mysql:host=" . $this->host . ";port=" . $this->port . ";dbname=" . $this->db_name,
16                 $this->username,
17                 $this->password
18             );
19             $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
20         } catch(PDOException $e) {
21             echo "Connection Error: " . $e->getMessage();
22             return null; // Return null if the connection fails
23         }
24
25         return $this->conn;
26     }
27 }
28
29 ?>
  
```

4. Membuat Model PHP untuk Operasi CRUD

Buat file models/User.php untuk menangani operasi CRUD untuk data pengguna.

```
1 <?php
2 require_once 'config/database.php';
3
4 class User {
5     private $conn;
6     private $table = 'users';
7
8     public $id;
9     public $nama;
10    public $email;
11    public $telepon;
12    public $alamat;
13
14    public function __construct() {
15        $database = new Database();
16        $this->conn = $database->connect();
17
18        // Check if connection was successful
19        if ($this->conn === null) {
20            throw new Exception("Database connection failed!");
21        }
22    }
23
24
25    // Create - Tambah user baru
26    public function create() {
27        $query = "INSERT INTO " . $this->table . "
28            SET nama = :nama, email = :email, telepon = :telepon, alamat = :alamat";
29
30        $stmt = $this->conn->prepare($query);
31
32        // Clean data
33        $this->nama = htmlspecialchars(strip_tags($this->nama));
34        $this->email = htmlspecialchars(strip_tags($this->email));
35        $this->telepon = htmlspecialchars(strip_tags($this->telepon));
36        $this->alamat = htmlspecialchars(strip_tags($this->alamat));
37
38        // Bind parameters
39        $stmt->bindParam(':nama', $this->nama);
40        $stmt->bindParam(':email', $this->email);
41        $stmt->bindParam(':telepon', $this->telepon);
42        $stmt->bindParam(':alamat', $this->alamat);
43
44        if($stmt->execute()) {
45            return true;
46        }
47        return false;
48    }
49
50    // Read - Ambil semua data user
51    public function read() {
52        $query = "SELECT * FROM " . $this->table . " ORDER BY created_at DESC";
53        $stmt = $this->conn->prepare($query);
54        $stmt->execute();
55        return $stmt;
56    }
57 }
```

Lanjutan function CRUD lainnya:

```
1 // Read Single - Ambil data user berdasarkan ID
2 public function readSingle() {
3     $query = "SELECT * FROM " . $this->table . " WHERE id = :id LIMIT 1";
4     $stmt = $this->conn->prepare($query);
5     $stmt->bindParam(':id', $this->id);
6     $stmt->execute();
7
8     $row = $stmt->fetch(PDO::FETCH_ASSOC);
9
10    if($row) {
11        $this->nama = $row['nama'];
12        $this->email = $row['email'];
13        $this->telepon = $row['telepon'];
14        $this->alamat = $row['alamat'];
15        return true;
16    }
17    return false;
18 }
19
20 // Update - Update data user
21 public function update() {
22     $query = "UPDATE " . $this->table . "
23         SET nama = :nama, email = :email, telepon = :telepon, alamat = :alamat
24         WHERE id = :id";
25
26     $stmt = $this->conn->prepare($query);
27
28     // Clean data
29     $this->nama = htmlspecialchars(strip_tags($this->nama));
30     $this->email = htmlspecialchars(strip_tags($this->email));
31     $this->telepon = htmlspecialchars(strip_tags($this->telepon));
32     $this->alamat = htmlspecialchars(strip_tags($this->alamat));
33     $this->id = htmlspecialchars(strip_tags($this->id));
34
35     // Bind parameters
36     $stmt->bindParam(':nama', $this->nama);
37     $stmt->bindParam(':email', $this->email);
38     $stmt->bindParam(':telepon', $this->telepon);
39     $stmt->bindParam(':alamat', $this->alamat);
40     $stmt->bindParam(':id', $this->id);
41
42     if($stmt->execute()) {
43         return true;
44     }
45     return false;
46 }
47
48 // Delete - Hapus user
49 public function delete() {
50     $query = "DELETE FROM " . $this->table . " WHERE id = :id";
51     $stmt = $this->conn->prepare($query);
52
53     $this->id = htmlspecialchars(strip_tags($this->id));
54     $stmt->bindParam(':id', $this->id);
55
56     if($stmt->execute()) {
57         return true;
58     }
59     return false;
60 }
61 }
62 ?>
```

5. Frontend - Halaman Index (Menampilkan Data)

Setelah membuat koneksi dan model, Anda perlu mengintegrasikan PHP ke dalam HTML yang sudah ada agar data bisa ditampilkan, ditambah, diperbarui, atau dihapus.

Menampilkan Data (Read) di index.php:

```

1  <?php
2  require_once 'models/User.php';
3
4  $user = new User();
5  $result = $user->read();
6
7  // Handle form submission
8  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
9      $user->nama = $_POST['nama'];
10     $user->email = $_POST['email'];
11     $user->telepon = $_POST['telepon'];
12     $user->alamat = $_POST['alamat'];
13
14     if ($user->create()) {
15         $success_message = "User berhasil ditambahkan!";
16         // Refresh data
17         $result = $user->read();
18     } else {
19         $error_message = "Gagal menambahkan user!";
20     }
21 }
22
23
24 <!DOCTYPE html>
25 <html lang="id">
26 <head>
27     <meta charset="UTF-8">
28     <meta name="viewport" content="width=device-width, initial-scale=1.0">
29     <title>CRUD App - Manajemen User</title>
30     <!-- Add Bootstrap CSS -->
31     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
32 </head>
33 <body>
34     <div class="container mt-5">
35         <h1 class="text-center mb-4">Manajemen Data User</h1>
36
37         <?php if (isset($success_message)): ?>
38         <div class="alert alert-success"><?php echo $success_message; ?></div>
39         <?php endif; ?>
40
41         <?php if (isset($error_message)): ?>
42         <div class="alert alert-danger"><?php echo $error_message; ?></div>
43         <?php endif; ?>
44
45         <!-- Form Tambah User -->
46         <div class="card p-4 mb-5">
47             <h2>Tambah User Baru</h2>
48             <form method="POST" action="">
49                 <div class="mb-3">
50                     <label for="nama" class="form-label">Nama:</label>
51                     <input type="text" id="nama" name="nama" class="form-control" required>
52                 </div>
53
54                 <div class="mb-3">
55                     <label for="email" class="form-label">Email:</label>
56                     <input type="email" id="email" name="email" class="form-control" required>
57                 </div>
58
59                 <div class="mb-3">
60                     <label for="telepon" class="form-label">Telepon:</label>
61                     <input type="text" id="telepon" name="telepon" class="form-control">
62                 </div>
63
64                 <div class="mb-3">
65                     <label for="alamat" class="form-label">Alamat:</label>
66                     <textarea id="alamat" name="alamat" class="form-control" rows="3"></textarea>
67                 </div>
68
69                 <button type="submit" class="btn btn-primary">Tambah User</button>
70             </form>
71         </div>
72
73         <!-- Tabel Data User -->
74         <div class="card p-4">
75             <h2>Data User</h2>
76             <table class="table table-bordered table-striped">
77                 <thead>
78                     <tr>
79                         <th>ID</th>
80                         <th>Nama</th>
81                         <th>Email</th>
82                         <th>Telepon</th>
83                         <th>Alamat</th>
84                         <th>Aksi</th>
85                     </tr>
86                 </thead>
87                 <tbody>
88                     <?php while ($row = $result->fetch(PDO::FETCH_ASSOC)): ?>
89                     <tr>
90                         <td><?php echo $row['id']; ?></td>
91                         <td><?php echo htmlspecialchars($row['nama']); ?></td>
92                         <td><?php echo htmlspecialchars($row['email']); ?></td>
93                         <td><?php echo htmlspecialchars($row['telepon']); ?></td>
94                         <td><?php echo htmlspecialchars($row['alamat']); ?></td>
95                         <td>
96                             <a href="edit.php?id=<?php echo $row['id']; ?>" class="btn btn-warning btn-sm">Edit</a>
97                             <a href="delete.php?id=<?php echo $row['id']; ?>" class="btn btn-danger btn-sm"
98                                 onclick="return confirm('Apakah Anda yakin ingin menghapus user ini?')">Hapus</a>
99                         </td>
100                     </tr>
101                 </tbody>
102             </table>
103         </div>
104     </div>
105
106     <!-- Add Bootstrap JS and Popper.js -->
107     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
108     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"></script>
109 </body>
110 </html>
111
112

```

6. Halaman Edit (Update Data)

```

1 <?php
2 require_once 'models/User.php';
3
4 if (!isset($_GET['id'])) {
5     header('Location: index.php');
6     exit;
7 }
8
9 $user = new User();
10 $user->id = $_GET['id'];
11
12 if (!$user->readSingle()) {
13     header('Location: index.php');
14     exit;
15 }
16
17 // Handle form submission
18 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
19     $user->id = $_POST['id'];
20     $user->nama = $_POST['nama'];
21     $user->email = $_POST['email'];
22     $user->telepon = $_POST['telepon'];
23     $user->alamat = $_POST['alamat'];
24
25     if ($user->update()) {
26         $success_message = "User berhasil diupdate!";
27         // Refresh data
28         $user->readSingle();
29     } else {
30         $error_message = "Gagal mengupdate user!";
31     }
32 }
33 ?>
34
35 <!DOCTYPE html>
36 <html lang="id">
37 <head>
38     <meta charset="UTF-8">
39     <meta name="viewport" content="width=device-width, initial-scale=1.0">
40     <title>Edit User - CRUD App</title>
41     <!-- Add Bootstrap CSS -->
42     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
43 </head>
44 <body>
45     <div class="container mt-5">
46         <h1 class="text-center mb-4">Edit User</h1>
47
48         <?php if (isset($success_message)): ?>
49             <div class="alert alert-success"><?php echo $success_message; ?></div>
50         <?php endif; ?>
51
52         <?php if (isset($error_message)): ?>
53             <div class="alert alert-danger"><?php echo $error_message; ?></div>
54         <?php endif; ?>
55
56         <div class="card p-4">
57             <form method="POST" action="">
58                 <input type="hidden" name="id" value="<?php echo $user->id; ?>">
59
60                 <div class="mb-3">
61                     <label for="nama" class="form-label">Nama:</label>
62                     <input type="text" id="nama" name="nama" class="form-control" value="<?php echo htmlspecialchars($user->nama); ?>" required>
63                 </div>
64
65                 <div class="mb-3">
66                     <label for="email" class="form-label">Email:</label>
67                     <input type="email" id="email" name="email" class="form-control" value="<?php echo htmlspecialchars($user->email); ?>" required>
68                 </div>
69
70                 <div class="mb-3">
71                     <label for="telepon" class="form-label">Telepon:</label>
72                     <input type="text" id="telepon" name="telepon" class="form-control" value="<?php echo htmlspecialchars($user->telepon); ?>">
73                 </div>
74
75                 <div class="mb-3">
76                     <label for="alamat" class="form-label">Alamat:</label>
77                     <textarea id="alamat" name="alamat" class="form-control" rows="3"><?php echo htmlspecialchars($user->alamat); ?></textarea>
78                 </div>
79
80                 <button type="submit" class="btn btn-primary">Update User</button>
81                 <a href="index.php" class="btn btn-secondary">Kembali</a>
82             </form>
83         </div>
84     </div>
85
86     <!-- Add Bootstrap JS and Popper.js -->
87     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
88     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"></script>
89 </body>
90 </html>
91

```

7. Halaman Delete (Menghapus Data)

```
1 <?php
2 require_once 'models/User.php';
3
4 if (isset($_GET['id'])) {
5     $user = new User();
6     $user->id = $_GET['id'];
7
8     if ($user->delete()) {
9         header('Location: index.php?message=deleted');
10    } else {
11        header('Location: index.php?error=delete_failed');
12    }
13 } else {
14     header('Location: index.php');
15 }
16 exit;
17 ?>
```

8. Hasil

Manajemen Data User

Tambah User Baru

Nama:

Email:

Telepon:

Alamat:

Data User

ID	Nama	Email	Telepon	Alamat	Aksi
4	Rama Soni Ravictor	ramaravictor07@gmail.com	087812415070	Jalan Kaliurang	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
1	John Doe	john@example.com	081234567890	Jl. Merdeka No. 1 jidopajdjapoda	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	Jane Smith	jane@example.com	081234567891	Jl. Sudirman No. 2	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	Bob Johnson	bob@example.com	081234567892	Jl. Thamrin No. 3	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>