2025

UJI KOMPETENSI KEAHLIAN PAKET 4

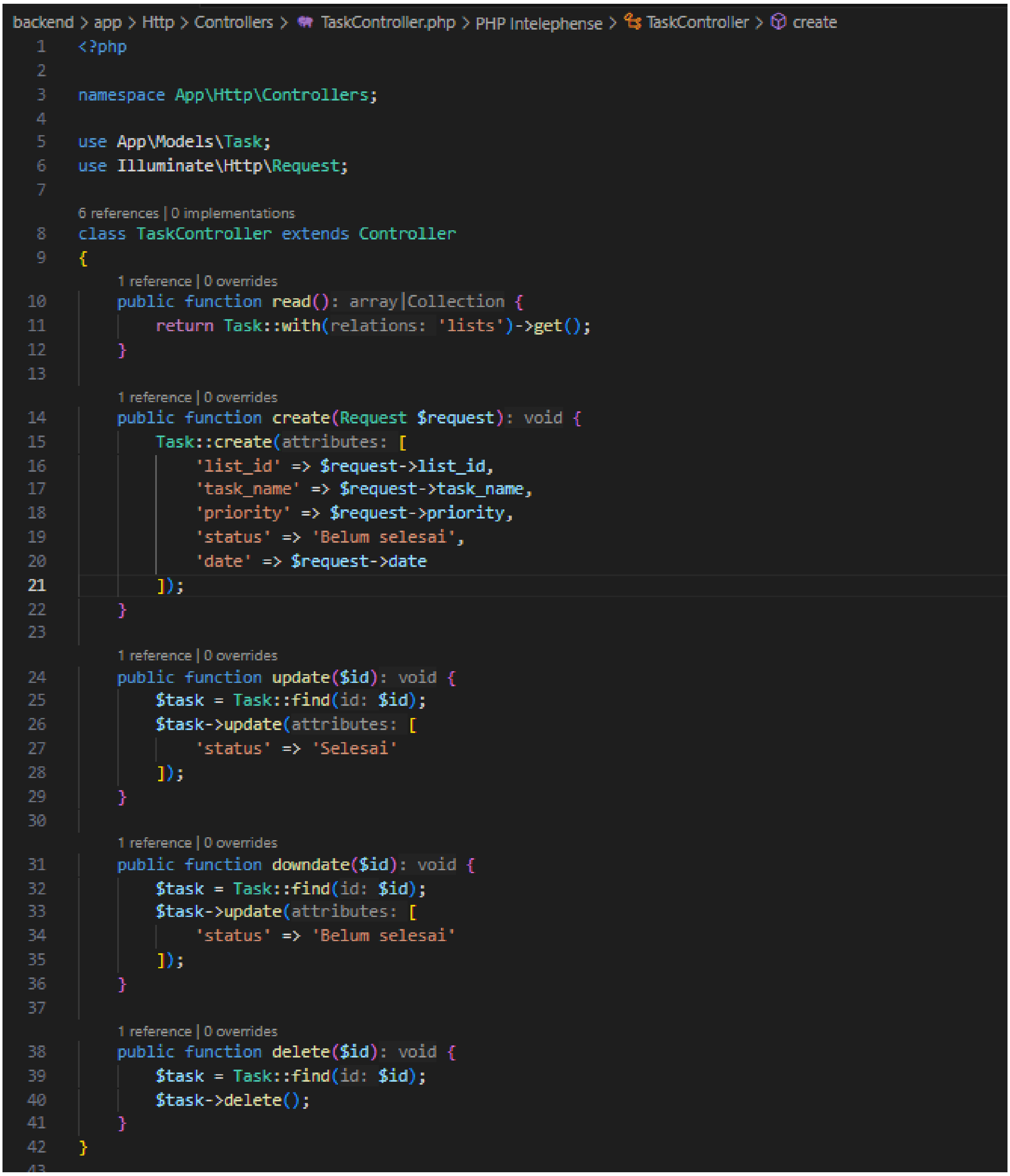
Pengembangan Perangkat Lunak dan Gim

Izzi Fawzan
0073976189

DAFTAR ISI

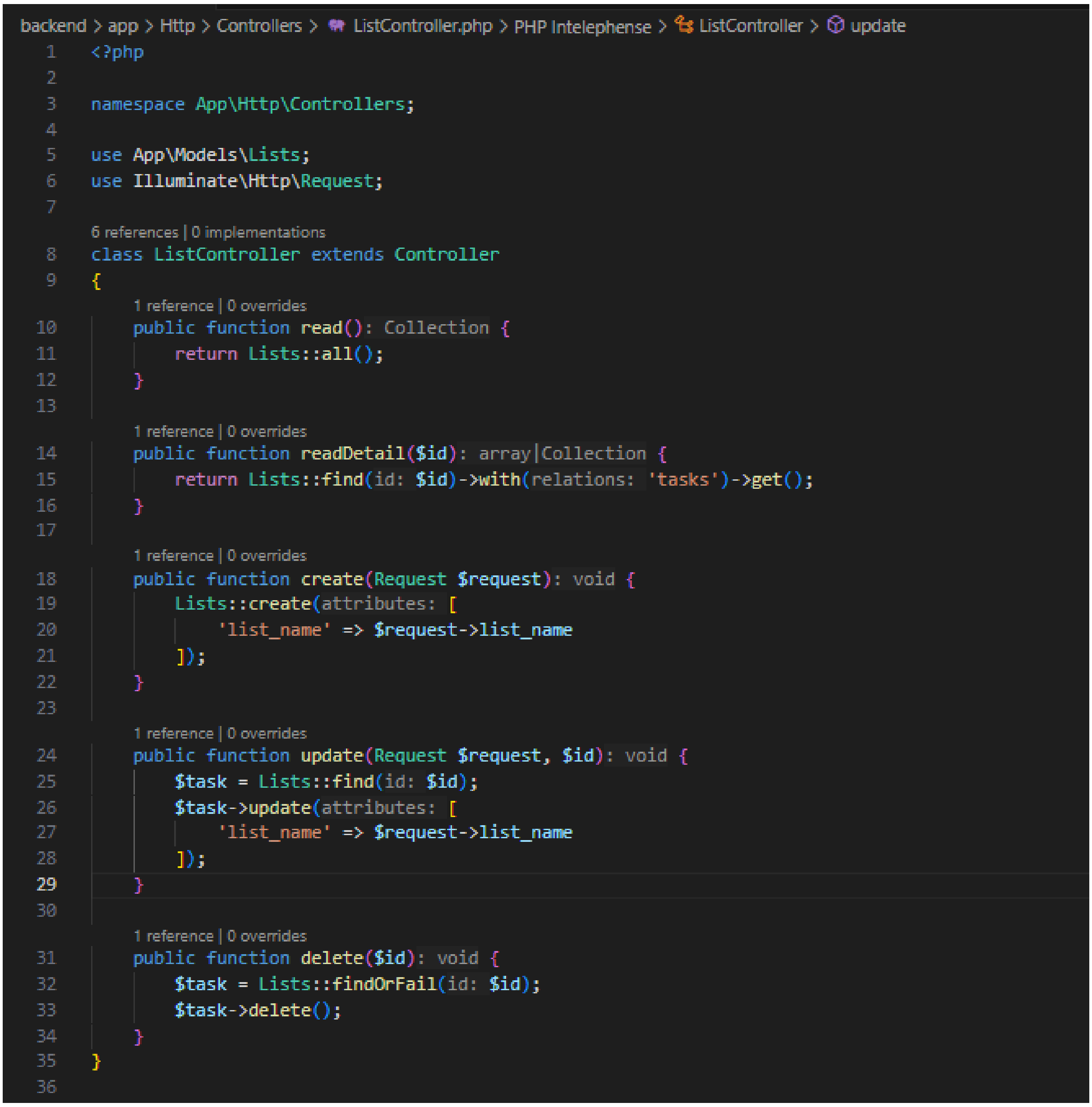
PEMBUKAAN

Ini adalah project ukk saya, paket 2 aplikasi TO DO LIST, saya menggunakan laravel rest api dan vue js. Udah itu aja


backend > app > Http > Controllers > TaskController.php > PHP Intelephense > TaskController > create

```php
<?php

namespace App\Http\Controllers;

use App\Models\Task;
use Illuminate\Http\Request;

6 references | 0 implementations
class TaskController extends Controller
{
    1 reference | 0 overrides
    public function read(): array|Collection {
        return Task::with(relations: 'lists')->get();
    }

    1 reference | 0 overrides
    public function create(Request $request): void {
        Task::create(attributes: [
            'list_id' => $request->list_id,
            'task_name' => $request->task_name,
            'priority' => $request->priority,
            'status' => 'Belum selesai',
            'date' => $request->date
        ]);
    }

    1 reference | 0 overrides
    public function update($id): void {
        $task = Task::find(id: $id);
        $task->update(attributes: [
            'status' => 'Selesai'
        ]);
    }

    1 reference | 0 overrides
    public function downdate($id): void {
        $task = Task::find(id: $id);
        $task->update(attributes: [
            'status' => 'Belum selesai'
        ]);
    }

    1 reference | 0 overrides
    public function delete($id): void {
        $task = Task::find(id: $id);
        $task->delete();
    }
}
```

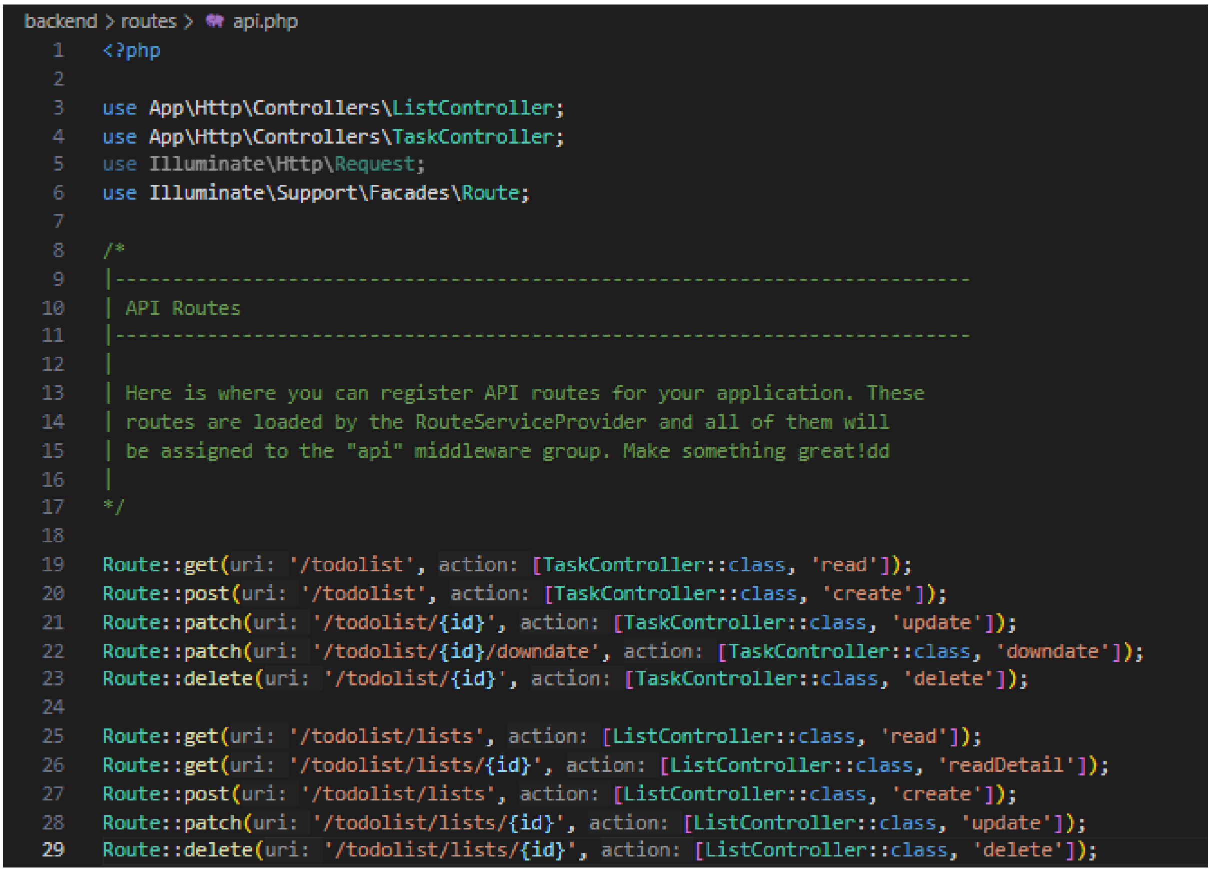
Ini adalah TaskController, yang berfungsi untuk membuat CRUD dari Task



backend > app > Http > Controllers > ListController.php > PHP Intelephense > ListController > update

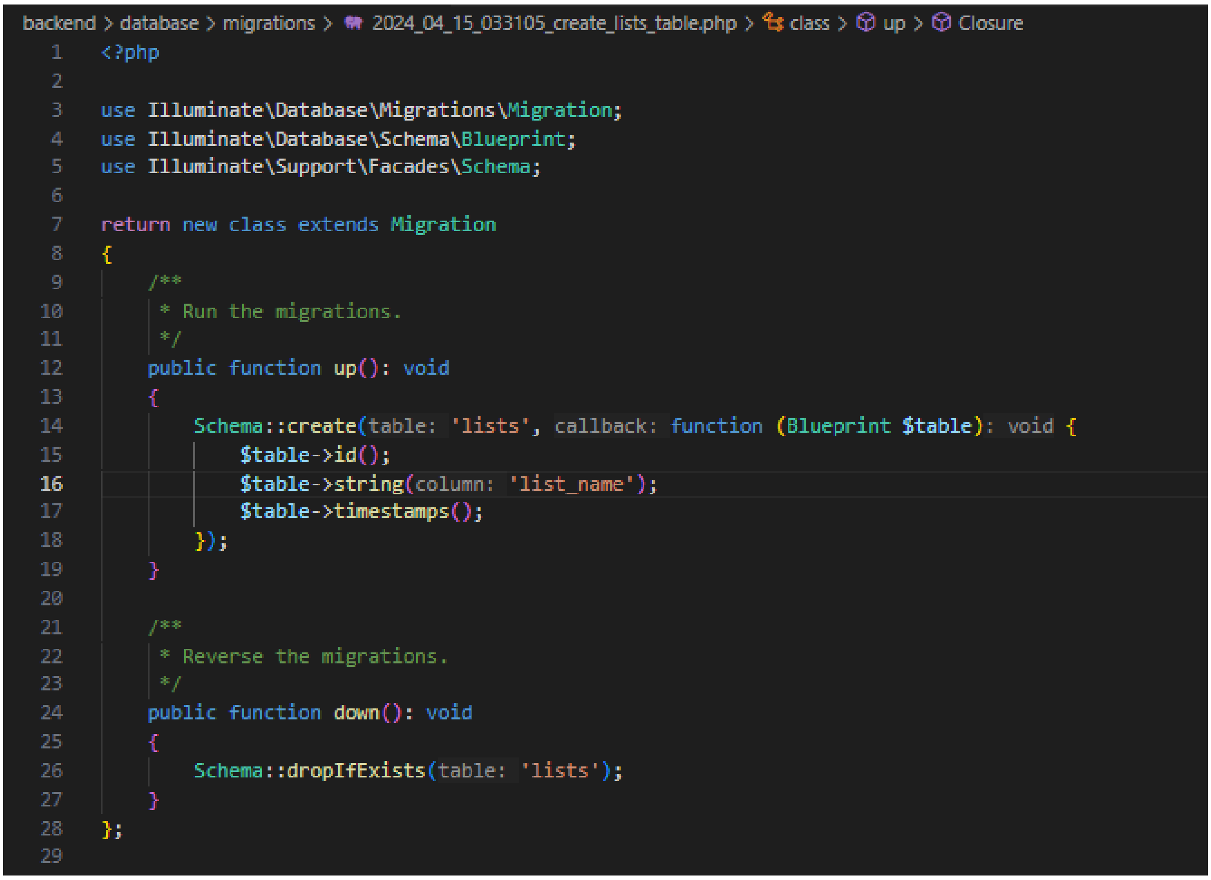
```
<?php

namespace App\Http\Controllers;

use App\Models\Lists;
use Illuminate\Http\Request;

6 references | 0 implementations
class ListController extends Controller
{
    1 reference | 0 overrides
    public function read(): Collection {
        return Lists::all();
    }

    1 reference | 0 overrides
    public function readDetail($id): array|Collection {
        return Lists::find(id: $id)->with(relations: 'tasks')->get();
    }

    1 reference | 0 overrides
    public function create(Request $request): void {
        Lists::create(attributes: [
            'list_name' => $request->list_name
        ]);
    }

    1 reference | 0 overrides
    public function update(Request $request, $id): void {
        $task = Lists::find(id: $id);
        $task->update(attributes: [
            'list_name' => $request->list_name
        ]);
    }

    1 reference | 0 overrides
    public function delete($id): void {
        $task = Lists::findOrFail(id: $id);
        $task->delete();
    }
}

```

Ini adalah ListController, yang berfungsi untuk membuat CRUD dari List



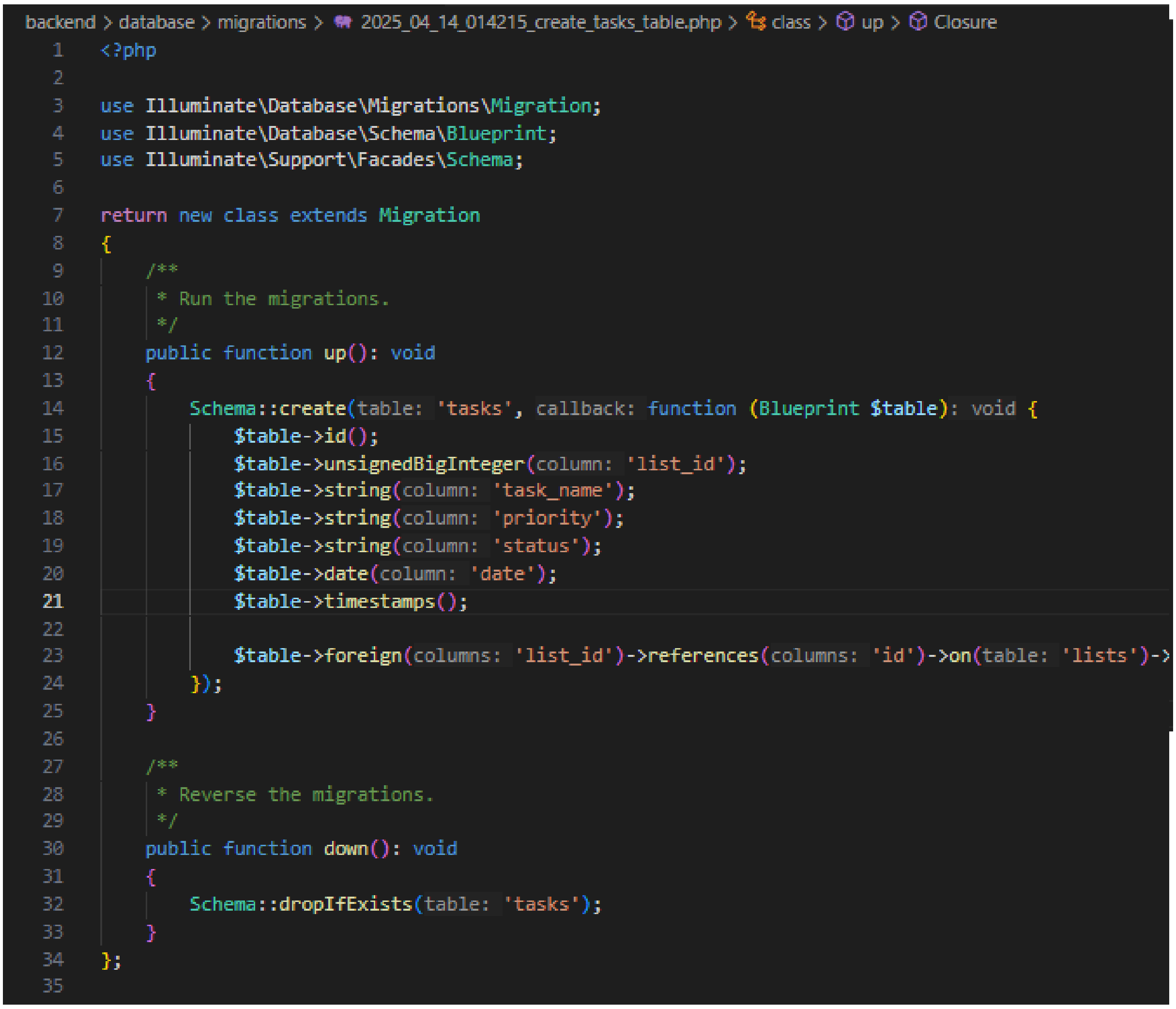
```php
<?php

use App\Http\Controllers\ListController;
use App\Http\Controllers\TaskController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!dd
|
*/

Route::get(uri: '/todolist', action: [TaskController::class, 'read']);
Route::post(uri: '/todolist', action: [TaskController::class, 'create']);
Route::patch(uri: '/todolist/{id}', action: [TaskController::class, 'update']);
Route::patch(uri: '/todolist/{id}/downdate', action: [TaskController::class, 'downdate']);
Route::delete(uri: '/todolist/{id}', action: [TaskController::class, 'delete']);

Route::get(uri: '/todolist/lists', action: [ListController::class, 'read']);
Route::get(uri: '/todolist/lists/{id}', action: [ListController::class, 'readDetail']);
Route::post(uri: '/todolist/lists', action: [ListController::class, 'create']);
Route::patch(uri: '/todolist/lists/{id}', action: [ListController::class, 'update']);
Route::delete(uri: '/todolist/lists/{id}', action: [ListController::class, 'delete']);
```

Ini ada route, berfungsi untuk memberikan url agar CRUD bisa berjalan sesuai url


backend > database > migrations > 2024_04_15_033105_create_lists_table.php > class > up > Closure

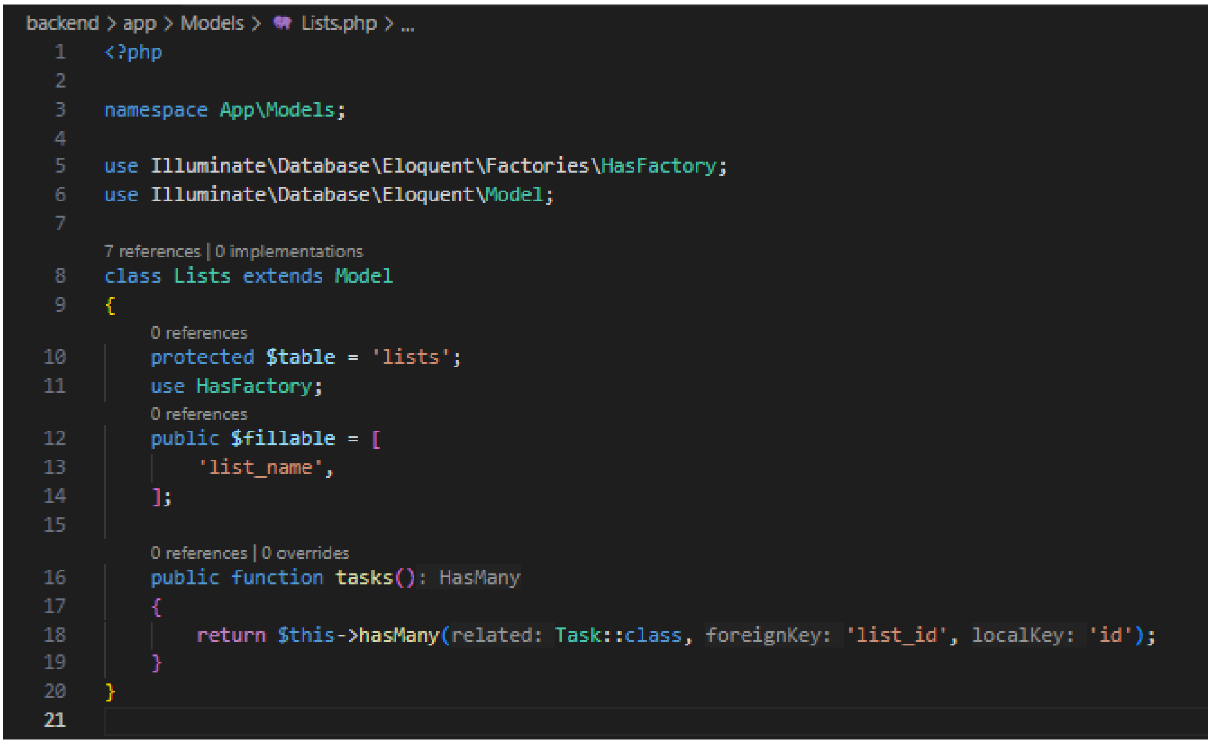
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'lists', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'list_name');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists(table: 'lists');
    }
};
```

Itu adalah migration dari Tabel lists, berfungsi untuk membuat tabel lists beserta colum nya


backend > database > migrations > 2025_04_14_014215_create_tasks_table.php > class > up > Closure

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'tasks', callback: function (Blueprint $table): void {
            $table->id();
            $table->unsignedBigInteger(column: 'list_id');
            $table->string(column: 'task_name');
            $table->string(column: 'priority');
            $table->string(column: 'status');
            $table->date(column: 'date');
            $table->timestamps();

            $table->foreign(columns: 'list_id')->references(columns: 'id')->on(table: 'lists')->
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists(table: 'tasks');
    }
};
```

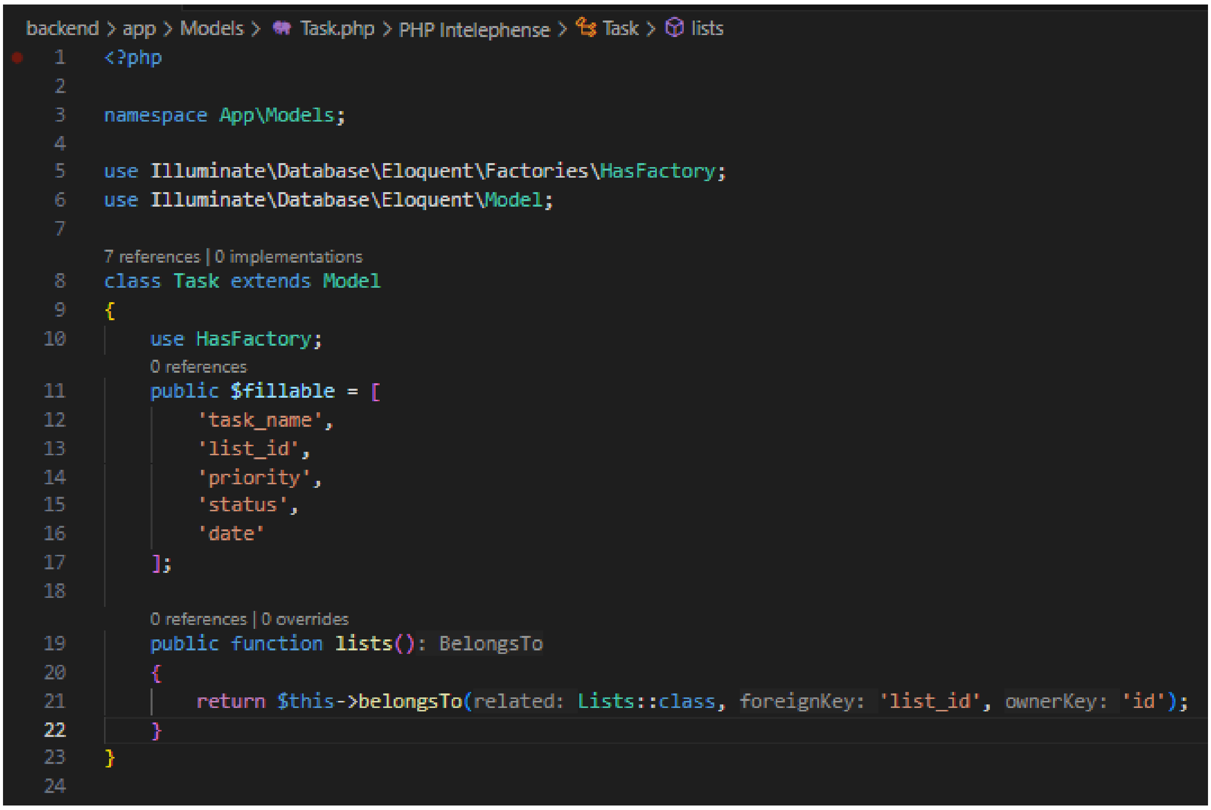
Sedangkan ini migrations dari tabel tasks


backend > app > Models > Lists.php > ...
7 references | 0 implementations
0 references
0 references
0 references | 0 overrides

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Lists extends Model
{
    protected $table = 'lists';
    use HasFactory;
    public $fillable = [
        'list_name',
    ];

    public function tasks(): HasMany
    {
        return $this->hasMany(related: Task::class, foreignKey: 'list_id', localKey: 'id');
    }
}
```

Itu merupakan model dari Lists, berfungsi untuk mengatur relasi, membuat colum dari tabel lists bisa diisi

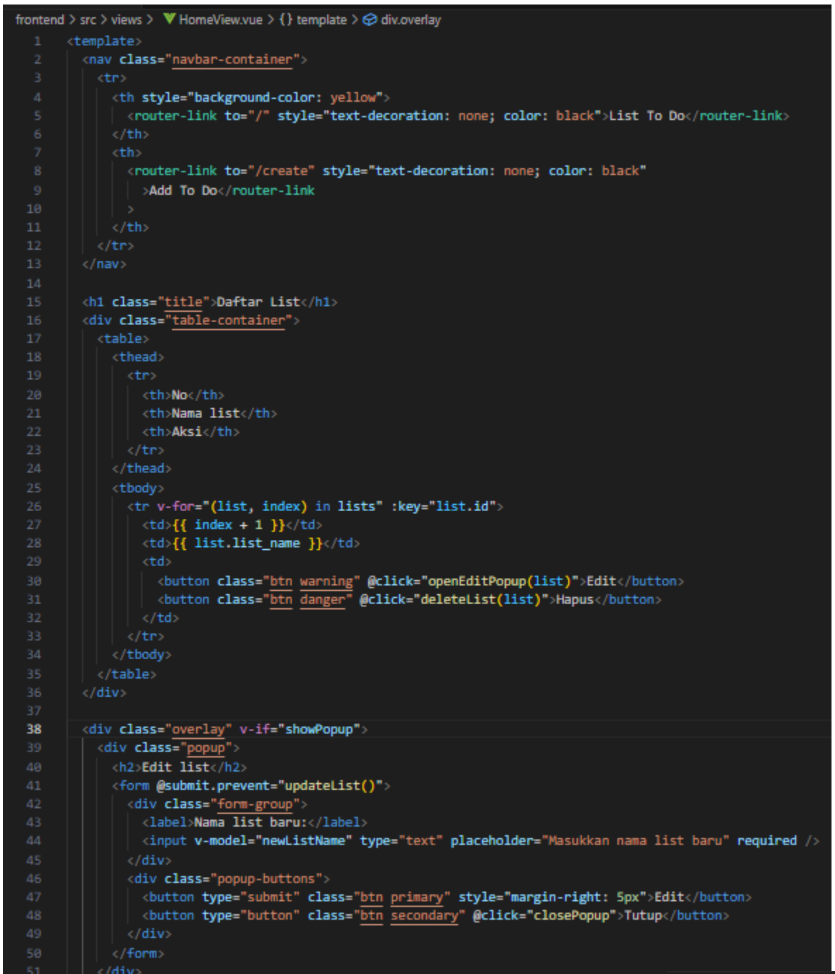
Sedangkan ini model dari Task


backend > app > Models > Task.php > PHP Intelephense > Task > lists
7 references | 0 implementations
0 references
0 references | 0 overrides

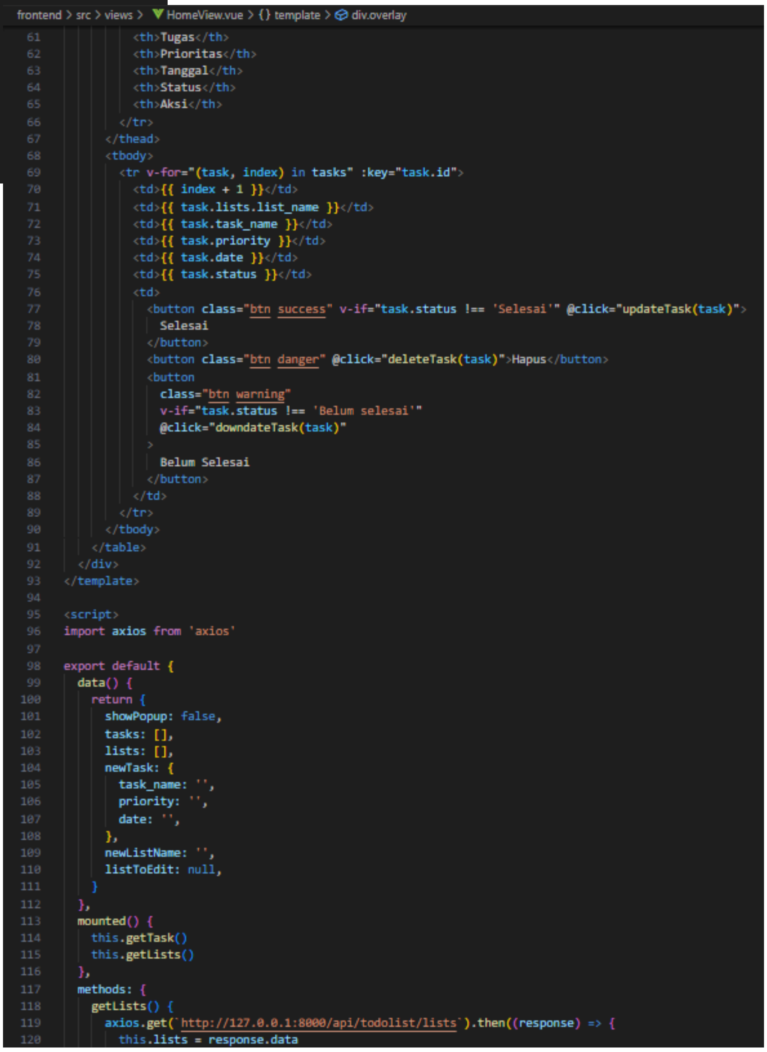
```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Task extends Model
{
    use HasFactory;
    public $fillable = [
        'task_name',
        'list_id',
        'priority',
        'status',
        'date'
    ];

    public function lists(): BelongsTo
    {
        return $this->belongsTo(related: Lists::class, foreignKey: 'list_id', ownerKey: 'id');
    }
}

```

Udah nampaknya itu saja semua hal yang ada di bagian laravel backend

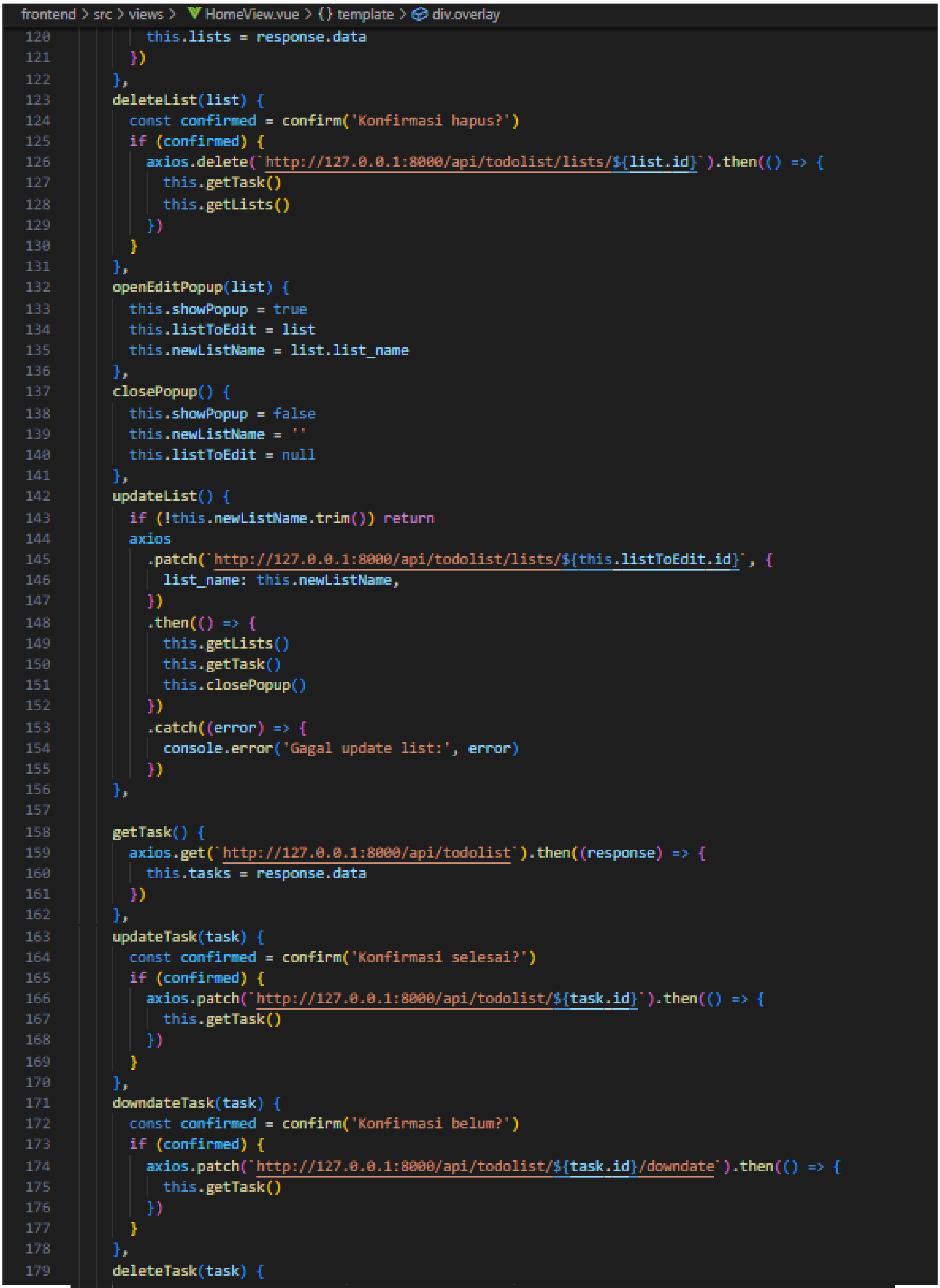
Ini adalah codingan HomeView.vue yang nantikan akan tampil sebagai beranda dari projectnya


frontend > src > views > HomeView.vue > {} template > div.overlay

```
<template>
  <nav class="navbar-container">
    <tr>
      <th style="background-color: yellow">
        <router-link to="/" style="text-decoration: none; color: black">List To Do</router-link>
      </th>
      <th>
        <router-link to="/create" style="text-decoration: none; color: black"
          >Add To Do</router-link
        >
      </th>
    </tr>
  </nav>

  <h1 class="title">Daftar List</h1>
  <div class="table-container">
    <table>
      <thead>
        <tr>
          <th>No</th>
          <th>Nama list</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <tr v-for="(list, index) in lists" :key="list.id">
          <td>{{ index + 1 }}</td>
          <td>{{ list.list_name }}</td>
          <td>
            <button class="btn warning" @click="openEditPopup(list)">Edit</button>
            <button class="btn danger" @click="deleteList(list)">Hapus</button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>

  <div class="overlay" v-if="showPopup">
    <div class="popup">
      <h2>Edit list</h2>
      <form @submit.prevent="updateList()">
        <div class="form-group">
          <label>Nama list baru:</label>
          <input v-model="newListName" type="text" placeholder="Masukkan nama list baru" required />
        </div>
        <div class="popup-buttons">
          <button type="submit" class="btn primary" style="margin-right: 5px">Edit</button>
          <button type="button" class="btn secondary" @click="closePopup">Tutup</button>
        </div>
      </form>
    </div>
  </div>

  <h1 class="title">Daftar Tugas</h1>
  <div class="table-container">
    <table>
      <thead>
        <tr>
          <th>No</th>
          <th>List</th>
```

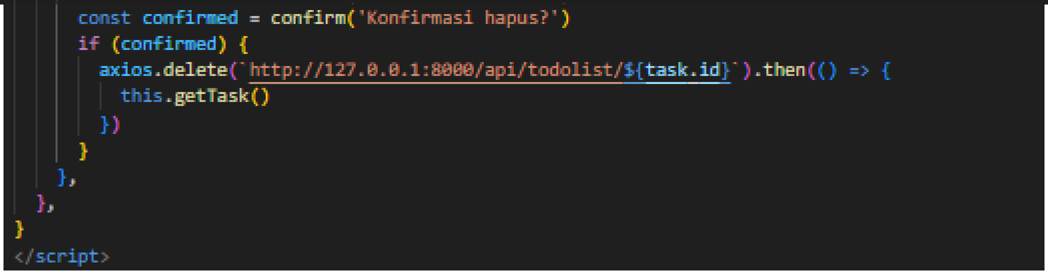

frontend > src > views > HomeView.vue > {} template > div.overlay

```
          <th>Tugas</th>
          <th>Prioritas</th>
          <th>Tanggal</th>
          <th>Status</th>
          <th>Aksi</th>
        </tr>
      </thead>
      <tbody>
        <tr v-for="(task, index) in tasks" :key="task.id">
          <td>{{ index + 1 }}</td>
          <td>{{ task.lists.list_name }}</td>
          <td>{{ task.task_name }}</td>
          <td>{{ task.priority }}</td>
          <td>{{ task.date }}</td>
          <td>{{ task.status }}</td>
          <td>
            <button class="btn success" v-if="task.status !== 'Selesai'" @click="updateTask(task)">
              Selesai
            </button>
            <button class="btn danger" @click="deleteTask(task)">Hapus</button>
            <button
              class="btn warning"
              v-if="task.status !== 'Belum selesai'"
              @click="downdateTask(task)"
            >
              Belum Selesai
            </button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</template>

<script>
import axios from 'axios'

export default {
  data() {
    return {
      showPopup: false,
      tasks: [],
      lists: [],
      newTask: {
        task_name: '',
        priority: '',
        date: '',
      },
      newListName: '',
      listToEdit: null,
    }
  },
  mounted() {
    this.getTask()
    this.getLists()
  },
  methods: {
    getLists() {
      axios.get(`http://127.0.0.1:8000/api/todolist/lists`).then((response) => {
        this.lists = response.data
```



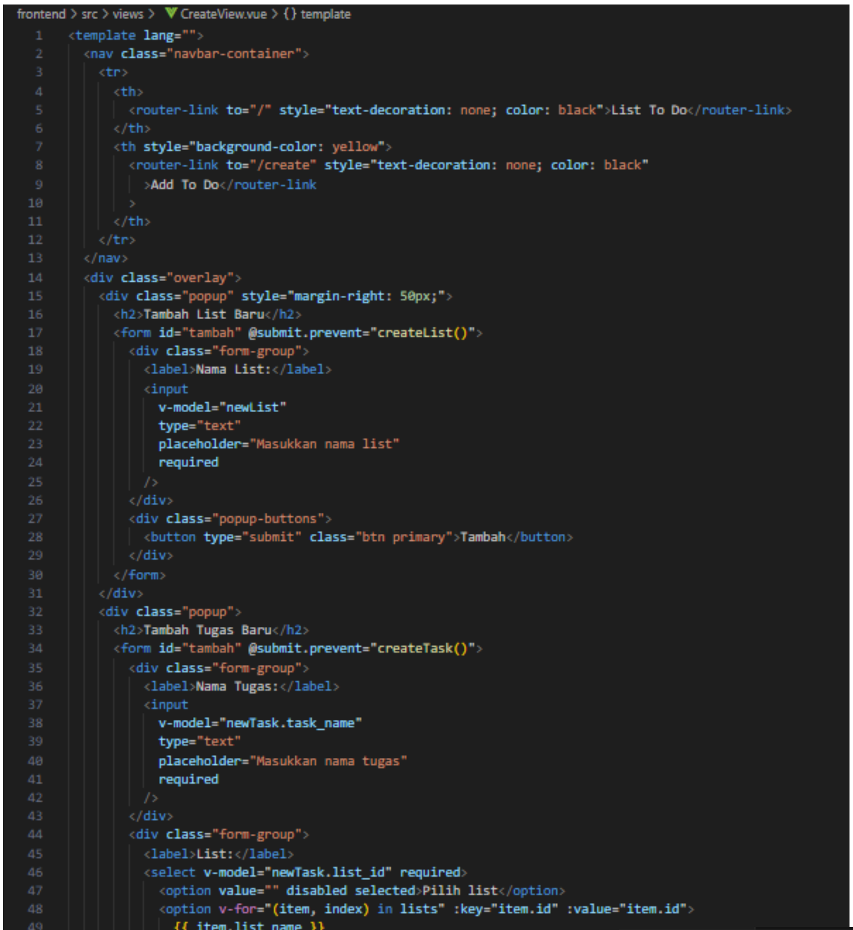
frontend > src > views > HomeView.vue > {} template > div.overlay

```
        this.lists = response.data
      })
    },
    deleteList(list) {
      const confirmed = confirm('Konfirmasi hapus?')
      if (confirmed) {
        axios.delete(`http://127.0.0.1:8000/api/todolist/lists/${list.id}`).then(() => {
          this.getTask()
          this.getLists()
        })
      }
    },
    openEditPopup(list) {
      this.showPopup = true
      this.listToEdit = list
      this.newListName = list.list_name
    },
    closePopup() {
      this.showPopup = false
      this.newListName = ''
      this.listToEdit = null
    },
    updateList() {
      if (!this.newListName.trim()) return
      axios
        .patch(`http://127.0.0.1:8000/api/todolist/lists/${this.listToEdit.id}`, {
          list_name: this.newListName,
        })
        .then(() => {
          this.getLists()
          this.getTask()
          this.closePopup()
        })
        .catch((error) => {
          console.error('Gagal update list:', error)
        })
    },

    getTask() {
      axios.get(`http://127.0.0.1:8000/api/todolist`).then((response) => {
        this.tasks = response.data
      })
    },
    updateTask(task) {
      const confirmed = confirm('Konfirmasi selesai?')
      if (confirmed) {
        axios.patch(`http://127.0.0.1:8000/api/todolist/${task.id}`).then(() => {
          this.getTask()
        })
      }
    },
    downdateTask(task) {
      const confirmed = confirm('Konfirmasi belum?')
      if (confirmed) {
        axios.patch(`http://127.0.0.1:8000/api/todolist/${task.id}/downdate`).then(() => {
          this.getTask()
        })
      }
    },
    deleteTask(task) {
```



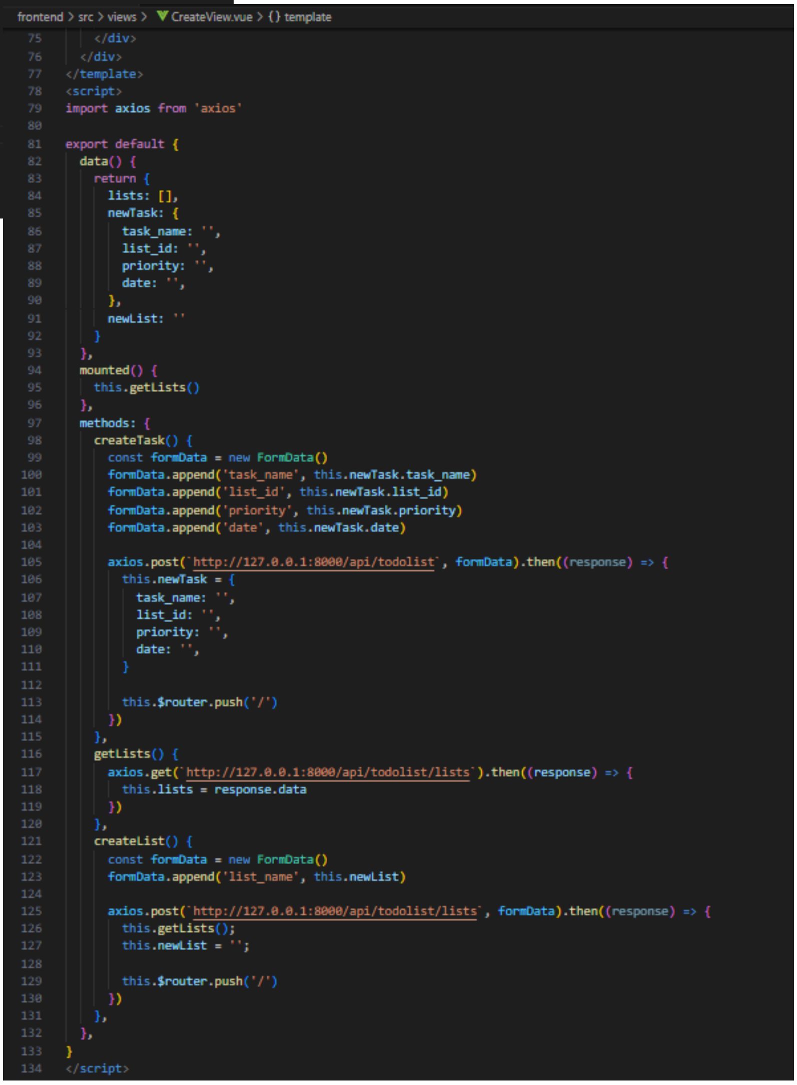
```
      const confirmed = confirm('Konfirmasi hapus?')
      if (confirmed) {
        axios.delete(`http://127.0.0.1:8000/api/todolist/${task.id}`).then(() => {
          this.getTask()
        })
      }
    },
  },
}
</script>
```

Sedangkan ini adalah codingan dari halaman untuk create, yah intinya begitulah



frontend > src > views > CreateView.vue > {} template

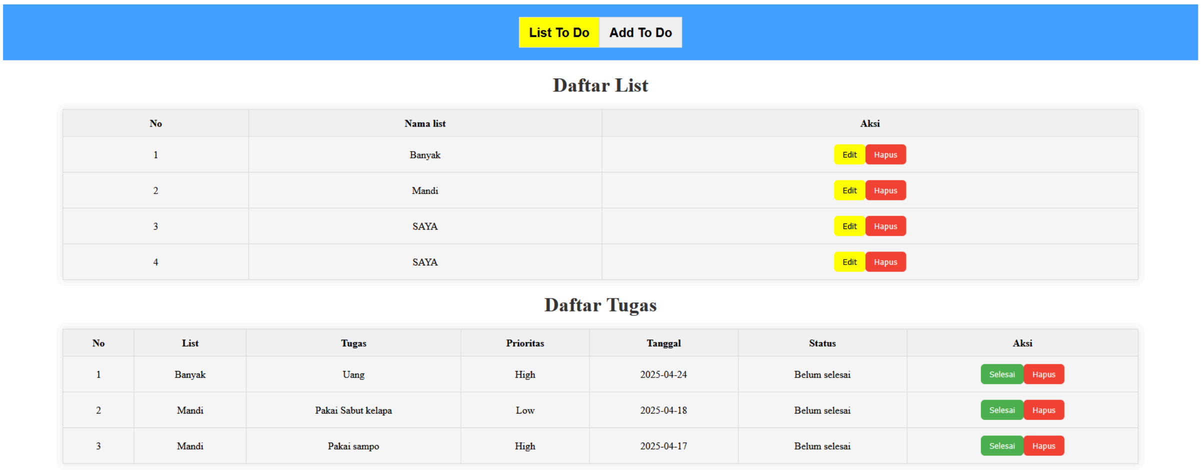
```
<template lang="">
  <nav class="navbar-container">
    <tr>
      <th>
        <router-link to="/" style="text-decoration: none; color: black">List To Do</router-link>
      </th>
      <th style="background-color: yellow">
        <router-link to="/create" style="text-decoration: none; color: black"
          >Add To Do</router-link
        >
      </th>
    </tr>
  </nav>
  <div class="overlay">
    <div class="popup" style="margin-right: 50px;">
      <h2>Tambah List Baru</h2>
      <form id="tambah" @submit.prevent="createList()">
        <div class="form-group">
          <label>Nama List:</label>
          <input
            v-model="newList"
            type="text"
            placeholder="Masukkan nama list"
            required
          />
        </div>
        <div class="popup-buttons">
          <button type="submit" class="btn primary">Tambah</button>
        </div>
      </form>
    </div>
    <div class="popup">
      <h2>Tambah Tugas Baru</h2>
      <form id="tambah" @submit.prevent="createTask()">
        <div class="form-group">
          <label>Nama Tugas:</label>
          <input
            v-model="newTask.task_name"
            type="text"
            placeholder="Masukkan nama tugas"
            required
          />
        </div>
        <div class="form-group">
          <label>List:</label>
          <select v-model="newTask.list_id" required>
            <option value="" disabled selected>Pilih list</option>
            <option v-for="(item, index) in lists" :key="item.id" :value="item.id">
              {{ item.list_name }}
            </option>
          </select>
        </div>
        <div class="form-group">
          <label>Prioritas:</label>
          <select v-model="newTask.priority" required>
            <option value="" selected disabled>Pilih prioritas</option>
            <option value="High">High</option>
            <option value="Medium">Medium</option>
            <option value="Low">Low</option>
          </select>
```



frontend > src > views > CreateView.vue > {} template

```
      </div>
    </div>
</template>
<script>
import axios from 'axios'

export default {
  data() {
    return {
      lists: [],
      newTask: {
        task_name: '',
        list_id: '',
        priority: '',
        date: '',
      },
      newList: ''
    }
  },
  mounted() {
    this.getLists()
  },
  methods: {
    createTask() {
      const formData = new FormData()
      formData.append('task_name', this.newTask.task_name)
      formData.append('list_id', this.newTask.list_id)
      formData.append('priority', this.newTask.priority)
      formData.append('date', this.newTask.date)

      axios.post(`http://127.0.0.1:8000/api/todolist`, formData).then((response) => {
        this.newTask = {
          task_name: '',
          list_id: '',
          priority: '',
          date: '',
        }

        this.$router.push('/')
      })
    },
    getLists() {
      axios.get(`http://127.0.0.1:8000/api/todolist/lists`).then((response) => {
        this.lists = response.data
      })
    },
    createList() {
      const formData = new FormData()
      formData.append('list_name', this.newList)

      axios.post(`http://127.0.0.1:8000/api/todolist/lists`, formData).then((response) => {
        this.getLists();
        this.newList = '';

        this.$router.push('/')
      })
    },
  },
}
</script>
```

Baiklah kita akan masuk ke tampilan dari project ini



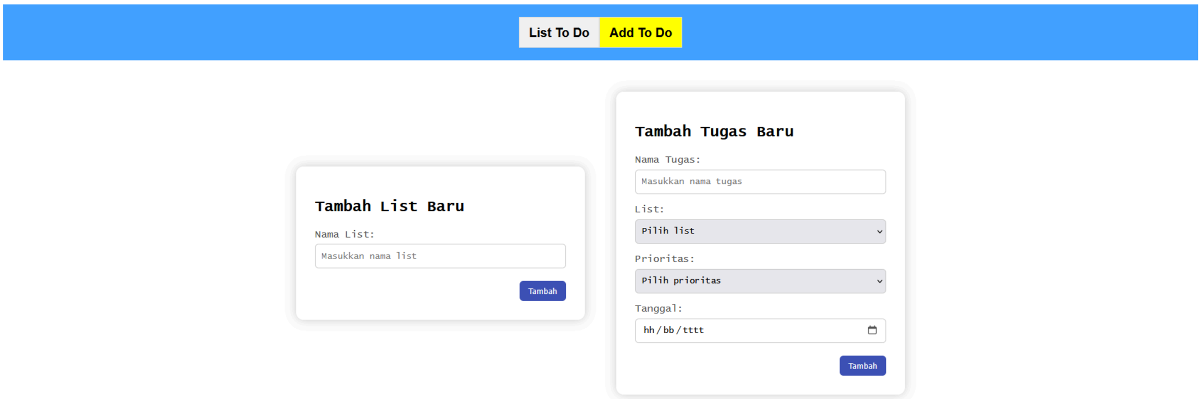
Daftar List

No	Nama list	Aksi
1	Banyak	Edit Hapus
2	Mandi	Edit Hapus
3	SAYA	Edit Hapus
4	SAYA	Edit Hapus

Daftar Tugas

No	List	Tugas	Prioritas	Tanggal	Status	Aksi
1	Banyak	Uang	High	2025-04-24	Belum selesai	Selesai Hapus
2	Mandi	Pakai Sabut kelapa	Low	2025-04-18	Belum selesai	Selesai Hapus
3	Mandi	Pakai sampo	High	2025-04-17	Belum selesai	Selesai Hapus

Ini adalah tampilan utama dari project saya karena disini kita bisa melihat semua daftar list dan daftar tugas, kita juga bisa edit delete listnya, tugasnya juga kita bisa selesaikan atau belum juga bisa hapus



Tambah List Baru

Nama List:

Masukkan nama list

Tambah

Tambah Tugas Baru

Nama Tugas:

Masukkan nama tugas

List:

Pilih list

Prioritas:

Pilih prioritas

Tanggal:

hh / bb / tttt

Tambah

Sedangkan ini halaman untuk mengcreate tugas dan list

Nampaknya itu saja mungkin dari saya, saya gak tau lagi bagaimana

TERIMA KASIH