



2025

UJI KOMPETENSI KEAHLIAN PAKET 4

Pengembangan Perangkat Lunak dan Gim

DAFTAR ISI

PEMBUKAAN.....	i
HALAMAN CODINGAN LARAVEL.....	ii
HALAMAN CODINGAN VUE JS.....	iii
HALAMAN TAMPILAN AKHIR.....	iv

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

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Task;
6  use Illuminate\Http\Request;
7
8  6 references | 0 implementations
9  class TaskController extends Controller
10 {
11     1 reference | 0 overrides
12     public function read(): array|Collection {
13         return Task::with('relations')->get();
14     }
15
16     1 reference | 0 overrides
17     public function create(Request $request): void {
18         Task::create(attributes: [
19             'list_id' => $request->list_id,
20             'task_name' => $request->task_name,
21             'priority' => $request->priority,
22             'status' => 'Belum selesai',
23             'date' => $request->date
24         ]);
25     }
26
27     1 reference | 0 overrides
28     public function update($id): void {
29         $task = Task::find(id: $id);
30         $task->update(attributes: [
31             'status' => 'Selesai'
32         ]);
33     }
34
35     1 reference | 0 overrides
36     public function downdate($id): void {
37         $task = Task::find(id: $id);
38         $task->update(attributes: [
39             'status' => 'Belum selesai'
40         ]);
41     }
42
43     1 reference | 0 overrides
44     public function delete($id): void {
45         $task = Task::find(id: $id);
46         $task->delete();
47     }
48 }
```

Ini adalah TaskController, yang berfungsi untuk membuat CRUD dari Task

```

backend > app > Http > Controllers > ListController.php > PHP Intelephense > ListController > update
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Lists;
6  use Illuminate\Http\Request;
7
8  6 references | 0 implementations
9  class ListController extends Controller
10 {
11     1 reference | 0 overrides
12     public function read(): Collection {
13         return Lists::all();
14     }
15
16     1 reference | 0 overrides
17     public function readDetail($id): array|Collection {
18         return Lists::find(id: $id)->with(relations: 'tasks')->get();
19     }
20
21     1 reference | 0 overrides
22     public function create(Request $request): void {
23         Lists::create(attributes: [
24             'list_name' => $request->list_name
25         ]);
26     }
27
28     1 reference | 0 overrides
29     public function update(Request $request, $id): void {
30         $task = Lists::find(id: $id);
31         $task->update(attributes: [
32             'list_name' => $request->list_name
33         ]);
34     }
35
36     1 reference | 0 overrides
37     public function delete($id): void {
38         $task = Lists::findOrFail(id: $id);
39         $task->delete();
40     }
41 }

```

Ini adalah ListController, yang berfungsi untuk membuat CRUD dari List

```

backend > routes > api.php
1  <?php
2
3  use App\Http\Controllers\ListController;
4  use App\Http\Controllers\TaskController;
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Route;
7
8  /*
9  |-----
10 | API Routes
11 |-----
12 |
13 | Here is where you can register API routes for your application. These
14 | routes are loaded by the RouteServiceProvider and all of them will
15 | be assigned to the "api" middleware group. Make something great!dd
16 |
17 */
18
19 Route::get(uri: '/todolist', action: [TaskController::class, 'read']);
20 Route::post(uri: '/todolist', action: [TaskController::class, 'create']);
21 Route::patch(uri: '/todolist/{id}', action: [TaskController::class, 'update']);
22 Route::patch(uri: '/todolist/{id}/downdate', action: [TaskController::class, 'downdate']);
23 Route::delete(uri: '/todolist/{id}', action: [TaskController::class, 'delete']);
24
25 Route::get(uri: '/todolist/lists', action: [ListController::class, 'read']);
26 Route::get(uri: '/todolist/lists/{id}', action: [ListController::class, 'readDetail']);
27 Route::post(uri: '/todolist/lists', action: [ListController::class, 'create']);
28 Route::patch(uri: '/todolist/lists/{id}', action: [ListController::class, 'update']);
29 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
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create(table: 'lists', callback: function (Blueprint $table): void {
15               $table->id();
16               $table->string(column: 'list_name');
17               $table->timestamps();
18           });
19       }
20
21       /**
22        * Reverse the migrations.
23        */
24       public function down(): void
25       {
26           Schema::dropIfExists(table: 'lists');
27       }
28   };
29

```

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

```
backend > database > migrations > 2025_04_14_014215_create_tasks_table.php > class > up > Closure
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create(table: 'tasks', callback: function (Blueprint $table): void {
15             $table->id();
16             $table->unsignedBigInteger(column: 'list_id');
17             $table->string(column: 'task_name');
18             $table->string(column: 'priority');
19             $table->string(column: 'status');
20             $table->date(column: 'date');
21             $table->timestamps();
22
23             $table->foreign(columns: 'list_id')->references(columns: 'id')->on(table: 'lists')->
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists(table: 'tasks');
33     }
34 };
35
```

Sedangkan ini migrations dari tabel tasks

```
backend > app > Models > Lists.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  7 references | 0 implementations
9  class Lists extends Model
10 {
11     0 references
12     protected $table = 'lists';
13     use HasFactory;
14     0 references
15     public $fillable = [
16         'list_name',
17     ];
18
19     0 references | 0 overrides
20     public function tasks(): HasMany
21     {
22         return $this->hasMany(related: Task::class, foreignKey: 'list_id', localKey: 'id');
23     }
24 }
```

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

Sedangkan ini model dari Task

```
backend > app > Models > Task.php > PHP Intelephense > Task > lists
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Task extends Model
9  {
10     use HasFactory;
11     public $fillable = [
12         'task_name',
13         'list_id',
14         'priority',
15         'status',
16         'date'
17     ];
18
19     public function lists(): BelongsTo
20     {
21         return $this->belongsTo(related: Lists::class, foreignKey: 'list_id', ownerKey: 'id');
22     }
23 }
24
```

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

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

```
frontend > src > views > HomeView.vue > {} template > div.overlay
1 <template>
2 <nav class="navbar-container">
3 <tr>
4 <th style="background-color: yellow">
5 <router-link to="/" style="text-decoration: none; color: black">List To Do</router-link>
6 </th>
7 <th>
8 <router-link to="/create" style="text-decoration: none; color: black">
9 >Add To Do</router-link>
10 </th>
11 </tr>
12 </nav>
13
14
15 <h1 class="title">Daftar List</h1>
16 <div class="table-container">
17 <table>
18 <thead>
19 <tr>
20 <th>No</th>
21 <th>Nama list</th>
22 <th>Aksi</th>
23 </tr>
24 </thead>
25 <tbody>
26 <tr v-for="(list, index) in lists" :key="list.id">
27 <td>{{ index + 1 }}</td>
28 <td>{{ list.list_name }}</td>
29 <td>
30 <button class="btn warning" @click="openEditPopup(list)">Edit</button>
31 <button class="btn danger" @click="deleteList(list)">Hapus</button>
32 </td>
33 </tr>
34 </tbody>
35 </table>
36 </div>
37
38 <div class="overlay" v-if="showPopup">
39 <div class="popup">
40 <h2>Edit list</h2>
41 <form @submit.prevent="updateList()">
42 <div class="form-group">
43 <label>Nama list baru:</label>
44 <input v-model="newListName" type="text" placeholder="Masukkan nama list baru" required />
45 </div>
46 <div class="popup-buttons">
47 <button type="submit" class="btn primary" style="margin-right: 5px">Edit</button>
48 <button type="button" class="btn secondary" @click="closePopup">Tutup</button>
49 </div>
50 </form>
51 </div>
52 </div>
53
54 <h1 class="title">Daftar Tugas</h1>
55 <div class="table-container">
56 <table>
57 <thead>
58 <tr>
59 <th>No</th>
60 <th>List</th>
```

```
frontend > src > views > HomeView.vue > {} template > div.overlay
61 <th>Tugas</th>
62 <th>Prioritas</th>
63 <th>Tanggal</th>
64 <th>Status</th>
65 <th>Aksi</th>
66 </tr>
67 </thead>
68 <tbody>
69 <tr v-for="(task, index) in tasks" :key="task.id">
70 <td>{{ index + 1 }}</td>
71 <td>{{ task.lists.list_name }}</td>
72 <td>{{ task.task_name }}</td>
73 <td>{{ task.priority }}</td>
74 <td>{{ task.date }}</td>
75 <td>{{ task.status }}</td>
76 <td>
77 <button class="btn success" v-if="task.status !== 'Selesai'" @click="updateTask(task)">
78 Selesai
79 </button>
80 <button class="btn danger" @click="deleteTask(task)">Hapus</button>
81 <button
82 class="btn warning"
83 v-if="task.status !== 'Belum selesai'"
84 @click="downDateTask(task)"
85 >
86 Belum Selesai
87 </button>
88 </td>
89 </tr>
90 </tbody>
91 </table>
92 </div>
93 </template>
94
95 <script>
96 import axios from 'axios'
97
98 export default {
99 data() {
100 return {
101 showPopup: false,
102 tasks: [],
103 lists: [],
104 newTask: {
105 task_name: '',
106 priority: '',
107 date: '',
108 },
109 newListName: '',
110 listToEdit: null,
111 },
112 },
113 mounted() {
114 this.getTask()
115 this.getLists()
116 },
117 methods: {
118 getLists() {
119 axios.get('http://127.0.0.1:8000/api/todolist/lists').then((response) => {
120 this.lists = response.data
```

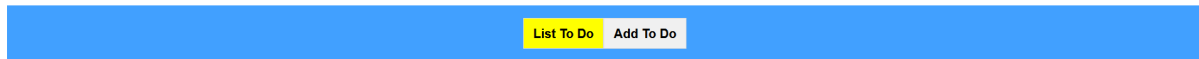

frontend > src > views > HomeView.vue > {} template > div.overlay

```
120     this.lists = response.data
121   })
122 },
123 deletelist(list) {
124   const confirmed = confirm('Konfirmasi hapus?')
125   if (confirmed) {
126     axios.delete(`http://127.0.0.1:8000/api/todolist/lists/${list.id}`).then(() => {
127       this.getTask()
128       this.getLists()
129     })
130   }
131 },
132 openEditPopup(list) {
133   this.showPopup = true
134   this.listToEdit = list
135   this.newListName = list.list_name
136 },
137 closePopup() {
138   this.showPopup = false
139   this.newListName = ''
140   this.listToEdit = null
141 },
142 updatelist() {
143   if (!this.newListName.trim()) return
144   axios
145     .patch(`http://127.0.0.1:8000/api/todolist/lists/${this.listToEdit.id}`, {
146       list_name: this.newListName,
147     })
148     .then(() => {
149       this.getLists()
150       this.getTask()
151       this.closePopup()
152     })
153     .catch((error) => {
154       console.error('Gagal update list:', error)
155     })
156 },
157
158 getTask() {
159   axios.get('http://127.0.0.1:8000/api/todolist').then((response) => {
160     this.tasks = response.data
161   })
162 },
163 updateTask(task) {
164   const confirmed = confirm('Konfirmasi selesai?')
165   if (confirmed) {
166     axios.patch(`http://127.0.0.1:8000/api/todolist/${task.id}`).then(() => {
167       this.getTask()
168     })
169   }
170 },
171 downdateTask(task) {
172   const confirmed = confirm('Konfirmasi belum?')
173   if (confirmed) {
174     axios.patch(`http://127.0.0.1:8000/api/todolist/${task.id}/downdate`).then(() => {
175       this.getTask()
176     })
177   }
178 },
179 deleteTask(task) {
180   const confirmed = confirm('Konfirmasi hapus?')
181   if (confirmed) {
182     axios.delete(`http://127.0.0.1:8000/api/todolist/${task.id}`).then(() => {
183       this.getTask()
184     })
185   }
186 },
187 },
188 }
</script>
```

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

```
frontend > src > views > CreateView.vue > {} template
1 <template lang="">
2 <nav class="navbar-container">
3 <tr>
4 <th>
5 <router-link to="/" style="text-decoration: none; color: black">List To Do</router-link>
6 </th>
7 <th style="background-color: yellow">
8 <router-link to="/create" style="text-decoration: none; color: black">
9 >Add To Do</router-link>
10 </th>
11 </tr>
12 </nav>
13
14 <div class="overlay">
15 <div class="popup" style="margin-right: 50px;">
16 <h2>Tambah List Baru</h2>
17 <form id="tambah" @submit.prevent="createList()">
18 <div class="form-group">
19 <label>Nama List:</label>
20 <input
21 v-model="newList"
22 type="text"
23 placeholder="Masukkan nama list"
24 required
25 />
26 </div>
27 <div class="popup-buttons">
28 <button type="submit" class="btn primary">Tambah</button>
29 </div>
30 </form>
31 </div>
32 <div class="popup">
33 <h2>Tambah Tugas Baru</h2>
34 <form id="tambah" @submit.prevent="createTask()">
35 <div class="form-group">
36 <label>Nama Tugas:</label>
37 <input
38 v-model="newTask.task_name"
39 type="text"
40 placeholder="Masukkan nama tugas"
41 required
42 />
43 </div>
44 <div class="form-group">
45 <label>List:</label>
46 <select v-model="newTask.list_id" required>
47 <option value="" disabled selected>Pilih list</option>
48 <option v-for="(item, index) in lists" :key="item.id" :value="item.id">
49 {{ item.list_name }}
50 </option>
51 </select>
52 </div>
53 <div class="form-group">
54 <label>Prioritas:</label>
55 <select v-model="newTask.priority" required>
56 <option value="" selected disabled>Pilih prioritas</option>
57 <option value="High">High</option>
58 <option value="Medium">Medium</option>
59 <option value="Low">Low</option>
60 </select>
61 </div>
62 </form>
63 </div>
64 </template>
65
66 <script>
67 import axios from 'axios'
68
69 export default {
70   data() {
71     return {
72       lists: [],
73       newTask: {
74         task_name: '',
75         list_id: '',
76         priority: '',
77         date: '',
78       },
79       newList: ''
80     }
81   },
82   mounted() {
83     this.getLists()
84   },
85   methods: {
86     createTask() {
87       const formData = new FormData()
88       formData.append('task_name', this.newTask.task_name)
89       formData.append('list_id', this.newTask.list_id)
90       formData.append('priority', this.newTask.priority)
91       formData.append('date', this.newTask.date)
92
93       axios.post('http://127.0.0.1:8000/api/todolist', formData).then((response) => {
94         this.newTask = {
95           task_name: '',
96           list_id: '',
97           priority: '',
98           date: '',
99         }
100       })
101       this.$router.push('/')
102     },
103     getLists() {
104       axios.get('http://127.0.0.1:8000/api/todolist/lists').then((response) => {
105         this.lists = response.data
106       })
107     },
108     createList() {
109       const formData = new FormData()
110       formData.append('list_name', this.newList)
111
112       axios.post('http://127.0.0.1:8000/api/todolist/lists', formData).then((response) => {
113         this.getLists()
114         this.newList = ''
115       })
116       this.$router.push('/')
117     }
118   },
119 }
120 </script>
```

Baiklah kita akan masuk ke tampilan dari project ini



Daftar List

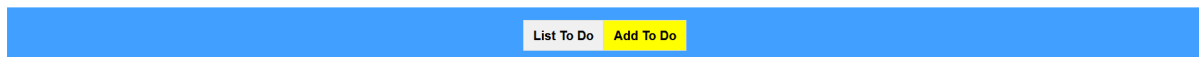
No	Nama list	Aksi
1	Banyak	Edit Hapus
2	Mandi	Edit Hapus
3	SAYA	Edit Hapus
4	SAYA	Edit Hapus

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:

[Tambah](#)

Tambah Tugas Baru

Nama Tugas:

List:

Pilih list

Prioritas:

Pilih prioritas

Tanggal:

hh/bb/yyyy

[Tambah](#)



Sedangkan ini halaman untuk mengcreate tugas dan list

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

TERIMA KASIH