

## **LAPORAN IMPAL – WEEK 1 – 2**

Anggota kelompok :

1. M Faruq Rantisi: (1203230122)
2. Galang Ubaidillah: (1203230056)
3. Rahmadi Rafiansyah: (1203230075)
4. Ando Ammar Rizqullah: (1203230102)
5. Surya Adi Arga Widana: (1203230101)

Rahmadi Rafianasyah

Halaman Welcome

```
1 import { PlaceholderPattern } from '@components/ui/placeholder-pattern';
2 import AppLayout from '@layouts/app-layout';
3 import { dashboard } from '@routes';
4 import { type BreadcrumbItem } from '@types';
5 import { Head } from '@inertias/react';
6
7 const breadcrumbs: BreadcrumbItem[] = [
8   {
9     title: 'Dashboard',
10    href: dashboard().url,
11  },
12 ];
13
14 export default function Dashboard() {
15   return (
16     <AppLayout breadcrumbs={breadcrumbs}>
17       <Head title="Dashboard" />
18       <div className="relative w-full flex flex-col items-center">
19         { /* Notifikasi */ }
20         <button className="absolute top-0 right-0 mt-2 mr-4 rounded-full border-2 border-[#bf1206] p-2">
21           <svg
22             className="text-[#bf1206]"
23             width="24"
24             height="24"
25             fill="none"
26             stroke="currentColor"
27             strokeWidth="2"
28             viewBox="0 0 24 24"
29           >
30             <path d="M18 8a6 6 0 10-12 0c0 7-3 9-3 9h18s-3-2-3-9" />
31             <path d="M13.73 21a2 2 0 01-3.46 0" />
32           </svg>
33         </button>
34         <div className="flex flex-col items-center w-full max-w-md gap-3 mt-8">
35           { /* Search Bar */ }
36           <div className="relative w-full">
37             <input
38               type="text"
39               placeholder="Cari Item"
40               className="w-full rounded-lg bg-[#bf1206] bg-opacity-90 text-white placeholder-white px-4 py-2 focus:outline-none"
41             />
42             <svg
43               className="absolute left-3 top-1/2 -translate-y-1/2 text-white opacity-80"
44               width="18"
45               height="18"
46               fill="none"
47               stroke="currentColor"
48               strokeWidth="2"
49               viewBox="0 0 24 24"
50             >
51               <circle cx="11" cy="11" r="8" />
52               <line x1="21" y1="21" x2="16.65" y2="16.65" />
53             </svg>
54           </div>
55           { /* Saldo */ }
56           <div className="flex items-center bg-white rounded-lg shadow px-4 py-2 w-full justify-center">
57             <svg
58               className="text-[#bf1206] mr-2"
59               width="20"
60               height="20"
61               fill="none"
62               stroke="currentColor"
63               strokeWidth="2"
64               viewBox="0 0 24 24"
65             >
66               <rect x="2" y="7" width="20" height="14" rx="4" />
67               <path d="M16 3v4M8 3v4" />
68             </svg>
69             <span className="font-medium text-[#bf1206]">Rp 45.000</span>
70           </div>
71         </div>
72       </div>
73     </AppLayout>
74   );
75 }
76
```

Halaman Dashboard

```

1 import { dashboard, login, register } from '@routes';
2 import { type sharedData } from '@types';
3 import { Head, Link, script } from '@reactjs/react';
4
5 export default function Welcome() {
6   const { auth } = usePageSharedData().props;
7
8   return (
9     <>
10      <Head title="Welcome">
11        <link rel="preconnect" href="https://fonts.bunny.net" />
12        <link
13          href="https://fonts.bunny.net/css?family=instrument-sans:400,500,600"
14          rel="stylesheet"
15        />
16      </Head>
17      <div className="flex min-h-screen flex-col items-center justify-center bg-[#F0F0F0] p-6 text-[#1B1B1B] dark:bg-[#2A2A2A]>
18        <div className="w-full max-w-sm rounded-lg bg-white p-8 shadow-lg flex flex-col items-center dark:bg-[#1B1B1B] dark:text-[#F0F0F0]>
19          <h1 className="text-3xl font-bold mb-2 text-center">Koperasi Tel-U/HS
20          <p className="text-center text-[#F0F0F0] mb-6 dark:text-[#1B1B1B]>
21            Selamat datang di Koperasi Tel-U, anda bisa membeli apa yang anda cari disini.
22          </p>
23          <div className="mt-8 flex flex-col w-full max-w-sm gap-2">
24            {auth.user ? (
25              <Link
26                href="/dashboard/"
27                className="w-full rounded-md bg-[#F0F0F0] px-5 py-2 text-white text-center font-medium hover:bg-[#A8A8A8] dark:bg-[#F0F0F0] dark:text-[#1B1B1B] dark:hover:bg-[#D6D6D6]"
28              >
29                Dashboard
30            </Link>
31            ) : (
32              <Link
33                href="/login/"
34                className="w-full rounded-md bg-[#F0F0F0] px-5 py-2 text-white text-center font-medium hover:bg-[#A8A8A8] dark:bg-[#F0F0F0] dark:text-[#1B1B1B] dark:hover:bg-[#D6D6D6]"
35              >
36                Login
37            </Link>
38            <Link
39                href="/register/"
40                className="w-full rounded-md border border-[#F0F0F0] px-5 py-2 text-[#F0F0F0] text-center font-medium hover:bg-[#F0F0F0] dark:border-[#F0F0F0] dark:text-[#F0F0F0] dark:hover:bg-[#232323]"
41              >
42                Register
43            </Link>
44          </div>
45        </div>
46      </div>
47    </>
48  );
49 }

```

M Faruq Rantisi

Halaman Profile



Surya

Pemesanana pembayaran

```
1  <?php
2
3  namespace App\Domain\Order;
4
5  class CatalogService
6  {
7      // Contoh item umum di koperasi kampus
8      public function all(): array
9      {
10         return [
11             // MINUMAN
12             ['id' => 'DR-AQUA-600', 'name' => 'Aqua 600ml', 'category' => 'MINUMAN', 'price' => 5000],
13             ['id' => 'DR-TEHBOT', 'name' => 'Teh Botol Sosro 350ml', 'category' => 'MINUMAN', 'price' => 6000],
14             ['id' => 'DR-GOODDAY', 'name' => 'Good Day Cappuccino', 'category' => 'MINUMAN', 'price' => 8000],
15             ['id' => 'DR-POCARI', 'name' => 'Pocari Sweat 350ml', 'category' => 'MINUMAN', 'price' => 9000],
16             // MAKANAN
17             ['id' => 'FD-INDOMIEG', 'name' => 'Indomie Goreng (cup)', 'category' => 'MAKANAN', 'price' => 7000],
18             ['id' => 'FD-ROTI', 'name' => 'Roti Tawar/Isian', 'category' => 'MAKANAN', 'price' => 8000],
19             ['id' => 'FD-RISOL', 'name' => 'Risol Mayo', 'category' => 'MAKANAN', 'price' => 6000],
20             ['id' => 'FD-PASTEL', 'name' => 'Pastel', 'category' => 'MAKANAN', 'price' => 6000],
21         ];
22     }
23
24     public function find(string $productId): ?array
25     {
26         foreach ($this->all() as $p) {
27             if ($p['id'] === $productId)
28                 return $p;
29         }
30         return null;
31     }
32 }
```

```

1  <?php
2
3  namespace App\Domain\Order;
4
5  use App\Infrastructure\OrderFileStore;
6  use Illuminate\Support\Str;
7  use RuntimeException;
8
9  class OrderService
10 {
11     public function __construct(
12         private CatalogService $catalog,
13         private OrderFileStore $store
14     ) {
15     }
16
17     public function create(array $input): array
18     {
19         // 1) Validasi product_id + hitung harga
20         $lines = [];
21         $subtotal = 0;
22
23         foreach ($input['items'] as $i) {
24             $prod = $this->catalog->find($i['product_id']);
25             if (!$prod) {
26                 throw new RuntimeException('PRODUCT_NOT_FOUND: ' . $i['product_id']);
27             }
28             $qty = (int) $i['qty'];
29             $lineTotal = $qty * $prod['price'];
30             $lines[] = [
31                 'product_id' => $prod['id'],
32                 'name' => $prod['name'],
33                 'category' => $prod['category'],
34                 'price' => $prod['price'],
35                 'qty' => $qty,
36                 'line_total' => $lineTotal,
37             ];
38             $subtotal += $lineTotal;
39         }
40
41         // 2) Hitung total
42         $total = $subtotal;
43
44         // 3) Bentuk order
45         $id = $this->store->nextId();
46         $code = 'ORD-' . Str::upper(Str::random(6));
47
48         $order = [
49             'id' => $id,
50             'code' => $code,
51             'customer_id' => $input['customer_id'],
52             'status' => 'UNPAID',
53             'items' => $lines,
54             'subtotal' => $subtotal,
55             'total' => $total,
56             'notes' => $input['notes'] ?? null,
57             'payment' => null,
58             'created_at' => now()->toIso8601String(),
59             'updated_at' => now()->toIso8601String(),
60         ];
61
62         // 4) Simpan
63         $this->store->put($order);
64         return $order;
65     }
66
67     public function pay(array $order, string $method): array
68     {
69         if ($order['status'] !== 'UNPAID') {
70             throw new RuntimeException('INVALID_STATUS_TRANSITION');
71         }
72
73         $order['status'] = 'PAID';
74         $order['payment'] = ['method' => $method, 'paid_at' => now()->toIso8601String()];
75         $order['updated_at'] = now()->toIso8601String();
76
77         $this->store->put($order);
78         return $order;
79     }
80 }
81

```



```
1  <?php
2
3  namespace App\Http\Controllers\Api\V1;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Resources\CatalogResource;
7  use App\Domain\Order\CatalogService;
8
9  class CatalogController extends Controller
10 {
11     public function __construct(private CatalogService $catalog)
12     {
13     }
14
15     public function index()
16     {
17         return new CatalogResource($this->catalog->all());
18     }
19 }
20
```



```

1  <?php
2
3  namespace App\Http\Controllers\Api\V1;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Requests\Order\CreateOrderRequest;
7  use App\Http\Requests\Order\PayOrderRequest;
8  use App\Http\Resources\OrderResource;
9  use App\Domain\Order\OrderService;
10 use App\Domain\Order\CatalogService;
11 use App\Infrastructure\OrderFileStore;
12 use Illuminate\Http\Request;
13 use Throwable;
14
15 class OrderController extends Controller
16 {
17     private OrderService $service;
18     private OrderFileStore $store;
19
20     public function __construct()
21     {
22         $catalog = new CatalogService();
23         $this->store = new OrderFileStore();
24         $this->service = new OrderService($catalog, $this->store);
25     }
26
27     public function store(CreateOrderRequest $request)
28     {
29         try {
30             $order = $this->service->create($request->validated());
31             return new OrderResource($order);
32         } catch (Throwable $e) {
33             return response()->json([
34                 'status' => 'error',
35                 'reason' => str_contains($e->getMessage(), 'PRODUCT_NOT_FOUND')
36                     ? 'PRODUCT_NOT_FOUND' : 'PROCESS_FAILED',
37                 'message' => app()->hasDebugModeEnabled() ? $e->getMessage() : null,
38             ], 422);
39         }
40     }
41
42     public function show(string $id)
43     {
44         $order = $this->store->get($id);
45         if (!$order) {
46             return response()->json(['status' => 'error', 'reason' => 'NOT_FOUND'], 404);
47         }
48         return new OrderResource($order);
49     }
50
51     public function pay(PayOrderRequest $request, string $id)
52     {
53         $order = $this->store->get($id);
54         if (!$order) {
55             return response()->json(['status' => 'error', 'reason' => 'NOT_FOUND'], 404);
56         }
57         try {
58             $updated = $this->service->pay($order, $request->validated()['payment_method']);
59             return new OrderResource($updated);
60         } catch (Throwable $e) {
61             return response()->json([
62                 'status' => 'error',
63                 'reason' => 'INVALID_STATUS_TRANSITION',
64                 'message' => app()->hasDebugModeEnabled() ? $e->getMessage() : null,
65             ], 422);
66         }
67     }
68 }
69

```

```
1  <?php
2
3  namespace App\Http\Requests\Order;
4
5  use Illuminate\Foundation\Http\FormRequest;
6
7  class CreateOrderRequest extends FormRequest
8  {
9      public function rules(): array
10     {
11         return [
12             'customer_id' => ['required','string','max:64'],
13             'items'       => ['required','array','min:1'],
14             'items.*.product_id'=> ['required','string','max:32'],
15             'items.*.qty'  => ['required','integer','min:1','max:50'],
16             'notes'       => ['sometimes','string','max:500'],
17         ];
18     }
19 }
20
```

```
1  <?php
2
3  namespace App\Http\Requests\Order;
4
5  use Illuminate\Foundation\Http\FormRequest;
6
7  class PayOrderRequest extends FormRequest
8  {
9      public function rules(): array
10     {
11         return [
12             'payment_method' => ['required', 'string', 'max:32'], // e.g. CASH, QRIS
13         ];
14     }
15 }
```



```
1  <?php
2
3  namespace App\Http\Resources;
4
5  use Illuminate\Http\Resources\Json\JsonResource;
6
7  class CatalogResource extends JsonResource
8  {
9      public function toArray($request): array
10     {
11         return ['status' => 'ok', 'data' => $this->resource];
12     }
13 }
14
```



```
1  <?php
2
3  namespace App\Http\Resources;
4
5  use Illuminate\Http\Resources\Json\JsonResource;
6
7  class OrderResource extends JsonResource
8  {
9      public function toArray($request): array
10     {
11         return ['status' => 'ok', 'data' => $this->resource];
12     }
13 }
14
```

```

1  <?php
2
3  namespace App\Infrastructure;
4
5  use Illuminate\Support\Facades\Storage;
6  use Illuminate\Support\Str;
7
8  class OrderFileStore
9  {
10     private string $file = 'orders.json';
11
12     private function read(): array
13     {
14         if (!Storage::exists($this->file))
15             return [];
16         return json_decode(Storage::get($this->file), true) ?: [];
17     }
18
19     private function write(array $data): void
20     {
21         Storage::put($this->file, json_encode($data, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE));
22     }
23
24     public function nextId(): string
25     {
26         return (string) Str::uuid();
27     }
28
29     public function put(array $order): void
30     {
31         $data = $this->read();
32         $data[$order['id']] = $order;
33         $this->write($data);
34     }
35
36     public function get(string $id): ?array
37     {
38         $data = $this->read();
39         return $data[$id] ?? null;
40     }
41 }
42

```

Galang ubaidillah  
 menyimpan data pengguna kopras





```
1  <?php
2
3  namespace App\Models;
4
5  // use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use Laravel\Fortify\TwoFactorAuthenticatable;
10
11 class User extends Authenticatable
12 {
13     /** @use HasFactory<\Database\Factories\UserFactory> */
14     use HasFactory, Notifiable, TwoFactorAuthenticatable;
15
16     /**
17      * The attributes that are mass assignable.
18      *
19      * @var list<string>
20      */
21     protected $fillable = [
22         'name',
23         'email',
24         'password',
25     ];
26
27     /**
28      * The attributes that should be hidden for serialization.
29      *
30      * @var list<string>
31      */
32     protected $hidden = [
33         'password',
34         'remember_token',
35     ];
36
37     /**
38      * Get the attributes that should be cast.
39      *
40      * @return array<string, string>
41      */
42     protected function casts(): array
43     {
44         return [
45             'email_verified_at' => 'datetime',
46             'password' => 'hashed',
47         ];
48     }
49 }
50
```

