LAMPIRAN

Lampiran-1 Login

```
<script setup lang="ts">
import InputError from
'@/components/InputError.vue';
import TextLink from
'@/components/TextLink.vue';
import { Button } from
'@/components/ui/button';
import { Checkbox } from
'@/components/ui/checkbox';
import { Input } from
'@/components/ui/input';
import { Label } from
'@/components/ui/label';
import AuthBase from
'@/layouts/AuthLayout.vue';
import { Head, useForm } from
'@inertiajs/vue3';
import { LoaderCircle } from
'lucide-vue-next';

defineProps<{
    status?: string;
    canResetPassword: boolean;
}>();

const form = useForm({
    email: '',
    password: '',
    remember: false,
});

const submit = () => {
    form.post(route('login'),
{
        onFinish: () => {
            form.password =
'';

console.log('oke');
        },
    });
};
</script>

<template>
    <AuthBase title="Masuk ke
Akun Anda"
description="Masukkan email
dan kata sandi Anda di bawah
ini untuk masuk">
        <Head title="Log in"
/>

        <div v-if="status"
class="mb-4 text-center text-
sm font-medium text-green-
600">
            {{ status }}
        </div>

        <form
@submit.prevent="submit"
class="flex flex-col gap-6">
            <div class="grid
gap-6">
                <div
class="grid gap-2">
                    <Label
for="email">Email</Label>
                    <Input

id="email"

type="email"

required

autofocus

:tabindex="1"

autocomplete="off"
                        v-
model="form.email"

placeholder="email@example.com
"
                    />

<InputError
:message="form.errors.email"
/>
                </div>

                <div
class="grid gap-2">
                    <div
class="flex items-center
justify-between">
                        <Label
for="password">Password</Label
>
                        <!--
<TextLink v-
if="canResetPassword"
:href="route('password.request
')" class="text-sm"
:tabindex="5">
```

```
Forgot password?

</TextLink> -->
                    </div>
                    <Input

id="password"

type="password"

required

:tabindex="2"

autocomplete="current-
password"
                        v-
model="form.password"

placeholder="Password"
                    />

<InputError
:message="form.errors.password
" />
                </div>

                <div
class="flex items-center
justify-between"
:tabindex="3">
                    <Label
for="remember" class="flex
items-center space-x-3">

<Checkbox id="remember" v-
model="form.remember"
:tabindex="4" />

<span>Remember me</span>
                    </Label>
                </div>

                <Button
type="submit" class="mt-4 w-
full" :tabindex="4"
:disabled="form.processing">

<LoaderCircle v-
if="form.processing" class="h-
4 w-4 animate-spin" />
                    Log in
                </Button>
                <div
class="text-muted-foreground
text-center text-sm">
                    Belum
punya akun?
                    <TextLink
:href="route('register')"
class="underline underline-
offset-4" :tabindex="6">Daftar
Sekarang</TextLink>
                </div>
            </div>
        </form>
    </AuthBase>
</template>

Lampiran-2 Home <script setup
lang="ts">
import { Card, CardContent }
from '@/components/ui/card';
import AppLayout from
'@/layouts/AppLayout.vue';
import { SharedData, type
BreadcrumbItem } from
'@/types';
import { Head, usePage } from
'@inertiajs/vue3';
import { StoreIcon } from
'lucide-vue-next';

const { data } = defineProps<{
data: { umkmCount: number }
}>();
const { auth } =
usePage<SharedData>().props;

const breadcrumbs:
BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
];
</script>

<template>
    <Head title="Dashboard" />

    <AppLayout
:breadcrumbs="breadcrumbs">
        <div class="flex h-
full flex-1 flex-col gap-4
rounded-xl p-4">
            <div v-
if="auth.roles?.includes('supe
r-admin')" class="grid auto-
```

```
rows-min gap-4 md:grid-cols-
3">
                <Card
class="rounded-xl bg-gray-100
p-6 shadow-md dark:bg-black
dark:text-white">

<CardContent class="flex
items-center justify-between">
                        <div>
                            <p
class="mb-1 text-sm text-gray-
400">UMKM</p>

<h2 class="text-3xl font-
bold">{{ data.umkmCount
}}</h2>

<!-- <p class="mt-1 text-sm
text-green-400">+180.1% from
last month</p> -->
                        </div>

<StoreIcon class="h-6 w-6
text-gray-400" />

</CardContent>
                </Card>
            </div>
            <div v-else>
                <h2
class="text-xl font-bold
tracking-wide md:text-
3xl">Selamat Datang Kembali di
Aplikasi Pengelolaan <br
/>Laporan Keuangan!</h2>

                <!-- cara
penggunanaan -->
                <div
class="mt-2 w-full max-w-3xl
text-sm tracking-wide text-
gray-500">
                    Aplikasi
ini digunakan untuk membantu
UMKM dalam mengelola laporan
keuangan mereka. Anda dapat
melakukan berbagai hal seperti
                    membuat
laporan keuangan. Aplikasi ini
juga dilengkapi dengan fitur
untuk mencetak laporan
keuangan dalam format PDF.
                    <span
class="text-red-500"
                        >Hanya
dengan membuat jurnal umum
data yang lain akan
tergenerate sesuai pedoman
pembuatan laporan
keuangan</span
                    >
                </div>
            </div>
        </div>
    </AppLayout>
</template>
```

Lampiran 3 – controller DashboardController.php

```
<?php

namespace
App\Http\Controllers;

use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;
use Inertia\Inertia;

class DashboardController
extends Controller
{
    /**
     * Handle the incoming
request.
     */
    public function
__invoke(Request $request)
    {
        // hitung jumlah umkm
        $user = Auth::user();
        $umkmCount = $user-
>hasRole('super-admin') ?
\App\Models\Umkm::count() : 0;

        return
Inertia::render('Dashboard', [
            'title' =>
'Dashboard',
            'description' =>
'Halaman untuk mengelola
dashboard',
            'data' =>
compact('umkmCount'),
        ]);
    }
}
```

JurnalUmumController.php

```
<?php

namespace
App\Http\Controllers;

use App\Models\Akun;
use App\Models\JurnalUmum;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;
use
Illuminate\Support\Facades\Cac
he;
use Inertia\Inertia;

class JurnalUmumController
extends Controller
{
    protected $month;
    public function
__construct()
    {
        $this->month =
request('month') ?? date('Y-
m');

        Inertia::share([
            'title' =>
'Jurnal Umum',
            'month' => $this-
>month,
        ]);
    }


    public function index()
    {
        $umkm_id =
Auth::user()->umkm_id;
        $data =
JurnalUmum::with('akun')-
>orderBy('tanggal')-
>orderBy('jurnal_id')-
>orderBy('debit', 'desc')
            ->where('umkm_id',
$umkm_id)
            ->where('tanggal',
'like', $this->month . '%')
            ->orderBy('id')-
>get();

        return
Inertia::render('JurnalUmum/In
dex', ['data' => $data]);
    }

    public function create()
    {

        return
Inertia::render(

'JurnalUmum/Create',
            [
                'akuns' =>
Akun::orderBy('kode_akun')-
>get()
            ]
        );
    }

    public function
store(Request $request)
    {


        $umkm_id =
Auth::user()->umkm_id;


        foreach ($request-
>jurnal as $data) {
            $kredit =
$data['kredit'] ?? [];
            $debit =
$data['debit'] ?? [];

            $JurnalUmum =
JurnalUmum::create([
                'kode_akun' =>
$debit['kode_akun'],
                'tanggal' =>
$data['tanggal'],
                'ref' =>
$debit['ref'] ?? null,
                'debit' =>
$debit['total'] ?? 0,
                'kredit' => 0,
                'umkm_id' =>
$umkm_id
            ]);

            $JurnalUmum-
>update([
                'jurnal_id' =>
$JurnalUmum->id,
            ]);
```

```
JurnalUmum::create([
                'tanggal' =>
$JurnalUmum->tanggal,
                'umkm_id' =>
$JurnalUmum->umkm_id,
                'kode_akun' =>
$kredit['kode_akun'],
                'debit' => 0,
                'kredit' =>
$kredit['total'] ?? 0,
                'jurnal_id' =>
$JurnalUmum->id,
            ]);
        }



        if ($request->action
=== 'save_and_create_another')
{
            return redirect()-
>back()->with($this-
>flash('Data berhasil
disimpan. Silakan tambah data
baru.'));
        }

        return redirect()-
>route('jurnal.index', [
            'month' => $this-
>month
        ])->with($this-
>flash('Data berhasil
disimpan.'));
    }

    public function edit($id)
    {

        $debit =
JurnalUmum::findOrFail($id);
        $kredit = $debit-
>jurnalChildren;
        $akuns =
Akun::orderBy('kode_akun')-
>get();

        return
Inertia::render('JurnalUmum/Ed
it', compact('debit',
'kredit', 'akuns'));
    }

    public function
update(Request $request, $id)
    {

        $jurnal = (object)
$request->jurnal;


        $model =
JurnalUmum::findOrFail($id);
        $model->update([
            'tanggal' =>
$jurnal->tanggal,
            'kode_akun' =>
$jurnal->debit['kode_akun'],
            'ref' => $jurnal-
>debit['ref'] ?? null,
            'debit' =>
$jurnal->debit['total'] ?? 0,
            'kredit' =>  0,
        ]);


JurnalUmum::where('jurnal_id',
$model->id)->whereNot('id',
$model->id)->update([
            'tanggal' =>
$jurnal->tanggal,
            'kode_akun' =>
$jurnal->kredit['kode_akun'],
            'debit' => 0,
            'kredit' =>
$jurnal->kredit['total'] ?? 0,
        ]);

        return redirect()-
>route('jurnal.index', [
            'month' => $this-
>month
        ])->with($this-
>flash('Data berhasil
diperbarui.'));
    }

    public function
destroy(Request $request, $id)
    {

JurnalUmum::where('id', $id)-
>orWhere('jurnal_id', $id)-
>delete();
        return redirect()-
>route('jurnal.index',
['month' => $request->month])-
>with($this->flash('Data
berhasil dihapus.'));
```

```
    }
}

/** end
JurnalUmumController.php */


/** start
JurnalPenyesuaianController */

<?php

namespace
App\Http\Controllers;

use App\Models\Akun;
use
App\Models\JurnalPenyesuaian;
use Illuminate\Http\Request;
use
Illuminate\Support\Facades\Aut
h;
use
Illuminate\Support\Facades\Cac
he;
use Inertia\Inertia;

class
JurnalPenyesuaianController
extends Controller
{

    protected $month;
    public function
__construct()
    {
        $this->month =
request('month') ?? date('Y-
m');
        Inertia::share([
            'title' =>
'Jurnal Penyesuaian',
            'month' => $this-
>month,
        ]);
    }


    public function index()
    {
        $umkm_id =
Auth::user()->umkm_id;
        $data =
JurnalPenyesuaian::with('akun'
)
            ->where('umkm_id',
$umkm_id)
            ->where('tanggal',
'like', $this->month . '%')
            -
>orderBy('tanggal')
            -
>orderBy('jurnal_id')
            ->orderBy('debit',
'desc')
            ->orderBy('id')-
>get();
        return
Inertia::render('JurnalPenyesu
aian/Index', ['data' =>
$data]);
    }

    public function create()
    {

        return
Inertia::render(

'JurnalPenyesuaian/Create',
            ['akuns' =>
Akun::orderBy('kode_akun')-
>get()]
        );
    }

    public function
store(Request $request)
    {


        $umkm_id =
Auth::user()->umkm_id;



        foreach ($request-
>jurnal as $data) {
            $kredit =
$data['kredit'] ?? [];
            $debit =
$data['debit'] ?? [];

            $JurnalPenyesuaian
= JurnalPenyesuaian::create([
                'kode_akun' =>
$debit['kode_akun'],
                'tanggal' =>
$data['tanggal'],
                'ref' =>
$debit['ref'] ?? null,
```

```
                'debit' =>
$debit['total'] ?? 0,
                'kredit' => 0,
                'umkm_id' =>
$umkm_id
            ]);


$JurnalPenyesuaian->update([
                'jurnal_id' =>
$JurnalPenyesuaian->id,
            ]);


JurnalPenyesuaian::create([
                'tanggal' =>
$JurnalPenyesuaian->tanggal,
                'umkm_id' =>
$JurnalPenyesuaian->umkm_id,
                'kode_akun' =>
$kredit['kode_akun'],
                'debit' => 0,
                'kredit' =>
$kredit['total'] ?? 0,
                'jurnal_id' =>
$JurnalPenyesuaian->id,
            ]);
        }



        if ($request->action
=== 'save_and_create_another')
{
            return redirect()-
>back()->with($this-
>flash('Data berhasil
disimpan. Silakan tambah data
baru.'));
        }

        return redirect()-
>route('jurnal-
penyesuaian.index', [
            'month' => $this-
>month
        ])->with($this-
>flash('Data berhasil
disimpan.'));
    }

    public function edit($id)
    {
        $debit =
JurnalPenyesuaian::findOrFail(
$id);
        $kredit = $debit-
>jurnalChildren;
        $akuns =
Akun::orderBy('kode_akun')-
>get();
        return
Inertia::render('JurnalPenyesu
aian/Edit', compact('debit',
'kredit', 'akuns'));
    }

    public function
update(Request $request, $id)
    {

        $jurnal = (object)
$request->jurnal;



        $model =
JurnalPenyesuaian::findOrFail(
$id);
        $model->update([
            'tanggal' =>
$jurnal->tanggal,
            'kode_akun' =>
$jurnal->debit['kode_akun'],
            'ref' => $jurnal-
>debit['ref'] ?? null,
            'debit' =>
$jurnal->debit['total'] ?? 0,
            'kredit' =>  0,
        ]);


JurnalPenyesuaian::where('jurn
al_id', $model->id)-
>whereNot('id', $model->id)-
>update([
            'tanggal' =>
$jurnal->tanggal,
            'kode_akun' =>
$jurnal->kredit['kode_akun'],
            'debit' => 0,
            'kredit' =>
$jurnal->kredit['total'] ?? 0,
        ]);

        return redirect()-
>route('jurnal-
penyesuaian.index', [
            'month' => $this-
>month
        ])->with($this-
>flash('Data di perbarui.'));
    }
```

```
    public function
destroy($id)
    {

JurnalPenyesuaian::where('id',
$id)->orWhere('jurnal_id',
$id)->delete();
        return redirect()-
>route('jurnal-
penyesuaian.index', [
            'month' => $this-
>month
        ])->with($this-
>flash('Data berhasil
dihapus.'));
    }
}


/** end
JurnalPenyesuaianController.ph
p */

Lampiran 4 – Kelola Pengguna
UMKM
<template>
    <AppLayout
:breadcrumbs="breadcrumbs">
        <Head title="Data
Pengguna UMKM" />
        <div class="w-full p-
4">
            <div class="mb-4
flex items-center justify-
between">
                <h1
class="text-2xl font-
bold">Data Pengguna UMKM</h1>
                <Link
:href="route('users.create')">
                    <Button>+
Tambah Data</Button>
                </Link>
            </div>

            <Table v-
if="data.length > 0">
                <TableHeader>
                    <TableRow>

<TableHead>Nama
Pengguna</TableHead>

<TableHead>Nama
UMKM</TableHead>

<TableHead>Email</TableHead>

<TableHead>Status</TableHead>

<TableHead>Aksi</TableHead>

</TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow
v-for="item in data"
:key="item.id">

<TableCell>{{ item.name || '-'
}}</TableCell>

<TableCell>{{ item.umkm.name
}}</TableCell>

<TableCell>{{ item.email
}}</TableCell>

<TableCell>

<div class="flex items-center
space-x-2">

<Label>

<Switch

@update:model-value="(status)
=> handleChange(status,
item.id)"

:default-
value="item.umkm.status ==
'active'"

id="airplane-mode"

:disabled="isDisabled"

/>

<span :id="String(item.id)"
class="text-sm font-medium
text-gray-900">

{{ item.umkm.status ==
'active' ? 'Aktif' : 'Non-
Aktif' }}

</span>
```

```
</Label>

</div>

</TableCell>

<TableCell>

<div class="flex items-center
gap-2">

<Link
:href="route('users.edit',
item.id)">

<Button size="icon"><Pen
/></Button>

</Link>

<Button @click="() =>
deleteItem(item.id)"
variant="destructive"
size="icon"><Trash /></Button>

</div>

</TableCell>

</TableRow>
                </TableBody>
            </Table>
            <div v-else
class="mt-20 text-center">
                <p
class="text-gray-500">Tidak
ada data pengguna.</p>
            </div>
        </div>
    </AppLayout>
</template>

<script setup lang="ts">
import { Button } from
'@/components/ui/button';
import { Label } from
'@/components/ui/label';
import { Switch } from
'@/components/ui/switch';
import { Table, TableBody,
TableCell, TableHead,
TableHeader, TableRow } from
'@/components/ui/table';
import AppLayout from
'@/layouts/AppLayout.vue';
import { BreadcrumbItem } from
'@/types';
import { Head, Link, router }
from '@inertiajs/vue3';
import { Pen, Trash } from
'lucide-vue-next';
import { ref } from 'vue';
import { User } from
'./types';

const { data } = defineProps<{
data: User[] }>();

const breadcrumbs:
BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
        title: 'Kelola
Pengguna UMKM',
        href:
route('users.index'),
    },
];
const isDisabled = ref(false);

const handleChange = (status:
boolean, id: number) => {
    const user =
data.find((user) => user.id
=== id) as User;
    if (user) {
        isDisabled.value =
true;
        const label =
document.getElementById(String
(id)) as HTMLAnchorElement;
        label.innerHTML =
'<span class="animate-
pulse">Loading...</span>';
        router.put(

route('users.update.status',
user.umkm.id),
            {
                status:
!status ? 'inactive' :
'active',
            },
            {
                onFinish: ()
=> {

isDisabled.value = false;
```

```
label.innerHTML = status ?
'Aktif' : 'Non-Aktif';
                },

preserveScroll: true,
                preserveState:
true,
            },
        );
    }
};

const deleteItem = (id:
number) => {
    if (confirm('Apakah Anda
yakin ingin menghapus pengguna
ini?')) {

router.delete(route('users.des
troy', id), {
            onSuccess: () => {
                // Optionally,
you can show a success message
or refresh the page
            },
            onError: (error)
=> {

console.error('Error deleting
user:', error);
            },
        });
    }
};
</script>

Lampiran 5 – Jurnal Umum
// □ File:
resources/js/Pages/JurnalUmum/
Index.vue
<template>
    <AppLayout
:breadcrumbs="breadcrumbs">
        <div class="p-4">
            <div class="mb-4
flex items-center justify-
between">
                <h1
class="text-2xl font-bold">
                    Jurnal
Umum <span class="hidden
print:inline">- {{
formatMonth(month as string)
}}</span>
                </h1>
                <div
class="flex items-center gap-
2">

<PrintButton />
                    <Link

:href="

route('jurnal.create', {

month: month,
                                })
                            "

class="print:hidden"
                    >

<Button>+ Tambah Jurnal
Umum</Button>
                    </Link>
                </div>
            </div>

            <div class="mt-4
flex items-center gap-2
print:hidden">
                <Input

type="month"

placeholder="Cari berdasarkan
keterangan, ref, debit, atau
kredit"
                    class="mb-
4 w-max"
                    v-
model="month"

@input="filterMonth"
                />
            </div>

            <Table v-
if="items.length > 0"
class="w-full">
                <TableHeader>
                    <TableRow>

<TableHead>Tanggal</TableHead>

<TableHead>Keterangan</TableHe
ad>

<TableHead>Ref</TableHead>
```

```
<TableHead class="text-
right">Debit</TableHead>

<TableHead class="text-
right">Kredit</TableHead>

<TableHead class="text-center
print:hidden">Aksi</TableHead>

</TableRow>
                </TableHeader>
                <TableBody>
                    <TableRow
v-for="(item, index) in items"
:key="item.id">

<TableCell rowspan="2" v-
if="index % 2 == 0">{{
item.formatted_date
}}</TableCell>

<TableCell :class="{ 'text-
center': index % 2 == 1 }">{{
item.keterangan ??
item.akun?.nama_akun ?? '-'
}}</TableCell>

<TableCell>{{ item.kode_akun
|| '-' }}</TableCell>

<TableCell class="text-
right">{{
formatRupiah(item.debit)
}}</TableCell>

<TableCell class="text-
right">{{
formatRupiah(item.kredit)
}}</TableCell>

<TableCell rowspan="2" v-
if="index % 2 == 0"
class="text-center
print:hidden">

<div class="flex justify-
center gap-2">

<Link
:href="route('jurnal.edit',
item.jurnal_id ?? item.id) +
`?month=${month}`">

<Button size="icon"><Pen
/></Button>
</Link>

<Button
@click="hapus(item.jurnal_id
?? item.id)"
variant="destructive"
size="icon"><Trash /></Button>

</div>

</TableCell>

</TableRow>
                    <TableRow>

<TableCell></TableCell>

<TableCell class="text-center
font-bold">Total</TableCell>

<TableCell></TableCell>

<TableCell class="text-right
font-bold">{{
formatRupiah(sumJurnalUmum('de
bit')) }}</TableCell>

<TableCell class="text-right
font-bold">{{
formatRupiah(sumJurnalUmum('kr
edit')) }}</TableCell>

<TableCell
class="print:hidden"></TableCe
ll>

</TableRow>
                </TableBody>
            </Table>
            <Alert v-else
variant="destructive">
                <InfoIcon
class="h-4 w-4" />

<AlertTitle>Tidak ada data
Jurnal Umum.</AlertTitle>

<AlertDescription> Silakan
tambahkan transaksi jurnal
umum baru untuk bulan {{
formatMonth(month as string)
}}. </AlertDescription>
            </Alert>
        </div>
    </AppLayout>
```

```
</template>

<script setup lang="ts">
import { Alert,
AlertDescription, AlertTitle }
from '@/components/ui/alert';
import { Button } from
'@/components/ui/button';
import PrintButton from
'@/components/ui/button/PrintB
utton.vue';
import { Input } from
'@/components/ui/input';
import { Table, TableBody,
TableCell, TableHead,
TableHeader, TableRow } from
'@/components/ui/table';
import AppLayout from
'@/layouts/AppLayout.vue';
import { formatMonth } from
'@/lib/utils';
import { BreadcrumbItem } from
'@/types';
import { Link, router } from
'@inertiajs/vue3';
import { InfoIcon, Pen, Trash
} from 'lucide-vue-next';
import { onMounted, ref } from
'vue';
import { JurnalUmum } from
'./types';

const props = defineProps<{
data: any[]; month: string
}>();

const formatRupiah = (val:
number | string): string => {
    const n =
parseInt(String(val).replace(/
[^\d]/g, '')) || 0;
    return 'Rp. ' +
n.toLocaleString('id-ID');
};

const hapus = (id: string) =>
{
    if (confirm('Yakin ingin
menghapus data ini?')) {

router.delete(route('jurnal.de
stroy', id), {
            data: {
                month:
month.value,
            },
            preserveScroll:
true,
            preserveState:
false,
        });
    }
};

const breadcrumbs:
BreadcrumbItem[] = [
    {
        title: 'Dashboard',
        href: '/dashboard',
    },
    {
        title: 'Jurnal Umum',
        href: '/jurnal-umum',
    },
];

const month =
ref<string>(props.month);

const filterMonth = () => {
    // update path url
    router.get(
        route('jurnal.index'),
        { month: month.value
},
        {
            preserveScroll:
true,
        },
    );
};
const items =
ref<JurnalUmum[]>([]);

const sumJurnalUmum = (column:
'kredit' | 'debit') => {
    return
items.value.reduce((acc, item)
=> acc + parseInt(item[column]
?? 0), 0);
};

onMounted(() => {
    items.value = props.data;
});
</script>
```