

LAMPIRAN

Lampiran 1. Kode Program atau *Source Code*

Berikut merupakan *source code* tampilan halaman login

```
import { Head, useForm } from '@inertiajs/react';
import { LoaderCircle } from 'lucide-react';
import { FormEventHandler } from 'react';
import InputError from '@components/input-error';
import TextLink from '@components/text-link';
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 AuthLayout from '@layouts/auth-layout';

type LoginForm = {
  email: string;
  password: string;
  remember: boolean;
};

interface LoginProps {
  status?: string;
  canResetPassword: boolean;
}

export default function Login({ status, canResetPassword }:
LoginProps) {
  const { data, setData, post, processing, errors, reset
} = useForm<Required<LoginForm>>(<{
  email: '',
```

```

        password: '',
        remember: false,
    });

    const submit: FormEventHandler = (e) => {
        e.preventDefault();
        post(route('login'), {
            onFinish: () => reset('password'),
        });
    };

    return (
        <AuthLayout title="Log in to your account"
        description="Enter your email and password below to log in">
            <Head title="Log in" />

            <form className="flex flex-col gap-6"
            onSubmit={submit}>
                <div className="grid gap-6">
                    <div className="grid gap-2">
                        <Label htmlFor="email">Email
address</Label>
                        <Input
                            id="email"
                            type="email"
                            required
                            autoFocus
                            tabIndex={1}
                            autoComplete="email"
                            value={data.email}

```



```

                                onChange={(e) =>
setData('password', e.target.value)}
                                placeholder="Password"
                                />
                                <InputError
message={errors.password} />
                                </div>

                                <div className="flex items-center
space-x-3">
                                <Checkbox
                                id="remember"
                                name="remember"
                                checked={data.remember}
                                onClick={() =>
setData('remember', !data.remember)}
                                tabIndex={3}
                                />
                                <Label htmlFor="remember">Remember
me</Label>
                                </div>

                                <Button type="submit" className="mt-4
w-full" tabIndex={4} disabled={processing}>
                                {processing} && <LoaderCircle
className="h-4 w-4 animate-spin" />
                                Log in
                                </Button>
                                </div>
                                </form>

```

```

        {status && <div className="mb-4 text-center
text-sm font-medium text-green-600">{status}</div>}
      </AuthLayout>
    );
  }
}

```

Berikut merupakan *source code* tampilan halaman dashboard admin dan user

```

import Header from '@components/Header';
import InfoSection from '@components/InfoSection';
import SensorDataCard from '@components/SensorDataCard';
import StatusPanel from '@components/StatusPanel';
import useRealtimeSensor from '@hooks/use-realtime-
subscribe';
import AppLayout from '@layouts/app-layout';
import { formatFullDateTime } from '@lib/utils';
import { SharedData, type BreadcrumbItem } from '@types';
import { Head, usePage } from '@inertiajs/react';
import { Droplets, Gauge, FlaskRound as Flask, Waves } from
' lucide-react';
import { useState } from 'react';

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

```

```

export default function Dashboard({ devices }: { devices:
string[] }) {
  const { auth } = usePage<SharedData>().props;
  const isAdmin = auth.user.role === 'admin';
  const [sensorValue, setSensorValue] =
useState<Record<string, any>>({});
  const [status, setStatus] = useState<string>('');
  useRealtimeSensor(devices,
    (deviceId, data) => {
      setSensorValue(prev => ({
        ...prev,
        [deviceId]: {
          data,
        }
      })))
    },
    (status) => {
      setStatus(status)
    }
  )
  // Function untuk menentukan style/status pH
  const getPhStatus = () => {
    const ph = sensorValue[devices[0]]?.data.ph;

    if (ph === undefined) return 'good';

    if (ph < 6 || ph > 8.5) {
      return 'alert';
    } else if (ph >= 6.5 && ph <= 7.5) {
      return 'good';
    }
  }
}

```

```

    } else {
        return 'warning';
    }
};

// Function untuk menentukan style/status TDS
const getTdsStatus = () => {
    const tds = sensorValue[devices[0]]?.data.tds;

    if (tds === undefined) return 'good';

    if (tds > 800) {
        return 'alert';
    } else if (tds >= 300 && tds <= 800) {
        return 'warning';
    } else {
        return 'good';
    }
};

// Function untuk menentukan style/status Turbidity
const getTurbidityStatus = () => {
    const turbidity =
sensorValue[devices[0]]?.data.turbidity;

    if (turbidity === undefined) return 'good';

    if (turbidity > 30) {
        return 'alert';
    } else if (turbidity >= 20 && turbidity <= 30) {

```

```

        return 'warning';
    } else {
        return 'good';
    }
};

const monitoringData = [
    {
        id: 1,
        value: sensorValue[devices[0]]?.data.ph || 0,
// Ambil pH dari data sensor
        unit: 'pH',
        location: 'Distribution Line',
        status: getPhStatus(),
        timestamp: formatFullDateTime(Date.now()),
        icon: Flask,
        description: 'Acidity Level'
    },
    {
        id: 2,
        value: sensorValue[devices[0]]?.data.tds || 0,
// Ambil TDS dari data sensor
        unit: 'ppm',
        location: 'Distribution Line',
        status: getTdsStatus(),
        timestamp: formatFullDateTime(Date.now()),
        icon: Droplets,
        description: 'Total Dissolved Solids'
    },
    {

```

```

        id: 3,
        value: sensorValue[devices[0]]?.data.turbidity
        || 0, // Ambil turbidity dari data sensor
        unit: 'NTU',
        location: 'Distribution Line',
        status: getTurbidityStatus(),
        timestamp: formatFullDateTime(Date.now()),
        icon: Waves,
        description: 'Turbidity Level'
    },
    {
        id: 4,
        value:
sensorValue[devices[1]]?.data.flow_m3_second || 0,
        unit: 'm³/h',
        location: 'Home Line',
        status: 'moderate',
        timestamp: formatFullDateTime(Date.now()),
        icon: Gauge,
        description: 'Flow Rate'
    }
];

const filteredData = isAdmin ?
monitoringData.filter(sensor => sensor.id !== 4) :
monitoringData;

return (
    <AppLayout breadcrumbs={breadcrumbs}>
        <Head title="Dashboard" />

```

```

        <div className="space-y-8 mb-8 p-4">
            <Header />
            <section>
                <h2 className="text-xl font-semibold
dark:text-gray-100 text-slate-700 mb-4">Water
Monitoring</h2>
                <div className="grid grid-cols-1
md:grid-cols-2 lg:grid-cols-4 gap-4">
                    {filteredData.map((data) => (
                        <SensorDataCard key={data.id}
data={data} />
                    )))}
                </div>
            </section>
            <section>
                <h2 className="text-xl font-semibold
dark:text-gray-100 text-slate-700 mb-4">Water Condition
Status</h2>
                <StatusPanel status={status || 'AMAN'}
timestamp={formatFullDateTime(Date.now())} />
            </section>
            <InfoSection />
        </div>
    </AppLayout>
);
}

```

Berikut merupakan *source code* halaman tagihan admin

```
import AppLayout from '@layouts/app-layout';
```

```

import { Filters, Pagination, Tagihan as Fine, type
BreadcrumbItem } from '@types';
import { Head } from '@inertiajs/react';
import { Separator } from '@components/ui/separator';
import Heading from '@components/heading';
import DataTable from './create/data-table';
import { Button } from '@components/ui/button';
import      {      DatePickerWithPresets      }      from
'@components/ui/date-picker';
import { useState } from 'react';
import { format } from 'date-fns';

const breadcrumbs: BreadcrumbItem[] = [
  {
    title: 'Daftar Tagihan',
    href: '/admin/warga',
  },
];

type Props = {
  tagihans: {
    data: Fine[];
  };
  filter_by_months: string[];
  filters: Filters;
  pagination: Pagination;
}

```

```

export default function Tagihan({ tagihans,
filter_by_months, filters, pagination }: Props) {
  const [selectedDate, setSelectedDate] =
useState<Date>();
  const printReport = () => {
    const periode = selectedDate ? format(selectedDate,
"yyyy-MM") : ""

window.open(`/admin/tagihan/report?periode=${periode}`);
  };
  return (
    <AppLayout breadcrumbs={breadcrumbs}>
      <Head title="Tagihan" />
      <div className="px-4 py-6">
        <Heading title="Daftar Tagihan"
description="Halaman untuk menambahkan daftar warga sistem"
/>
        <div className='flex gap-2'>
          <DatePickerWithPresets
            months={filter_by_months}
            value={selectedDate}
            onChange={setSelectedDate}
          />
          <Button
            onClick={printReport}
          >Cetak Laporan</Button>
        </div>
        <Separator className="my-4" />

        <DataTable

```

```

tagihans={tagihans}
pagination={pagination}
total={pagination.total}
filters={filters}

/>
</div>
</AppLayout>
);
}

```

Berikut merupakan *source code* halaman tagihan user

```

import AppLayout from '@layouts/app-layout';
import { SharedData, Tagihan, type BreadcrumbItem } from
'@types';
import { Head, useForm, usePage } from '@inertiajs/react';
import { Separator } from '@components/ui/separator';
import Heading from '@components/heading';
import AppLogo from '@components/app-logo';
import DataTable from './data-table';
import { useRef, useState } from 'react';
import { ImageUp, LoaderCircle } from 'lucide-react';
import { Label } from '@components/ui/label';
import { Button } from '@components/ui/button';
import { toast } from 'sonner';

const breadcrumbs: BreadcrumbItem[] = [
  {
    title: 'Daftar Tagihan',
    href: '/admin/warga',
  },

```

```

];

type TagihanProps = {
  tagihan: Tagihan;
}

export default function TagihanUser({ tagihan }:
TagihanProps) {
  const { auth } = usePage<SharedData>().props;
  const user = auth.user;

  const { setData, post, processing, reset, errors } =
useForm({
    bukti_pembayaran: null as File | null, // Explicitly
type the field
  });
  const fileInputRef = useRef<HTMLInputElement>(null);

  const [previewUrl, setPreviewUrl] = useState<string |
null>(null);

  const handleImageChange = (e:
React.ChangeEvent<HTMLInputElement>) => {
    const file = e.target.files?.[0];
    if (file) {
      setData('bukti_pembayaran', file);

      const reader = new FileReader();

```

```
        reader.onloadend = () => {
            setPreviewUrl(reader.result as string);
        };
        reader.readAsDataURL(file);
    }
};

const handlePreview = () => {
    fileInputRef.current?.click();
};

const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    if (!tagihan?.tagihan_id) {
        return toast.error("Tagihan tidak ditemukan");
    }

    post(route('proof.payment', tagihan.tagihan_id), {
        forceFormData: true,
        preserveScroll: true,
        preserveState: false,
        onSuccess: () => {
            reset();
            setPreviewUrl(null);
            toast.success("Bukti Pembayaran Berhasil
diupload");
        },
        onError: (error) => {
            toast.error("Gagal Mengupload Bukti
Pembayaran");
        }
    });
};
```

```

        console.log(error);
    }
    });
};
return (
    <AppLayout breadcrumbs={breadcrumbs}>
        <Head title="Tagihan" />
        <div className="px-4 py-6 flex flex-col gap-4">
            <div className='text-slate-600 dark:text-slate-200 bg-blue-50 dark:bg-accent border border-blue-100 dark:border-border shadow-sm rounded-md p-6'>
                <div className='flex justify-between'>
                    <div className='flex-1'>
                        <Heading title="Tagihan"
                            className='text-blue-500 dark:text-slate-200' />
                        <p className="mt-4 font-medium tracking-tight text-sm -mt-6">Status :
                            {tagihan && tagihan.status
                                === "belum_lunas" ? (
                                    <span className='px-2 py-1 rounded-md bg-red-100 text-red-700'>Belum Lunas</span>
                                ) : tagihan ? (
                                    <span className='px-2 py-1 rounded-md bg-green-100 text-green-700'>Lunas</span>
                                ) : (
                                    <span className='px-2 py-1 rounded-md bg-gray-700 text-gray-100 dark:bg-gray-100 dark:text-gray-700'>Belum ditetapkan</span>
                                )}
                        </p>
                    </div>
                </div>
            </div>
        </div>
    </AppLayout>
);

```

```

        </div>
        <div>
            <AppLogo />
        </div>
    </div>
    <div className='flex justify-between
space-y-2 my-6'>
        <div className='flex-1'>
            <p className="font-semibold
tracking-tight text-sm">Ditagihkan Kepada : </p>
            <p className="tracking-tight
text-sm">Nama : {user.name}</p>
            <p className="tracking-tight
text-sm"> Email : {user.email}</p>
            <p className="tracking-tight
text-sm">
                No          Telepon          :
                {user?.warga?.no_telp || '-'}
            </p>
            <p className="tracking-tight
text-sm break-all">
                Alamat          :
                {user?.warga?.alamat || '-'}
            </p>
        </div>
        <div className='flex-2'>
            <p className="font-semibold
tracking-tight text-sm text-right">Periode</p>
            <p className="font-medium
tracking-tight text-sm text-right">

```

```

{tagihan ? (
  `${tagihan.tanggal_mulai} - ${tagihan.tanggal_akhir}`
  ) : (
    <span className='px-2
py-1 rounded-md bg-gray-700 text-gray-100 dark:bg-gray-100
dark:text-gray-700'>Belum ditetapkan</span>
  )}
</p>
</div>
</div>
<div className='-mt-6'>
  <p className="font-semibold
tracking-tight text-sm">Metode Pembayaran</p>
  <p className="tracking-tight text-
sm">Transfer Manual <span className='font-
semibold'>Mandiri</span></p>
  <p className="tracking-tight text-
sm">No. Rekening : <span className='font-
semibold'>123456789</span></p>
  <p className="tracking-tight text-
sm">Atas Nama : <span className='font-semibold'>Maulana
Feri Setyawan</span></p>
</div>
</div>
<div className='text-slate-600 dark:text-
slate-200 bg-blue-50 dark:bg-accent border border-blue-100
dark:border-border shadow-sm rounded-md p-6'>

```

```

<h3 className="text-xl font-semibold
tracking-tight text-blue-500 dark:text-slate-
200">Detail</h3>

<Separator className="my-2" />
<DataTable
    tagihan={tagihan}
/>
<div className='mb-4'>
    <p className="font-medium
tracking-tight text-sm mt-6 text-right">Total Bayar</p>
    <h2 className="text-xl font-
semibold tracking-tight text-right">
        `${tagihan?.total_bayar ??
"Rp 0,00"}`
    </h2>
</div>
<form onSubmit={handleSubmit}
className="flex gap-4 flex-col">
    <Label htmlFor="picture"
className='font-semibold'>Upload Bukti Pembayaran</Label>

    {/* Gambar Preview */}
    <div
        className="flex items-center
justify-center w-40 h-40 object-cover rounded mt-2 bg-blue-
100 dark:bg-blue-950 border border-blue-950 dark:border-
blue-100 cursor-pointer overflow-hidden"
        onClick={handlePreview}
    >
        {previewUrl ? (

```

```

<img
    src={previewUrl}
    alt="Preview"
    className="w-full h-
full object-cover rounded"
/>
) : tagihan?.bukti_pembayaran ?
(
    <img
src={` /storage/${tagihan.bukti_pembayaran}`}
    alt="Bukti Pembayaran"
    className="w-full h-
full object-cover"
/>
    ) : (
        <ImageUp size={50}
className="text-blue-900 dark:text-blue-100" />
    )}
</div>

{/* File Input Hidden */}
<input
    id="bukti_pembayaran"
    type="file"
    className="hidden"
    ref={fileInputRef}
    onChange={handleImageChange}
    accept="image/jpg, image/jpeg,
image/png"

```

```

        />
        <Button                                type="submit"
className="w-40">
                                {processing && <LoaderCircle
className="h-4 w-4 animate-spin mr-2" />}
                                Submit
        </Button>

                                {errors.bukti_pembayaran && (
                                <div    className="text-red-500
text-sm">{errors.bukti_pembayaran}</div>
                                )}

        </form>
    </div>
</div>
</AppLayout>
);
}

```

Berikut merupakan *source code* tampilan halaman fuzzy mamdani

```

import AppLayout from '@layouts/app-layout';
import { Filters, Pagination, Sensor, type BreadcrumbItem }
from '@types';
import { Head } from '@inertiajs/react';
import { Separator } from '@components/ui/separator';
import Heading from '@components/heading';
import DataTable from './create/data-table';

const breadcrumbs: BreadcrumbItem[] = [

```

```

    {
      title: 'Perhitungan Fuzzy Mamdani',
      href: '/admin/fuzzy',
    },
  ];

type Props = {
  sensorDatas: {
    data: Sensor[];
  };
  filters: Filters;
  pagination: Pagination;
}

export default function Perangkat({ sensorDatas, filters,
pagination }: Props) {
  return (
    <AppLayout breadcrumbs={breadcrumbs}>
      <Head title="Sensor" />
      <div className="px-4 py-6">
        <Heading title="Data Fuzzy"
description="Halaman untuk melihat data dari sensor yang
tersimpan didatabase" />

        <Separator className="my-4" />

        <DataTable
          sensors={sensorDatas}
          pagination={pagination}

```

```

        total={pagination.total}
        filters={filters}

    />

</div>
</AppLayout>

);
}

```

Berikut merupakan *source code* tampilan halaman detail fuzzy

```

<AppLayout breadcrumbs={breadcrumbs}>
    <Head title="Detail Perhitungan" />
    <div className='px-4 py-6 gap-2 flex flex-col
lg:flex-row space-y-8 lg:space-y-0 lg:space-x-12'>
        <div className='flex-2 shadow-md p-6
rounded-xl'>
            <div className='flex-1 mb-2'>
                <Heading className=''
title={`Detail Perhitungan Data ID ke ${sensorData.id}`}
description="Halaman untuk melihat detail perhitungan fuzzy
mamdani menggunakan metode center of gravity" />
                <div className="grid grid-cols-1
md:grid-cols-3 gap-4">
                    {ListData.map((data) => (
                        <DataPanel key={data.id}
data={data} />
                    )))}
                </div>
            </div>
            <div className='flex-1 mb-2'>

```

```

<Heading title={`Fuzzifikasi`}
description="Menentukan Nilai Fuzzifikasi" />
<div className="grid grid-cols-1
md:grid-cols-3 gap-4">
  {FuzzifikasiData.map((data) =>
    (
      <PanelFuzzy key={data.id}
data={data} />
    ))}
</div>
</div>
<div className='flex-1 mb-2'>
  <Heading title={`Rules`}
description="Menentukan Nilai Rules Yang Digunakan" />
  <div className="overflow-auto
rounded-md">
    <div className="inline-block
w-full overflow-x-auto">
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead
className="text-center">
              <div
className="flex items-center justify-center gap-1 text-
center">
                Rules
              </div>
            </TableHead>
          </TableRow>
        </TableHeader>
      </Table>
    </div>
  </div>

```

```

className="text-center"
    >
        <div
className="flex items-center justify-center gap-1 text-
center">
            Nilai
Rules
        </div>
    </TableHead>
</TableRow>
</TableHeader>
<TableBody>
{fuzzyMamdani.rules.map((rule, index) => (
    <TableRow
className="text-center" key={rule + index}>
<TableCell>`Rules ${index + 1}`</TableCell>
<TableCell>{useTruncateNumber(rule)}</TableCell>
    </TableRow>
))
}
</TableBody>
</Table>
<div />
</div>
</div>
</div>

```

```

<div className='flex-1 mb-2'>
  <Heading title={`Agregasi Output`}
description="Menentukan Nilai Agregasi Output" />
  <div className="grid grid-cols-1
md:grid-cols-3 gap-4">
    {AgregasiData.map((data) => (
      <PanelFuzzy key={data.id}
data={data} />
    ))}
  </div>
</div>
<div className='flex-1 mb-2'>
  <Heading title={`Himpunan Baru`}
description="Menentukan Himpunan Baru" />
  <div className="grid grid-cols-1
md:grid-cols-3 gap-4">
    {HimpunanData.map((data) => (
      <PanelFuzzy key={data.id}
data={data} />
    ))}
  </div>
</div>
<div className='flex-1 mb-2'>
  <Heading title={`Deffuzifikasi`}
description="Menentukan Nilai Deffuzifikasi Menggunakan
Metode COG" />
  <div className="grid grid-cols-1
md:grid-cols-3 gap-4">
    {DeffuzifikasiData.map((data)
=> (

```

```

<PanelFuzzy key={data.id}
data={data} />
    )}}
  </div>
</div>
<div className='flex-1 mb-2'>
  <Heading title={`Nilai
Deffuzifikasi`} description="Menentukan Nilai Deffuzifikasi
Menggunakan Metode COG" />
  <div className="grid grid-cols-1
md:grid-cols-2 gap-4">
    {ValueFuzzyData.map((data) =>
  (
    <PanelFuzzy key={data.id}
data={data} />
    )}}
  </div>
</div>
</div>
</div>
</AppLayout>

```