

Lampiran 1. Surat Kesiediaan Membimbing TA Dosen Pembimbing I

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Arif Rakhman, SE, S.Pd, M.Kom
NIPY : 05.016.291
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Azmy Mutiara Zanzabila
NIM : 22040092
Program Studi : Diploma III Teknik Komputer

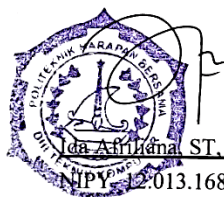
Judul TA : SISTEM MONITORING KUALITAS AIR DAN KONTROL POMPA AIR
UNTUK PENGURASAN DAN PENGISIAN KOLAM IKAN KOI BERBASIS
WEBSITE

Dengan ini menyatakan ini dibuat agar dilaksanakan sebagaimana mestinya.

Tegal, 17 Februari 2025

Mengetahui

Ka Prodi DIII Teknik Komputer,


da Afrihana, ST, M.Kom
NIPY. 12.013.168

Dosen Pembimbing I,


Arif Rakhman, SE, S.Pd, M.Kom
NIPY. 05.016.291

Lampiran 2. Surat Ketersediaan Membimbing TA Dosen Pembimbing II

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Arfan Haqiqi Sulasmoro, M.Kom
NIPY : 02.009.054
Jabatan Struktural : -
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Azmy Mutiara Zanzabila
NIM : 22040092
Program Studi : Diploma III Teknik Komputer

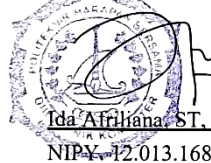
Judul TA : SISTEM MONITORING KUALITAS AIR DAN KONTROL POMPA
AIR UNTUK PENGURASAN DAN PENGISIAN KOLAM IKAN KOI
BERBASIS WEBSITE

Dengan ini menyatakan ini dibuat agar dilaksanakan sebagaimana mestinya.

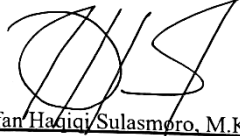
Tegal, 5 Mei 2025

Mengetahui

Ka Prodi DIII Teknik Komputer,


Ida Afriliana ST. M.Kom
NIPY. 12.013.168

Dosen Pembimbing II,


Arfan Haqiqi Sulasmoro, M.Kom
NIPY. 02.009.054

Lampiran 3. *Source Code Program*

```

<?= $this->extend('layout/app') ?>

<?= $this->section('content') ?>
<main class="content">
    <div class="container-fluid p-0">
        <h1 class="h3 mb-3"><strong>Monitoring</strong>
Dashboard</h1>

        <div class="row">
            <div class="col-12">
                <div class="row">
                    <div class="col-sm-4">
                        <div class="card">
                            <div class="card-
body">
                                <div
class="row">
                                    <div
class="col mt-0">
                                        <h4
class="card-title">Tingkat Kekeruhan</h4>
                                        </div>
                                        <div
class="col-auto">
                                            <div
class="stat text-danger">
                                                <i class="align-middle" data-feather="sun"></i>
                                                </div>
                                            </div>
                                        </div>
                                    </div>
                                    <?php if
(!empty($sensors)): ?>
                                        <?php
$latestSensor = end($sensors); ?>
                                        <h1
class="mt-1 mb-3"><?= $latestSensor['turbidity']; ?> NTUs</h1>
                                        <?php else: ?>
                                        <h1
class="mt-1 mb-3">0 NTUs</h1>
                                        <?php endif; ?>
                                    </div>
                                </div>
                            </div>
                        <div class="col-sm-4">
                            <div class="card">
                                <div class="card-
body">
                                    <div
class="row">
                                        <div
class="col mt-0">

```

```

class="card-title">pH Air</h4>
</div>
<div
class="col-auto">
<div
class="stat text-warning">
<i class="align-middle" data-feather="thermometer"></i>
</div>
</div>
</div>
<?php if
(!empty($sensors)): ?>
<?php
$latestSensor = end($sensors); ?>
<h1
class="mt-1 mb-3"><?= $latestSensor['pH']; ?> pH</h1>
<?php else: ?>
<h1
class="mt-1 mb-3">0 pH</h1>
<?php endif; ?>
</div>
</div>
<div class="col-sm-4">
<div class="card">
<div class="card-
body">
<div
class="row">
<div
class="col mt-0">
<h4
class="card-title">Partikel Terlarut</h4>
</div>
<div
class="col-auto">
<div
class="stat text-warning">
<i class="align-middle" data-feather="droplet"></i>
</div>
</div>
</div>
<?php if
(!empty($sensors)): ?>
<?php
$latestSensor = end($sensors); ?>
<h1
class="mt-1 mb-3"><?= $latestSensor['tds']; ?> ppm</h1>
<?php else: ?>

```



```

        // Group sensor data by date (YYYY-MM-DD)
        const dataByDate = {};
        let uniqueDates = [];
        let chart = null;

        // Create fixed hour labels from 00:00 to 23:00
        const hourLabels = Array.from({
            length: 24
        }, (_, i) => `${String(i).padStart(2, '0')}:00`);

        // Process and organize sensor data by date
        if (sensorData && sensorData.length > 0) {
            sensorData.forEach(function(reading) {
                const readingDate = new
Date(reading.created_at);
                const dateKey =
readingDate.toISOString().split('T')[0]; // YYYY-MM-DD format

                // Create date entry if it doesn't exist
                if (!dataByDate[dateKey]) {
                    dataByDate[dateKey] = {
                        turbidity:
Array(24).fill(null),
                        pH: Array(24).fill(null),
                        tds: Array(24).fill(null)
                    };
                    uniqueDates.push(dateKey);
                }

                // Get the hour and update data for that
hour
                const hour = readingDate.getHours();
                dataByDate[dateKey].turbidity[hour] =
Number(reading.turbidity);
                dataByDate[dateKey].pH[hour] =
Number(reading.pH);
                dataByDate[dateKey].tds[hour] =
Number(reading.tds);
            });

            // Sort dates chronologically
            uniqueDates.sort();
        }

        // Function to update the chart for a selected date
        function updateChart(selectedDate) {
            // Get data for the selected date
            const dateData = dataByDate[selectedDate] || {
                turbidity: Array(24).fill(null),
                pH: Array(24).fill(null),
                tds: Array(24).fill(null)
            };

            // Format date for display
            const displayDate = new

```

```

Date(selectedDate).toLocaleDateString('id-ID', {
    weekday: 'long',
    day: 'numeric',
    month: 'long',
    year: 'numeric'
});

// If chart already exists, destroy it before
creating a new one
if (chart) {
    chart.destroy();
}

// Create the chart
chart = new Chart(ctx, {
    type: "line",
    data: {
        labels: hourLabels,
        datasets: [{
            label: "Tingkat Kekeruhan
(NTUs)",
            fill: true,
            backgroundColor: gradient1,
            borderColor:
window.theme.danger,
            data: dateData.turbidity,
            spanGaps: true
        }, {
            label: "pH Air pada Kolam
Ikan Koi",
            fill: true,
            backgroundColor: gradient2,
            borderColor:
window.theme.warning,
            data: dateData.pH,
            spanGaps: true
        }, {
            label: "Partikel Terlarut
pada Kolam Ikan Koi (ppm)",
            fill: true,
            backgroundColor: gradient3,
            borderColor:
window.theme.success,
            data: dateData.tds,
            spanGaps: true
        }]
    },
    options: {
        responsive: true,
        maintainAspectRatio: false,
        legend: {
            display: true
        },
        tooltips: {
            mode: 'index',
            intersect: false,

```

```

callbacks: {
  title:
function(tooltipItem, data) {
  return `Tanggal:
  ${displayDate} - Jam: ${tooltipItem[0].xLabel}`;
  },
  label:
function(tooltipItem, data) {
  const
datasetLabel = data.datasets[tooltipItem.datasetIndex].label ||
'';
  const value =
tooltipItem.yLabel;
  return
datasetLabel + ': ' + value;
  }
  },
  hover: {
    mode: 'nearest',
    intersect: true
  },
  scales: {
    xAxes: [{
      display: true,
      gridLines: {
        color:
"transparent"
      },
      scaleLabel: {
        display: true,
        labelString:
'Waktu (Jam)'
      }
    }],
    yAxes: [{
      display: true,
      gridLines: {
        borderDash: [3,
3],
        color:
"transparent"
      },
      ticks: {
        beginAtZero:
true,
        // Dynamic scale
        based on data
        max: Math.max(
          Math.max(...dateData.turbidity.filter(v => v !==
null).length > 0 ? dateData.turbidity.filter(v => v !== null) :
[0]),
          Math.max(...dateData.pH.filter(v => v !== null).length > 0
? dateData.pH.filter(v => v !== null) : [0]),

```



```

        Math.max(...dateData.tds.filter(v => v !== null).length >
0 ? dateData.tds.filter(v => v !== null) : [0]),
                                100 //
Minimum max value
                                ) + 20
                                },
                                scaleLabel: {
                                    display: true,
                                    labelString:
'Nilai'
                                }
                                ]]
                                }
                                }
                                });
                                }

                                // Get current date in YYYY-MM-DD format
                                const today = new Date();
                                const todayFormatted =
today.toISOString().split('T')[0];

                                // Set the date input to today's date
                                const dateSelector = document.getElementById('date-
selector');
                                dateSelector.value = todayFormatted;

                                // Event listener for date selection
                                dateSelector.addEventListener('change', function() {
                                    updateChart(this.value);
                                });

                                // Initialize chart with today's date, or most
recent data date if available
                                const initialDate =
uniqueDates.includes(todayFormatted) ? todayFormatted :
                                (uniqueDates.length > 0 ?
uniqueDates[uniqueDates.length - 1] : todayFormatted);

                                // Set the input value again to ensure it shows the
correct date
                                dateSelector.value = initialDate;

                                // Update chart with initial date
                                updateChart(initialDate);

                                // Auto-refresh the chart every minute
                                const refreshInterval = setInterval(function() {
                                    window.location.reload();
                                }, 60000); // Refresh every minute
                                });
</script>

<?= $this->endSection('content') ?>

```