

LAMPIRAN

Lampiran 1 Source Code Bot

```
const express = require('express');
const cron = require('node-cron');
const mysql = require('mysql');
const moment = require('moment');
const axios = require('axios');

const app = express();
const port = 3000;

app.set('view engine', 'ejs');
app.use(express.json());

// Setup MySQL connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'perpustakaan',
});

db.connect((err) => {
  if (err) {
    console.error('Error connecting to the database:', err);
  } else {
    console.log('Connected to the MySQL database.');
```

```

});

// Ganti dengan token Fonnte kamu
const FONNTE_TOKEN = 'A3CnQCWXjPUFBWNGBwRY';

// Fungsi kirim pesan ke Fonnte
async function sendWhatsAppMessage(number, message) {
  try {
    const response = await axios.post(
      'https://api.fonnte.com/send',
      {
        target: number,
        message: message,
      },
      {
        headers: {
          Authorization: FONNTE_TOKEN,
        },
      }
    );
    console.log('Pesan terkirim ke', number, response.data);
  } catch (error) {
    console.error(
      'Gagal kirim ke',
      number,
      error.response?.data || error.message
    );
  }
}

```

```
// Reminder 1 hari sebelum jatuh tempo

function checkLoansAndSendReminders() {

  const query = `

    SELECT m.member_name, m.member_phone, m.member_address,
    m.inst_name,

      m.birth_date, i.item_code, b.title, b.edition, b.isbn_issn,

      b.publish_year, l.due_date

    FROM loan l

    JOIN member m ON l.member_id = m.member_id

    JOIN item i ON l.item_code = i.item_code

    JOIN biblio b ON i.biblio_id = b.biblio_id

    WHERE l.due_date = DATE_ADD(CURDATE(), INTERVAL 1 DAY) AND
    l.is_return = 0;

  `;

  db.query(query, (err, results) => {

    if (err) throw err;

    results.forEach((result) => {

      const message = `

*REMINDER PENGEMBALIAN BUKU*

📖 *${result.title}* (Kode: ${result.item_code})

Edisi: ${result.edition}, ISBN/ISSN: ${result.isbn_issn}, Tahun: ${
      result.publish_year
    }

    }

    }

    🧑 *${result.member_name}*

    Alamat: ${result.member_address}
  `;

```

Institusi: \${result.inst_name}

Lahir: \${moment(result.birth_date).format('DD-MM-YYYY')}

 *Pengembalian besok!* Mohon dikembalikan tepat waktu yaa 😊

```
`;  
    sendWhatsAppMessage(result.member_phone, message);  
  });  
});  
}
```

// Reminder jika sudah telat

```
function checkLateLoansAndSendReminders() {  
  const query = `  
    SELECT m.member_name, m.member_phone, m.member_address,  
    m.inst_name,  
           m.birth_date, i.item_code, b.title, b.edition, b.isbn_issn,  
           b.publish_year, l.due_date,  
           DATEDIFF(CURDATE(), l.due_date) AS days_late  
    FROM loan l  
    JOIN member m ON l.member_id = m.member_id  
    JOIN item i ON l.item_code = i.item_code  
    JOIN biblio b ON i.biblio_id = b.biblio_id  
    WHERE l.due_date < CURDATE() AND l.is_return = 0;  
  `;  
  
  db.query(query, (err, results) => {  
    if (err) throw err;  
  
    results.forEach((result) => {
```

```

const message = `
*PENGINGAT TERLAMBAT*

📖 *${result.title}* (Kode: ${result.item_code})
Edisi: ${result.edition}, ISBN/ISSN: ${result.isbn_issn}, Tahun: ${
    result.publish_year
}

👤 *${result.member_name}*
Alamat: ${result.member_address}
Institusi: ${result.inst_name}
Lahir: ${moment(result.birth_date).format('DD-MM-YYYY')}

📅 Buku seharusnya dikembalikan pada *${moment(result.due_date).format(
    'DD-MM-YYYY'
)}*.

🕒 Sudah terlambat selama *${result.days_late} hari*.

Mohon segera dikembalikan 🙏
`;

sendWhatsAppMessage(result.member_phone, message);
});
});
}

// Cron Job: setiap hari jam 8 pagi
cron.schedule('19 22 * * *', () => {
    console.log('Menjalankan pengingat peminjaman...');
    checkLoansAndSendReminders();
});

```

```

    checkLateLoansAndSendReminders();
  });

  // Kirim pesan manual lewat endpoint POST
  app.post('/send-message', (req, res) => {
    const { number, message } = req.body;
    if (!number || !message) {
      return res.status(400).json({ error: 'Harap isi number dan message' });
    }

    sendWhatsAppMessage(number, message)
      .then(() => res.status(200).json({ success: true }))
      .catch((err) =>
        res.status(500).json({ success: false, error: err.message })
      );
  });

  app.get('/', (req, res) => {
    res.send('Server jalan dan siap kirim via Fonnte!');
  });

  app.listen(port, () => {
    console.log(`Server berjalan di http://localhost:${port}`);
  });

```