# System Design untuk Endpoint dengan Kode Unik dan Penanganan Race Condition

## Tujuan

Membuat endpoint untuk menyimpan data dengan kode unik (menggunakan format ITEM-<running\_number>) dan menangani race condition ketika banyak request terjadi bersamaan.

## Studi Kasus

Menyimpan entitas "item" dengan kode unik berformat ITEM-0001, ITEM-0002, dan seterusnya.

## Struktur Folder (Modular Design)

src/

├── config/

│ └── db.ts

├── controllers/

│ └── input\_items.controller.ts

├── routes/

│ └── input.route.ts

├── utils/

│ ├── logger.ts

│ └── errorHandler.ts

├── app.ts

└── server.ts

## Struktur Tabel PostgreSQL

CREATE TABLE items (

id SERIAL PRIMARY KEY,

code VARCHAR(20) UNIQUE NOT NULL,

name TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

## Strategi Penanganan Race Condition

Menggunakan PostgreSQL transaction dan locking dengan FOR UPDATE.

### Logika Endpoint (TypeScript)

export const createItem = async (req: Request, res: Response, next: NextFunction) => {

    const client = await db.connect();

    const { name } = req.body;

    try {

      await client.query('BEGIN');

      const result = await client.query(`SELECT nextval('item\_code\_seq') AS number`);

      const count = result.rows[0].number;

      const code = `ITEM-${String(count).padStart(4, '0')}`;

      await client.query(

        `INSERT INTO items (code, name) VALUES ($1, $2)`,

        [code, name]

      );

      await client.query('COMMIT');

      const timestamp = new Date().toISOString();

      logger.debug('Item created successfully', {

        code,

        name,

        date: timestamp

      });

      res.status(201).json({ code, name });

    } catch (err) {

      await client.query('ROLLBACK');

      const error = err as Error;

      logger.error('Failed to create item', {

        name,

        error: error.message

      });

      next(error);

    } finally {

      client.release();

    }

  };

## Endpoint

POST /api/items

Body:

{

  "name": "Keyboard"

}

Response:

{

    "code": "ITEM-0002",

    "name": "Keyboard"

}

## Logging dan Monitoring

Gunakan winston logger untuk mencatat log:

* Info: ketika data berhasil dibuat.
* Error: ketika transaksi gagal atau rollback.

Contoh:

logger.info(`Created item with code ${code}`);

logger.error('Failed to create item', { error });