UTS
EXPRESS

Albi Nur Rosif

3122522010

D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PENS PSDKU SUMENEP

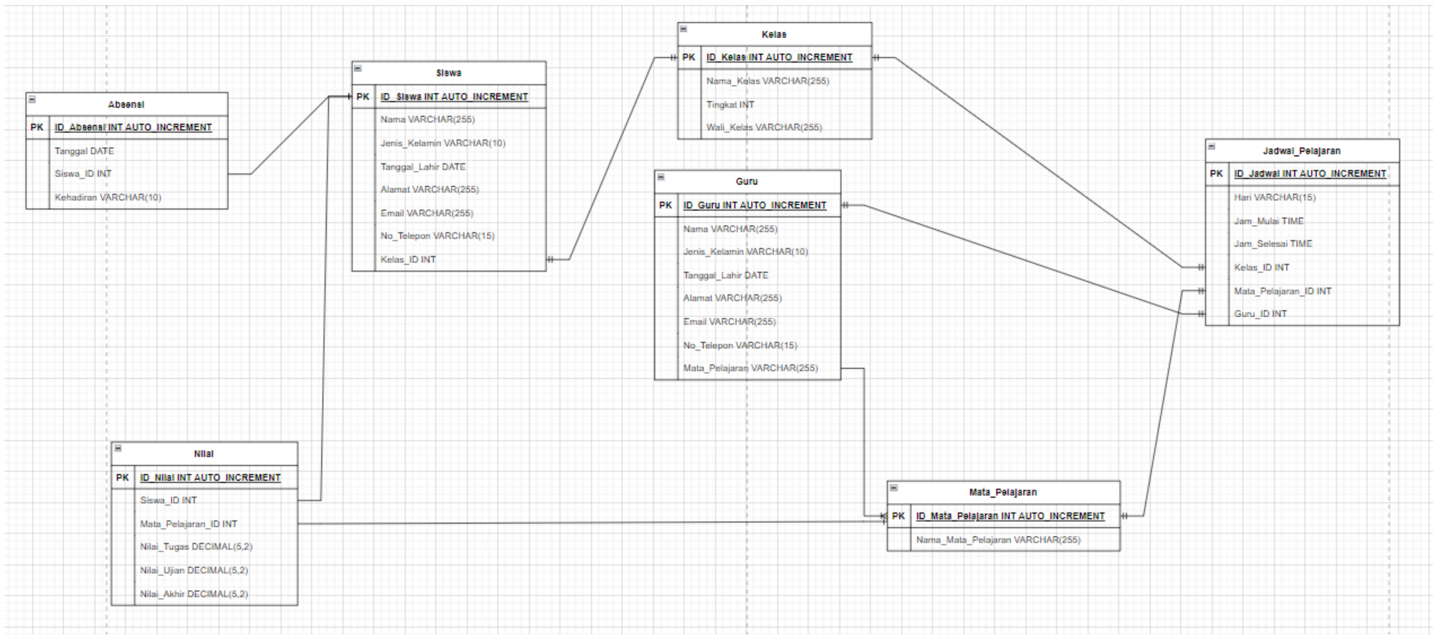
Judul Proyek: Aplikasi Manajemen Sekolah

Deskripsi Proyek: Aplikasi ini akan digunakan oleh sekolah untuk mengelola data siswa, guru, kelas, jadwal pelajaran, dan lainnya. Ini akan membantu sekolah dalam menjalankan operasi sehari-hari mereka dengan lebih efisien.

Tabel (Class) dalam Class Diagram:

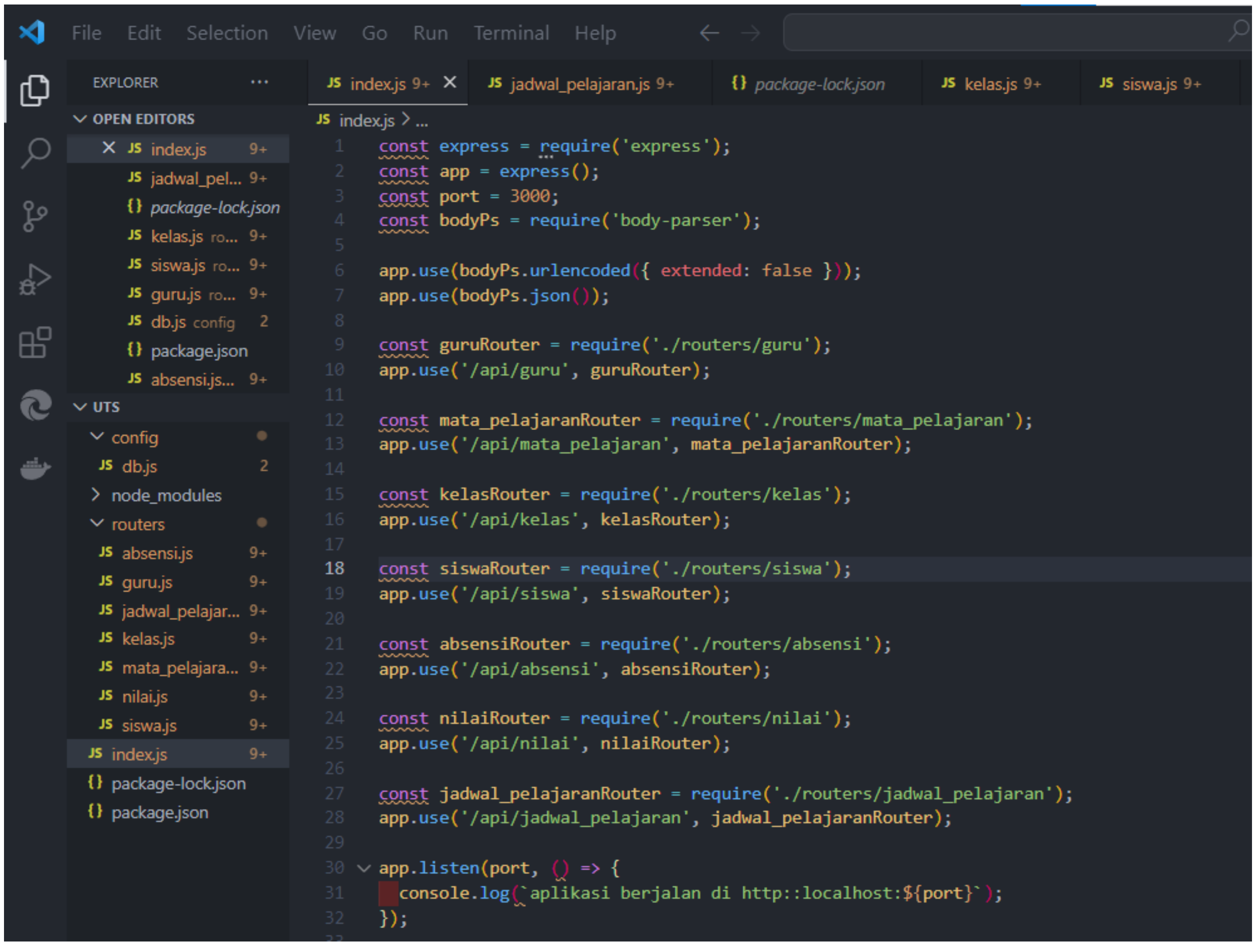
1. **Tabel Siswa**
    - Atribut: ID Siswa, Nama, Jenis Kelamin, Tanggal Lahir, Alamat, Email, No. Telepon, Kelas ID
    - Relasi: Satu siswa terdaftar dalam satu kelas.
2. **Tabel Guru**
    - Atribut: ID Guru, Nama, Jenis Kelamin, Tanggal Lahir, Email, No. Telepon, Mata Pelajaran
    - Relasi: Satu guru mengajar beberapa mata pelajaran**.**
3. **Tabel Kelas**
    - Atribut: ID Kelas, Nama Kelas, Tingkat, Wali Kelas
    - Relasi: Satu kelas memiliki satu wali kelas.
4. **Tabel Mata Pelajaran**
    - Atribut: ID Mata Pelajaran, Nama Mata Pelajaran
    - Relasi: Mata pelajaran dapat diajar oleh banyak guru.
5. **Tabel Jadwal Pelajaran**
    - Atribut: ID Jadwal, Hari, Jam Mulai, Jam Selesai, Kelas ID, Mata Pelajaran ID, Guru ID
    - Relasi: Setiap jadwal pelajaran terkait dengan satu kelas, satu mata pelajaran, dan satu guru.
6. **Tabel Absensi**
    - Atribut: ID Absensi, Tanggal, Siswa ID, Kehadiran (Hadir/Absen)
    - Relasi: Setiap absensi terkait dengan satu siswa.
7. **Tabel Nilai**
    - Atribut: ID Nilai, Siswa ID, Mata Pelajaran ID, Nilai Tugas, Nilai Ujian, Nilai Akhir
    - Relasi: Setiap nilai terkait dengan satu siswa dan satu mata pelajaran.

CLASS DIAGRAM



SCRIPT

Index.js



```javascript
const express = require('express');
const app = express();
const port = 3000;
const bodyPs = require('body-parser');

app.use(bodyPs.urlencoded({ extended: false }));
app.use(bodyPs.json());

const guruRouter = require('./routers/guru');
app.use('/api/guru', guruRouter);

const mata_pelajaranRouter = require('./routers/mata_pelajaran');
app.use('/api/mata_pelajaran', mata_pelajaranRouter);

const kelasRouter = require('./routers/kelas');
app.use('/api/kelas', kelasRouter);

const siswaRouter = require('./routers/siswa');
app.use('/api/siswa', siswaRouter);

const absensiRouter = require('./routers/absensi');
app.use('/api/absensi', absensiRouter);

const nilaiRouter = require('./routers/nilai');
app.use('/api/nilai', nilaiRouter);

const jadwal_pelajaranRouter = require('./routers/jadwal_pelajaran');
app.use('/api/jadwal_pelajaran', jadwal_pelajaranRouter);

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

Guru

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from guru', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  guru',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Nama').notEmpty(),
body('Jenis_Kelamin').notEmpty(), body('Tanggal_Lahir').notEmpty(),
body('Email').notEmpty(), body('No_Telepon').notEmpty(),
body('Mata_Pelajaran').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Nama: req.body.Nama,
    Jenis_Kelamin: req.body.Jenis_Kelamin,
    Tanggal_Lahir: req.body.Tanggal_Lahir,
    Email: req.body.Email,
    No_Telepon: req.body.No_Telepon,
    Mata_Pelajaran: req.body.Mata_Pelajaran,
  };
  connection.query('insert into guru set ?', data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from guru where ID_Guru = ${id}`, function (err,
rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Nama').notEmpty(),
body('Jenis_Kelamin').notEmpty(), body('Tanggal_Lahir').notEmpty(),
body('Email').notEmpty(), body('No_Telepon').notEmpty(),
body('Mata_Pelajaran').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Nama: req.body.Nama,
    Jenis_Kelamin: req.body.Jenis_Kelamin,
    Tanggal_Lahir: req.body.Tanggal_Lahir,
    Email: req.body.Email,
    No_Telepon: req.body.No_Telepon,
    Mata_Pelajaran: req.body.Mata_Pelajaran,
  };
  connection.query(`update guru set ? where ID_Guru = ${id}`, data, function
(err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from guru where ID_Guru = ${id}`, function (err, rows)
{
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Kelas

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from kelas', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  kelas',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Nama_Kelas').notEmpty(),
body('Tingkat').notEmpty(), body('Wali_Kelas').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Nama_Kelas: req.body.Nama_Kelas,
    Tingkat: req.body.Tingkat,
    Wali_Kelas: req.body.Wali_Kelas,
  };
  connection.query('insert into kelas set ?', data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from kelas where ID_kelas = ${id}`, function (err,
rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Nama_Kelas').notEmpty(),
body('Tingkat').notEmpty(), body('Wali_Kelas').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Nama_Kelas: req.body.Nama_Kelas,
    Tingkat: req.body.Tingkat,
    Wali_Kelas: req.body.Wali_Kelas,
  };
  connection.query(`update kelas set ? where ID_kelas = ${id}`, data, function
(err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from kelas where ID_kelas = ${id}`, function (err,
rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Siswa

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from siswa', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  siswa',
        data: rows[0],
      });
    }
  });
});

router.post(
  '/create',
  [body('Nama').notEmpty(), body('Jenis_Kelamin').notEmpty(),
body('Tanggal_Lahir').notEmpty(), body('Alamat').notEmpty(),
body('Email').notEmpty(), body('No_Telepon').notEmpty(),
body('Kelas_ID').notEmpty()],
  (req, res) => {
    const error = validationResult(req);
    if (!error.isEmpty()) {
      return res.status(400).json({
        error: error.array(),
      });
    }
    let data = {
      Nama: req.body.Nama,
      Jenis_Kelamin: req.body.Jenis_Kelamin,
      Tanggal_Lahir: req.body.Tanggal_Lahir,
      Alamat: req.body.Alamat,
      Email: req.body.Email,
      No_Telepon: req.body.No_Telepon,
      Kelas_ID: req.body.Kelas_ID,
    };
    connection.query('insert into siswa set ?', data, function (err, rows) {
      if (err) {
        return res.status(500).json({
          status: false,
          message: 'server gangguan',
        });
      } else {
        return res.status(201).json({
          status: true,
          message: 'data berhasil di buat',
          data: rows[0],
        });
      }
    });
  }
);

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from siswa where ID_siswa = ${id}`, function (err,
rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});
```

```
router.patch(
  '/update/:id',
  [body('Nama').notEmpty(), body('Jenis_Kelamin').notEmpty(),
body('Tanggal_Lahir').notEmpty(), body('Alamat').notEmpty(),
body('Email').notEmpty(), body('No_Telepon').notEmpty(),
body('Kelas_ID').notEmpty()],
  (req, res) => {
    const error = validationResult(req);
    if (!error.isEmpty()) {
      return res.status(422).json({
        error: error.array(),
      });
    }
    let id = req.params.id;
    let data = {
      Nama: req.body.Nama,
      Jenis_Kelamin: req.body.Jenis_Kelamin,
      Tanggal_Lahir: req.body.Tanggal_Lahir,
      Alamat: req.body.Alamat,
      Email: req.body.Email,
      No_Telepon: req.body.No_Telepon,
      Kelas_ID: req.body.Kelas_ID,
    };
    connection.query(`update siswa set ? where ID_siswa = ${id}`, data, function
(err, rows) {
      if (err) {
        return res.status(500).json({
          status: false,
          message: 'server error',
        });
      } else {
        return res.status(200).json({
          status: true,
          message: 'update berhasil',
        });
      }
    });
  }
);

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from siswa where ID_siswa = ${id}`, function (err,
rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Mata_Pelajaran

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from mata_pelajaran', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  mata_pelajaran',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Nama_Mata_Pelajaran').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Nama_Mata_Pelajaran: req.body.Nama_Mata_Pelajaran,
  };
  connection.query('insert into mata_pelajaran set ?', data, function (err, rows)
{
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from mata_pelajaran where ID_mata_pelajaran =
${id}`, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status(200).json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Nama_Mata_Pelajaran').notEmpty()], (req, res)
=> {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Nama_Mata_Pelajaran: req.body.Nama_Mata_Pelajaran,
  };
  connection.query(`update mata_pelajaran set ? where ID_mata_pelajaran = ${id}`,
data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from mata_pelajaran where ID_mata_pelajaran = ${id}`,
function (err, rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Absensi

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from absensi', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  absensi',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Tanggal').notEmpty(), body('Siswa_ID').notEmpty(),
body('Kehadiran').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Tanggal: req.body.Tanggal,
    Siswa_ID: req.body.Siswa_ID,
    Kehadiran: req.body.Kehadiran,
  };
  connection.query('insert into absensi set ?', data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from absensi where ID_absensi = ${id}`, function
(err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Tanggal').notEmpty(),
body('Siswa_ID').notEmpty(), body('Kehadiran').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Tanggal: req.body.Tanggal,
    Siswa_ID: req.body.Siswa_ID,
    Kehadiran: req.body.Kehadiran,
  };
  connection.query(`update absensi set ? where ID_absensi = ${id}`, data,
function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from absensi where ID_absensi = ${id}`, function (err,
rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Nilai

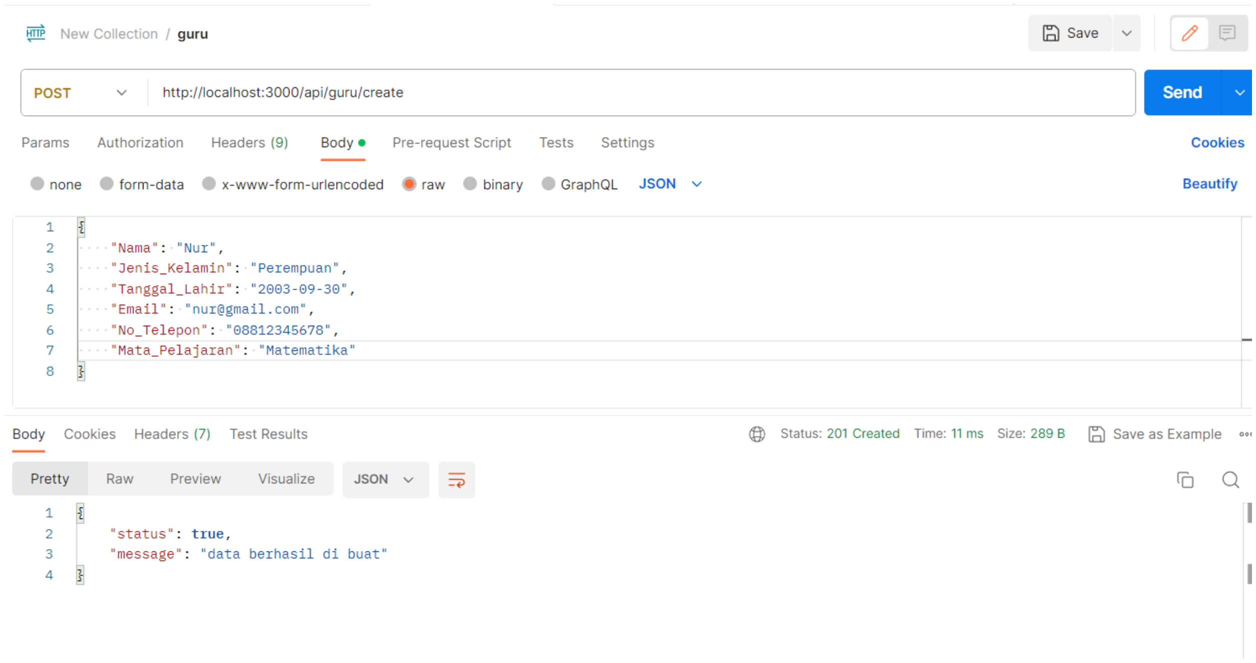
```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from nilai', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  nilai',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Siswa_ID').notEmpty(),
body('Mata_Pelajaran_ID').notEmpty(), body('Nilai_Tugas').notEmpty(),
body('Nilai_Ujian').notEmpty(), body('Nilai_Akhir').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Siswa_ID: req.body.Siswa_ID,
    Mata_Pelajaran_ID: req.body.Mata_Pelajaran_ID,
    Nilai_Tugas: req.body.Nilai_Tugas,
    Nilai_Ujian: req.body.Nilai_Ujian,
    Nilai_Akhir: req.body.Nilai_Akhir,
  };
  connection.query('insert into nilai set ?', data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from nilai where ID_nilai = ${id}`, function (err,
rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Siswa_ID').notEmpty(),
body('Mata_Pelajaran_ID').notEmpty(), body('Nilai_Tugas').notEmpty(),
body('Nilai_Ujian').notEmpty(), body('Nilai_Akhir').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Siswa_ID: req.body.Siswa_ID,
    Mata_Pelajaran_ID: req.body.Mata_Pelajaran_ID,
    Nilai_Tugas: req.body.Nilai_Tugas,
    Nilai_Ujian: req.body.Nilai_Ujian,
    Nilai_Akhir: req.body.Nilai_Akhir,
  };
  connection.query(`update nilai set ? where ID_nilai = ${id}`, data, function
(err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from nilai where ID_nilai = ${id}`, function (err,
rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

Jadwal_Pelajaran

```
const express = require('express');
const router = express.Router();
const { body, validationResult } = require('express-validator');
const connection = require('../config/db');

router.get('/', function (req, res) {
  connection.query('select * from jadwal_pelajaran', (err, rows) => {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gagal',
        Error: err,
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data  jadwal_pelajaran',
        data: rows[0],
      });
    }
  });
});

router.post('/create', [body('Hari').notEmpty(), body('Jam_Mulai').notEmpty(),
body('Jam_Selesai').notEmpty(), body('Kelas_ID').notEmpty(),
body('Mata_Pelajaran_ID').notEmpty(), body('Guru_ID').notEmpty()], (req, res) =>
{
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(400).json({
      error: error.array(),
    });
  }
  let data = {
    Hari: req.body.Hari,
    Jam_Mulai: req.body.Jam_Mulai,
    Jam_Selesai: req.body.Jam_Selesai,
    Kelas_ID: req.body.Kelas_ID,
    Mata_Pelajaran_ID: req.body.Mata_Pelajaran_ID,
    Guru_ID: req.body.Guru_ID,
  };
  connection.query('insert into jadwal_pelajaran set ?', data, function (err,
rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server gangguan',
      });
    } else {
      return res.status(201).json({
        status: true,
        message: 'data berhasil di buat',
        data: rows[0],
      });
    }
  });
});

router.get('/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`select * from jadwal_pelajaran where ID_jadwal_pelajaran =
${id}`, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error ',
        error: err,
      });
    }
    if (rows.length <= 0) {
      return res.status.json({
        status: false,
        message: 'not found',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'alat tangkap ada',
        data: rows[0],
      });
    }
  });
});

router.patch('/update/:id', [body('Hari').notEmpty(),
body('Jam_Mulai').notEmpty(), body('Jam_Selesai').notEmpty(),
body('Kelas_ID').notEmpty(), body('Mata_Pelajaran_ID').notEmpty(),
body('Guru_ID').notEmpty()], (req, res) => {
  const error = validationResult(req);
  if (!error.isEmpty()) {
    return res.status(422).json({
      error: error.array(),
    });
  }
  let id = req.params.id;
  let data = {
    Hari: req.body.Hari,
    Jam_Mulai: req.body.Jam_Mulai,
    Jam_Selesai: req.body.Jam_Selesai,
    Kelas_ID: req.body.Kelas_ID,
    Mata_Pelajaran_ID: req.body.Mata_Pelajaran_ID,
    Guru_ID: req.body.Guru_ID,
  };
  connection.query(`update jadwal_pelajaran set ? where ID_jadwal_pelajaran =
${id}`, data, function (err, rows) {
    if (err) {
      return res.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'update berhasil',
      });
    }
  });
});

router.delete('/delete/(:id)', function (req, res) {
  let id = req.params.id;
  connection.query(`delete from jadwal_pelajaran where ID_jadwal_pelajaran =
${id}`, function (err, rows) {
    if (err) {
      return req.status(500).json({
        status: false,
        message: 'server error',
      });
    } else {
      return res.status(200).json({
        status: true,
        message: 'data berhasil dihapus',
      });
    }
  });
});

module.exports = router;
```

POSTMAN

Guru



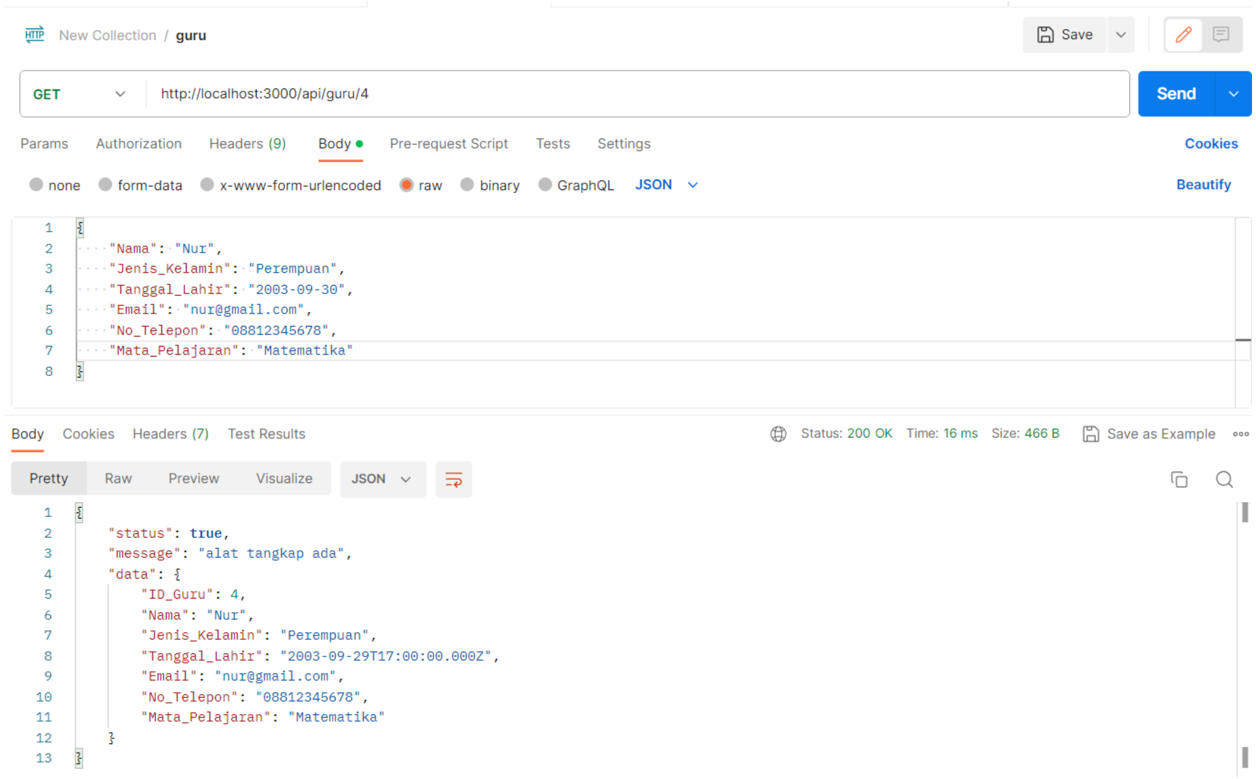
POST http://localhost:3000/api/guru/create

```
{
    "Nama": "Nur",
    "Jenis_Kelamin": "Perempuan",
    "Tanggal_Lahir": "2003-09-30",
    "Email": "nur@gmail.com",
    "No_Telepon": "08812345678",
    "Mata_Pelajaran": "Matematika"
}
```

Status: 201 Created Time: 11 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



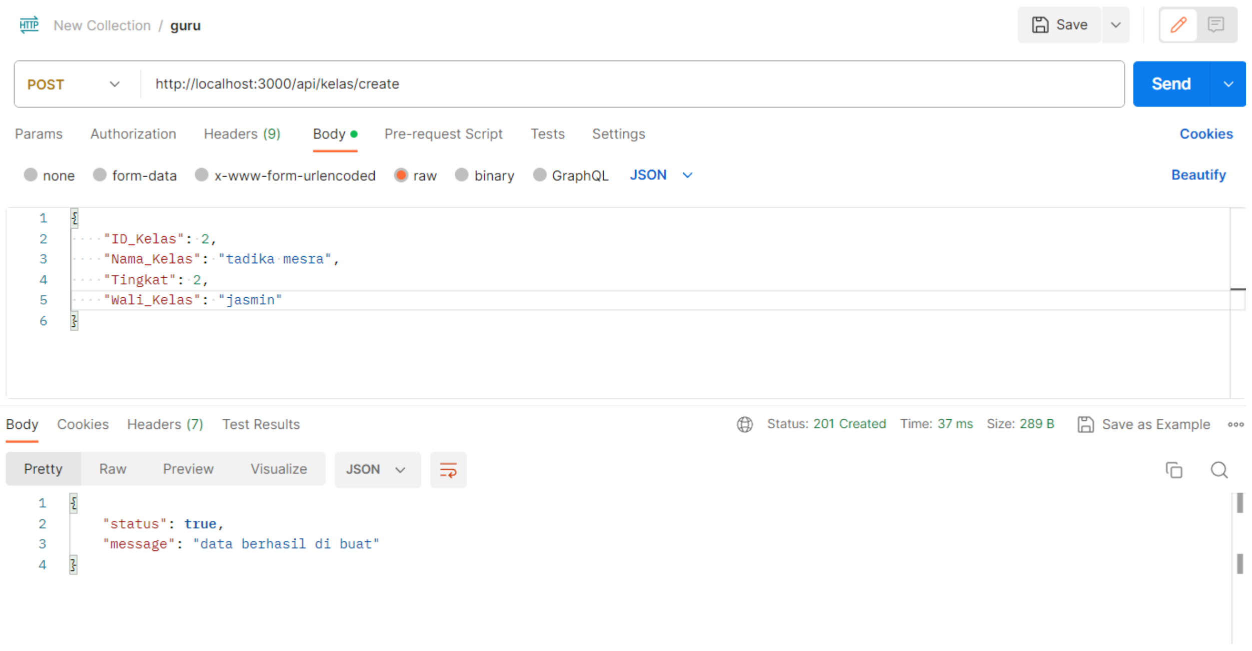
GET http://localhost:3000/api/guru/4

```
{
    "Nama": "Nur",
    "Jenis_Kelamin": "Perempuan",
    "Tanggal_Lahir": "2003-09-30",
    "Email": "nur@gmail.com",
    "No_Telepon": "08812345678",
    "Mata_Pelajaran": "Matematika"
}
```

Status: 200 OK Time: 16 ms Size: 466 B

```
{
    "status": true,
    "message": "alat tangkap ada",
    "data": {
        "ID_Guru": 4,
        "Nama": "Nur",
        "Jenis_Kelamin": "Perempuan",
        "Tanggal_Lahir": "2003-09-29T17:00:00.000Z",
        "Email": "nur@gmail.com",
        "No_Telepon": "08812345678",
        "Mata_Pelajaran": "Matematika"
    }
}
```

Kelas



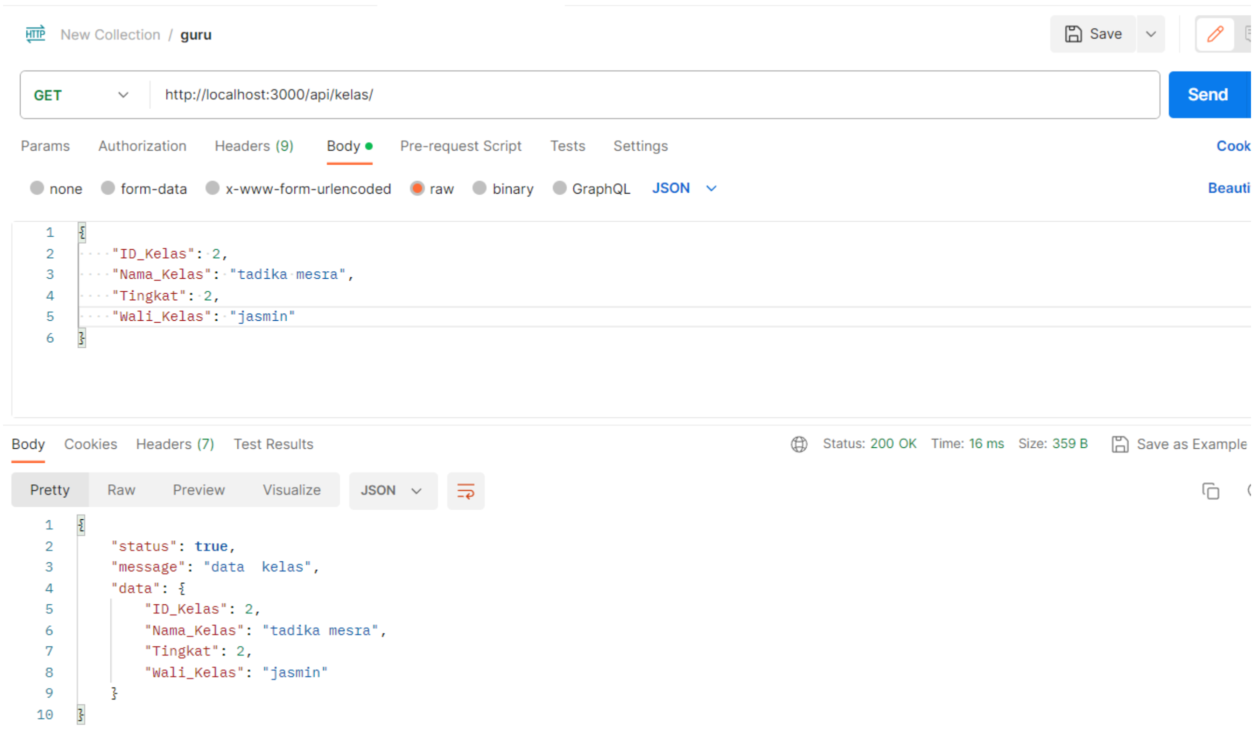
POST http://localhost:3000/api/kelas/create

```
{
    "ID_Kelas": 2,
    "Nama_Kelas": "tadika mesra",
    "Tingkat": 2,
    "Wali_Kelas": "jasmin"
}
```

Status: 201 Created Time: 37 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



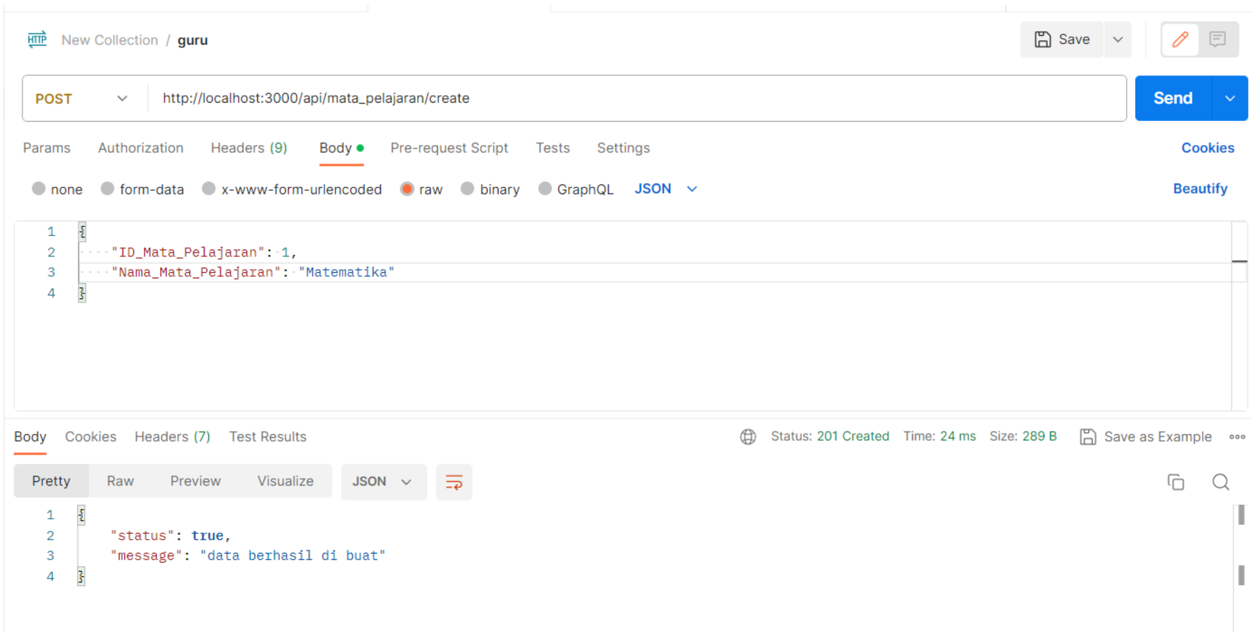
GET http://localhost:3000/api/kelas/

```
{
    "ID_Kelas": 2,
    "Nama_Kelas": "tadika mesra",
    "Tingkat": 2,
    "Wali_Kelas": "jasmin"
}
```

Status: 200 OK Time: 16 ms Size: 359 B

```
{
    "status": true,
    "message": "data  kelas",
    "data": {
        "ID_Kelas": 2,
        "Nama_Kelas": "tadika mesra",
        "Tingkat": 2,
        "Wali_Kelas": "jasmin"
    }
}
```

Mata_Pelajaran



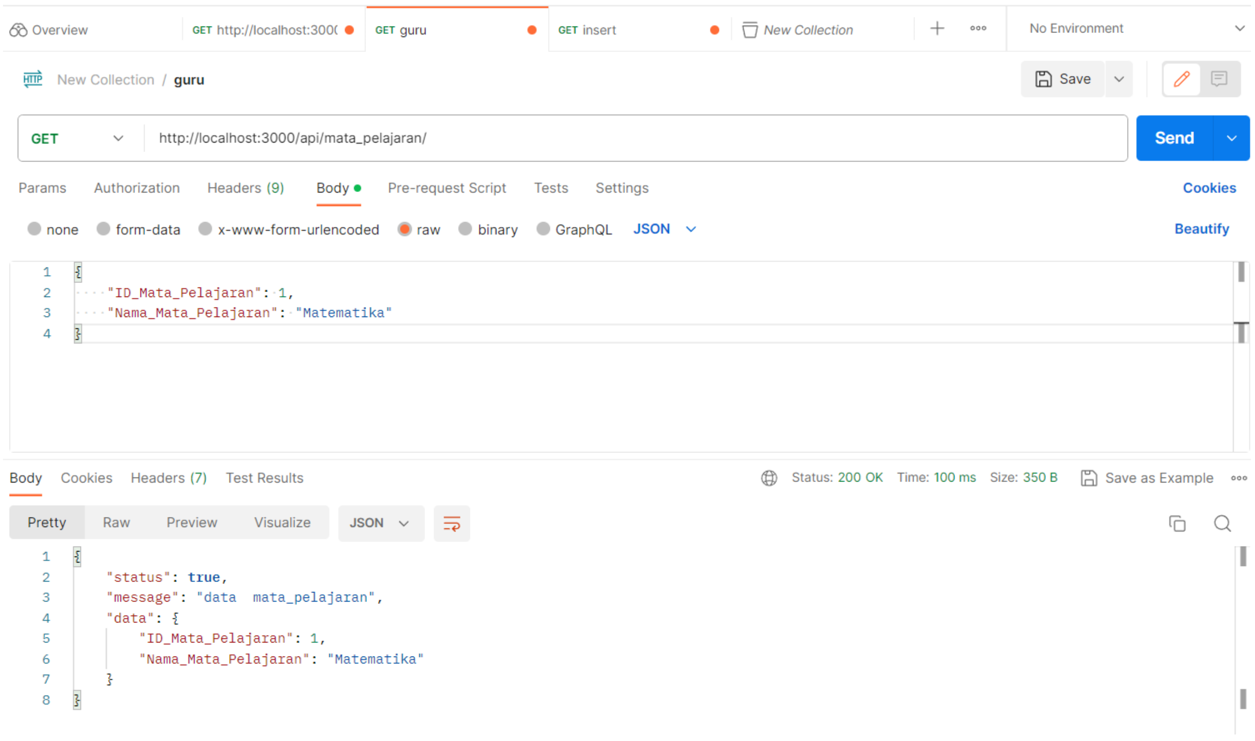
POST http://localhost:3000/api/mata_pelajaran/create

```
{
    "ID_Mata_Pelajaran": 1,
    "Nama_Mata_Pelajaran": "Matematika"
}
```

Status: 201 Created Time: 24 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



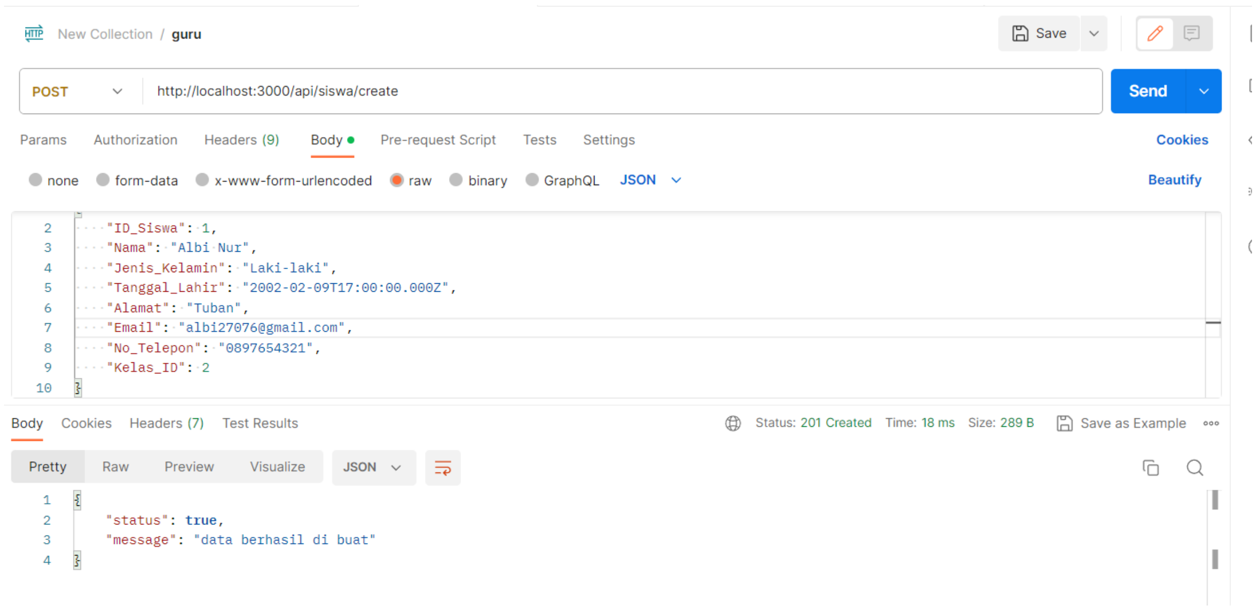
GET http://localhost:3000/api/mata_pelajaran/

```
{
    "ID_Mata_Pelajaran": 1,
    "Nama_Mata_Pelajaran": "Matematika"
}
```

Status: 200 OK Time: 100 ms Size: 350 B

```
{
    "status": true,
    "message": "data  mata_pelajaran",
    "data": {
        "ID_Mata_Pelajaran": 1,
        "Nama_Mata_Pelajaran": "Matematika"
    }
}
```

Siswa



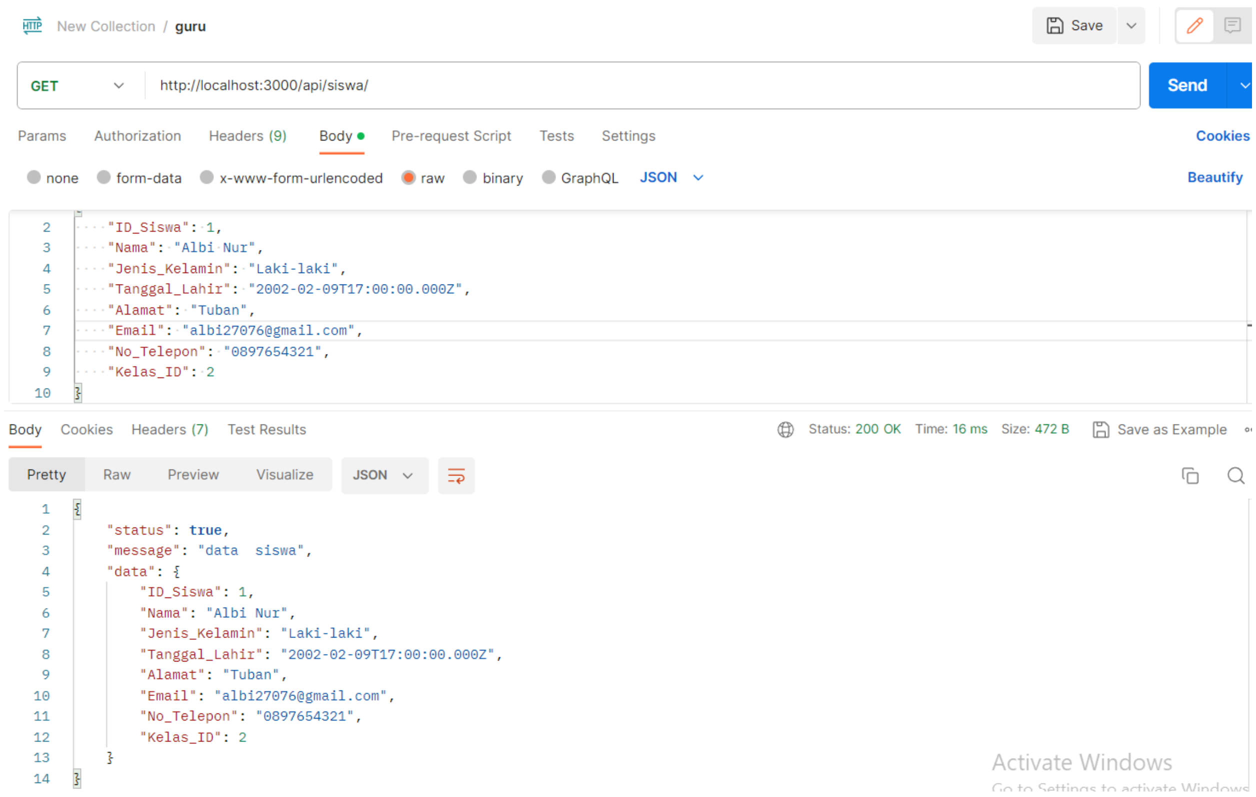
POST http://localhost:3000/api/siswa/create

```
    "ID_Siswa": 1,
    "Nama": "Albi Nur",
    "Jenis_Kelamin": "Laki-laki",
    "Tanggal_Lahir": "2002-02-09T17:00:00.000Z",
    "Alamat": "Tuban",
    "Email": "albi27076@gmail.com",
    "No_Telepon": "0897654321",
    "Kelas_ID": 2
}
```

Status: 201 Created Time: 18 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



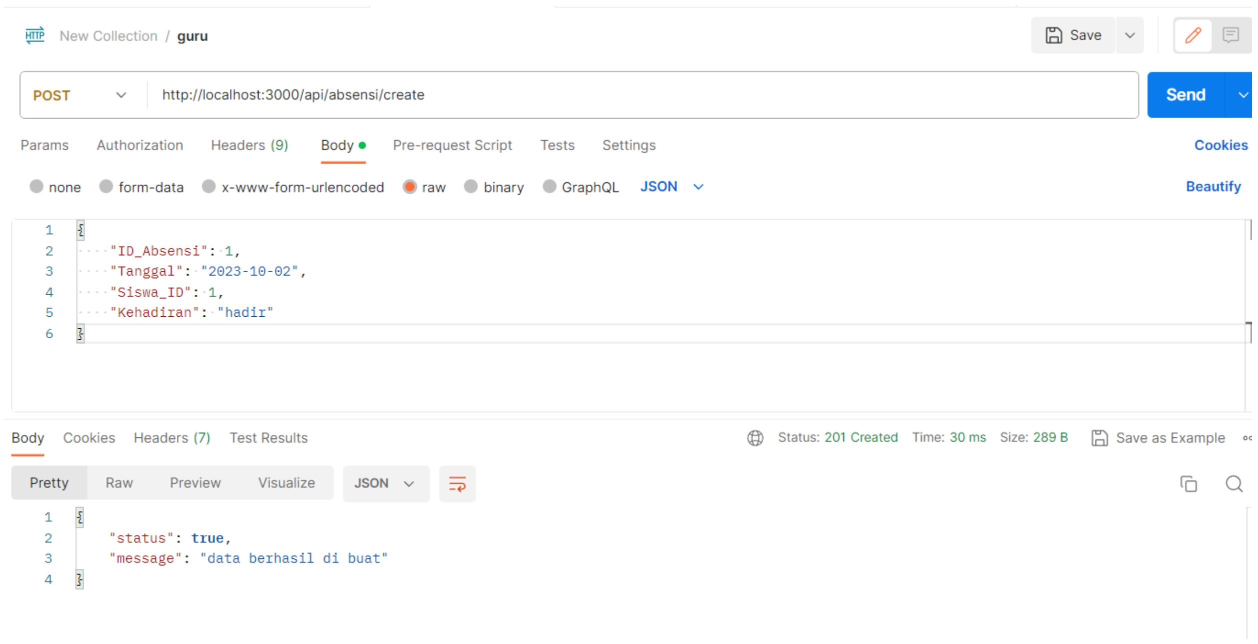
GET http://localhost:3000/api/siswa/

```
    "ID_Siswa": 1,
    "Nama": "Albi Nur",
    "Jenis_Kelamin": "Laki-laki",
    "Tanggal_Lahir": "2002-02-09T17:00:00.000Z",
    "Alamat": "Tuban",
    "Email": "albi27076@gmail.com",
    "No_Telepon": "0897654321",
    "Kelas_ID": 2
}
```

Status: 200 OK Time: 16 ms Size: 472 B

```
{
    "status": true,
    "message": "data  siswa",
    "data": {
        "ID_Siswa": 1,
        "Nama": "Albi Nur",
        "Jenis_Kelamin": "Laki-laki",
        "Tanggal_Lahir": "2002-02-09T17:00:00.000Z",
        "Alamat": "Tuban",
        "Email": "albi27076@gmail.com",
        "No_Telepon": "0897654321",
        "Kelas_ID": 2
    }
}
```

Absensi



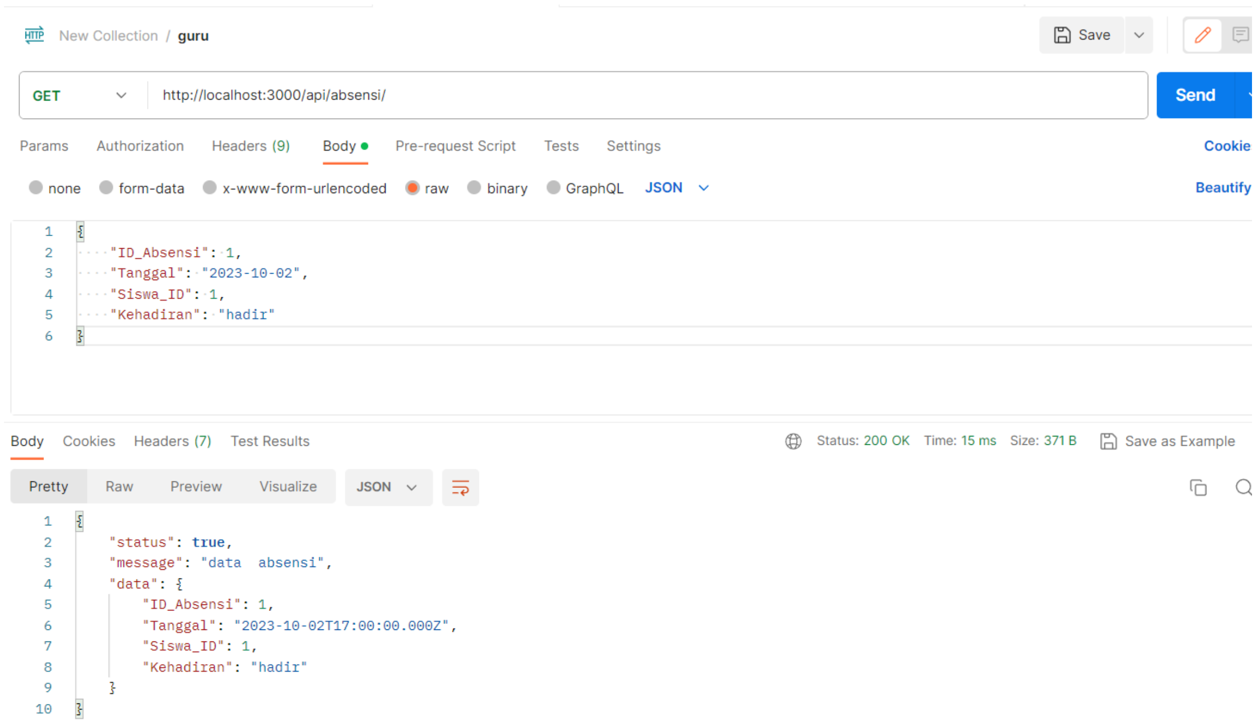
POST http://localhost:3000/api/absensi/create

```
{
    "ID_Absensi": 1,
    "Tanggal": "2023-10-02",
    "Siswa_ID": 1,
    "Kehadiran": "hadir"
}
```

Status: 201 Created Time: 30 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



GET http://localhost:3000/api/absensi/

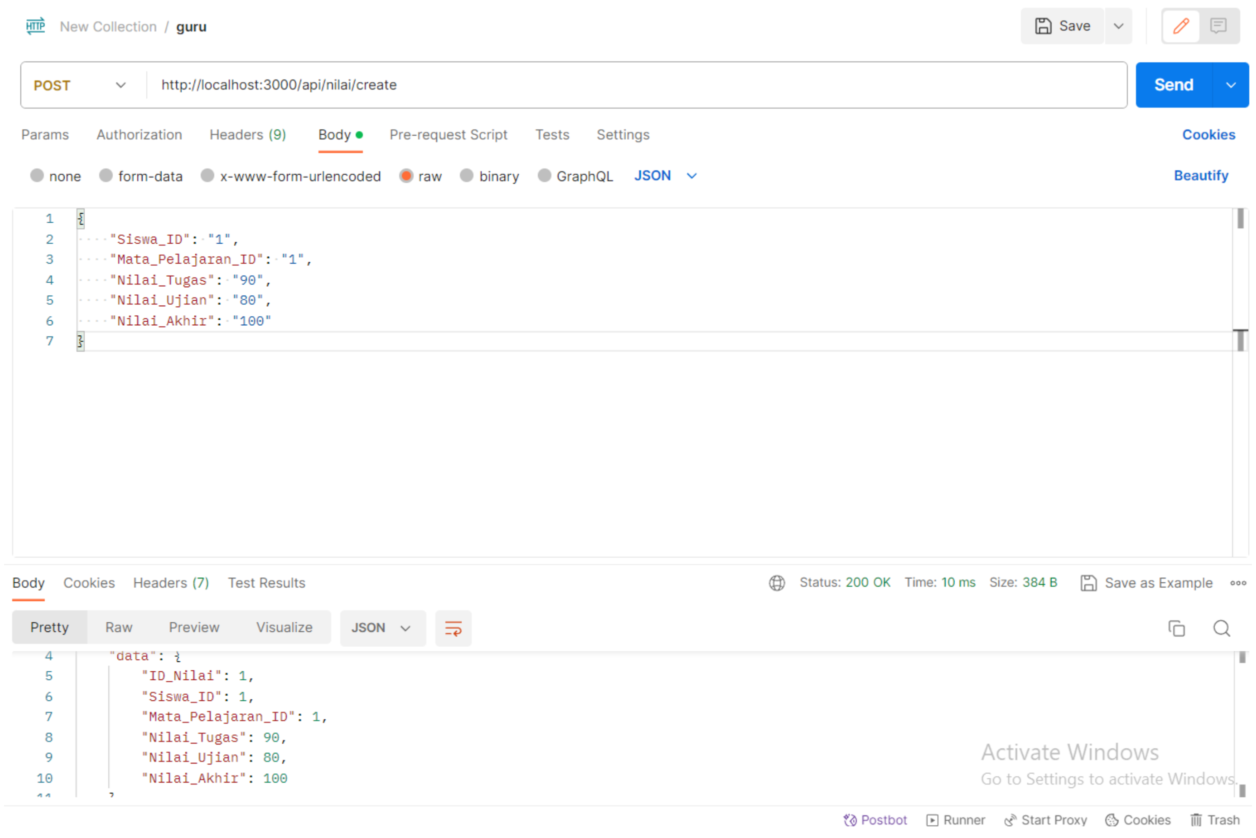
```
{
    "ID_Absensi": 1,
    "Tanggal": "2023-10-02",
    "Siswa_ID": 1,
    "Kehadiran": "hadir"
}
```

Status: 200 OK Time: 15 ms Size: 371 B

```
{
    "status": true,
    "message": "data  absensi",
    "data": {
        "ID_Absensi": 1,
        "Tanggal": "2023-10-02T17:00:00.000Z",
        "Siswa_ID": 1,
        "Kehadiran": "hadir"
    }
}
```

Nilai

Cek


New Collection / guru

Save

POST http://localhost:3000/api/nilai/create

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "Siswa_ID": "1",
    "Mata_Pelajaran_ID": "1",
    "Nilai_Tugas": "90",
    "Nilai_Ujian": "80",
    "Nilai_Akhir": "100"
}
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 10 ms Size: 384 B Save as Example

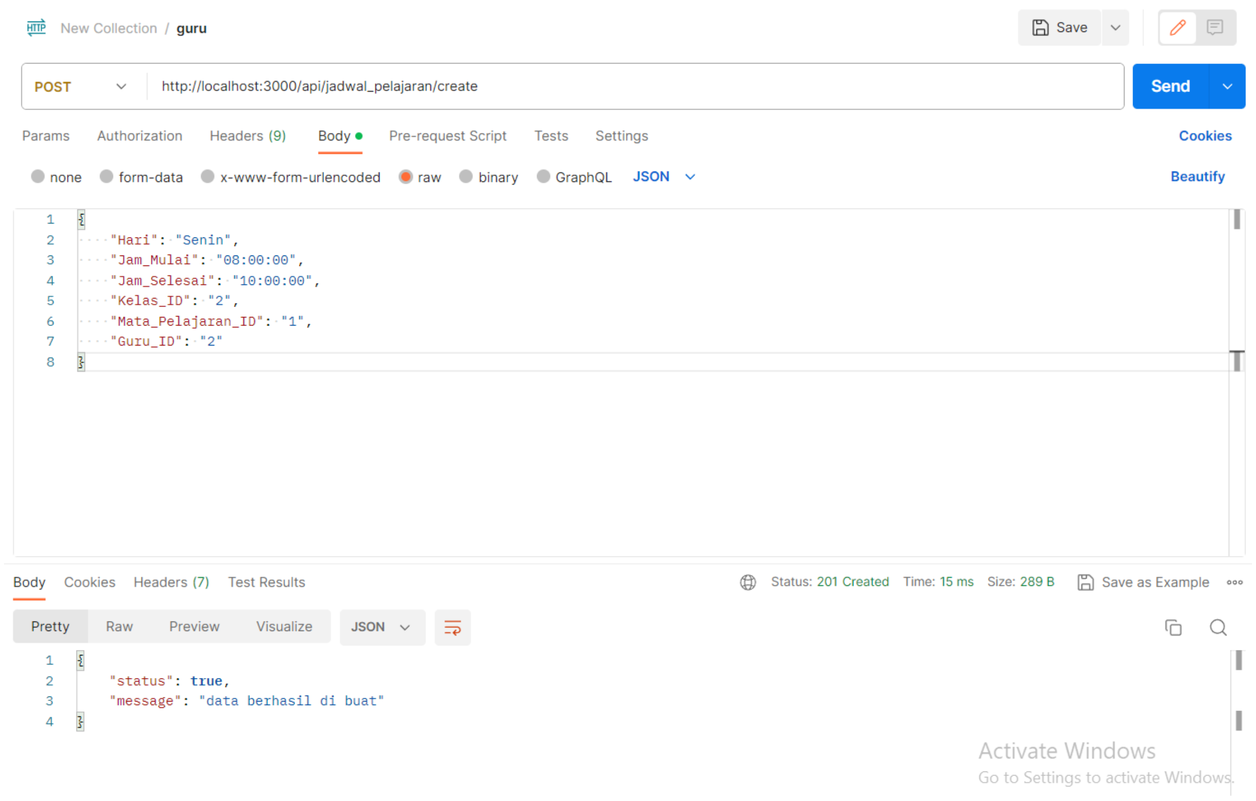
Pretty Raw Preview Visualize JSON

```
    "data": {
        "ID_Nilai": 1,
        "Siswa_ID": 1,
        "Mata_Pelajaran_ID": 1,
        "Nilai_Tugas": 90,
        "Nilai_Ujian": 80,
        "Nilai_Akhir": 100
```

Activate Windows
Go to Settings to activate Windows.

Postbot Runner Start Proxy Cookies Trash

Jadwal_Pelajaran



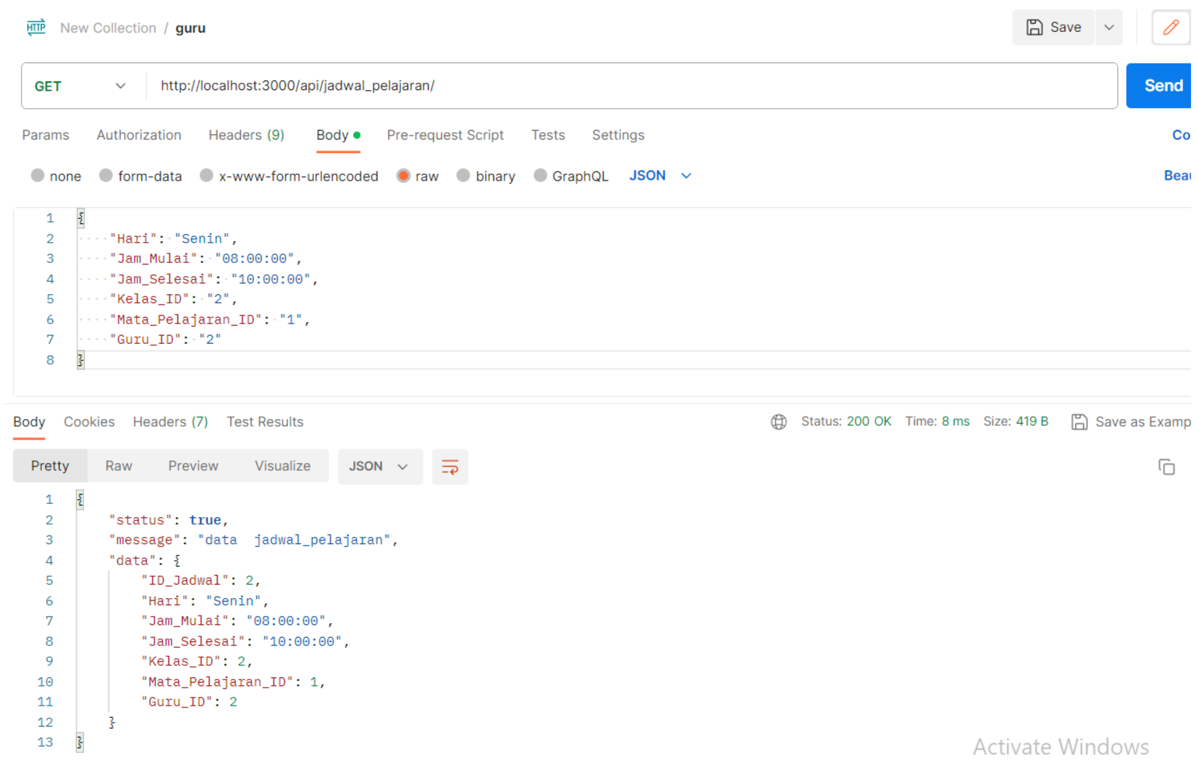
POST http://localhost:3000/api/jadwal_pelajaran/create

```
{
    "Hari": "Senin",
    "Jam_Mulai": "08:00:00",
    "Jam_Selesai": "10:00:00",
    "Kelas_ID": "2",
    "Mata_Pelajaran_ID": "1",
    "Guru_ID": "2"
}
```

Status: 201 Created Time: 15 ms Size: 289 B

```
{
    "status": true,
    "message": "data berhasil di buat"
}
```

Cek



GET http://localhost:3000/api/jadwal_pelajaran/

```
{
    "Hari": "Senin",
    "Jam_Mulai": "08:00:00",
    "Jam_Selesai": "10:00:00",
    "Kelas_ID": "2",
    "Mata_Pelajaran_ID": "1",
    "Guru_ID": "2"
}
```

Status: 200 OK Time: 8 ms Size: 419 B

```
{
    "status": true,
    "message": "data  jadwal_pelajaran",
    "data": {
        "ID_Jadwal": 2,
        "Hari": "Senin",
        "Jam_Mulai": "08:00:00",
        "Jam_Selesai": "10:00:00",
        "Kelas_ID": 2,
        "Mata_Pelajaran_ID": 1,
        "Guru_ID": 2
    }
}
```

Belum pakai inner join

Link github

https://github.com/albinurrosif/10_MID_UTS_express_sekolah.git