

LANJUTAN PRAKTIKUM API DI ALTIM PEMROGRAMAN WEB



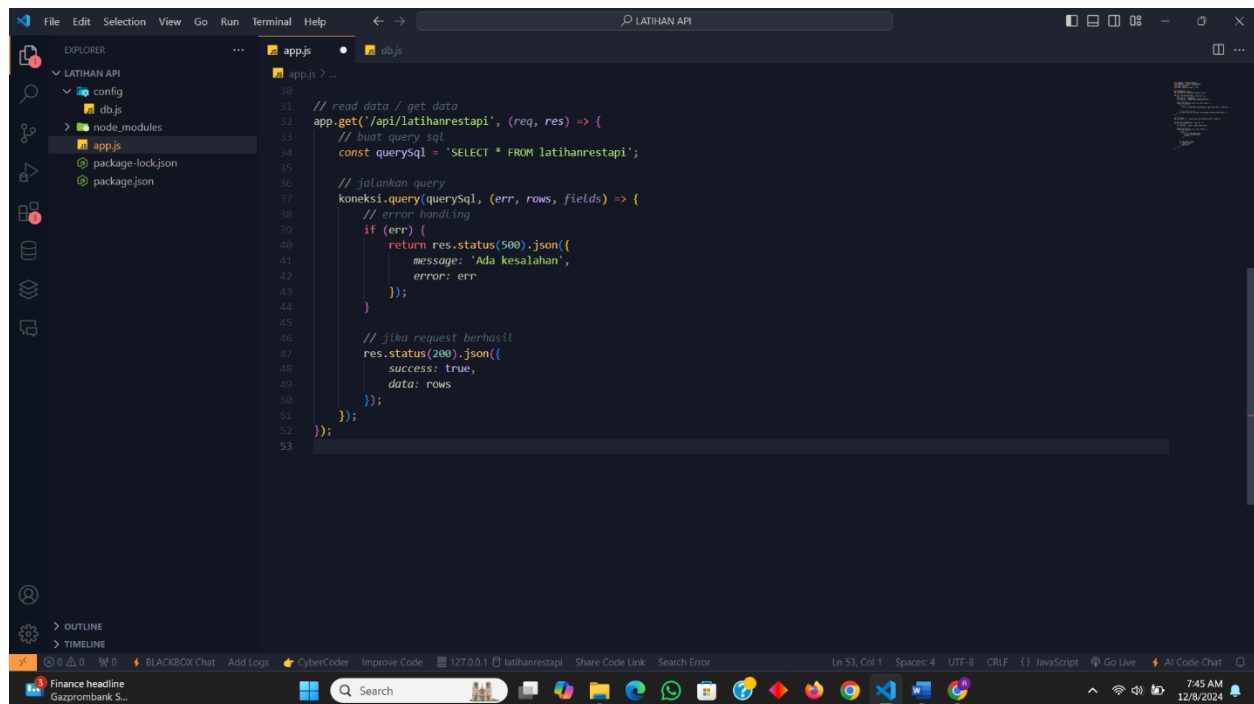
RIYADI TRI WALUYA BUDI

2308065

SIK A3

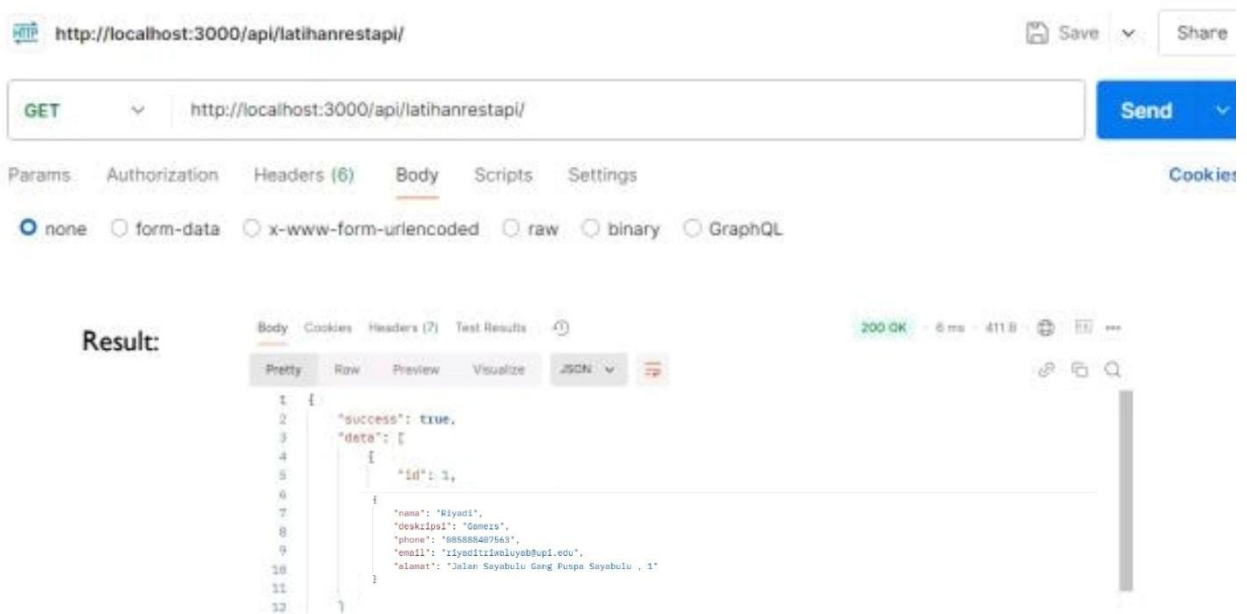
READ DATA

Di dalam file **app.js**, tambahkan kode ini dibawah fungsi createdata



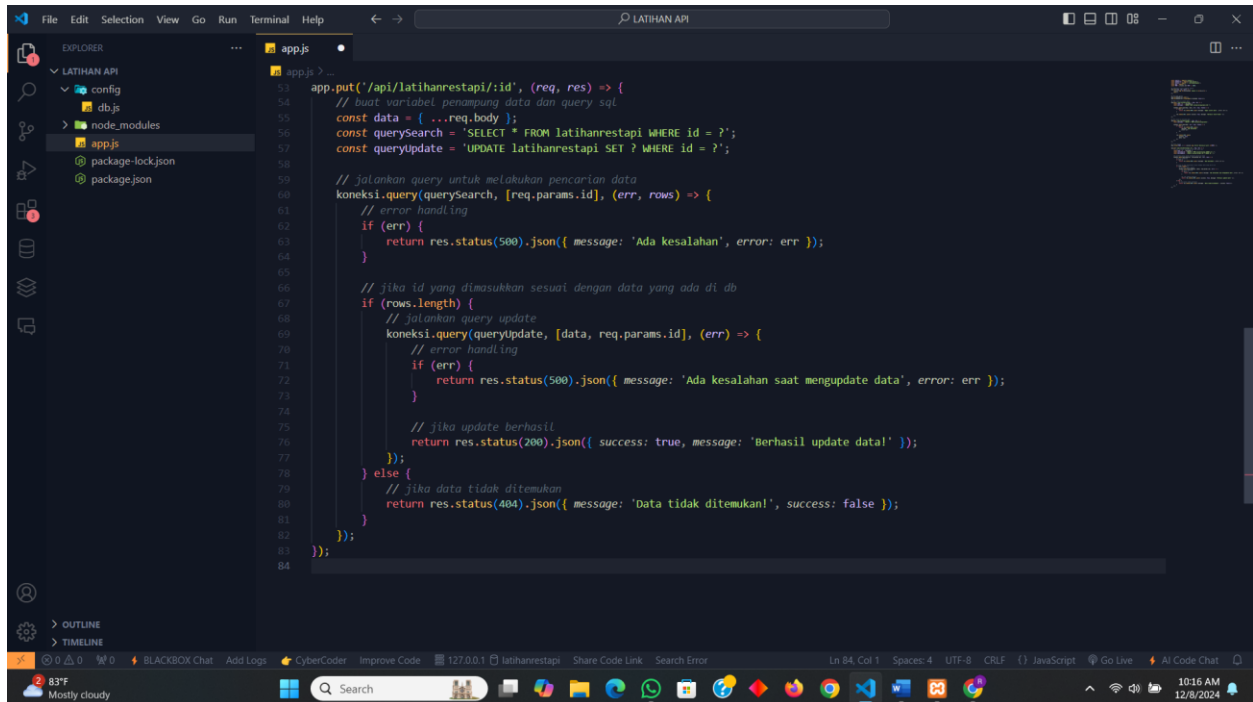
```
30
31 // read data / get data
32 app.get('/api/latihanrestapi', (req, res) => {
33   // buat query sql
34   const querySql = 'SELECT * FROM latihanrestapi';
35
36   // jalankan query
37   koneksi.query(querySql, (err, rows, fields) => {
38     // error handling
39     if (err) {
40       return res.status(500).json({
41         message: 'Ada kesalahan',
42         error: err
43       });
44     }
45
46     // jika request berhasil
47     res.status(200).json({
48       success: true,
49       data: rows
50     });
51   });
52 });
53
```

*Untuk Tampilan Postman



UPDATE DATA

Di dalam file **app.js**, tambahkan kode ini dibawah fungsi Insertdata



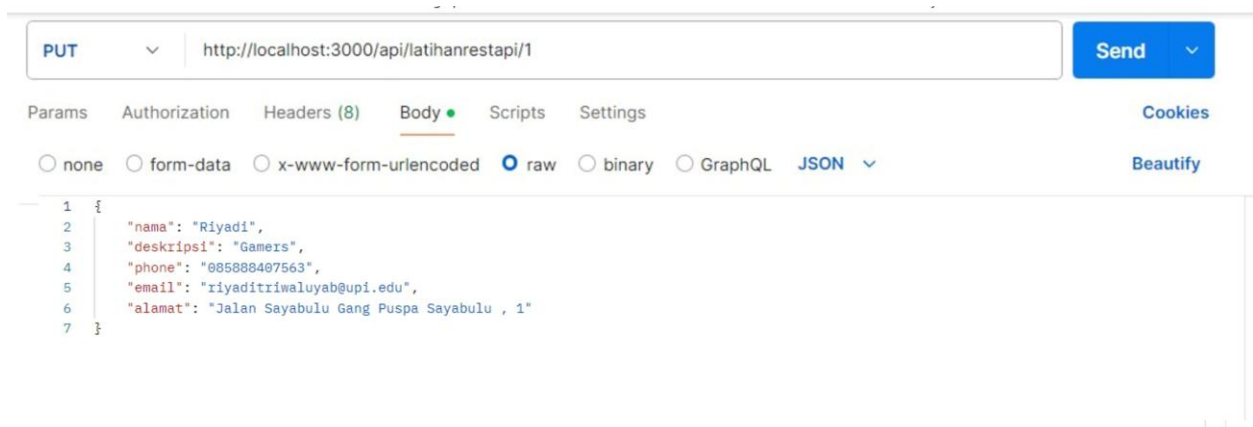
```
app.put('/api/latihanrestapi/:id', (req, res) => {
  // buat variabel penampung data dan query sql
  const data = { ...req.body };
  const querySearch = 'SELECT * FROM latihanrestapi WHERE id = ?';
  const queryUpdate = 'UPDATE latihanrestapi SET ? WHERE id = ?';

  // jalankan query untuk melakukan pencarian data
  koneksi.query(querySearch, [req.params.id], (err, rows) => {
    // error handling
    if (err) {
      return res.status(500).json({ message: 'Ada kesalahan', error: err });
    }

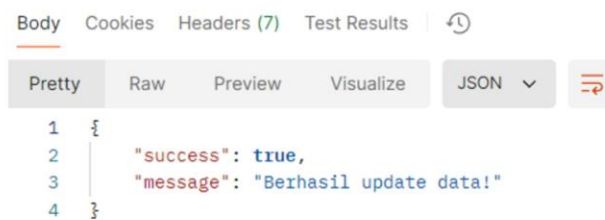
    // jika id yang dimasukkan sesuai dengan data yang ada di db
    if (rows.length) {
      // jalankan query update
      koneksi.query(queryUpdate, [data, req.params.id], (err) => {
        // error handling
        if (err) {
          return res.status(500).json({ message: 'Ada kesalahan saat mengupdate data', error: err });
        }

        // jika update berhasil
        return res.status(200).json({ success: true, message: 'Berhasil update data!' });
      });
    } else {
      // jika data tidak ditemukan
      return res.status(404).json({ message: 'Data tidak ditemukan!', success: false });
    }
  });
});
```

Tampilan untuk Postman

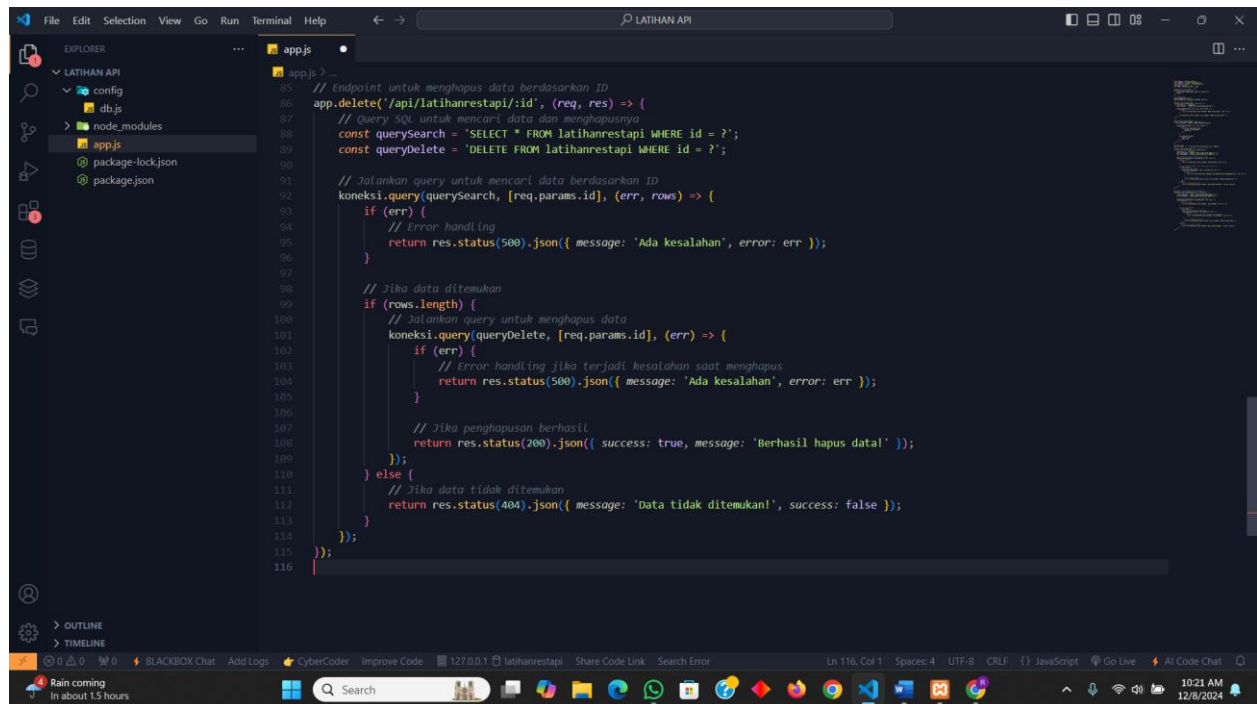


Result:



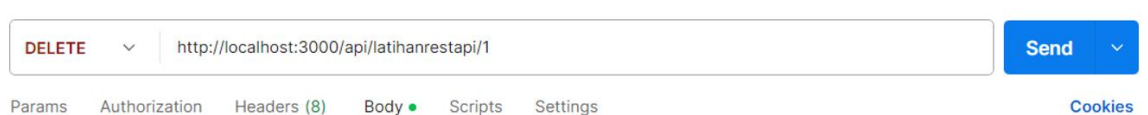
DELETE DATA

Di dalam file **app.js**, tambahkan kode ini dibawah fungsi update data



```
85 // Endpoint untuk menghapus data berdasarkan ID
86 app.delete('/api/latihanrestapi/:id', (req, res) => {
87   // Query SQL untuk mencari data dan menghapusnya
88   const querySearch = 'SELECT * FROM latihanrestapi WHERE id = ?';
89   const queryDelete = 'DELETE FROM latihanrestapi WHERE id = ?';
90
91   // Jalankan query untuk mencari data berdasarkan ID
92   koneksi.query(querySearch, [req.params.id], (err, rows) => {
93     if (err) {
94       // Error handling
95       return res.status(500).json({ message: 'Ada kesalahan', error: err });
96     }
97
98     // Jika data ditemukan
99     if (rows.length) {
100       // Jalankan query untuk menghapus data
101       koneksi.query(queryDelete, [req.params.id], (err) => {
102         if (err) {
103           // Error handling jika terjadi kesalahan saat menghapus
104           return res.status(500).json({ message: 'Ada kesalahan', error: err });
105         }
106
107         // Jika penghapusan berhasil
108         return res.status(200).json({ success: true, message: 'Berhasil hapus data!' });
109       });
110     } else {
111       // Jika data tidak ditemukan
112       return res.status(404).json({ message: 'Data tidak ditemukan!', success: false });
113     }
114   });
115 });
116
```

Tampilan Postman



Result:

