

M3 MULTI PRIMARY KEY & PROCEDURE

1. TUJUAN

CPMK: Mahasiswa mengetahui dan memahami dasar dari web dan dasar microservices.

Sub-CPMK:

- Mahasiswa mampu mengimplementasikan Rest API sederhana dengan menggunakan bahasa Java Spring
- Mahasiswa mampu mengimplementasikan Rest API dengan database sederhana

2. DURASI WAKTU

1 pertemuan x 3 jam

3. DASAR TEORI

Java Web

4. PERCOBAAN

A. Membuat Database

- Bukalah Aplikasi SQL Server Manajemen Studio
- Buatlah database dengan nama DB_Sertifikasi
- Buatlah tabel dengan nama stf_msdetailskema dengan struktur seperti dibawah ini, set sebagai primary key untuk dsk_id, skm_id, dan pro_id (**semua PK tidak perlu di auto increment**).

| UPTIF-078.DB_Serti....stf_msdetailskema | | | |
|---|----------------------|-----------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| 🔑 | dsk_id | int | <input type="checkbox"/> |
| 🔑 | skm_id | int | <input type="checkbox"/> |
| 🔑 | pro_id | int | <input type="checkbox"/> |
| | dsk_tanggal_mulai | datetime | <input checked="" type="checkbox"/> |
| | dsk_tanggal_berakhir | datetime | <input checked="" type="checkbox"/> |
| | dsk_total_peserta | int | <input checked="" type="checkbox"/> |
| | dsk_kompeten | int | <input checked="" type="checkbox"/> |
| | dsk_belum_kompeten | int | <input checked="" type="checkbox"/> |
| | dsk_tidak_hadir | int | <input checked="" type="checkbox"/> |
| | dsk_status | int | <input checked="" type="checkbox"/> |
| | dsk_creaby | int | <input checked="" type="checkbox"/> |
| | dsk_credate | date | <input checked="" type="checkbox"/> |
| | dsk_modiby | int | <input checked="" type="checkbox"/> |
| | dsk_modidate | date | <input checked="" type="checkbox"/> |

- Buatlah sebuah stored procedure sederhana untuk menampilkan list dari detail skema yang aktif

```

CREATE PROCEDURE [dbo].[get_active_detail_skema]
    @status INT
AS
BEGIN
    SELECT *
    FROM stf_msdetailskema
    WHERE dsk_status = @status;
END
GO

```

5. Bukalah projek yang sebelumnya (PRG7_Sertifikasi), tambahkan model dengan nama DetailSkemaPK.java di package model, lalu ketikkan kode seperti dibawah ini

```

@Embeddable
public class DetailSkemaPK implements Serializable {
    3 usages
    @Column(name = "dsk_id")
    private Integer dsk_id;
    3 usages
    @Column(name = "skm_id")
    private Integer skm_id;
    3 usages
    @Column(name = "pro_id")
    private Integer pro_id;

    public DetailSkemaPK(){}

    no usages
    public DetailSkemaPK(Integer dsk_id, Integer skm_id, Integer pro_id) {
        this.dsk_id = dsk_id;
        this.skm_id = skm_id;
        this.pro_id = pro_id;
    }
}

```

6. Dalam hal ini kita menggunakan anotasi @Embeddable, ketika sebuah kelas ditandai dengan @Embeddable, itu berarti bahwa objek dari kelas tersebut dapat menjadi bagian dari entitas lainnya, biasanya sebagai bagian dari entitas yang lebih besar.
7. Selanjutnya, kita akan membuat DetailSkema.java di package yang sama. @EmbeddedId digunakan untuk menandai properti detailSkemaPK di kelas DetailSkema sebagai kunci utama yang terdiri dari objek kelas DetailSkemaPK. Dengan kata lain, kunci utama entitas DetailSkema terdiri dari kombinasi nilai skm_id dan pro_id yang disimpan dalam kelas DetailSkemaPK.

```

12  @Entity
13  @Table(name = "stf_msdetailskema")
14  public class DetailSkema implements Serializable {
15      3 usages
16      @EmbeddedId
17      private DetailSkemaPK detailSkemaPK;
18
19      3 usages
20      @Column(name = "dsk_tanggal_mulai")
21      private Date dsk_tanggal_mulai;
22
23      3 usages
24      @Column(name = "dsk_tanggal_berakhir")
25      private Date dsk_tanggal_berakhir;
26
27      3 usages
28      @Column(name = "dsk_total_peserta")
29      private Integer dsk_total_peserta;
30
31      3 usages
32      @Column(name = "dsk_kompeten")
33      private Integer dsk_kompeten;
34
35      3 usages
36      @Column(name = "dsk_belum_kompeten")
37      private Integer dsk_belum_kompeten;

```

```

38      @Column(name = "dsk_tidak_hadir")
39      private Integer dsk_tidak_hadir;
40
41      3 usages
42      @Column(name = "dsk_status")
43      private Integer dsk_status;
44
45      3 usages
46      @Column(name = "dsk_creaby")
47      private Integer dsk_creaby;
48
49      3 usages
50      @Column(name = "dsk_creadate")
51      private Date dsk_creadate;
52
53      3 usages
54      @Column(name = "dsk_modiby")
55      private Integer dsk_modiby;
56
57      3 usages
58      @Column(name = "dsk_modidate")
59      private Date dsk_modidate;

```

8. Setelah itu, buatlah sebuah Vo dengan nama DetailSkema.java untuk menampilkan data nantinya. Jangan lupa untuk menambahkan konstruktor dan enkapsulasi.

```

public class DetailSkemaVo implements Serializable {
    3 usages
    private Integer dskId;
    3 usages
    private Integer skmId;
    2 usages
    private String skmDesc;
    3 usages
    private Integer proId;
    2 usages
    private String proDesc;
    3 usages
    private Date tanggalMulai;
    3 usages
    private Date tanggalBerakhir;
    3 usages
    private Integer totalPeserta;
    3 usages
    private Integer kompeten;
    3 usages
    private Integer belumKompeten;
    3 usages
    private Integer tidakHadir;
    3 usages
    private Integer status;

    no usages
    public DetailSkemaVo() {
    }
}

```

```

public DetailSkemaVo(DetailSkema DetailSkema) {
    this.dskId = DetailSkema.getDetailSkemaPK().getDsk_id();
    this.skmId = DetailSkema.getDetailSkemaPK().getSkm_id();
    this.proId = DetailSkema.getDetailSkemaPK().getPro_id();
    this.tanggalMulai = DetailSkema.getDsk_tanggal_mulai();
    this.tanggalBerakhir = DetailSkema.getDsk_tanggal_berakhir();
    this.totalPeserta = DetailSkema.getDsk_total_peserta();
    this.kompeten = DetailSkema.getDsk_kompeten();
    this.belumKompeten = DetailSkema.getDsk_belum_kompeten();
    this.tidakHadir = DetailSkema.getDsk_tidak_hadir();
    this.status = DetailSkema.getDsk_status();
}

```

9. Selanjutnya, tambahkan DetailSkemaVoForm yang nantinya untuk form data.

```

public class DetailSkemaVoForm {
    3 usages
    private Integer dskId;
    3 usages
    private Integer skmId;
    3 usages
    private Integer proId;
    3 usages
    private Date tanggalMulai;
    3 usages
    private Date tanggalBerakhir;
    3 usages
    private Integer totalPeserta;
    3 usages
    private Integer kompeten;
    3 usages
    private Integer belumKompeten;
    3 usages
    private Integer tidakHadir;
    3 usages
    private Integer status;
    3 usages
    private Integer creaby;
    3 usages
    private Date creadate;
    3 usages
    private Integer modiby;
    3 usages
    private Date modidate;

```

```

public DetailSkemaVoForm() {
}

no usages
public DetailSkemaVoForm(DetailSkema DetailSkema) {
    this.dskId = DetailSkema.getDetailSkemaPK().getDsk_id();
    this.skmId = DetailSkema.getDetailSkemaPK().getSkm_id();
    this.proId = DetailSkema.getDetailSkemaPK().getPro_id();
    this.tanggalMulai = DetailSkema.getDsk_tanggal_mulai();
    this.tanggalBerakhir = DetailSkema.getDsk_tanggal_berakhir();
    this.totalPeserta = DetailSkema.getDsk_total_peserta();
    this.kompeten = DetailSkema.getDsk_kompeten();
    this.belumKompeten = DetailSkema.getDsk_belum_kompeten();
    this.tidakHadir = DetailSkema.getDsk_tidak_hadir();
    this.status = DetailSkema.getDsk_status();
    this.creaby = DetailSkema.getDsk_creaby();
    this.creadate = DetailSkema.getDsk_creadate();
    this.modiby = DetailSkema.getDsk_modiby();
    this.modidate = DetailSkema.getDsk_modidate();
}

```

10. Kita juga akan membuat DetailSkemaConstant.java di package constant

```

1 package id.co.prg7_sertifikasi.constant;
2
3 public class DetailSkemaConstant {
4     public static final String mNotFound = "Detail Skema not found";
5     public static final String mEmptyData = "No data available";
6     public static final String mCreateSuccess = "Detail Skema created successfully";
7     public static final String mCreateFailed = "Failed to create Detail Skema";
8     public static final String mUpdateSuccess = "Detail Skema updated successfully";
9     public static final String mUpdateFailed = "Failed to update Detail Skema";
10    public static final String mDeleteSuccess = "Detail Skema deleted successfully";
11    public static final String mDeleteFailed = "Failed to delete Detail Skema";
12 }

```

11. Selanjutnya, bukalah DetailRepository.java, disini kita akan membuat DetailSkemaRepository.java. Seperti yang dijelaskan di sebelumnya, disini kita mengextends JpaRepository<Tabel, Primary Key> dimana dalam kasus ini berupa tabel DetailSkema dan menggunakan primary key DetailSkemaPK

```

11 @Repository
12 public interface DetailSkemaRepository extends JpaRepository<DetailSkema, DetailSkemaPK> {
13     1 usage
14     @Procedure(procedureName = "get_active_detail_skema")
15     List<DetailSkema> findByStatus(int status);
16 }

```

12. Disini kita menggunakan @Procedure yang menandakan bahwa kita memanggil stored procedure yang bernama get_active_detail_skema.
13. Selanjutnya, buatlah sebuah DetailSkemaDao.java, deklarasikan beberapa fungsi.

```

1 package id.co.prg7_sertifikasi.dao;
2
3 import id.co.prg7_sertifikasi.vo.DetailSkemaVo;
4
5 import java.util.List;
6
7 public interface DetailSkemaDao {
8     2 usages 1 implementation
9     List<DetailSkemaVo> getAllDetailSkemas();
10    2 usages 1 implementation
11    List<DetailSkemaVo> getDetailSkemaActive();
12 }

```

14. Selanjutnya, didalam DetailSkemaDaoImpl.java, implementasikan method-method tersebut.

```

17  @Repository
18  public class DetailSkemaDaoImpl implements DetailSkemaDao {
19      2 usages
20      @Autowired
21      private DetailSkemaRepository detailSkemaRepository;
22      2 usages
23      @Autowired
24      private ProdiRepository prodiRepository;
25
26      2 usages
27      @Override
28      public List<DetailSkemaVo> getAllDetailSkemas() {
29          Iterable<DetailSkema> detailSkemas = detailSkemaRepository.findAll();
30          List<DetailSkemaVo> detailSkemaVos = new ArrayList<>();
31          for (DetailSkema item: detailSkemas) {
32              DetailSkemaVo detailSkemaVo = new DetailSkemaVo(item);
33
34              Skema skema = skemaRepository.findById(detailSkemaVo.getSkmId()).orElseThrow();
35              detailSkemaVo.setSkmDesc(skema.getSkm_nama());
36              Prodi prodi = prodiRepository.findById(detailSkemaVo.getProId()).orElseThrow();
37              detailSkemaVo.setProDesc(prodi.getPro_nama());
38
39              detailSkemaVos.add(detailSkemaVo);
40          }
41          return detailSkemaVos;
42      }
43  }

```

```

43  @Override
44  public List<DetailSkemaVo> getDetailSkemaActive() {
45      Iterable<DetailSkema> detailSkemas = detailSkemaRepository.findByStatus(1);
46      List<DetailSkemaVo> detailSkemaVos = new ArrayList<>();
47      for (DetailSkema item: detailSkemas) {
48          DetailSkemaVo detailSkemaVo = new DetailSkemaVo(item);
49
50          Skema skema = skemaRepository.findById(detailSkemaVo.getSkmId()).orElseThrow();
51          detailSkemaVo.setSkmDesc(skema.getSkm_nama());
52          Prodi prodi = prodiRepository.findById(detailSkemaVo.getProId()).orElseThrow();
53          detailSkemaVo.setProDesc(prodi.getPro_nama());
54
55          detailSkemaVos.add(detailSkemaVo);
56      }
57      return detailSkemaVos;
58  }
59  }

```

15. Selanjutnya, buatlah DetailSkemaService.java, deklarasikan beberapa fungsi dibawah ini

```

9  public interface DetailSkemaService {
10      1 usage 1 implementation
11      DtoResponse getAllDetailSkemas();
12      1 usage 1 implementation
13      DtoResponse getDetailSkemaActive();
14  }

```

16. Selanjutnya, didalam DetailSkemaServiceImpl.java, tambahkan method-method dibawah ini

```

16  @Service
17  @Transactional
18  public class DetailSkemaServiceImpl implements DetailSkemaService {
19      4 usages
20      @Autowired
21      private DetailSkemaDao detailSkemaDao;
22      4 usages
23      @Autowired
24      private DetailSkemaRepository detailSkemaRepository;
25      1 usage
26      @Override
27      public DtoResponse getAllDetailSkemas() {
28          if (detailSkemaDao.getAllDetailSkemas() != null) {
29              return new DtoResponse( status: 200, detailSkemaDao.getAllDetailSkemas());
30          }
31          return new DtoResponse( status: 200, data: null, mEmptyData);
32      }
33      1 usage
34      @Override
35      public DtoResponse getDetailSkemaActive() {
36          if (detailSkemaDao.getDetailSkemaActive() != null) {
37              return new DtoResponse( status: 200, detailSkemaDao.getDetailSkemaActive());
38          }
39          return new DtoResponse( status: 200, data: null, mEmptyData);
40      }

```

17. Selanjutnya, tambahkan DetailSkemaRest.java, disini tambahkan kode seperti dibawah ini

```

8  @RestController
9  @RequestMapping("/detailskemas")
10 public class DetailSkemaRest {
11     3 usages
12     @Autowired
13     private DetailSkemaService detailSkemaService;
14     public DetailSkemaRest(DetailSkemaService detailSkemaService) { this.detailSkemaService = detailSkemaService; }
15
16     @GetMapping("/getDetailSkemas")
17     public DtoResponse getDetailSkemas() { return detailSkemaService.getAllDetailSkemas(); }
18
19     @GetMapping("/getDetailSkemaActive")
20     public DtoResponse getDetailSkemaActive() { return detailSkemaService.getDetailSkemaActive(); }
21
22 }

```

18. Cobalah untuk menjalankan fungsi-fungsi tersebut dalam postman. Dalam hal ini kita telah berhasil mengambil data dari gabungan DetailSkema dan DetailSkemaPK


```
GET http://localhost:8080/detailskemas/getDetailSkemas Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Body 200 OK 73 ms 463 B Save as example

Pretty Raw Preview Visualize JSON

1 {
2   "status": 200,
3   "data": [
4     {
5       "dskId": 1,
6       "skmId": 1,
7       "skmDesc": "Programmer",
8       "proId": 1,
9       "proDesc": "Manajemen Informatika",
10      "tanggalMulai": "2024-03-17T00:00:00.000+00:00",
11      "tanggalBerakhir": "2024-03-27T00:00:00.000+00:00",
12      "totalPeserta": 10,
13      "kompeten": 0,
14      "belumKompeten": 0,
15      "tidakHadir": 0,
16      "status": 1
17    }
18  ],
19  "message": null
20 }
```

19. Dalam `getDetailSkemaActive`, kita juga berhasil memanggil melalui `@Procedure`

```
GET http://localhost:8080/detailskemas/getDetailSkemaActive Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Body 200 OK 40 ms 463 B Save as example

Pretty Raw Preview Visualize JSON

1 {
2   "status": 200,
3   "data": [
4     {
5       "dskId": 1,
6       "skmId": 1,
7       "skmDesc": "Programmer",
8       "proId": 1,
9       "proDesc": "Manajemen Informatika",
10      "tanggalMulai": "2024-03-17T00:00:00.000+00:00",
11      "tanggalBerakhir": "2024-03-27T00:00:00.000+00:00",
12      "totalPeserta": 10,
13      "kompeten": 0,
14      "belumKompeten": 0,
15      "tidakHadir": 0,
16      "status": 1
17    }
18  ],
19  "message": null
20 }
```

20. Dalam menambahkan data, kita perlu beberapa modifikasi dan logika dalam DetailSkemaServiceImpl. Langkah yang pertama yang harus dilakukan adalah menambahkan method dalam DetailSkemaService.java

```
12 1 implementation
    DtoResponse saveDetailSkema(DetailSkemaVoForm detailSkemaVoForm);
13 1 implementation
    DtoResponse updateDetailSkema(DetailSkemaVoForm detailSkemaVoForm);
14 1 implementation
    DtoResponse deleteDetailSkema(DetailSkemaVoForm detailSkemaVoForm);
```

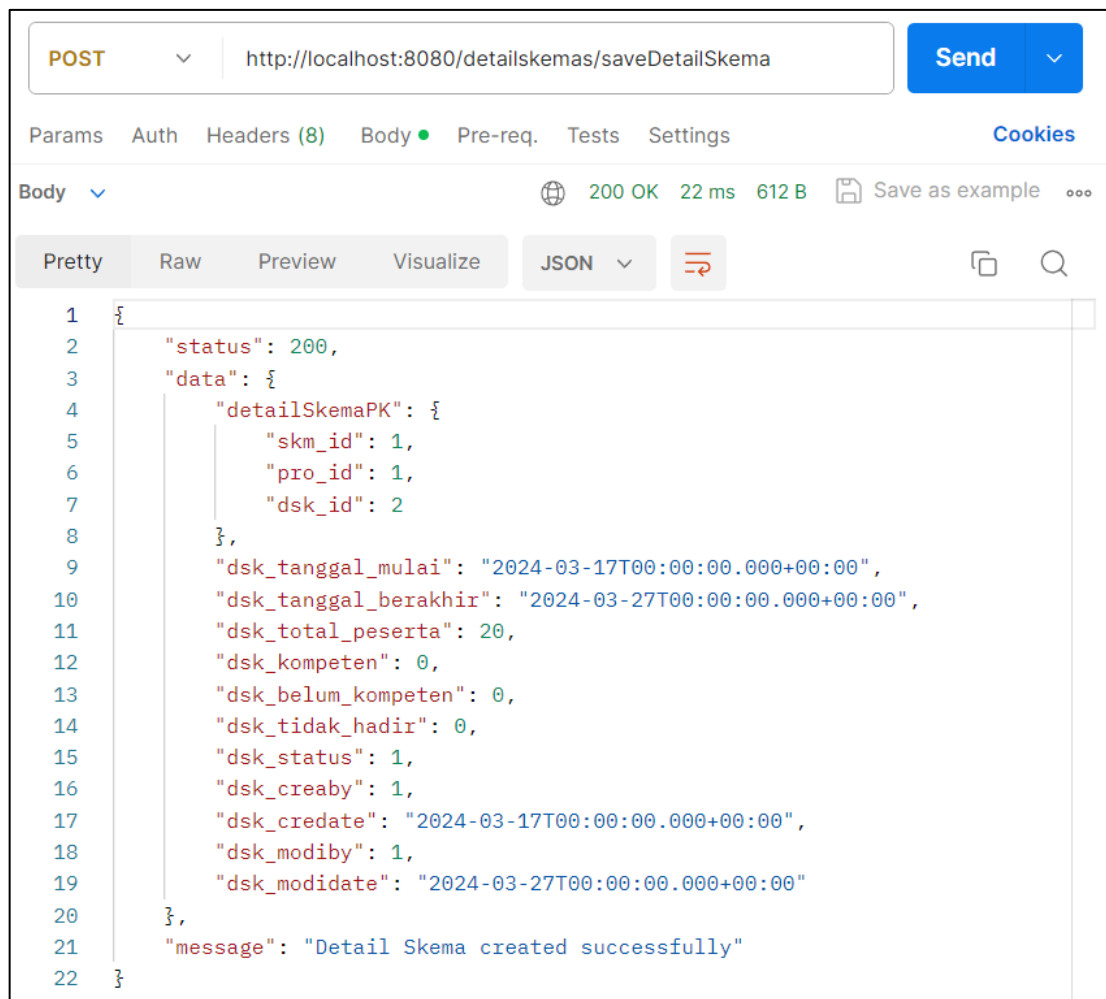
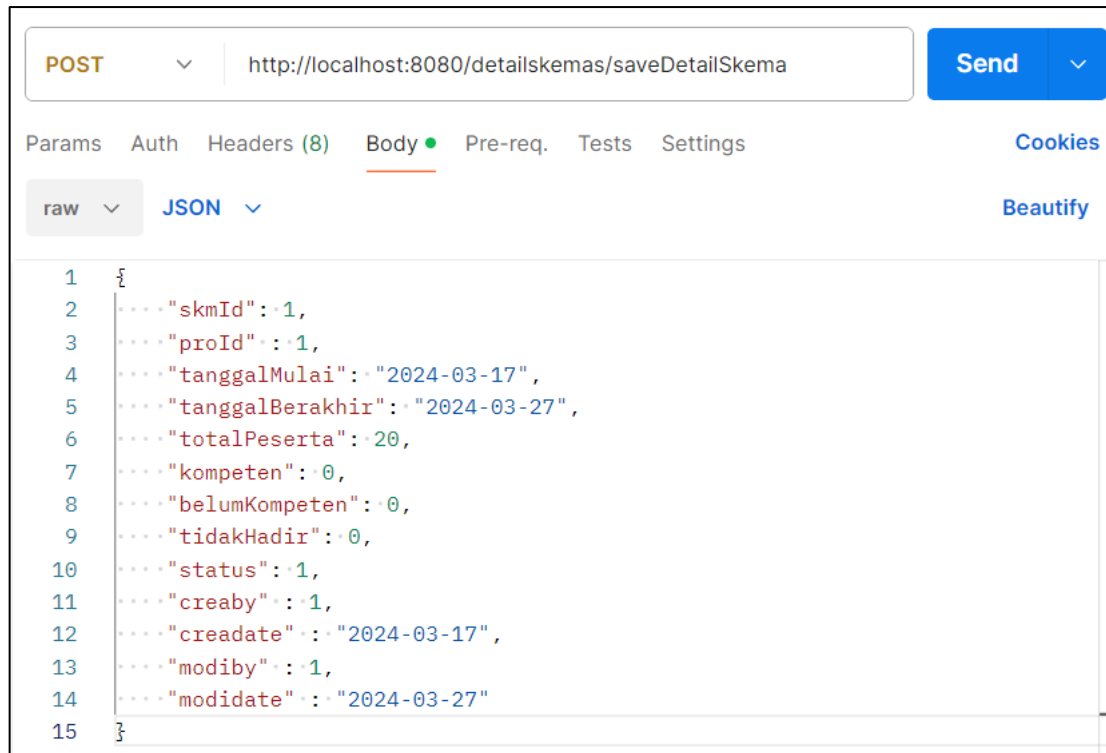
21. Implementasikan didalam DetailSkemaServiceImpl dengan kode seperti dibawah ini

```
43 @Override
44 public DtoResponse saveDetailSkema(DetailSkemaVoForm detailSkemaVoForm) {
45     try{
46         DetailSkemaPK detailSkemaPK = new DetailSkemaPK();
47         detailSkemaPK.setPro_id(detailSkemaVoForm.getProId());
48         detailSkemaPK.setSkm_id(detailSkemaVoForm.getSkmId());
49         List<DetailSkemaVo> listData = detailSkemaDao.getAllDetailSkemas();
50         if(listData.isEmpty()) {
51             detailSkemaPK.setDsk_id(1);
52         } else {
53             DetailSkemaVo lastData = listData.get(listData.size() - 1);
54             detailSkemaPK.setDsk_id(lastData.getDskId()+1);
55         }
56
57         DetailSkema detailSkema = new DetailSkema();
58         detailSkema.setDetailSkemaPK(detailSkemaPK);
59         detailSkema.setDsk_tanggal_mulai(detailSkemaVoForm.getTanggalMulai());
60         detailSkema.setDsk_tanggal_berakhir(detailSkemaVoForm.getTanggalBerakhir());
61         detailSkema.setDsk_total_peserta(detailSkemaVoForm.getTotalPeserta());
62         detailSkema.setDsk_kompeten(detailSkemaVoForm.getKompeten());
63         detailSkema.setDsk_belum_kompeten(detailSkemaVoForm.getBelumKompeten());
64         detailSkema.setDsk_tidak_hadir(detailSkemaVoForm.getTidakHadir());
65         detailSkema.setDsk_status(detailSkemaVoForm.getStatus());
66         detailSkema.setDsk_creaby(detailSkemaVoForm.getCreaby());
67         detailSkema.setDsk_credate(detailSkemaVoForm.getCredate());
68         detailSkema.setDsk_modiby(detailSkemaVoForm.getModiby());
69         detailSkema.setDsk_modidate(detailSkemaVoForm.getModidate());
70
71         id.co.prg7_sertifikasi.model.DetailSkema detailSkema = detailSkemaRepository.save(detailSkema);
72         return new DtoResponse( status: 200, detailSkema, mCreateSuccess);
73
74     }catch (Exception e){
75         return new DtoResponse( status: 500, detailSkemaVoForm, mCreateFailed);
76     }
77 }
```

22. Dalam method save ini kita perlu membuat instantiasi kelas DetailSkemaPK terlebih dahulu, lalu kita isikan datanya berdasarkan dari parameter yang ada, setelah itu baru dimasukkan kedalam DetailSkema.
23. Selanjutnya, tambahkan method di DetailSkemaRest.java

```
33 @PostMapping("/saveDetailSkema")
34 public DtoResponse createDetailSkema(@RequestBody DetailSkemaVoForm detailSkemaVoForm) {
35     return detailSkemaService.saveDetailSkema(detailSkemaVoForm);
36 }
```

24. Cobalah melakukan inputan dari postman



25. Kita akan mencoba untuk membuat bagian edit dan delete, tambahkan kode di DetailSkemaServiceImpl seperti dibawah ini

```

@Override
public DtoResponse updateDetailSkema(DetailSkemaVoForm detailSkemaVoForm) {
    try{
        DetailSkemaPK detailSkemaPK = new DetailSkemaPK();
        detailSkemaPK.setPro_id(detailSkemaVoForm.getProId());
        detailSkemaPK.setSkem_id(detailSkemaVoForm.getSkemId());
        detailSkemaPK.setDsk_id(detailSkemaVoForm.getDskId());

        DetailSkema detailSkema = new DetailSkema();
        detailSkema.setDetailSkemaPK(detailSkemaPK);
        detailSkema.setDsk_tanggal_mulai(detailSkemaVoForm.getTanggalMulai());
        detailSkema.setDsk_tanggal_berakhir(detailSkemaVoForm.getTanggalBerakhir());
        detailSkema.setDsk_total_peserta(detailSkemaVoForm.getTotalPeserta());
        detailSkema.setDsk_kompeten(detailSkemaVoForm.getKompeten());
        detailSkema.setDsk_belum_kompeten(detailSkemaVoForm.getBelumKompeten());
        detailSkema.setDsk_tidak_hadir(detailSkemaVoForm.getTidakHadir());
        detailSkema.setDsk_status(detailSkemaVoForm.getStatus());
        detailSkema.setDsk_creaby(detailSkemaVoForm.getCreaby());
        detailSkema.setDsk_creadate(detailSkemaVoForm.getCreadate());
        detailSkema.setDsk_modiby(detailSkemaVoForm.getModiby());
        detailSkema.setDsk_modidate(detailSkemaVoForm.getModidate());

        DetailSkema updatedDetailSkema = detailSkemaRepository.save(detailSkema);
        if(updatedDetailSkema != null){
            return new DtoResponse( status: 200, updatedDetailSkema, mUpdateSuccess);
        } else {
            return new DtoResponse( status: 404, data: null, mNotFound);
        }
    } catch (Exception e){
        return new DtoResponse( status: 500, data: null, mUpdateFailed);
    }
}

```

26. Untuk delete, tambahkan kode seperti dibawah ini

```

@Override
public DtoResponse deleteDetailSkema(DetailSkemaVoForm detailSkemaVoForm) {
    DetailSkemaPK detailSkemaPK = new DetailSkemaPK();
    detailSkemaPK.setPro_id(detailSkemaVoForm.getProId());
    detailSkemaPK.setSkem_id(detailSkemaVoForm.getSkemId());
    detailSkemaPK.setDsk_id(detailSkemaVoForm.getDskId());

    DetailSkema detailSkemaData = detailSkemaRepository.findById(detailSkemaPK).orElseThrow();
    if(detailSkemaData != null){
        try {
            detailSkemaRepository.delete(detailSkemaData);
            return new DtoResponse( status: 200, detailSkemaVoForm, mDeleteSuccess);
        } catch (Exception e){
            return new DtoResponse( status: 500, detailSkemaVoForm, mDeleteFailed);
        }
    }

    return new DtoResponse( status: 404, data: null, mNotFound);
}

```

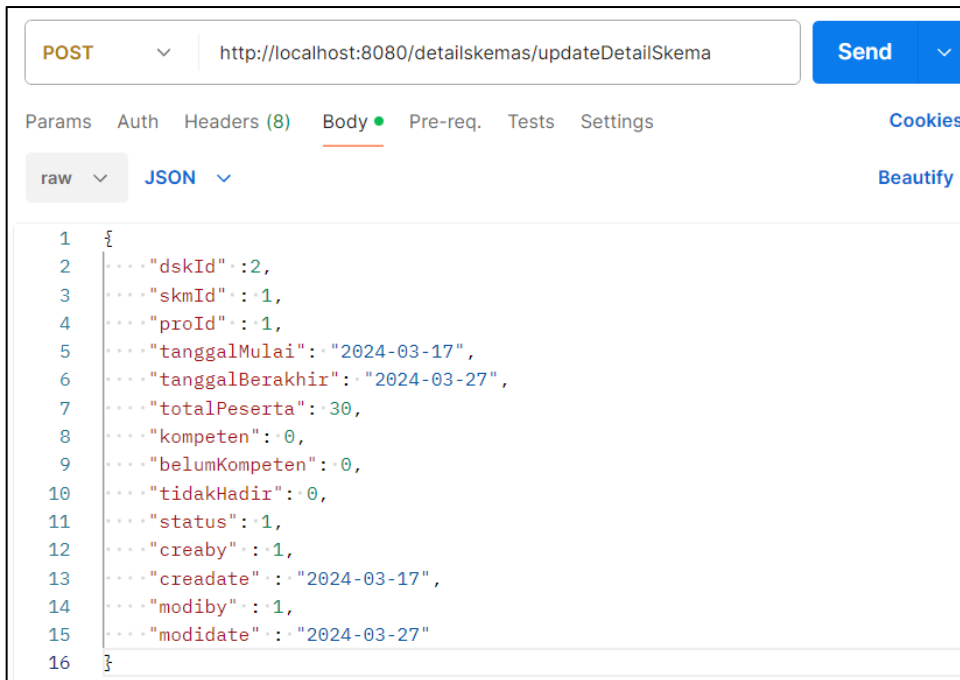
27. Selanjutnya tambahkan dalam DetailSkemaRest kode untuk update dan delete

```

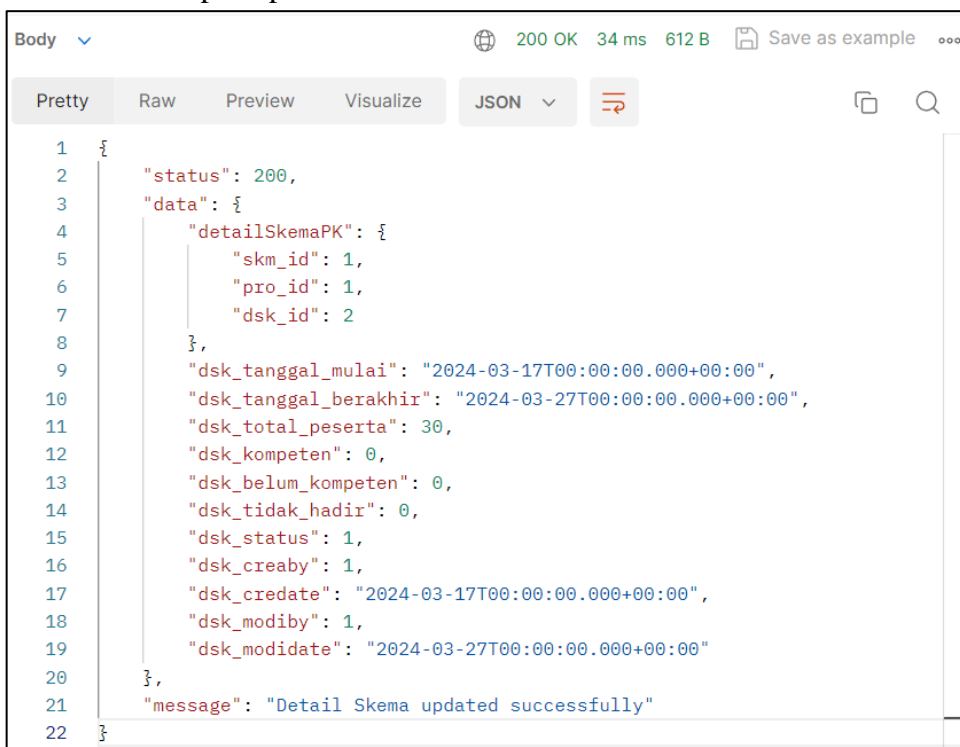
38     @PostMapping("/updateDetailSkema")
39     public DtoResponse updateProdi(@RequestBody DetailSkemaVoForm detailSkemaVoForm) {
40         return detailSkemaService.updateDetailSkema(detailSkemaVoForm);
41     }
42
43     @PostMapping("/deleteDetailSkema")
44     public DtoResponse deleteDetailSkema(@RequestBody DetailSkemaVoForm detailSkemaVoForm) {
45         return detailSkemaService.deleteDetailSkema(detailSkemaVoForm);
46     }

```

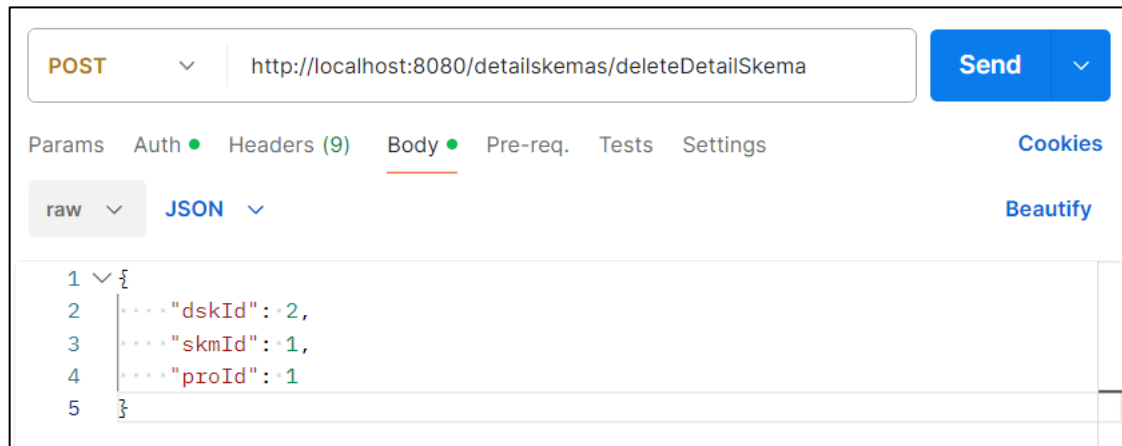
28. Coba jalankan untuk update di dalam Postman



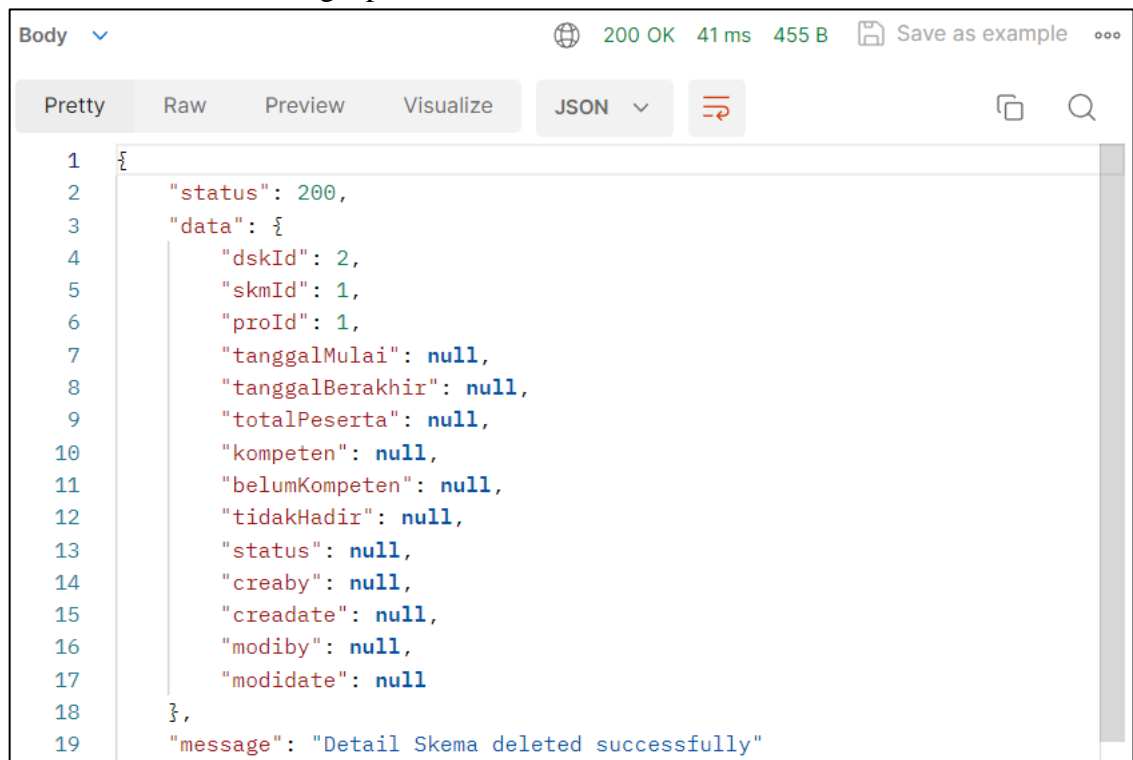
29. Maka akan tampil seperti dibawah ini



30. Selanjutnya, untuk delete cobalah kembali untuk menghapus salah satu data

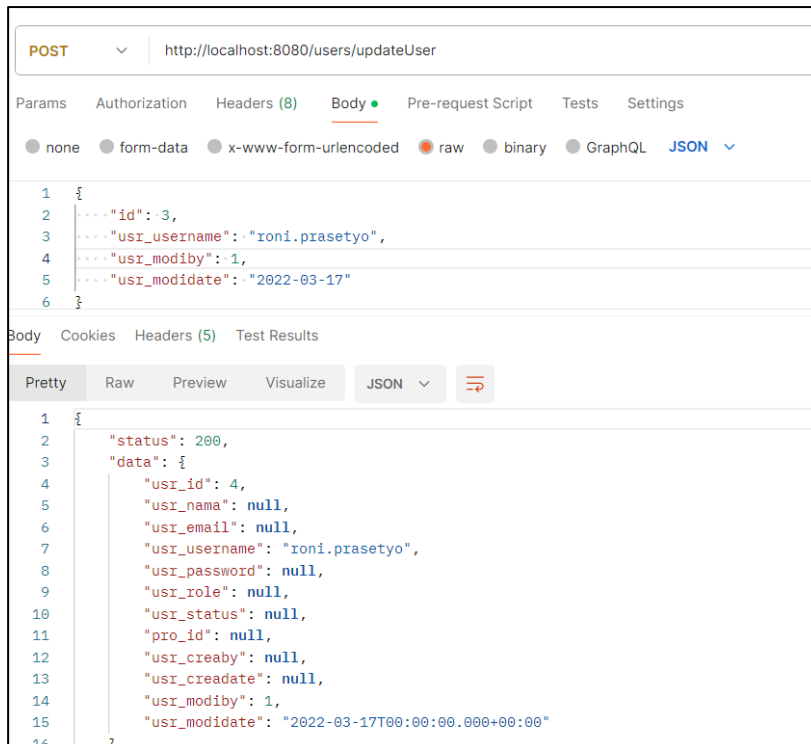


31. Maka akan berhasil menghapus salah satu data



5. LATIHAN

1. Pada saat melakukan update data, apabila kita hanya menginputkan id dan juga data yang ingin diupdate, maka data yang tidak ingin kita update menjadi null. Didatabase pun juga akan berubah menjadi null.



```

POST http://localhost:8080/users/updateUser

{
  "id": 3,
  "usr_username": "roni.prasetyo",
  "usr_modiby": 1,
  "usr_modidate": "2022-03-17"
}

{
  "status": 200,
  "data": {
    "usr_id": 4,
    "usr_name": null,
    "usr_email": null,
    "usr_username": "roni.prasetyo",
    "usr_password": null,
    "usr_role": null,
    "usr_status": null,
    "pro_id": null,
    "usr_creaby": null,
    "usr_creadate": null,
    "usr_modiby": 1,
    "usr_modidate": "2022-03-17T00:00:00.000+00:00"
  }
}

```

- Ubahlah untuk fungsi setiap update menjadi menanggulangi kasus ini dimana apabila kita hanya menginputkan 4 field, maka hanya 4 field itu saja yang akan diupdate, untuk yang lainnya akan mempertahankan data yang lama.
- Tambahkan validasi apabila kita menambahkan detail skema, maka perlu mengecek skm_id dan pro_id nya apakah ada didalam tabel skema dan prodi atau tidak. Jika tidak maka mengembalikan pesan “Scheme or Study Program is invalid”.
- Buatlah CR untuk tabel dengan nama stf_trsertifikasi.

| UPTIF-013.DB_Sertifi...bo.stf_trsertifikasi* | | | |
|--|--------------|-----------|-------------------------------------|
| | Column Name | Data Type | Allow Nulls |
| 🔑 | tsk_id | int | <input type="checkbox"/> |
| 🔑 | usr_id | int | <input type="checkbox"/> |
| 🔑 | dsk_id | int | <input type="checkbox"/> |
| | tsk_status | int | <input checked="" type="checkbox"/> |
| | tsk_creaby | int | <input checked="" type="checkbox"/> |
| | tsk_creadate | date | <input checked="" type="checkbox"/> |
| | tsk_modiby | int | <input checked="" type="checkbox"/> |
| | tsk_modidate | date | <input checked="" type="checkbox"/> |
| ▶ | | | <input type="checkbox"/> |