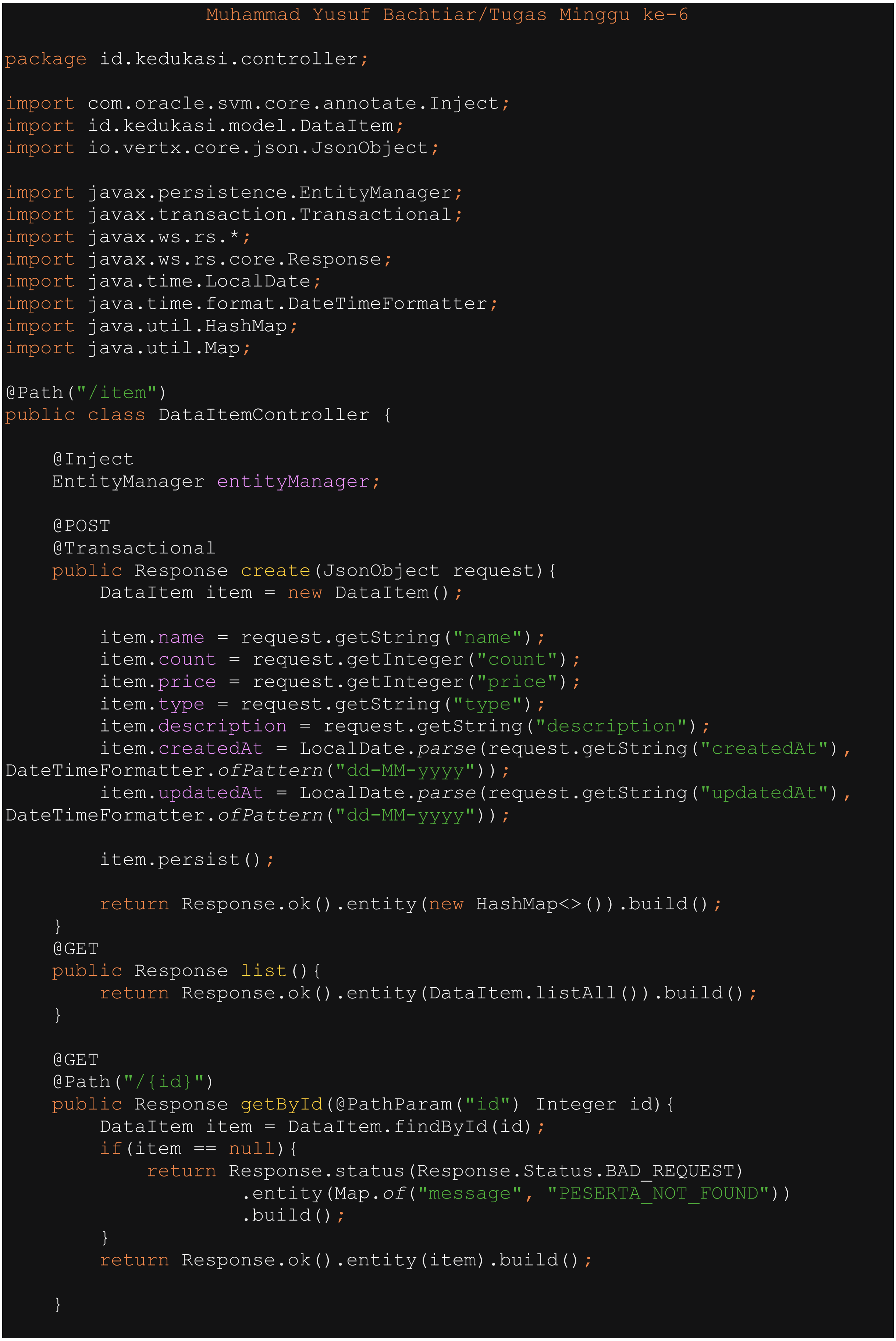
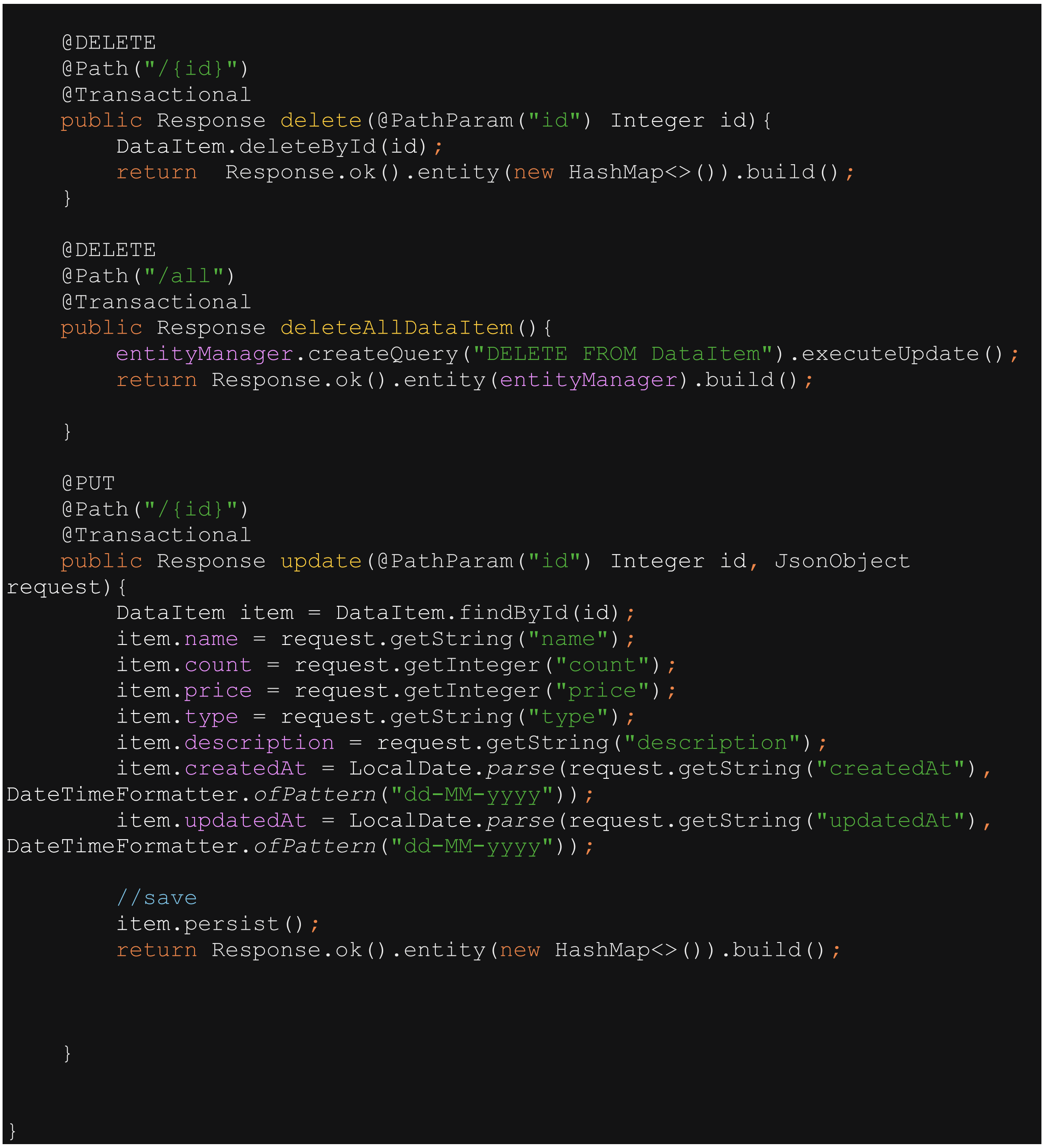
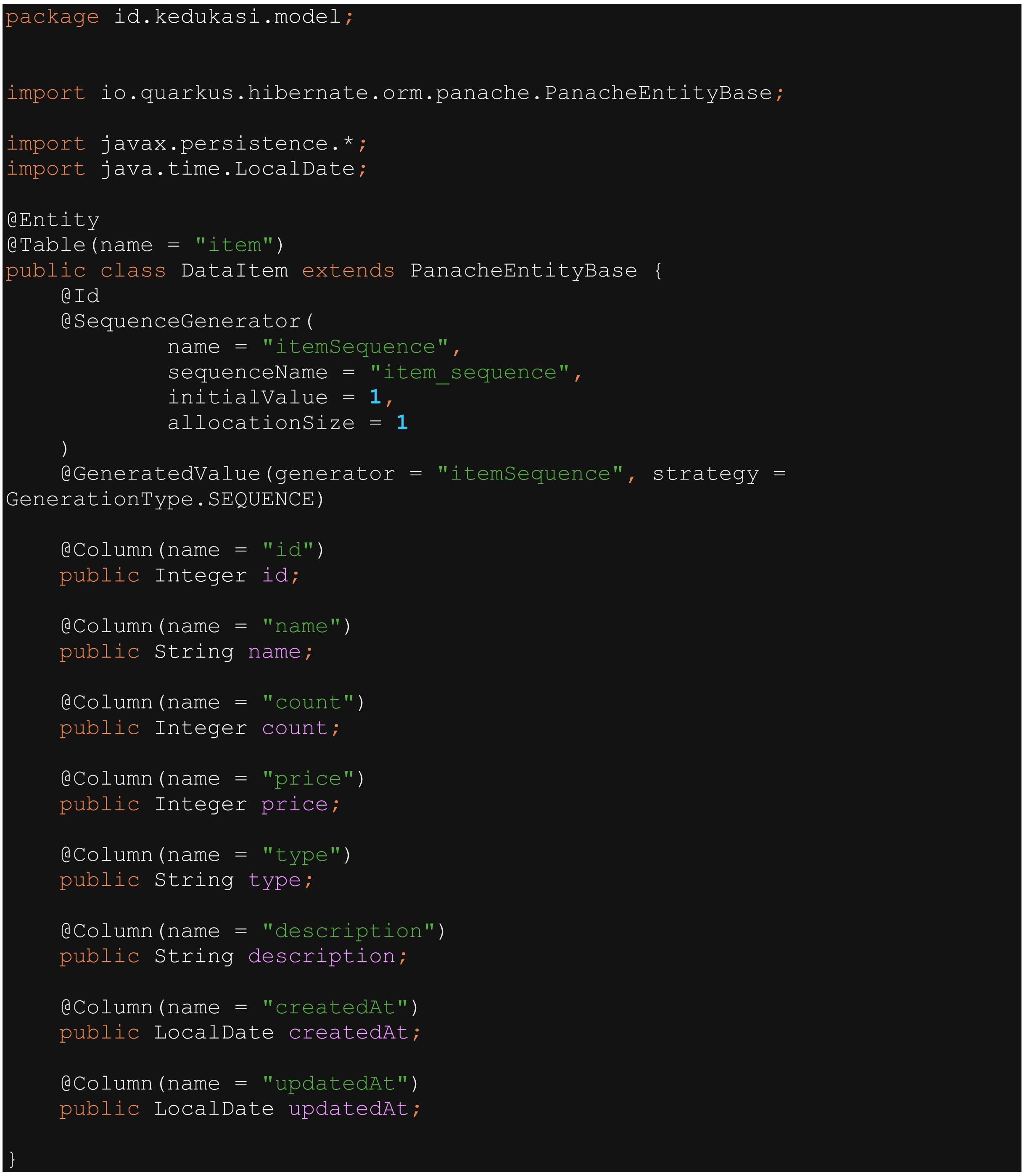


```
                 Muhammad Yusuf Bachtiar/Tugas Minggu ke-6

package id.kedukasi.controller;

import com.oracle.svm.core.annotate.Inject;
import id.kedukasi.model.DataItem;
import io.vertx.core.json.JsonObject;

import javax.persistence.EntityManager;
import javax.transaction.Transactional;
import javax.ws.rs.*;
import javax.ws.rs.core.Response;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.HashMap;
import java.util.Map;

@Path("/item")
public class DataItemController {

    @Inject
    EntityManager entityManager;

    @POST
    @Transactional
    public Response create(JsonObject request){
        DataItem item = new DataItem();

        item.name = request.getString("name");
        item.count = request.getInteger("count");
        item.price = request.getInteger("price");
        item.type = request.getString("type");
        item.description = request.getString("description");
        item.createdAt = LocalDate.parse(request.getString("createdAt"),
DateTimeFormatter.ofPattern("dd-MM-yyyy"));
        item.updatedAt = LocalDate.parse(request.getString("updatedAt"),
DateTimeFormatter.ofPattern("dd-MM-yyyy"));

        item.persist();

        return Response.ok().entity(new HashMap<>()).build();
    }
    @GET
    public Response list(){
        return Response.ok().entity(DataItem.listAll()).build();
    }

    @GET
    @Path("/{id}")
    public Response getById(@PathParam("id") Integer id){
        DataItem item = DataItem.findById(id);
        if(item == null){
            return Response.status(Response.Status.BAD_REQUEST)
                    .entity(Map.of("message", "PESERTA_NOT_FOUND"))
                    .build();
        }
        return Response.ok().entity(item).build();

    }
```



```
    @DELETE
    @Path("/{id}")
    @Transactional
    public Response delete(@PathParam("id") Integer id){
        DataItem.deleteById(id);
        return  Response.ok().entity(new HashMap<>()).build();
    }

    @DELETE
    @Path("/all")
    @Transactional
    public Response deleteAllDataItem(){
        entityManager.createQuery("DELETE FROM DataItem").executeUpdate();
        return Response.ok().entity(entityManager).build();

    }

    @PUT
    @Path("/{id}")
    @Transactional
    public Response update(@PathParam("id") Integer id, JsonObject
request){
        DataItem item = DataItem.findById(id);
        item.name = request.getString("name");
        item.count = request.getInteger("count");
        item.price = request.getInteger("price");
        item.type = request.getString("type");
        item.description = request.getString("description");
        item.createdAt = LocalDate.parse(request.getString("createdAt"),
DateTimeFormatter.ofPattern("dd-MM-yyyy"));
        item.updatedAt = LocalDate.parse(request.getString("updatedAt"),
DateTimeFormatter.ofPattern("dd-MM-yyyy"));

        //save
        item.persist();
        return Response.ok().entity(new HashMap<>()).build();



    }


}
```



```java
package id.kedukasi.model;


import io.quarkus.hibernate.orm.panache.PanacheEntityBase;

import javax.persistence.*;
import java.time.LocalDate;

@Entity
@Table(name = "item")
public class DataItem extends PanacheEntityBase {
    @Id
    @SequenceGenerator(
            name = "itemSequence",
            sequenceName = "item_sequence",
            initialValue = 1,
            allocationSize = 1
    )
    @GeneratedValue(generator = "itemSequence", strategy =
GenerationType.SEQUENCE)

    @Column(name = "id")
    public Integer id;

    @Column(name = "name")
    public String name;

    @Column(name = "count")
    public Integer count;

    @Column(name = "price")
    public Integer price;

    @Column(name = "type")
    public String type;

    @Column(name = "description")
    public String description;

    @Column(name = "createdAt")
    public LocalDate createdAt;

    @Column(name = "updatedAt")
    public LocalDate updatedAt;

}
```

```
package id.kedukasi;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/hello")
public class GreetingResource {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String hello() {
        return "Hello RESTEasy";
    }
}
```

```
package id.kedukasi;

import io.quarkus.test.junit.QuarkusIntegrationTest;

@QuarkusIntegrationTest
public class GreetingResourceIT extends GreetingResourceTest {
    // Execute the same tests but in packaged mode.
}
```

```
package id.kedukasi;

import io.quarkus.test.junit.QuarkusTest;
import org.junit.jupiter.api.Test;

import static io.restassured.RestAssured.given;
import static org.hamcrest.CoreMatchers.is;

@QuarkusTest
public class GreetingResourceTest {

    @Test
    public void testHelloEndpoint() {
        given()
          .when().get("/hello")
          .then()
             .statusCode(200)
             .body(is("Hello RESTEasy"));
    }

}
```

POST: http://localhost:9090/item

GET http://localhost:9090/item

GET http://localhost:9090/item/2

PUT http://localhost:9090/item/2

DELETE http://localhost:9090/item/2