

```

import mongoose from "mongoose";

const Schema = mongoose.Schema;

const InformationSchema = new Schema({
  banner: [
    {
      banner_name: { type: String },
      banner_image: { type: String },
      description: { type: String },
    },
  ],
  service: [
    {
      service_code: { type: String },
      service_name: { type: String },
      service_icon: { type: String },
      service_tariff: { type: Number },
    },
  ],
});

5 references
export default mongoose.model("Informations",
InformationSchema);

```

```

import mongoose from "mongoose";

const Schema = mongoose.Schema;

const MembershipSchema = new Schema({
  email: String,
  first_name: String,
  last_name: String,
  password: String,
  profile_image: {
    type: String,
    default:
      "https://res.cloudinary.com/dymv3cmhq/image/upload/v1674914217/cld-sample-5.jpg",
  },
  balance: { type: Number, default: 0 },
});

12 references
export default mongoose.model("Membership",
MembershipSchema);

```

```

import mongoose from "mongoose";

const Schema = mongoose.Schema;

const TransactionSchema = new Schema({
  nameTransaksionMember: { type: Schema.Types.ObjectId,
ref: "Membership" },
  produkServices: { type: Schema.Types.ObjectId, ref:
"Informations" },
  record: [
    {
      invoice_number: String,
      transaction_type: String,
      description: String,
      total_amount: Number,
      created_on: Date,
    },
  ],
});

5 references
export default mongoose.model("Transaction",
TransactionSchema);

```

1. Ada 3 shema yaitu MembershipSchema, TransactionSchema dan InformationSchema.
2. TransactionSchema mempunyai 2 relasi, dan untuk mendapatkan data dari tiap-tiap relasi menggunakan `_id` yang sudah digenerate oleh mongodb dan mongoose secara otomatis.

```
nameTransaksiMember: { type: Schema.Types.ObjectId,
  ref: "Membership" },

produkServices: { type: Schema.Types.ObjectId, ref:
  "Informations" },
```

3. Cara menambah data shema membership registrasi:

```
await new shemaMembership({
  email,
  last_name,
  first_name,
  password: hash,
}).save();
```

4. Mencari data document shema membership by email

```
const getData = await shemaMembership.findOne(
  {
    email: req.userData.email,
  },
  "-password -_id -balance"
);
```

5. Update data document shema membership

```
filterData.first_name = await first_name;
filterData.last_name = await last_name;

await filterData.save();

res.status(200).json({
  status: 200,
  message: "Sukses",
  data: {
    first_name: filterData.first_name,
    last_name: filterData.last_name,
  },
});
```