### Repo Mini Projek gen sebelumnya

- Jeremy: <https://github.com/Jeremy250700/point-of-sale>

- Rizqi: <https://github.com/RizqiAhmadAbdillah/PointOfSalesBimBim>

### Wireframe

Freehand: <https://muhammadabidkhairy127182.invisionapp.com/freehand/Point-of-Sale-Mini-Project-Wireframe-G1Vs08FJo>

ERD Database:

<https://dbdiagram.io/d/Point-Of-Sale-65e9455ab1f3d4062c539a2e>

### Project Requirement

1. Halaman Order Produk

Halaman ini memiliki 2 bagian, yaitu Daftar Produk dan Daftar Pesanan

1. Bagian Daftar Produk memiliki fitur sebagai berikut:
2. Daftar produk yang menampilkan "image", "name", dan "price" produk
3. Ketika produk ditekan maka produk akan ditambahkan ke Daftar Pesanan
4. Produk dapat diurutkan berdasarkan "name"
5. Produk dapat diurutkan berdasarkan "price"
6. Produk dapat difilter berdasarkan “category”
7. Tersedia fitur pencarian produk berdasarkan "name"
8. Bagian Daftar Pesanan memiliki fitur sebagai berikut:
9. Daftar produk yang telah dipesan yang menampilkan "name", "quantity", dan "price" produk
10. Terdapat tombol Hapus pada setiap produk yang dipesan
11. Total harga dari semua pesanan
12. Tombol “Payment” yang ketika ditekan akan mengarah ke halaman Pembayaran
13. Halaman Pembayaran
14. Halaman ini menampilkan “total price” yang akan dibayar, Jumlah Uang yang konsumer beri “Amount of paid”, dan Jumlah Kembalian “change”, dan Tombol “Order”
15. Jumlah Kembalian berubah secara otomatis ketika Input Jumlah Uang Konsumer berubah
16. Tombol “order” dalam keadaan disabled jika total uang yang dibayarkan kurang dari total price
17. Ketika tombol “order” ditekan:
18. Mengirimkan data ke endpoint API transaksi
19. menampilkan pop up pembelian sukses jika berhasil dibuat transaksi baru
20. me-reset Daftar Pesanan dan mengembalikan ke halaman Order Produk
21. Halaman Riwayat Transaksi
22. Halaman ini menampilkan tabel riwayat transaksi yang berisi "transaction date", “transaction id”, "total price", “total paid”, dan "action" yang berisi tombol "transaction detail"
23. ketika tombol “transaction detail” ditekan, maka akan diarahkan ke halaman detail transaksi
24. Halaman Detail Transaksi
25. Menampilkan “transaction id”,”transaction date”, “total price”, “total paid” produk yang dibeli
26. Menampilkan detail dari transaksi yang berisi daftar produk yang dibeli
27. Modul Produk (CRUD)
28. Halaman List Product
29. Menampilkan daftar product dalam tabel dengan kolom “Product Id”, “Product Name”, “Price”, “Category”, dan “Action” yang berisi tombol “Detail”, “Edit” dan “Delete” produk
30. Terdapat tombol “Add Product” untuk menambahkan product baru
31. Halaman Add Product/Edit Product
32. Terdapat fitur untuk add product dengan field input “Product Name”, “Category, “Price”, “Image URL”, dan tombol “Add Product”
33. Terdapat validasi untuk tiap input field harus diisi (required)
34. Saat edit product, field input sudah terisi dengan data yang sudah ada dan tombolnya “Edit Product”
35. Halaman Detail Product
36. Menampilkan “Product Id”, “Product name”, “price”, “Image URL”, “Category id”, “category name”
37. Menampilkan gambar
38. Modul Category (CRUD)
39. Halaman list category
40. Menampilkan daftar category dalam tabel dengan kolom “Id Category”, “Category Name”, “Total related products”, “Action” yang berisi tombol “Detail”, “Edit” dan “Delete” category
41. Terdapat tombol “Add category” untuk menambahkan category baru
42. Category yang memiliki product tidak dapat didelete, hanya bisa diedit
43. Halaman add/edit category
44. Terdapat fitur untuk add category dengan field input “Category Name” dengan validasi harus diisi (required)
45. Saat edit field sudah terisi dengan data yang sudah ada
46. Halaman detail category
47. Menampilkan “category id”, “category name”, “total related products”

### Project References

* <https://gdm-catalog-fmapi-prod.imgix.net/ProductScreenshot/2f8634de-fff1-4b10-bfb7-b30c129b0778.png>
* <https://odoocdn.com/openerp_website/static/src/img/2016/pos/pos_screenshot_01.gif>
* <https://webkul.com/wp-content/uploads/2021/01/pos-hero-img-final.png>
* <https://gdm-catalog-fmapi-prod.imgix.net/ProductScreenshot/2d6c059a-d449-4cc5-873c-97a64495cba2.png>
* <https://image.winudf.com/v2/image1/Y29tLmR6LnllZXNjYW4ucG9pbnQub2Yuc2FsZS5wb3Nfc2NyZWVuXzBfMTU1NDMwNjczOF8wMzY/screen-0.webp?fakeurl=1&type=.webp>

### Todo

- buat trello

- masukin daftar fitur yang akan dibuat ke dalam backlog di trello

**Design API**

Product

1. List Produk

Method : GET

Path Default : localhost/pos/api/listproduct

( jika tidak ada parameter sort\_by dan sort\_order gunakan default sort dari Spring )

Path Sort :

localhost/pos/api/listproduct?sort\_by=title&sort\_order=asc

Path Search by title :

localhost/pos/api/listproduct?title=Capucino&sort\_by=title&sort\_order=asc

( searching query menggunakan like )

Request

Response Sukses

[

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

Id:

},

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

id:

},

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

id:

}

]

Response Failed

[

]

2. List Produk by Categori ( Gunakan API yang sama listproduk )

Method : GET

Path : localhost/pos/api/listproduct?category\_id=1

Request

Response Sukses

[

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

id

},

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

id

},

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

id

}

]

Response Failed

[

]

3. Add Product

Method : POST

Path : localhost/pos/api/addproduct

Request

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

}

Response

{

status : ok,

message : sucess

}

4. Update Product

Method : PUT

Path : localhost/pos/api/updateproduct/{id}

Request

{

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

}

Response

{

status : ok,

message : sucess

}

5. Delete Product

Method : DELETE

Path : localhost/pos/api/deleteproduct/{id}

Request

Response

{

status : ok,

message : sucess

}

6. Detail Product

Method : GET

Path : localhost/pos/api/detailproduct/{id}

Request

Response

{

id:

title: xxxx,

image: url ,

price: integerharga,

category\_id : xxxx

category\_name: xxx

}

7. Add Transaksi

Method : POST

Path : localhost/pos/api/addtransaction

Request

{

total\_amount: 2000,

total\_pay: 3000,

transaction\_details: [

{

product\_id: 4,

quantity: 5,

subtotal: 500

},

{

product\_id: 7,

quantity: 9,

subtotal: 1500

},

]

}

Response

{

status : ok,

message : sucess

}