

Laporan Development

Web Service

II 3160 Pemrograman Integratif

FoodfFinder



Dipersiapkan oleh

Ikhsan Widi Adyatma / 18215047

PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

2017

Daftar Isi

Daftar Isi	2
Bab 1 Pendahuluan	3
1. Latar Belakang	3
2. Deskripsi Sistem	3
3. Deskripsi API	4
Bab 2 Spesifikasi Kebutuhan Perangkat Lunak	6
1. Kebutuhan Fungsional.....	6
2. Kebutuhan Non-Fungsional	6
3. Hardware	6
4. Software	6
5. Infoware	7
Bab 3 Deskripsi Perancangan Perangkat Lunak	8
1. Lingkungan Implementasi	8
2. Use Case	8
3. Implementasi Web-service	9
Bab 4 Hasil Implementasi dan Pengujian Perangkat Lunak.....	18
1. Hasil Pengujian	18
2. Evaluasi Pengujian	20

Bab 1 Pendahuluan

1. Latar Belakang

Dalam berkegiatan sehari-hari di ITB, setiap civitas akademika di Institut Teknologi Bandung ini pada dasarnya memiliki kebutuhan primer yang sama, yaitu kebutuhan makanan. Dalam memenuhi kebutuhan ini, baik mahasiswa, dosen, ataupun civitas akademika lainnya di ITB memiliki cara mereka sendiri dalam memenuhi kebutuhannya. Sebagian diantara mereka memilih untuk membawa makanan sendiri dari rumah, sebagian lain memesan catering, dan sebagian lainnya memilih untuk makan di kantin-kantin yang tersedia di ITB.

Bagi mereka yang memilih untuk makan di kantin-kantin yang tersedia di ITB, tersedia sangat banyak pilihan tempat di kantin ITB sebagai tempat pilihan untuk memenuhi kebutuhan makanan tersebut. Tak jarang terjadi kondisi dimana kita merasa kebingungan harus makan dimana, apa saja menu yang tersedia di suatu kantin, dan juga masalah harga yang seringkali menjadi pertimbangan kita dalam memilih makanan. Akan tetapi, seringkali informasi-informasi mengenai makanan-makanan yang tersedia di suatu kantin, beserta harga-harga barang nya tidak kita ketahui, hingga kita tiba di kantin tersebut, menyebabkan kita akhirnya harus memilih makanan makanan seadanya yang berada di suatu kantin.

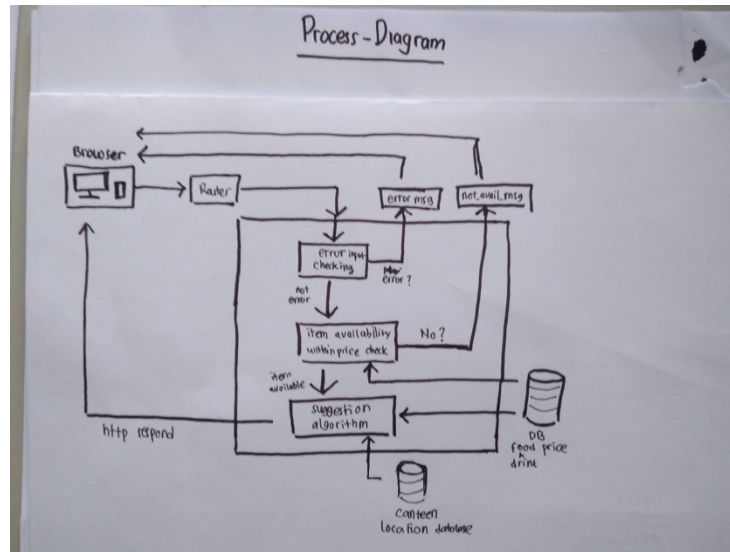
Berawal dari kebutuhan akan informasi mengenai makanan, beserta harga makanan dan lokasi makanan tersebut, saya terdorong untuk menciptakan solusi berupa webservice “*foodfinder*”, yang menyediakan kebutuhan atas data makanan dan minuman yang ada di ITB, sekaligus dapat memberikan saran pembelian makanan berdasarkan input harga yang dimasukkan oleh pengguna.

2. Deskripsi Sistem

"Food Finder" Merupakan sebuah web service yang menerima data berupa besaran uang dan mengembalikan data berupa suggestion barang yang dapat dibeli dengan sejumlah uang tersebut. Pengguna web service ini dapat mengakses layanan secara manual dengan menggunakan HTTP GET. Ide layanan "Food Finder" ini juga bersifat open, sehingga siapapun dapat menambahkan, dan menghapus data makanan yang ada supaya data yang berada dalam basis data selalu terbaru.

3. Deskripsi API

Berikut merupakan cara kerja dari *web service food finder* yang dibuat.



Web service FoodFinder ini dibangun dengan menggunakan RESTful API menggunakan Bahasa pemrograman Go versi 1.9. Berikut ini merupakan daftar Fungsi API yang terdapat dalam web service “*foodfinder*”.

Fungsi	Method	URL	Deskripsi
GetAllMakanan	GET	http://167.205.67.253/food	Mengembalikan data seluruh makanan yang ada di database
GetAllMinuman	GET	http://167.205.67.253/drink	Mengembalikan data seluruh minuman yang ada di database
GetAllCanteen	GET	http://167.205.67.253/canteen	Mengembalikan data seluruh canteen yang ada di database
GetMakanan	GET	<a href="http://167.205.67.253/food/<id>">http://167.205.67.253/food/<id>	Mengembalikan data 1 makanan yang ada di database sesuai dengan id
GetMinuman	GET	<a href="http://167.205.67.253/drink/<id>">http://167.205.67.253/drink/<id>	Mengembalikan data 1 minuman yang ada di database sesuai dengan id
GetCanteen	GET	<a href="http://167.205.67.253/canteen/<id>">http://167.205.67.253/canteen/<id>	Mengembalikan data 1 canteen yang ada di database sesuai dengan id
AddMakanan	POST	<a href="http://167.205.67.253/food/<id>/<nama_makanan>:<harga>/<nama_kantin>">http://167.205.67.253/food/<id>/<nama_makanan>:<harga>/<nama_kantin>	Menambahkan data 1 makanan ke dalam database
AddMinuman	POST	<a href="http://167.205.67.253/drink/<id>/<nama_minuman>:<harga>/<nama_kantin>">http://167.205.67.253/drink/<id>/<nama_minuman>:<harga>/<nama_kantin>	Menambahkan data 1 minuman ke dalam database
AddCanteen	POST	<a href="http://167.205.67.253/canteen/<id>/<nama_kantin>:<lokasi>">http://167.205.67.253/canteen/<id>/<nama_kantin>:<lokasi>	Menambahkan data 1 canteen ke dalam database
DeleteFood	DELETE	<a href="http://167.205.67.253/food/<id>">http://167.205.67.253/food/<id>	Menghapus data 1 makanan dari database

Fungsi	Method	URL	Deskripsi
DeleteDrink	DELETE	<a href="http://167.205.67.253/drink/<id>">http://167.205.67.253/drink/<id>	Menghapus data 1 minuman dari database
DeleteCanteen	DELETE	<a href="http://167.205.67.253/canteen/<id>">http://167.205.67.253/canteen/<id>	Menghapus data 1 canteen dari database
SuggestFood	GET	<a href="http://167.205.67.253/SuggestFood/<price>">http://167.205.67.253/SuggestFood/<price>	Mengembalikan data berupa saran makanan yang dapat dibeli dengan harga yang dimasukkan user
SuggestDrink	GET	<a href="http://167.205.67.253/SuggestDrink/<price>">http://167.205.67.253/SuggestDrink/<price>	Mengembalikan data berupa saran minuman yang dapat dibeli dengan harga yang dimasukkan user

Bab 2 Spesifikasi Kebutuhan Perangkat Lunak

1. Kebutuhan Fungsional

Berikut merupakan kebutuhan fungsional dari web service “food finder”.

- Web Service mampu memberikan saran pembelian barang sesuai dengan jumlah uang yang dimiliki oleh user
- Web Service mampu untuk memberikan saran pembelian makanan + minuman (lebih dari 1 barang)
- Web service menampilkan pesan error jika masukan tidak sesuai
- Web service menampilkan pesan tidak tersedia jika ternyata tidak terdapat makanan/minuman yang dapat dibeli sesuai jumlah uang pengguna

2. Kebutuhan Non-Fungsional

Berikut merupakan kebutuhan non-fungsional dari web service “foodfinder”.

- Software Web service harus mampu untuk memproses request kurang dari 5 detik
- Database mampu menyimpan data untuk seluruh kantin ITB

3. Hardware

Kebutuhan terkait perangkat keras pada layanan “food finder” dapat dibedakan menjadi dua jenis, yaitu dari sisi *client* dan dari sisi *server*. Berikut merupakan kebutuhan perangkat keras dari *client* & *server*.

- Server (http server & host database)
 - o CPU Clockspeed min 1,6 Ghz Dual Core (2 Ghz or more is recommended)
 - o RAM Min. 4 GB
 - o Disk Space min. 40 GB
- Client
 - o CPU Clockspeed min 1 Ghz Dual Core
 - o RAM min. 2 GB (4 GB Recommended)
 - o Video Card min. 16 bit color video, monitor capable of 800x600 resolution

4. Software

Berikut merupakan perangkat lunak yang dibutuhkan dalam membangun *web service* “food finder”.

- Client
 - o OS dengan GUI (MacOS, Windows, Linux, dll)
 - o Browser (Chrome, Safari, Firefox, Opera, dll)
- Server
 - o OS Server (Windows server, linux server)
- Development
 - o Database : MySQL / PostgreSQL
 - o Programming language : GoLang
 - o IDE/Text Editor : Atom
 - o Repository : Github
 - o Project Management : Slack, Trello

5. Infoware

Berikut merupakan informasi yang dibutuhkan untuk membangun web service “food finder”

- Informasi Kantin di sekitar ITB
- Informasi Harga Makanan/Minuman di setiap kantin

Bab 3 Deskripsi Perancangan Perangkat Lunak

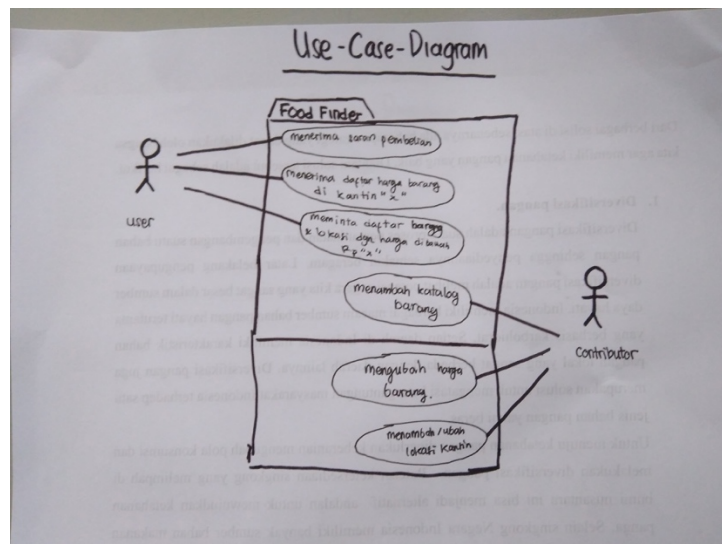
1. Lingkungan Implementasi

Berikut adalah rincian informasi mengenai lingkungan implementasi dari *web service foodfinder* :

Operating System	Linux Ubuntu 16.4 LTS 32-bit
DBMS	phpmyadmin
Database server	167.205.67.251:3306
Development tools	Sublime Text, Postman, Cyberduck, Atom,
Bahasa pemrograman	Go Programming Language

2. Use Case

Berikut merupakan diagram use case diagram dari web service yang dibuat.

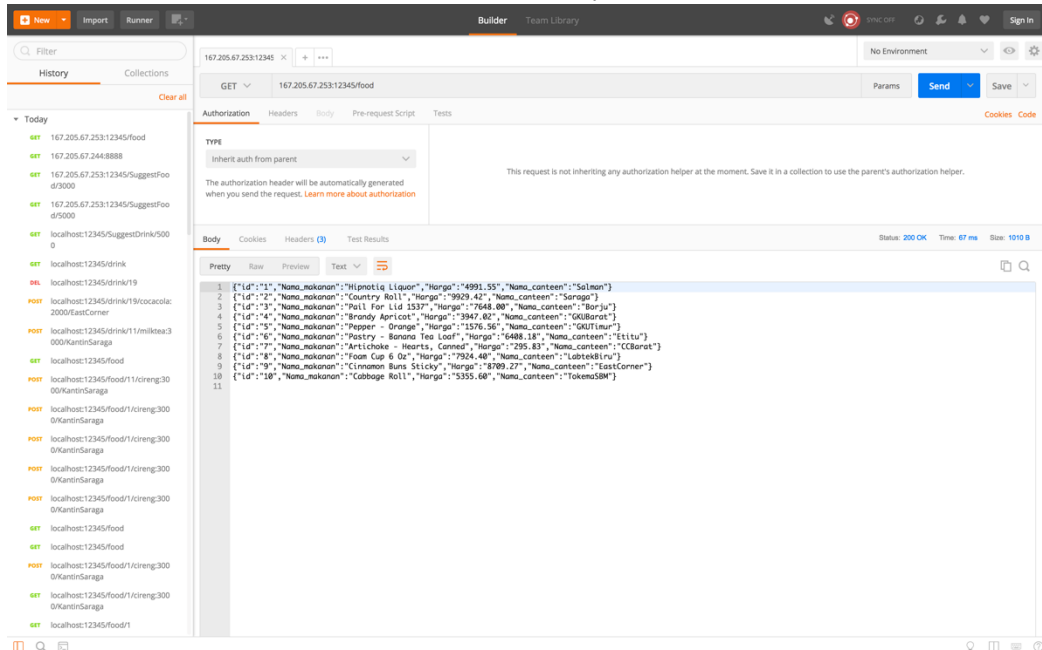


Berdasarkan gambar diatas, dapat dilihat bahwa terdapat 2 aktor dalam web service yang dibuat, yaitu user dan contributor. Terdapat beberapa perubahan dari use case pada gambar diatas, yaitu seorang user dapat melakukan beberapa aksi yaitu menerima saran pembelian, & menerima melihat data yang ada di database. Sementara seorang Contributor dapat melihat data, menambah data dan menghapus data.

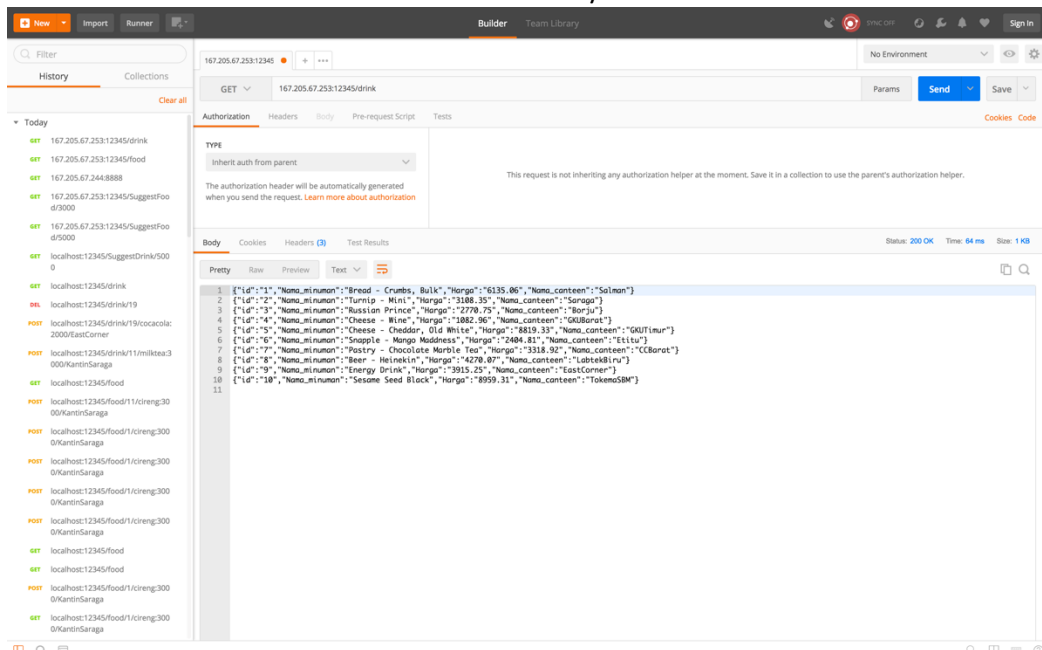
3. Implementasi Web-service

Berikut merupakan implementasi dari masing-masing fungsi menggunakan http request berdasarkan deskripsi API yang berada pada bagian sebelumnya.

167.205.67.253/food



167.205.67.253/drink



167.205.67.253/canteen

REST client interface showing a GET request to `167.205.67.253:12345/canteen`. The response body is a JSON array of 11 objects, each representing a menu item with fields like `id`, `name`, `location`, and `price`.

```
[{"id": "0", "name": "Kantin Saraga", "location": "Saraga"}, {"id": "1", "name": "Kantin Borju", "location": "Gedung Kita"}, {"id": "2", "name": "Kantin GiliBora", "location": "GiliBora"}, {"id": "3", "name": "Kantin GUTimur", "location": "GUTimur"}, {"id": "4", "name": "Kantin Titu", "location": "CTimur"}, {"id": "5", "name": "Kantin Cileur", "location": "CCileur"}, {"id": "6", "name": "Kantin LabtekBiru", "location": "SITB"}, {"id": "7", "name": "Kantin East Corner", "location": "FTTB"}, {"id": "8", "name": "Kantin Solman", "location": "Solman"}, {"id": "9", "name": "Kantin SBM", "location": "SBM"}, {"id": "10", "name": "Kantin Saraga", "location": "Saraga"}, {"id": "11", "name": "Kantin Borju", "location": "Gedung Kita"}]
```

167.205.67.253/food/1

REST client interface showing a GET request to `167.205.67.253:12345/food/1`. The response body is a JSON object representing a specific menu item with fields like `id`, `name`, `location`, and `price`.

```
{ "id": "1", "name": "Kantin Borju", "location": "Gedung Kita", "price": 4991.55 }
```

167.205.67.253/drink/1

Swagger UI interface for the endpoint `167.205.67.253/drink/1`. The interface shows a GET request with a status of 200 OK, a response time of 41 ms, and a response size of 208 B. The response body is a JSON object: `{\"id\":1,\"Nama_namun\":\"Bread - Crumbs, Bulk\",\"Korpa\":\"6135.06\",\"Nama_canteen\":\"Solomon\"}`. The left sidebar shows a list of endpoints, including `167.205.67.253/drink/1`, `167.205.67.253/food/1`, `167.205.67.253/canteen`, `167.205.67.253/drink`, `167.205.67.253/food`, `167.205.67.244.8888`, `167.205.67.253/12345/SuggestFood/9000`, `167.205.67.253/12345/SuggestFood/5000`, `localhost:12345/drink`, `localhost:12345/drink/19`, `localhost:12345/drink/19/cocacola:2000/EastCorner`, `localhost:12345/drink/11/milktea:3000/KantinSaraga`, `localhost:12345/food`, `localhost:12345/food/11/cireng:3000/KantinSaraga`, `localhost:12345/food/11/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food`, and `localhost:12345/food`.

167.205.67.253/canteen/1

Swagger UI interface for the endpoint `167.205.67.253/canteen/1`. The interface shows a GET request with a status of 200 OK, a response time of 73 ms, and a response size of 162 B. The response body is a JSON object: `{\"id\":1,\"Nama_canteen\":\"KantinRorju\",\"Location\":\"GedungKmla\"}`. The left sidebar shows a list of endpoints, including `167.205.67.253/12345`, `167.205.67.253/drink/1`, `167.205.67.253/food/1`, `167.205.67.253/canteen`, `167.205.67.253/drink`, `167.205.67.253/food`, `167.205.67.244.8888`, `167.205.67.253/12345/SuggestFood/9000`, `167.205.67.253/12345/SuggestFood/5000`, `localhost:12345/drink`, `localhost:12345/drink/19`, `localhost:12345/drink/19/cocacola:2000/EastCorner`, `localhost:12345/drink/11/milktea:3000/KantinSaraga`, `localhost:12345/food`, `localhost:12345/food/11/cireng:3000/KantinSaraga`, `localhost:12345/food/11/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food/1/cireng:3000/KantinSaraga`, `localhost:12345/food`, and `localhost:12345/food`.

- 167.205.67.253:12345/food/milktea:3000/KantinSaraga

The screenshot shows the Postman interface with a POST request to the URL `167.205.67.253:12345/food/milktea:3000/KantinSaraga`. The request is in the 'Builder' tab, and the 'Body' tab is selected, showing a status of 200 OK. The left sidebar shows a list of requests, with the current request highlighted. The top bar shows the 'Builder' tab and the 'Team Library' button. The bottom bar shows the status '200 OK' and the time '139 ms'.

hasil update data dengan method get all

The screenshot shows the Postman interface with a GET request to the URL `167.205.67.253:12345/food`. The request is in the 'Builder' tab, and the 'Body' tab is selected, showing a status of 200 OK and a large JSON response. The left sidebar shows a list of requests, with the current request highlighted. The top bar shows the 'Builder' tab and the 'Team Library' button. The bottom bar shows the status '200 OK' and the time '42 ms'.

```
1 [{"id":1,"Nama_makanan":"Hipnotic Liquor","Harga":4991.55,"Nama_canteen":"Salmon"}
2 [{"id":2,"Nama_makanan":"Country Roll","Harga":9929.42,"Nama_canteen":"Saraga"}
3 [{"id":3,"Nama_makanan":"Pell For Lid 1537","Harga":7648.00,"Nama_canteen":"Borju"}
4 [{"id":4,"Nama_makanan":"Brandy Apricot","Harga":3947.02,"Nama_canteen":"GUBURU"}
5 [{"id":5,"Nama_makanan":"Pepper - Orange","Harga":1576.56,"Nama_canteen":"GUTInur"}
6 [{"id":6,"Nama_makanan":"Pastry - Banana Tea Loaf","Harga":6488.18,"Nama_canteen":"titiu"}
7 [{"id":7,"Nama_makanan":"Artichoke - Hearts, Corned","Harga":295.83,"Nama_canteen":"Cobaret"}
8 [{"id":8,"Nama_makanan":"Foam Cup 6 Oz","Harga":7924.40,"Nama_canteen":"Labtek8iru"}
9 [{"id":9,"Nama_makanan":"Cinnamon Buns Sticky","Harga":8709.27,"Nama_canteen":"EastCorner"}
10 [{"id":10,"Nama_makanan":"Cabbage Roll","Harga":5335.00,"Nama_canteen":"Tokend8M"}
11 [{"id":11,"Nama_makanan":"milktea","Harga":3000,"Nama_canteen":"KantinSaraga"}
12
```

- 167.205.67.253:12345/minuman/milktea:3000/KantinSaraga

Postman interface showing a POST request to 167.205.67.253:12345/minuman/milktea:3000/KantinSaraga. The request is in the 'Authorization' tab, showing 'Inherit auth from parent'. The 'Body' tab is selected, showing a 'Pretty' view of the request body. The status bar indicates 'Status: 200 OK', 'Time: 128 ms', and 'Size: 118 B'.

hasil update data dengan get all

Postman interface showing a GET request to 167.205.67.253:12345/minuman/milktea:3000/KantinSaraga. The request is in the 'Authorization' tab, showing 'Inherit auth from parent'. The 'Body' tab is selected, showing a 'Pretty' view of the response body. The status bar indicates 'Status: 200 OK', 'Time: 41 ms', and 'Size: 1.08 KB'.

```
1 [{"id":1,"Nama_minuman":"Bread - Crumbs, Bulk","Harga":6135.06,"Nama_canteen":"Solnon"}
2 [{"id":2,"Nama_minuman":"Turnip - Mint","Harga":3188.35,"Nama_canteen":"Saraga"}
3 [{"id":3,"Nama_minuman":"Russian Prince","Harga":2778.75,"Nama_canteen":"Barju"}
4 [{"id":4,"Nama_minuman":"Cheese - Wine","Harga":1882.96,"Nama_canteen":"GOUBarot"}
5 [{"id":5,"Nama_minuman":"Cheese - Cheddar, Old White","Harga":8819.33,"Nama_canteen":"GOUBarot"}
6 [{"id":6,"Nama_minuman":"Sopote - Nongu Modress","Harga":2588.81,"Nama_canteen":"Teltu"}
7 [{"id":7,"Nama_minuman":"Postry - Chocolate Marble Tea","Harga":3318.92,"Nama_canteen":"CCBarot"}
8 [{"id":8,"Nama_minuman":"Beer - Heinekken","Harga":4278.00,"Nama_canteen":"LuhkeBlin"}
9 [{"id":9,"Nama_minuman":"Energy Drink","Harga":3915.25,"Nama_canteen":"EastCorner"}
10 [{"id":10,"Nama_minuman":"Sesame Seed Block","Harga":8959.31,"Nama_canteen":"TokemSBM"}
11 [{"id":11,"Nama_minuman":"milktea","Harga":3000,"Nama_canteen":"KantinSaraga"}
12
```

- 167.205.67.253:12345/canteen/12/KantinSR:ParkiranSR

Postman interface showing a POST request to `167.205.67.253:12345/canteen/12/KantinSR:ParkiranSR`. The request is in the 'Builder' tab, showing the 'Body' tab with a 'Pretty' view. The response status is 200 OK, Time: 68 ms, Size: 116 B.

hasil update data dengan get All canteen

Postman interface showing a GET request to `167.205.67.253:12345/canteen`. The request is in the 'Builder' tab, showing the 'Body' tab with a 'Pretty' view. The response status is 200 OK, Time: 40 ms, Size: 776 B. The response body is a JSON array of 12 canteen objects.

```
[{"id": "0", "Name_canteen": "KantinSraga", "Location": "Sraga"}, {"id": "1", "Name_canteen": "KantinBorju", "Location": "Sedunggilur"}, {"id": "2", "Name_canteen": "KantinGubaret", "Location": "Gubaret"}, {"id": "3", "Name_canteen": "KantinGulimur", "Location": "Gulimur"}, {"id": "4", "Name_canteen": "KantinLitlu", "Location": "CCLitlu"}, {"id": "5", "Name_canteen": "KantinCCBaret", "Location": "CCBaret"}, {"id": "6", "Name_canteen": "KantinAlteablu", "Location": "SITM"}, {"id": "7", "Name_canteen": "KantinEastCorner", "Location": "FTTM"}, {"id": "8", "Name_canteen": "KantinSalmon", "Location": "Salmon"}, {"id": "9", "Name_canteen": "Tokomoh", "Location": "SBM"}, {"id": "10", "Name_canteen": "KantinSR", "Location": "ParkiranSR"}, {"id": "11", "Name_canteen": "KantinSR", "Location": "ParkiranSR"}, {"id": "12", "Name_canteen": "KantinSR", "Location": "ParkiranSR"}]
```

-167.205.67.253:12345/food/1

The screenshot shows the Swagger UI interface for the endpoint `167.205.67.253:12345/food/1`. The interface is divided into several sections:

- Header:** Includes a "New" button, "Import", "Runner", and "Builder" tabs. The "Builder" tab is active.
- Filter:** A search bar with the text "Filter".
- History:** A list of endpoints with status indicators (e.g., "DEL", "POST", "GET"). The selected endpoint is `167.205.67.253:12345/food/1`.
- Main Panel:** Contains tabs for "Authorization", "Headers", "Body", "Pre-request Script", and "Tests". The "Authorization" tab is active, showing a dropdown menu for "Inherit auth from parent".
- Bottom Panel:** Contains tabs for "Body", "Cookies", "Headers", and "Test Results". The "Body" tab is active, showing a "Pretty" view of the request body.

-167.205.67.253:12345/drink/1

The screenshot shows the Swagger UI interface for the endpoint `167.205.67.253:12345/drink/1`. The interface is similar to the previous one, but the selected endpoint is `167.205.67.253:12345/drink/1`. The request is currently set to DELETE and the body is empty.

-167.205.67.253:12345/canteen/1

167.205.67.253:12345/canteen/12

DELETE

Params Send Save

Authorization Headers Body Pre-request Script Tests

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies Headers Test Results

Status: 200 OK Time: 111 ms Size: 116 B

Pretty Raw Preview Text

167.205.67.253:12345/SuggestFood/5000

167.205.67.253:12345/SuggestFood/5000

GET

Params Send Save

Authorization Headers Body Pre-request Script Tests

TYPE

Inherit auth from parent

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.

Body Cookies Headers Test Results

Status: 200 OK Time: 229 ms Size: 472 B

Pretty Raw Preview Text

```
1 [{"id": "14", "Name_makanan": "Brandy Apricot", "Harga": "3947.80", "Nama_canteen": "OGIBarat"}]
2 [{"id": "5", "Name_makanan": "Pepper - Orange", "Harga": "1576.56", "Nama_canteen": "OGUItan"}]
3 [{"id": "77", "Name_makanan": "Artichoke - Hearts, Canned", "Harga": "795.83", "Nama_canteen": "CCBarot"}]
4 [{"id": "12", "Name_makanan": "Milktea", "Harga": "3000", "Nama_canteen": "KontinSaraga"}]
5
```


167.205.67.253:12345/SuggestDrink/5000

The screenshot displays the Postman API client interface. At the top, the URL bar shows the request: `167.205.67.253:12345/SuggestDrink/5000`. The interface is divided into several sections:

- Left Sidebar:** Contains a 'History' tab with a list of recent requests. The selected request is a GET request to `167.205.67.253:12345/SuggestDrink/5000`.
- Main Panel:** Features tabs for 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Authorization' tab is active, showing a dropdown menu set to 'inherit auth from parent'. Below this, a message states: 'The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)'. To the right, a note says: 'This request is not inheriting any authorization helper at the moment. Save it in a collection to use the parent's authorization helper.'
- Bottom Panel:** Displays the response body in JSON format. The response is a list of 11 items, each containing an 'id', 'Nama_minuman', 'Harga', and 'Nama_canteen'. The items are:
 - 1: `{ "id": "2", "Nama_minuman": "Turnip - Mini", "Harga": "3188.35", "Nama_canteen": "Saraga" }`
 - 2: `{ "id": "3", "Nama_minuman": "Russian Prince", "Harga": "2770.75", "Nama_canteen": "Borju" }`
 - 3: `{ "id": "4", "Nama_minuman": "Cheese - Wine", "Harga": "1082.96", "Nama_canteen": "GulBarat" }`
 - 4: `{ "id": "6", "Nama_minuman": "Snapple - Mango Madness", "Harga": "2404.81", "Nama_canteen": "Ettitu" }`
 - 5: `{ "id": "7", "Nama_minuman": "Pastry - Chocolate Marble Tea", "Harga": "3318.92", "Nama_canteen": "CCBarat" }`
 - 6: `{ "id": "8", "Nama_minuman": "Beer - Heinekken", "Harga": "4270.07", "Nama_canteen": "LabtekBiru" }`
 - 7: `{ "id": "9", "Nama_minuman": "Energy Drink", "Harga": "3915.25", "Nama_canteen": "EastCorner" }`
 - 8: `{ "id": "11", "Nama_minuman": "Milktea", "Harga": "3000", "Nama_canteen": "KantinSaraga" }`

Bab 4 Hasil Implementasi dan Pengujian Perangkat Lunak

1. Hasil Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapatkan	kesimpulan
PF-01	Pengujian untuk menampilkan seluruh data makanan	Menuliskan request http GET pada aplikasi postman	http://167.205.67.253/food	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-02	Pengujian untuk menampilkan seluruh data minuman	Menuliskan request http GET pada aplikasi postman	http://167.205.67.253/drink	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-03	Pengujian untuk menampilkan seluruh data canteen	Menuliskan request http GET pada aplikasi postman	http://167.205.67.253/canteen	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-04	Pengujian untuk menampilkan 1 data makanan	Menuliskan request http GET pada aplikasi postman	<a href="http://167.205.67.253/food/<id>">http://167.205.67.253/food/<id>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik

PF-05	Pengujian untuk menampilkan 1 data minuman	Menuliskan request http GET pada aplikasi postman	<a href="http://167.205.67.253/drink/<id>">http://167.205.67.253/drink/<id>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-06	Pengujian untuk menampilkan 1 data canteen	Menuliskan request http GET pada aplikasi postman	<a href="http://167.205.67.253/canteen/<id>">http://167.205.67.253/canteen/<id>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-07	Pengujian untuk memasukkan 1 data makanan ke database	Menuliskan request http POST pada aplikasi postman	<a href="http://167.205.67.253/food/<id>/<nama_makanan>:<harga>/<nama_kantin>">http://167.205.67.253/food/<id>/<nama_makanan>:<harga>/<nama_kantin>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-08	Pengujian untuk memasukkan 1 data minuman ke database	Menuliskan request http POST pada aplikasi postman	<a href="http://167.205.67.253/drink/<id>/<nama_minuman>:<harga>/<nama_kantin>">http://167.205.67.253/drink/<id>/<nama_minuman>:<harga>/<nama_kantin>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-09	Pengujian untuk memasukkan 1 data kantin ke dalam database	Menuliskan request http POST pada aplikasi postman	<a href="http://167.205.67.253/canteen/<id>/<nama_kantin>:<lokasi>">http://167.205.67.253/canteen/<id>/<nama_kantin>:<lokasi>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-10	Pengujian untuk menghapus 1 data makanan dari dalam database	Menuliskan request http DELETE pada aplikasi postman	<a href="http://167.205.67.253/food/<id>">http://167.205.67.253/food/<id>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-11	Pengujian untuk menghapus 1	Menuliskan request http	<a href="http://167.205.67.253/drink/<id>">http://167.205.67.253/drink/<id>	Data makanan	Data makanan	Fungsi berjalan

	data minuman dari dalam database	DELET pada aplikasi postman		dengan format json	dengan format json	dengan baik
PF-12	Pengujian untuk menghapus 1 data kantin dari dalam database	Menuliskan request http DELETE pada aplikasi postman	<a href="http://167.205.67.253/canteen/<id>">http://167.205.67.253/canteen/<id>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-13	Pengujian untuk menampilkan data suggestion pembelian makanan	Menuliskan request http GET pada aplikasi postman	<a href="http://167.205.67.253/SuggestFood/<price>">http://167.205.67.253/SuggestFood/<price>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik
PF-14	Pengujian untuk menampilkan data suggestion pembelian minuman	Menuliskan request http GET pada aplikasi postman	<a href="http://167.205.67.253/SuggestDrink/<price>">http://167.205.67.253/SuggestDrink/<price>	Data makanan dengan format json	Data makanan dengan format json	Fungsi berjalan dengan baik

2. Evaluasi Pengujian

Dalam pengujian yang dilakukan pada tahap sebelumnya, hanya dilakukan pengujian berdasarkan input yang benar, dengan kata lain input dalam pengujian yang dilakukan dalam tahap diatas tidak menguji eror-eror, karena dalam web service yang dibuat, eror-eror minor seperti kesalahan input ke database, duplikasi id, dan beberapa kesalahan yang tidak menyebabkan web service mati belum dibuat handlernya. Jadi apabila mencoba untuk menguji dengan memasukkan input tidak sesuai dengan seharusnya, maka tidak akan mendapat suatu respons. Web service disini akan tetap berjalan jika input tidak sesuai, namun tidak memberikan pesan eror atau apapun jika terjadi kegagalan.