

Api First Development Portal Informasi Produk UMKM Kerupuk Di Kecamatan Tulangan, Kabupaten Sidoarjo Berbasis Microservices

Muhammad Yusriza Ardiansyah¹, Ricky Eka Putra²

^{1,2}Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹muhammad.17051204059@mhs.unesa.ac.id

²rickyeka@unesa.ac.id

Abstrak — UMKM kerupuk di Kecamatan Tulangan Kabupaten Sidoarjo memiliki banyak produk kerupuk. Salah satu produk UMKM yang menjadi ikonik adalah usaha rumahan yang menghasilkan aneka ragam kerupuk melalui proses pengolahan. Namun dalam hal ini masih memiliki salah satu hal yang kurang diperhatikan, yaitu dalam pemasaran dan promosi. Penyampaian informasi dilakukan melalui berbagai macam jenis media. Peneliti memilih media WebAPI untuk memberikan informasi yang bersifat publik serta dapat diakses dan dikelola oleh publik. Peneliti merancang sistem layanan informasi berbasis API yang menyediakan data produk kerupuk olahan UMKM di Kecamatan Tulangan, Kabupaten Sidoarjo. Sistem tersebut dapat diakses oleh *user* yang terdaftar untuk mendapatkan informasi produk. *User* yang telah terdaftar juga dapat berkontribusi untuk mengelola produk kerupuk oleh masing-masing *user*. Pengembangan sistem layanan informasi kerupuk menggunakan metode *API first development* digunakan agar pengembangan tersebut dilakukan secara terfokus dengan kebutuhan fungsional sistem.

Berdasarkan penelitian yang sudah dilaksanakan, dengan mengambil dari banyaknya data yang diuji coba hingga membantu berjalannya bisnis kerupuk, pengembangan layanan sistem informasi berbasis *WebAPI*. Karena pada UMKM terdapat beberapa produk kerupuk unggulan, peneliti membuat sistem layanan informasi yang dapat membantu memberikan detail informasi kerupuk produksi UMKM kepada klien/*user* yang mengakses sistem tersebut dengan *WebAPI*. Pertimbangan penggunaan arsitektur *microservice* agar sistem dapat terisolasi dan berfokus pada modul tertentu. Dengan arsitektur *microservice*, ketika terdapat pengembangan selanjutnya baik *minor* maupun *major* dapat dilakukan secara parsial ataupun paralel. Pengembangan dengan menerapkan *API first* juga dilakukan dengan tujuan tercapainya produk *WebAPI* sehingga pengembangan layanan agar terbangun sesuai kebutuhan.

Kata Kunci – UMKM, Web API, *Microservices*, CNN, *API First Developments*.

I. PENDAHULUAN

Bersamaan dengan waktu yang berjalan, pembangunan dan pertumbuhan ekonomi mempunyai peran krusial dalam meningkatkan pendapatan perekonomian wilayah, termasuk meningkatkan kesejahteraan rakyat. Peran UMKM memegang peran yang begitu krusial dalam kemajuan ekonomi Indonesia. Persentase tersebut mencapai 99% dari keseluruhan unit usaha. Sumbangan UMKM terhadap Produk Domestik Bruto (PDB) pun mencapai 60,5%, sementara penyerapan tenaga kerja mencapai 96,9% dari total tenaga kerja di negara ini [7]. Kabupaten Sidoarjo terutama di daerah Tulangan dikenal dengan produk-produknya untuk UMKM. Salah satu produk UMKM yang menjadi ikonik adalah usaha rumahan yang

menghasilkan aneka ragam kerupuk melalui proses pengolahan [8]. Tidak berbanding lurus dengan produksi besar dan berbagai macam jenis kerupuk yang di produksi dari daerah Sidoarjo, penjualan kerupuk hasil olahan daerah Tulangan masih belum mencapai angka penjualan yang signifikan karena banyaknya para pengusaha yang menggunakan metode pemasaran kuno dari mulut ke mulut atau iklan konvensional, sedangkan di era sekarang kita sudah dimudahkan dengan adanya teknologi untuk saling bertukar informasi dan melakukan jual beli di dunia digital. Namun hanya beberapa pengusaha kerupuk yang sudah menggunakan teknologi terbaru dikarenakan minimnya informasi dan banyaknya pengusaha yang masih bertahan dengan cara pemasaran kuno. Oleh karenanya peneliti ingin mendorong para pengusaha kerupuk terutama daerah Tulangan untuk mulai beradaptasi dengan adanya teknologi terbaru. Salah satunya penuliti akan mengangkat UMKM kerupuk di Kecamatan Tulangan dan berharap UMKM kerupuk pada kecamatan lainnya di Sidoarjo untuk beradaptasi dengan adanya teknologi terbaru saat ini.

Berdasarkan hasil kunjungan peneliti pada UMKM di Kecamatan Tulangan Kabupaten Sidoarjo, terdapat beberapa jenis kerupuk yang dapat dijadikan bahan penelitian yaitu kerupuk tahu, kerupuk bawang dan kerupuk keong. Peneliti melakukan pengembangan sistem informasi produk UMKM kerupuk secara *web API* untuk menyediakan informasi kerupuk yang ada di Kecamatan Tulangan, Kabupaten Sidoarjo agar dapat diakses oleh berbagai *client* secara terbuka.

Tentunya dalam langkah memajukan UMKM perlu menggunakan teknologi terbaru yang sudah tersedia saat ini, seperti teknologi *web* misalnya. Teknologi *web* merupakan suatu teknologi yang memiliki hubungan dengan antarmuka untuk menjadi jembatan antara *web server* dan klien. *Website* yakni kumpulan direktori dan berkas yang berisi berbagai perintah dan fungsi khusus. Oleh karena itu, diperlukan konsep pengembangan yang baik agar sistem informasi dapat beradaptasi. Dengan cara memisahkan logika dalam database dari logika antarmuka dalam sistem informasi dianggap sebagai cara ideal untuk menyelesaikan masalah ini. dan setiap logika dapat dikembangkan secara independen, sehingga memungkinkan lebih banyak fungsionalitas. Representasi fungsi dalam logika *database* yang diakses logika dala antarmuka secara terprogram disebut *Application Programming Interface (API)*.

Pengembangan *web* sebagai cara mudah untuk mentransfer informasi dari satu lokasi ke lokasi lain melalui jaringan komunikasi komputer. Pengguna sistem informasi pada jaringan komputer bisa meminta layanan data berupa

layanan *web* dari *server* sistem informasi, dan *server* sistem informasi dengan cepat memberikan layanan data yang diminta. Layanan *web* untuk komunikasi antar sistem informasi guna pertukaran informasi memakai JSON. JSON adalah sintaks sederhana yang memakai teks, dan tidak bergantung pada bahasa tertentu guna menetapkan format pertukaran data. Asal usul JSON berasal dari bahasa pemrograman *ECMAScript*, yang merupakan bahasa tujuan khusus. JSON menetapkan seperangkat aturan kecil guna membangun representasi portabel dari data terstruktur. JSON digunakan untuk memfasilitasi pengambilan informasi yang diperlukan dan memastikan keutuhan data yang dihasilkan oleh sistem informasi berbeda dalam suatu organisasi. Seorang peneliti mengembangkan *web service* pertukaran data antar sistem informasi dalam format menggunakan JSON yang diperlukan oleh sistem informasi lainnya. Peneliti berharap akan adanya layanan *web* yang menggunakan format JSON, yang memungkinkan sistem informasi memperoleh informasi yang dibutuhkannya dan menjamin keutuhan data yang dihasilkan di antara sistem informasi tersebut [6].

Aplikasi *web* bisa diakses di berbagai sistem operasi dan hampir semua jenis perangkat. Pembangunan *web* merupakan proses yang cukup kompleks karena melibatkan aliran data yang cukup banyak dan penyusunan kode program yang cukup kompleks untuk menangani itu penggunaan bahasa pemrograman Java dan *Framework Spring Boot* sangat menolong. Penggunaan *framework* Java yaitu *Spring Boot* semakin memudahkan pembangunan *web* karena *Spring Boot* tergolong *framework* yang mempunyai bobot cukup ringan untuk dipakai, bersifat *open source*, dan menyediakan berbagai modul yang bisa dipakai guna membantu tugas pengembang aplikasi dan *Spring Boot* sendiri bisa dikolaborasikan dengan bahasa pemrograman lainnya karena *Spring Boot* menunjang pembuatan aplikasi berbasis *restfull web Service*. *Spring Boot* memungkinkan memakai modul - modul yang kita butuhkan saja sehingga proyek yang dikerjakan menjadi lebih ringan [3].

Dalam pengembangan aplikasi ini penulis memilih untuk menggunakan bahasa pemrograman *java*, *Java* dipilih daripada bahasa pemrograman lain karena memiliki sifat fleksibel yang menjadikannya bahasa pemrograman yang mudah dikembangkan sesuai kebutuhan, bersifat *multiplatform* yang menjadikan bahasa pemrograman *Java* mudah digunakan di berbagai jenis sistem operasi. *Java* terus menerus dikembangkan dan ditambah dengan banyak fitur yang bisa digunakan para *programer* untuk membuat program aplikasi yang stabil dan aman sesuai dengan kebutuhan saat ini [4].

Microservice merupakan sekelompok proses kecil yang berdiri sendiri dan berinteraksi satu sama lain guna membentuk aplikasi kompleks yang tidak tergantung pada API bahasa tertentu. Layanan ini merupakan elemen penyusun kecil dan berfokus pada tugas-tugas ringan untuk mempromosikan pendekatan berbeda terhadap pengembangan sistem. *Mircoservice* ialah evolusi dari *Service Architecture*. Karena *microservice* yakni suatu sistem yang terdiri dari komponen-komponen berupa layanan modular, layanan independen yang mempunyai tujuan masing-masing tapi

saling berkoordinasi melalui mekanisme yang jelas untuk mencapai tujuan bersama. Hal ini terutama berlaku untuk pengembangan perangkat lunak [10].

Berdasarkan penelitian terdahulu yang pernah dilakukan oleh M. Dudjak & G. Martinović yang berjudul “*An API-first Methodology for Designing a Microservice-based Backend as a Service Platform*” Dengan menyediakan *backend* untuk aplikasi, memfasilitasi dan mempercepat proses pengembangan perangkat lunak. Untuk mencegah masalah besar dengan menggunakan penyedia BaaS pihak ketiga, *platform BaaS (Backend as A Service)* menghadirkan berbagai tantangan terkait arsitektur dan desain. Dalam penelitian tersebut membahas *platform BaaS* dan mengusulkan metode untuk menyediakan desain *platform BaaS* berdasarkan arsitektur *microservice*. Arsitektur *microservice* adalah gaya arsitektur pilihan untuk solusi *cloud* karena *loose coupling*, arsitektur *microservice* memberikan kemudahan penskalaan dan integrasi dengan layanan *platform BaaS*. Metodologi yang diadopsi dalam merancang *platform BaaS* berbasis *microservice* dibentuk sesuai dengan pendekatan API yang berusaha untuk merancang API yang sesuai dan representatif dari *platform*. Dalam jurnal ini mengusulkan desain *platform BaaS* yang menggambarkan *entity relations*, *integration pattern*, dan *communication style* [1].

Berdasarkan analisa dari penelitian tersebut, peneliti ingin melakukan penelitian dengan judul “*API First Development Portal Informasi Produk UMKM Kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo Berbasis Microservices*”. Penelitian ini akan menerapkan arsitektur *microservices* dalam proses pembuatan sistem layanan informasi produk UMKM kerupuk berbasis *web api* sehingga diharapkan dapat melakukan pengembangan *web api* yang sistematis serta sistem layanan informasi produk UMKM kerupuk yang lebih terstruktur, efisien dan terbuka.

II. METODOLOGI PENELITIAN

Untuk mendukung penyelenggaraan penelitian ini, peneliti memanfaatkan metode pengembangan perangkat lunak serta mengumpulkan data yang diperlukan. Metode yang dipakai pada pengembangan perangkat lunak “*Api First Development Portal Informasi Produk UMKM Kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo Berbasis Microservices*” dengan memakai metode *waterfall*. Metodologi *Waterfall* yakni pendekatan SDLC pertama yang dipakai dalam pengembangan aplikasi. Urutan metode air terjun berlangsung secara berkesinambungan, dimulai dari proses perencanaan, analisis, perancangan, implementasi dan pemeliharaan sistem. Metodologi air terjun adalah metodologi yang menyediakan siklus hidup perangkat lunak yang berurutan atau berurutan [5]. Beberapa langkah-langkah pada penelitian yaitu:

A. Analisa Kebutuhan

Dalam pengembangan *API first development* portal informasi produk UMKM kerupuk dengan arsitektur *microservice* menggunakan metode *waterfall* ini, terdapat beberapa kebutuhan yang harus dianalisa terlebih dahulu.

Karena kebutuhan tersebut akan dipakai sebagai patokan dalam mendukung penelitian ini. Analisa kebutuhan dibagi menjadi beberapa bagian, yakni:

1. Kebutuhan Data

Sumber data yang digunakan guna melakukan penelitian ini diambil dari beberapa referensi. Metode pengumpulan data dalam penelitian ini memakai dua jenis, yakni studi literatur dan juga observasi. Peneliti mengutip beberapa referensi yang relevan dari berbagai macam literatur tentang pembahasan API *First Development* portal informasi produk UMKM kerupuk untuk mendalami pengetahuan. Data yang digunakan diantaranya berupa data pada produk UMKM jurnal internasional maupun nasional, makalah, dan situs resmi serta beberapa sumber dari internet.

2. Kebutuhan Fungsional

Aplikasi yang dikembangkan diharapkan dapat memenuhi kebutuhan melakukan pengelolaan *user* termasuk autentikasi dan *authorisasi*, penyediaan data jenis dan bahan pembuatan kerupuk, *sharing* informasi penjual kerupuk, serta *generate* laporan data kerupuk sebagaimana berikut :

- a. Pengelolaan user melalui backend api untuk autentikasi dan *authorisasi*
- b. Menyediakan data jenis dan bahan pembuatan kerupuk
- c. *Sharing* informasi penjual kerupuk
- d. Membuat laporan data kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo.

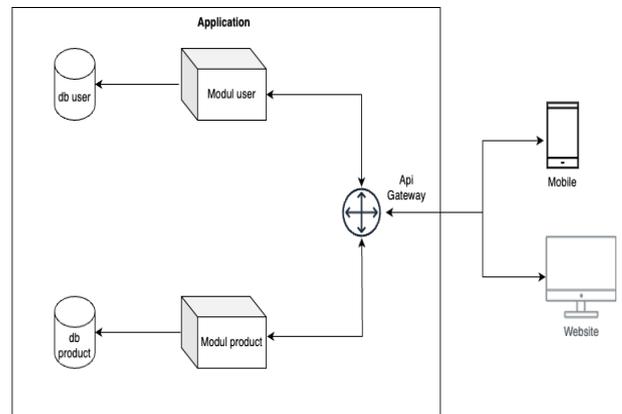
3. Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional terbagi menjadi dua, yakni analisis kebutuhan *software* dan analisis kebutuhan *hardware*. Analisis *hardware* bertujuan guna mempermudah tahap perancangan dan implementasi dalam pengembangan sistem. Spesifikasi perangkat yakni satu diantara faktor krusial yang dibutuhkan dalam mendukung proses penelitian pengembangan sistem portal informasi data UMKM kerupuk berbasis *microservice* ialah:

- Processor Intel Core i7-1165G7
 - RAM 8 GB
 - SSD 500 GB
 - Sistem Operasi Windows 11 64-bit
- Sementara *software* yang diperlukan pada penelitian ini yakni antara lain:
- IntelliJ Idea sebagai media untuk menulis dan menjalankan source code di sistem operasi
 - Dbeaver sebagai perangkat lunak administrasi database
 - PostgreSQL sebagai system penyimpanan basis data
 - Postman sebagai client untuk melakukan pengujian pengambilan data

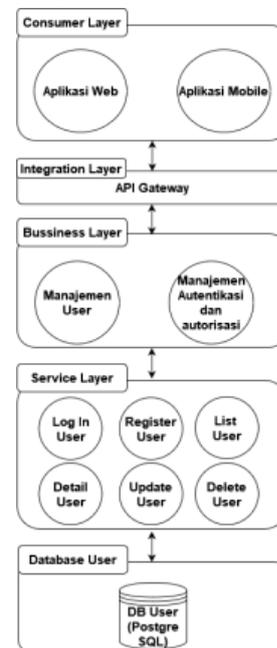
B. Desain Sistem

Pada penelitian ini peneliti akan merancang portal sistem informasi produk UMKM kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo berbasis *microservice*. Didalam desain sistem terdapat beberapa penjabaran terkait merancang dan pembuatan arsitektur *microservice*, arsitektur modul *user* dan arsitektur modul produk seperti pada gambar dibawah:



Gambar 1. Arsitektur Microservice

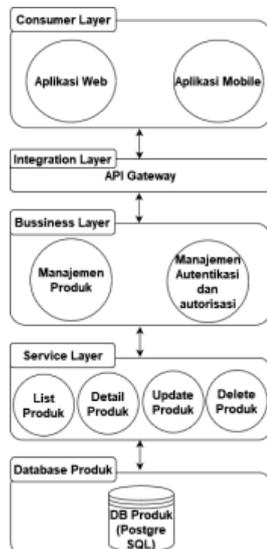
Pada gambar 1 yakni arsitektur *microservice*. Dalam arsitektur tersebut terdapat *client mobile* ataupun *web* melakukan *request* ke API gateway. Kemudian API gateway meneruskan *request* ke service yang memiliki API sesuai dengan *request*. Setiap modul akan terhubung kedalam masing-masing database. Sistem dapat melakukan *create*, *read*, *update* atau *delete* sesuai *request*nya. Pada modul *user* akan terhubung ke database *user*, dengan fungsi untuk manajemen *user*. Pada modul *product* akan terhubung ke database *product*, dengan fungsi untuk manajemen *product*.



Gambar 2. Arsitektur Modul User [2]

Pada gambar 2 merupakan arsitektur modul *user*. Terdapat *consumer layer*, *integration layer*, *business layer*, *service*

layer dan database user. Dalam *consumer layer* terdapat aplikasi web dan aplikasi mobile. Di dalam *integration layer* terdapat API gateway. Lalu *bussiness layer* terdapat manajemen user dan manajemen autentikasi dan otorisasi. Kemudian *service layer* terdapat *log in user*, *register user*, *list user*, *detail user*, *update user* dan juga *delete user*. Yang terakhir *database user* di dalamnya terdapat *database user* (Postgre SQL).



Gambar 3. Arsitektur Modul Produk [2]

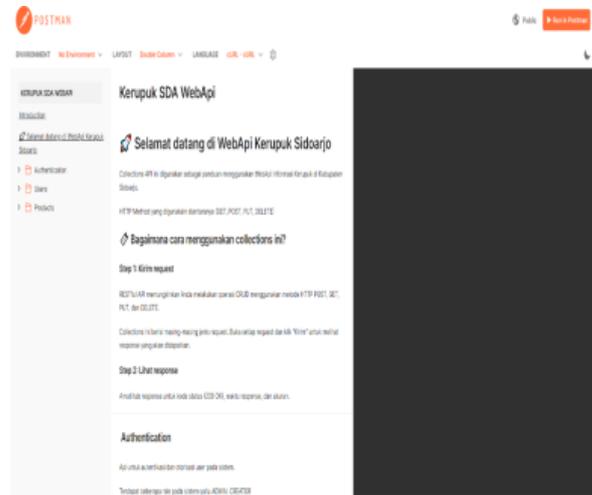
Pada gambar 3 merupakan arsitektur modul produk. Terdapat *consumer layer*, *integration layer*, *bussiness layer*, *service layer* dan *database produk*. Dalam *consumer layer* terdapat aplikasi web dan aplikasi mobile. Di dalam *integration layer* terdapat API gateway. Lalu *bussiness layer* terdapat manajemen produk dan manajemen autentikasi dan otorisasi. Kemudian *service layer* terdapat list produk, detail produk, update produk dan juga delete produk Yang terakhir *database produk* di dalamnya terdapat *database produk* (Postgre SQL).

C. Tahap Pengembangan Sistem (Coding)

Metode untuk mengembangkan sistem yang digunakan yaitu metode waterfall. Desain perlu diimplementasikan dalam bentuk program *software*, dan pada tahap ini dihasilkan program komputer yang selaras dengan desain yang sudah dibuat sebelumnya pada tahap desain. Selanjutnya, perancangan sistem akan merinci setiap perintah atau kode yang dipakai dalam pemrosesan data untuk pembuatan Portal Sistem Informasi Produk UMKM Kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo Berbasis Microservices. Pengembangan aplikasi ini memakai bahasa pemrograman php dengan framework laravel 8. Desain aplikasi yang dikembangkan menggunakan *model view controller*, model digunakan untuk adapter antara aplikasi dan database, dalam penelitian ini menggunakan database mysql 8. View berfungsi untuk menampilkan interface aplikasi web kepada user, dalam layer ini menggunakan html dan css. Manajemen tampilan dan data model diatur oleh controller yang merupakan alur bisnis

dari aplikasi. Penggunaan pola arsitektur *microservices* untuk memudahkan dalam pengembangan Portal Sistem Informasi Produk UMKM Kerupuk di Kecamatan Tulangan, Kabupaten Sidoarjo agar lebih terstruktur dan sistematis.

D. Rancangan UI



Gambar 4. Rancangan UI

Pada gambar 4 merupakan rancangan UI pada Web API kerupuk Sidoarjo. Di dalamnya terdapat *authentication*, *users* dan *products*.

E. Pengujian (Testing)

Dalam fase ini, program diuji memakai metode *BlackBox Testing* dengan tujuan agar implementasi yang sudah dirancang bisa berjalan selaras dengan rencana yang sudah dibuat. Teknik *black box testing* yakni teknik yang dipakai guna menguji aplikasi dengan tidak berfokus pada spesifikasi aplikasi. Tes ini memeriksa nilai *output* terhadap nilai masukannya. Tak ada upaya guna mencari tahu kode pemrograman apa yang memakai keluarannya. Proses pengujian *black box* adalah menguji program yang dirancang dengan menginputkan data ke tiap form. Pengujian ini dibutuhkan guna menguji aplikasi dengan *black box testing* dengan tujuan untuk memastikan bahwa proyek tersebut sesuai dengan manajemen proyek yang dijalankan sesuai kebutuhan perusahaan [9]. Berikut Langkah-langkah pengujian dengan metode *BlackBox testing*:

1. Memeriksa spesifikasi dan persyaratan sistem.
2. Memasukkan input yang valid dan apakah proses yang terjadi sudah sesuai. Selain itu juga menguji input yang tidak valid.
3. Melihat output apakah sesuai dengan yang diharapkan atau tidak.
4. Membuat kasus uji dengan input yang dipilih.
5. Kasus uji dijalankan.
6. Membandingkan output yang dihasilkan.
7. Mencatat dan memperbaiki.

III. HASIL DAN PEMBAHASAN

A. Implementasi

TABEL I
 API PATH

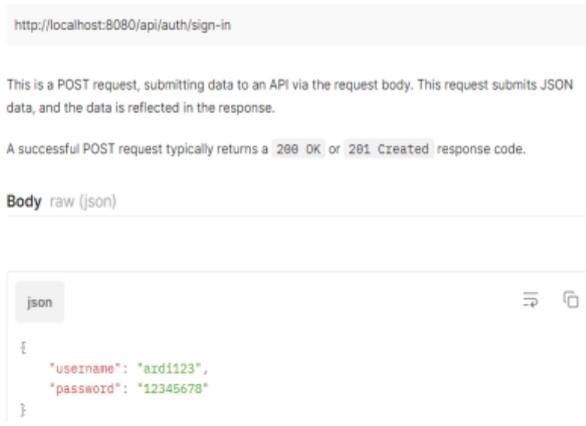
API Endpoint	HTTP Method	Semantics
/api/auth/sign-in	POST	Mengirimkan entitas login
/api/auth/sign-up	POST	Mengirimkan entitas register
/api/user/list	GET	Mendapatkan list user
/api/user/detail	GET	Mendapatkan detail user
/api/user/update	PUT	Menggantikan semua representasi saat ini dengan payload permintaan update user
/api/user/delete	DELETE	Menghapus data user yang telah dipilih
/api/product/list	GET	Mendapatkan list produk
/api/product/detail	GET	Mendapatkan detail produk
/api/product/tambah	POST	Mengirimkan entitas tambah produk
/api/product/update	PUT	Menggantikan semua representasi saat ini dengan payload permintaan update produk
/api/product/hapus	DELETE	Menghapus data product yang telah dipilih

Pada tabel 1 diatas merupakan penjelasan tentang API Path. Di dalamnya terdapat endpoint yang terdiri dari authentication, user dan product. Lalu HTTP Method yang didalamnya terdapat post, get, put dan delete. Kemudian semantics yang menjelaskan fungsi dari Endpoint serta HTTP Methodnya.

B. Pengembangan

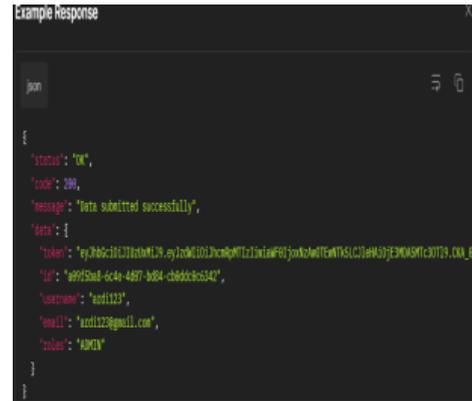
1. Authentication

POST Post Sign-In



Gambar 5. Sign In Request

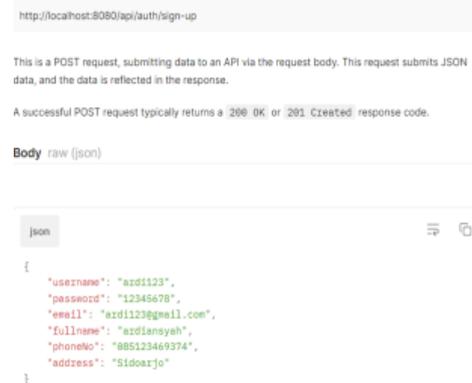
Gambar 5 merupakan API sign in digunakan untuk login dari klien ke server. Mekanisme login menggunakan field request username dan password.



Gambar 6. Sign In response

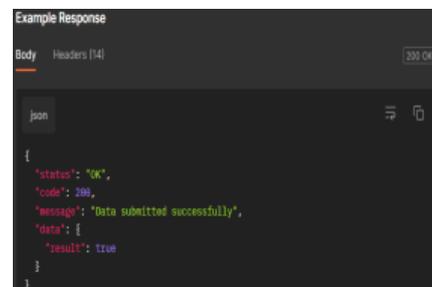
Pada gambar 6 merupakan respon API sign in ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API sign in yang berisi informasi kredensial user login. Diantaranya token adalah kredensial jwt. Id yang menunjukkan id user, username yang menunjukkan username dari user. Email yang menunjukkan email dari user serta roles merupakan roles dari user.

POST Post Sign-Up



Gambar 7. Sign Up Request

Pada gambar 7 merupakan API sign up digunakan untuk daftar akun klien. Mekanisme daftar akun menggunakan field request username, password, email, fullName, phoneNo dan address.



Gambar 8. Sign Up Response

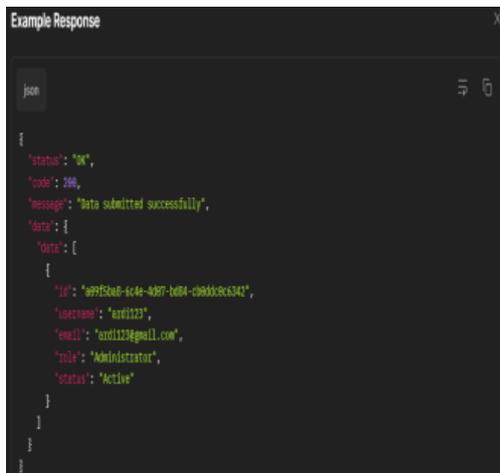
Pada gambar 8 merupakan respon API *sign up* ada beberapa *field* yang dikembalikan. Terdapat *field* status menunjukkan http status, *code* yang menunjukkan http status *code*, *message* yang merupakan informasi pesan dari *response*, dan *field data* merupakan objek respon API *sign up*. Lalu pada *field data* terdapat *result* yang merupakan *result* dari *user*.

2. User



Gambar 9. API Get List User Request

Pada gambar 9 merupakan API *get list user* digunakan untuk melihat *list data* klien yang terbaru. Terdapat *authorization* untuk melihat token.



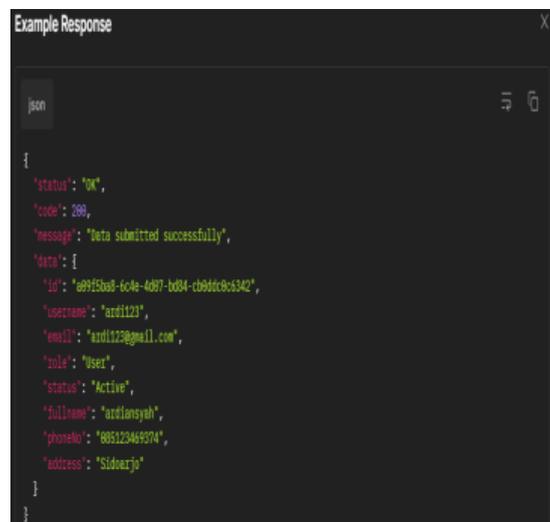
Gambar 10. API Get List User Response

Pada gambar 10 merupakan respon API *get list user* ada beberapa *field* yang dikembalikan. Terdapat *field* status menunjukkan http status, *code* yang menunjukkan http status *code*, *message* yang merupakan informasi pesan dari *response*, dan *field data* merupakan objek respon API *get list user*. Lalu pada *field data* terdapat *id*, *username*, *email*, *role* dan *status*.



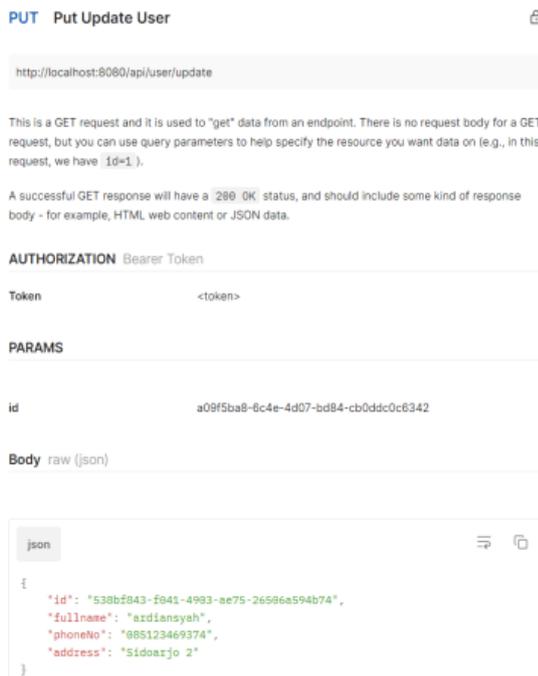
Gambar 11. API Get Detail User Request

Pada gambar 11 merupakan API *get detail user* digunakan untuk melihat detail data klien yang terbaru. Terdapat *authorization* untuk melihat token. Terdapat juga *params* untuk melihat *id* klien.



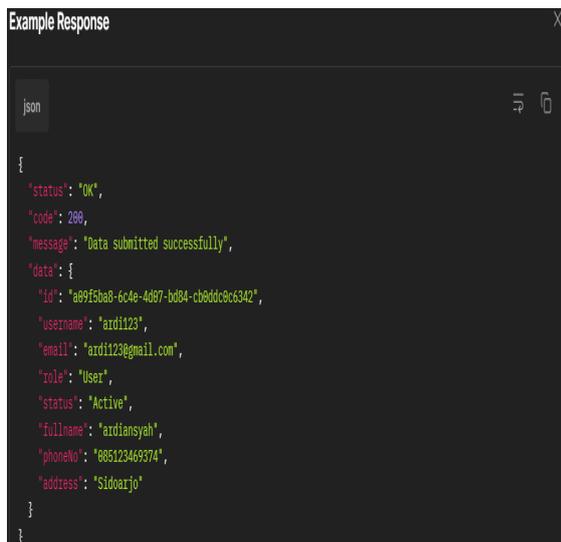
Gambar 12. API Get Detail User Response

Pada gambar 12 merupakan respon API *get detail user* ada beberapa *field* yang dikembalikan. Terdapat *field* status menunjukkan http status, *code* yang menunjukkan http status *code*, *message* yang merupakan informasi pesan dari *response*, dan *field data* merupakan objek respon API *get detail user*. Lalu pada *field data* terdapat *id*, *username*, *email*, *role*, *status*, *fullname*, *phoneNo* dan *address*.



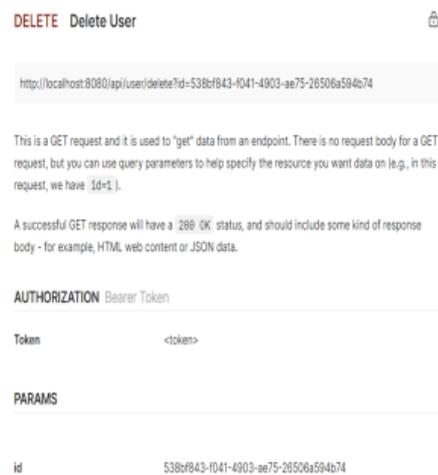
Gambar 13. API Put Update User Request

Pada gambar 13 merupakan request API put update user berfungsi melakukan update data user yang terbaru. Terdapat authorization untuk melihat token, params untuk melihat id klien dan body raw (json) yang didalamnya terdapat data id, fullname, phoneNo dan address user.



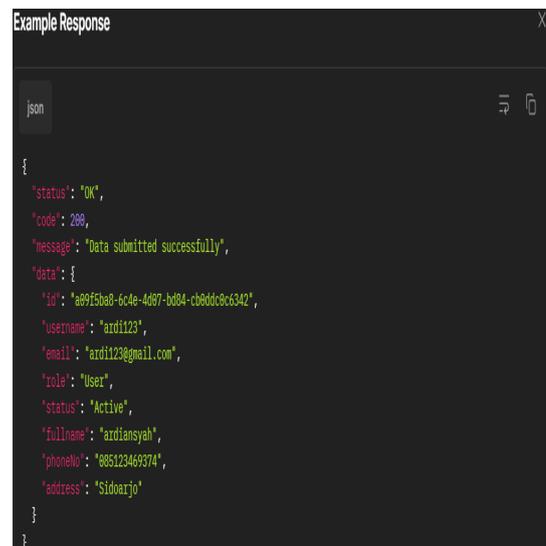
Gambar 14. API Put Update User Response

Pada gambar 14 merupakan respon API put update user ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API put update user. Lalu pada field data terdapat id, username, email, role, status, fullname, phoneNo dan address.



Gambar 15. API Delete User Request

Pada gambar 15 merupakan API delete user digunakan untuk melakukan penghapusan data user. Terdapat authorization untuk melihat token, params untuk melihat id klien dan body raw (json) yang didalamnya terdapat data id, fullname, phone No dan address dan user.

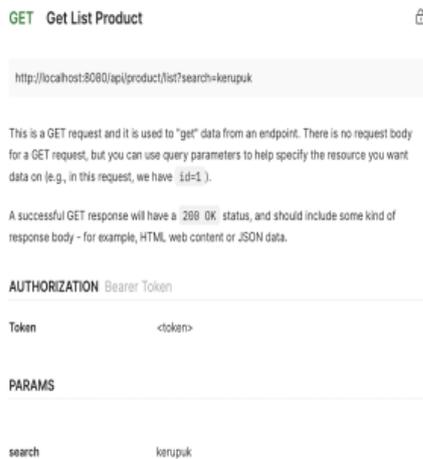


Gambar 16. API Delete User Response

Pada gambar 16 merupakan respon API put delete user ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API delete user. Lalu pada field data terdapat id, username, email, role, status, fullname, phoneNo dan address.

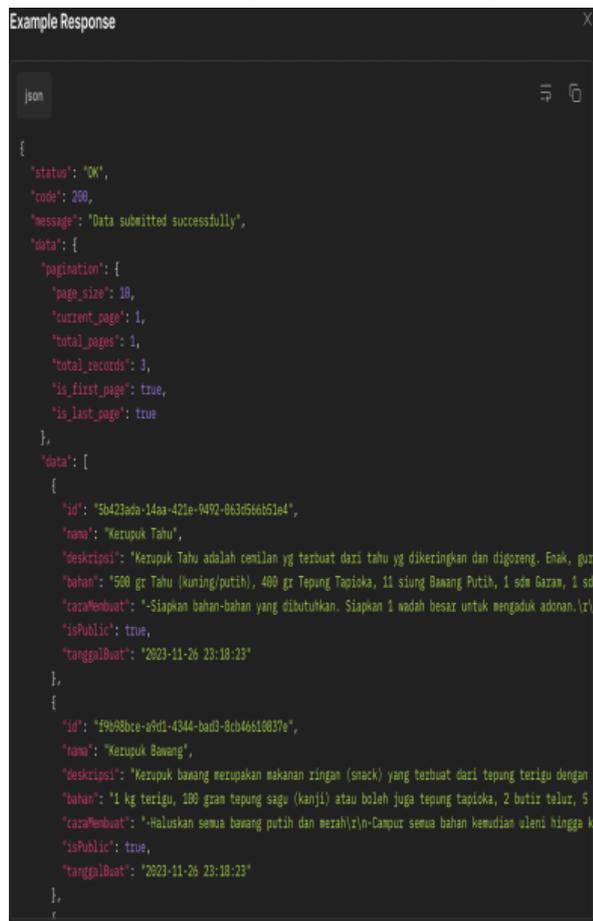
3. Product

a. Akses Authorize User



Gambar 17. API Get List Product Request

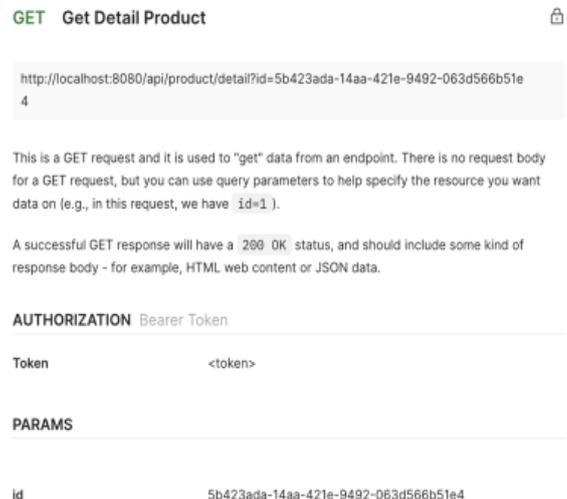
Pada gambar 17 API get list product digunakan untuk melihat list data product. Terdapat authorization untuk melihat token, params untuk melihat pencarian.



Gambar 18. API Get List Product Response

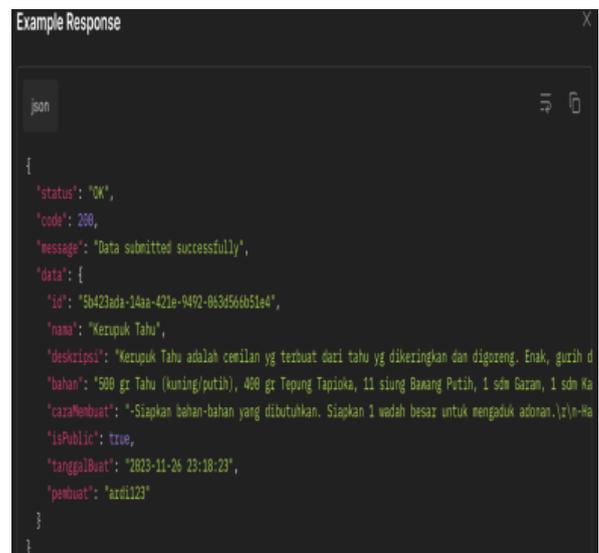
Pada gambar 18 merupakan respon API get list product ada beberapa field yang dikembalikan.

Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API get list product. Lalu pada field data terdapat id, nama, deskripsi, bahan, cara membuat, is public dan tanggal buat.



Gambar 19. API Get Detail Product Request

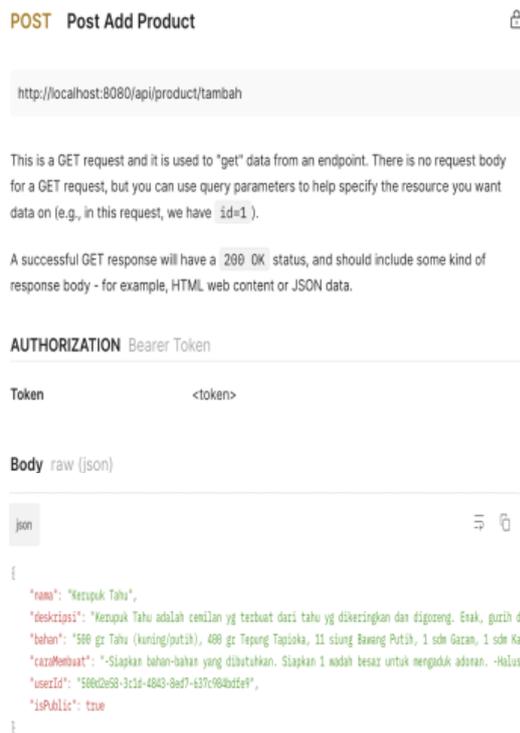
Pada gambar 19 merupakan API get detail product digunakan untuk melihat detail data product. Terdapat authorization untuk melihat token, params untuk melihat id.



Gambar 20. API Get Detail Product Response

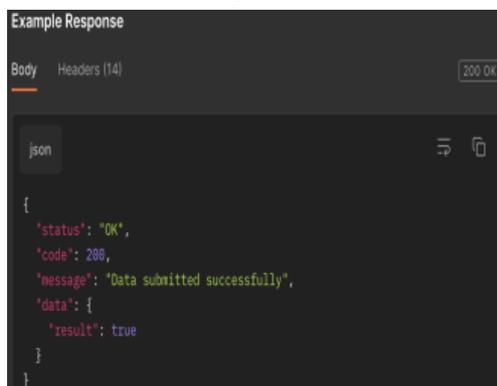
Pada gambar 20 merupakan respon API get detail product ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API get detail product. Lalu pada field data terdapat id, nama, deskripsi,

bahan, cara membuat, is public, tanggal buat dan pembuat.



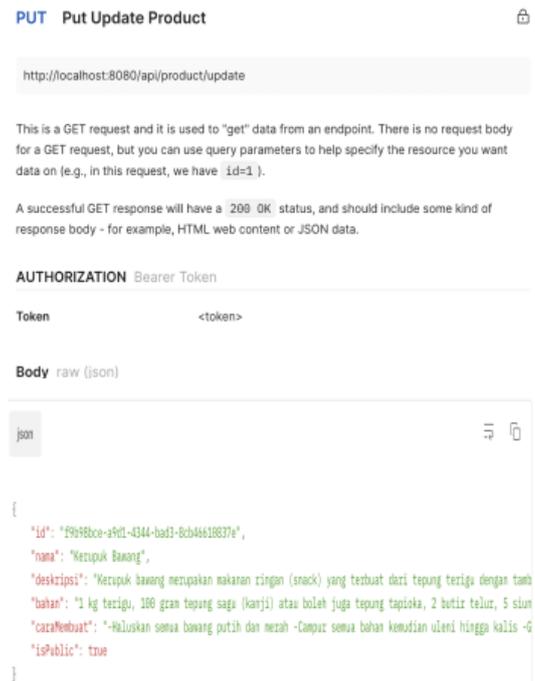
Gambar 21. API Post Add Product Request

Pada gambar 21 merupakan API post add product digunakan untuk menambah produk baru ke database. Untuk memakai atau request ke API ini diperlukan request type berupa body raw (json). Field dalam body request diantaranya nama, deskripsi, bahan, cara membuat, user id dan is public.



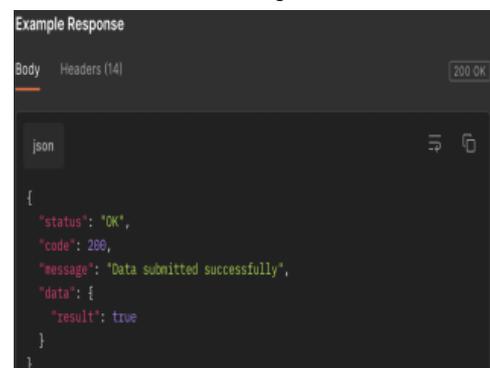
Gambar 22. API Post Add Product Response

Pada gambar 22 merupakan respon API post add product ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API post add product. Lalu pada field data terdapat result.



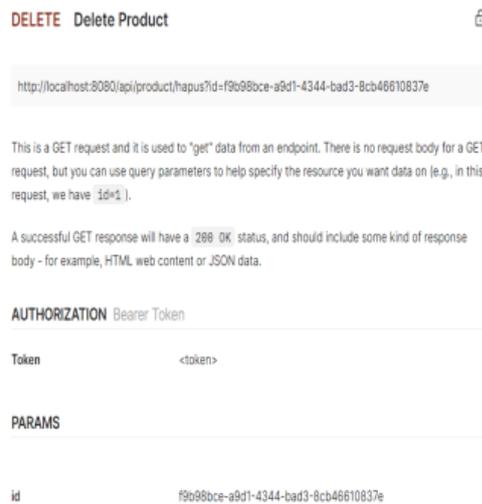
Gambar 23. API Put Update Product Request

Pada gambar 23 merupakan API put update product digunakan untuk melakukan update produk pada database. Untuk memakai atau request ke API ini diperlukan request type berupa body raw (json). Field dalam body request diantaranya id, nama, deskripsi, bahan, cara membuat dan is public.



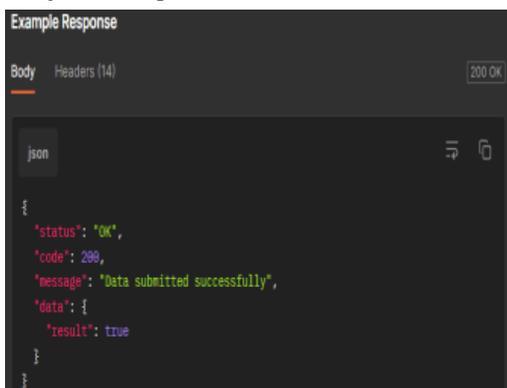
Gambar 24. API Post Add Product Response

Pada gambar 24 merupakan respon API put update produk ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API put update produk yang berisi informasi kredensial daftar produk.



Gambar 25. API Delete Product Request

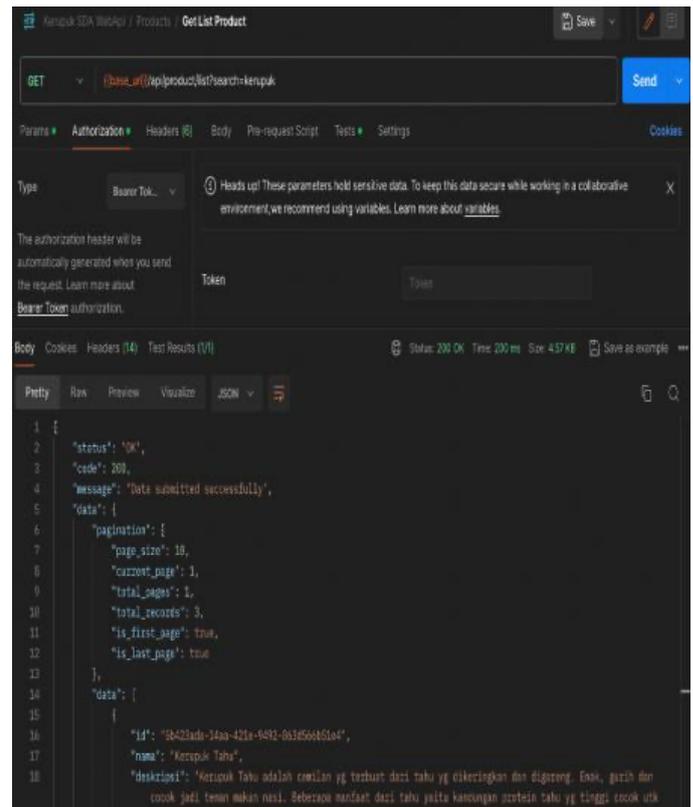
Pada gambar 25 merupakan respon API delete product digunakan untuk melakukan penghapusan produk pada server. Terdapat authorization yang menunjukkan token dan juga params yang menunjukkan id produk.



Gambar 26. API Delete Product Response

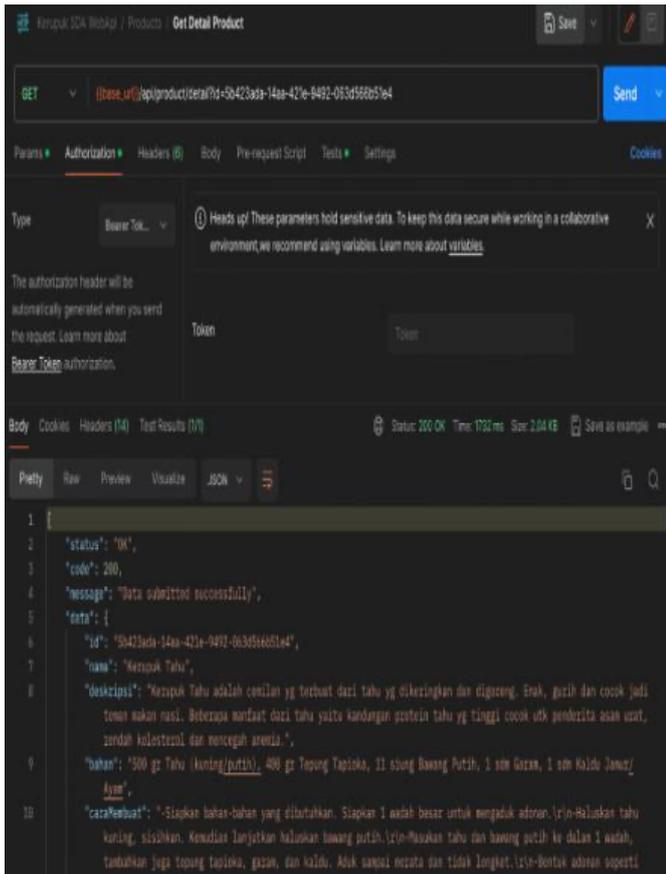
Pada gambar 26 merupakan respon API delete product ada beberapa field yang dikembalikan. Terdapat field status menunjukkan http status, code yang menunjukkan http status code, message yang merupakan informasi pesan dari response, dan field data merupakan objek respon API delete produk.

b. Akses Guest User



Gambar 27. Akses Guest User List Product

Pada gambar 27 merupakan akses *guest user* untuk *list product* ada beberapa field yang dikembalikan. Terdapat *field* status menunjukkan http status, *code* yang menunjukkan http status code, *message* yang merupakan informasi pesan dari *response*, dan *field data* merupakan objek respon API *list* produk. Dalam *field data* terdapat dua bagian yaitu *pagination* dan *data*. Dalam *pagination* terdapat *page size*, *current page*, *total pages*, *total records*, *is first page* dan *is last page*.



Gambar 28. Akses Guest User Detail Product

Pada gambar 28 merupakan akses *guest user* untuk *detail product* ada beberapa *field* yang dikembalikan. Terdapat *field* status menunjukkan http status, *code* yang menunjukkan http status code, *message* yang merupakan informasi pesan dari *response*, dan *field data* merupakan objek respon API *detail* produk. Dalam *field data* terdapat terdapat *id*, nama, deskripsi, bahan dan cara membuat.

C. Pengujian *Blackbox*

TABEL III
BLACKBOX TESTING AUTHENTICATION

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melakukan <i>sign in</i>	Akan menerima respon API <i>sign in</i> dan memunculkan informasi kredensial <i>user login</i>	Muncul informasi kredensial <i>user login</i>	Sukses
2	Melakukan <i>sign up</i>	Akan menerima respon API <i>sign up</i> untuk daftar akun klien dan memunculkan result yang	Muncul result yang merupakan sukses submit dari <i>sign up user</i>	Sukses

		merupakan sukses submit dari <i>sign up user</i>		
--	--	--	--	--

TABEL IIIII
BLACKBOX TESTING USER

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melakukan <i>get list user</i>	Akan menerima respon API <i>get list user</i> dan akan menampilkan list data klien yang terbaru yang berisi <i>id</i> , <i>username</i> , <i>email</i> , <i>role</i> dan status	Muncul informasi list data klien yang terbaru yang berisi <i>id</i> , <i>username</i> , <i>email</i> , <i>role</i> dan status	Sukses
2	Melakukan <i>get detail user</i>	Akan menerima respon API <i>get detail user</i> dan akan menampilkan detail data klien terbaru yang terdapat <i>id</i> , <i>username</i> , <i>email</i> , <i>role</i> , status, <i>fullname</i> , <i>phoneNo</i> dan <i>address</i>	Muncul informasi detail data klien terbaru yang terdapat <i>id</i> , <i>username</i> , <i>email</i> , <i>role</i> , status, <i>fullname</i> , <i>phoneNo</i> dan <i>address</i>	Sukses
3	Melakukan <i>put update user</i>	Akan menerima respon API <i>put update user</i> dan akan menampilkan update data user yang terbaru berupa data <i>id</i> , <i>fullname</i> , <i>phoneNo</i> dan <i>address user</i>	Muncul informasi update data user yang terbaru berupa data <i>id</i> , <i>fullname</i> , <i>phoneNo</i> dan <i>address user</i>	Sukses
4	Melakukan <i>delete user</i>	Akan menerima respon API <i>put delete user</i> untuk melakukan penghapusan data user dan memunculkan result yang merupakan sukses <i>delete data user</i>	Muncul result yang merupakan sukses <i>delete data user</i>	Sukses

TABEL IVV
BLACKBOX TESTING PRODUCT

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Melakukan <i>get list product</i>	Akan menerima respon API <i>get list product</i> dan akan menampilkan list data product berisi <i>id</i> , nama, deskripsi,	Muncul informasi list data product berisi <i>id</i> , nama, deskripsi,	Sukses

		deskripsi, bahan, cara membuat, is public dan tanggal buat	bahan, cara membuat, is public dan tanggal buat	
2	Melakukan <i>get detail product</i>	Akan menerima <i>respon API get detail product</i> dan akan menampilkan <i>detail data product</i> terdapat id, nama, deskripsi, bahan, cara membuat, is public, tanggal buat dan pembuat	Muncul informasi <i>detail data product</i> terdapat id, nama, deskripsi, bahan, cara membuat, is public, tanggal buat dan pembuat	Sukses
3	Melakukan <i>post add product</i>	Akan menerima <i>respon API post add product</i> digunakan untuk menambah produk baru ke <i>database</i> dan memunculkan <i>result</i> yang merupakan sukses <i>add product</i>	Muncul <i>result</i> yang merupakan sukses <i>add product</i>	Sukses
4	Melakukan <i>put update product</i>	Akan menerima <i>respon API put update produk</i> dan akan menampilkan <i>update data product</i> yang terbaru berupa id, nama, deskripsi, bahan, cara membuat dan is public	Muncul Informasi <i>update data product</i> yang terbaru berupa id, nama, deskripsi, bahan, cara membuat dan is public	Sukses
5	Melakukan <i>delete product</i>	Akan menerima <i>respon API delete product</i> digunakan untuk melakukan penghapusan produk pada server dan memunculkan <i>result</i> yang merupakan sukses <i>delete data product</i>	Muncul <i>result</i> yang merupakan sukses <i>delete data product</i>	Sukses

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, pengembangan layanan sistem informasi berbasis WebAPI. Karena terdapat beberapa produk kerupuk unggulan pada UMKM di Kecamatan Tulangan, Sidoarjo, peneliti membuat sistem layanan informasi yang dapat membantu memberikan

detail informasi kerupuk produksi UMKM kepada klien atau user yang mengakses sistem tersebut dengan WebAPI.

Pertimbangan penggunaan arsitektur microservice agar sistem dapat terisolasi dan berfokus pada modul tertentu. Dengan arsitektur microservice, ketika terdapat pengembangan selanjutnya baik minor maupun major dapat dilakukan secara parsial atau paralel. Pengembangan dengan menerapkan API first juga dilakukan dengan tujuan meningkatnya kualitas dokumentasi proyek agar memberikan pesan literasi service produk WebAPI sehingga pengembangan layanan agar terbangun sesuai kebutuhan.

V. SARAN

Dari penelitian dan pengembangan sistem layanan informasi berbasis OpenAPI yang telah dilakukan oleh peneliti, saran pengembangan penelitian selanjutnya adalah mempublikasikan Web API yang telah dibuat pada hosting internet agar dapat diintegrasikan oleh klien dan dikonsumsi publik secara langsung. Pengembangan WebAPI juga dapat diperbarui menggunakan User Interface sehingga memberi pengalaman pengguna yang baik.

UCAPAN TERIMA KASIH

Peneliti senantiasa mengucapkan puji syukur kepada Tuhan Yang Maha Esa lagi Maha Pengasih atas segala pertolonganNya, sehingga peneliti dapat menyelesaikan artikel ini dengan baik. Terima kasih juga peneliti sampaikan terhadap kedua orang tua yang selalu memberikan motivasi beserta nasihat-nasihat, Dosen Pembimbing Skripsi yang selalu membimbing peneliti serta menasihati, tidak lupa juga peneliti ucapkan kepada teman-teman seperjuangan yang selalu memberi dukungan terhadap peneliti dalam melakukan penelitian hingga selesai.

REFERENSI

- [1] M. Dudjak and G. Martinović. (2020). "An API-first methodology for designing a microservice-based backend as a service platform". *Inf. Technol. Control*, vol. 49, no. 2, pp. 206–223. doi: 10.5755/j01.itc.49.2.23757.
- [2] Samuel, & Girsang, A. S. (2020). "Implementation of service oriented architecture using web API & SOMA in e-commerce web application". *International Journal of Emerging Trends in Engineering Research*. Vol. 8, No. 7, 3410–3419.
- [3] Kurniawan Yosua & Yerymia Alfa S. (2020). "Perancangan Sistem Asset Management Berbasis Web Menggunakan Framework Spring Boot (Studi Kasus: PT XYZ)". Fakultas Teknologi Informasi Universitas Kristen Satya Wacana.
- [4] Sibarani, Niko Sumanda, Munawa, Ghifari, Wisnuadhi, Bambang. (2018). "Analisis Performa Aplikasi Android pada Bahasa Pemrograman Java dan Kotlin". *Industrial Research Workshop and National Seminar*. Vol. 9.
- [5] Juniardi Dermawan S. H. (2017). "Implementasi Model Waterfall Pada Pengembangan Sistem Informasi Perhitungan Nilai Mata Pelajaran Berbasis Web Pada Sekolah Dasar Al-Azhar Syifa Budi Jatibening". *Notes and Queries*. Vol. VII, No. 159, 37. <https://doi.org/10.1093/nq/s5-VII.159.37-a>

- [6] Buwono & Robby Cokro. (2019). “*Web Services Menggunakan Format JSON*”. *Jurnal Teknologi Informasi*, Vol. XIV No. 2, 1-10.
- [7] Kementerian Koordinator Bidang Perekonomian Republik Indonesia.
- [8] Abdul Halim. (2020). “Pengaruh Pertumbuhan Usaha Mikro Kecil Dan Menengah Terhadap Pertumbuhan Ekonomi Kabupaten Mamuju”. *Jurnal Ilmiah Ekonomi Pembangunan*. Vol. 1, No.2.
- [9] Bayu Aji P., Dwi Bayu Aji, Mukron Syahroni, Nurul Tri Sukma Aji, Aries Saifudin. (2020). “Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions”. *Jurnal Teknologi Sistem Informasi dan Aplikasi*. Vol.3 No.3.
- [10] Putra, R. A. (2018). “Analisa Implementasi Arsitektur Microservices Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka (*Open Day Light Dev Ops Community*)”. *Jurnal Sistem Informasi Teknologi Informasi Dan Komputer (Just It) Universitas Bina Nusantara Magister Manajemen Sistem Informasi Jakarta*, 150–162.