

Laporan Final Project
Mata Kuliah Komputasi Awan
Institut Teknologi Sepuluh Nopember



Anggota Kelompok

Bryan Yehuda Mannuel
(05311940000021)
I Gde Ardha Semaranatha
(05311940000034)
Prananda Nur Widiastara
(05311940000048)

Penjelasan Aplikasi

Kelompok kami berencana untuk melakukan pembuatan sebuah Serverless Computing API menggunakan kombinasi dari berbagai macam servis yang diberikan oleh Amazon AWS. Disini Serverless Computing API kami akan digunakan untuk memberikan berbagai macam informasi mengenai koleksi produk Gaming Gear mulai dari nama produk, tipe produk, harga, spesifikasi dan berbagai macam informasi menarik lainnya dari berbagai macam brand-brand yang terkenal di dunia Gaming. Serverless Computing API kami juga akan dilakukan hosting di dalam Amazon AWS sehingga nantinya API yang dihasilkan ini nanti akan bisa dilakukan akses lebih lanjut menggunakan Key untuk bisa mengirimkan data dalam bentuk JSON yang nantinya bisa dilakukan manipulasi lebih lanjut atau ditampilkan dalam sebuah website Front-End. Aplikasi ini akan kami namakan GamingPI yang terdiri dari 2 kata yaitu Gaming dan API.

Fitur Aplikasi

- Create : Melakukan penambahan data pada database GamingPI menggunakan method POST
- Read : Melakukan pembacaan data database GamingPI menggunakan method GET
- Update : Melakukan edit data pada database GamingPI menggunakan method PATCH
- Delete : Melakukan penghapusan data pada database GamingPI menggunakan method DELETE
- Authentication : GamingPI ini sudah diamankan menggunakan fitur Key Authentication sehingga keamanan datanya terjamin
- Serverless Computing : Menggunakan service dari AWS Lambda, GamingPI ini tidak membutuhkan sebuah server yang harus menyala 24 jam sehari agar bisa dilakukan akses. Hal ini mengakibatkan beban kerja dari GamingPI ini menjadi ringan sehingga lebih mudah saat melakukan handle Request yang banyak. Hal ini juga akan mengakibatkan biaya servis yang dikeluarkan untuk menyewa Server menjadi tidak ada dan akan menjadi lebih murah.
- Highly Scalable : Menggunakan berbagai macam service dalam AWS seperti Amazon DynamoDB, Amazon Lambda, dan juga Amazon API Gateway yang highly scalable, membuat GamingPI menjadi sangat highly scalable untuk bisa handle Request yang lebih banyak saat nanti sudah lebih berkembang.
- Kecepatan Pemrosesan Data : Karena GamingPI bersifat Serverless Computing, maka kecepatan Pemrosesan Data tidak akan terhambat oleh kecepatan server sehingga kecepatan Pemrosesan Data menjadi lebih cepat secara signifikan berkat adanya AWS Lambda.
- Secure Logging : Dengan penambahan Amazon Cloudwatch, setiap aktivitas yang terjadi dalam GamingPI akan tercatat, mengakibatkan reliability dan security dari GamingPI menjadi lebih meningkat dengan pencatatan yang akurat dan detail.
- API Maintenance : Menggunakan Amazon API Gateway, pemeliharaan data API yang ada di dalam database GamingPI menjadi lebih mudah dan cepat, mengakibatkan mudahnya data-data di dalam database untuk semakin berkembang.
- Accessible : Penggunaan GamingPI sangat mudah dan bisa dilakukan akses dimanapun dan kapanpun menggunakan 1 link dan 1 key saja.

Fungsionalitas

GamingPI bisa memberikan berbagai macam informasi yang menarik, update, dan juga akurat mengenai berbagai macam namun tidak terbatas pada Gaming Gear yang dibuat oleh berbagai macam manufacturer terkenal. GamingPI memberikan berbagai macam informasi yang lengkap seperti tipe produk, harga, merek, spesifikasi, material, dan berbagai macam informasi yang berguna lainnya untuk kita.

GamingPI bisa memberikan bantuan bagi kita semua yang ingin melakukan pembelian sebuah peralatan gaming ataupun peralatan komputer sehari-hari namun tidak tahu harus mencari dimana sebuah situs terintegrasi yang melakukan kurasi terhadap banyaknya tipe dan jenis Gaming Gear yang ada di dunia ini. Kita juga bisa melakukan perbandingan antara berbagai macam produk dari berbagai macam brand yang ada, mencari harga termurah yang sesuai budget, melihat opsi warna dan ukuran, dan berbagai macam hal lainnya untuk menghemat waktu kita.

GamingPI juga bisa memberikan bantuan bagi para penjual Gaming Gear untuk bisa melakukan pencarian informasi mengenai berbagai macam spesifikasi produk yang mereka jual tanpa perlu melakukan pencarian satu-persatu. Mereka juga bisa melihat harga jual yang disarankan oleh manufacturer sehingga bisa lebih bersaing di pasaran. Dengan adanya GamingPI, niscaya para penjual barang bisa mendapatkan lebih banyak lagi keuntungan.

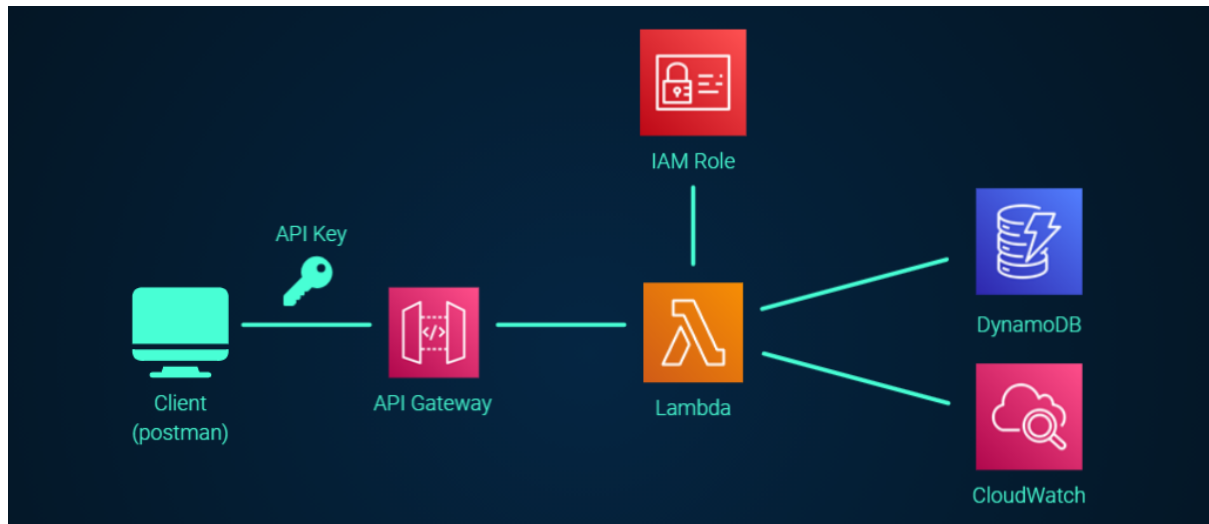
Penggunaan Servis AWS

1. AWS Lambda : AWS Lambda adalah sebuah servis yang disediakan oleh AWS agar dapat dilakukan komputasi / menjalankan kode tanpa perlu adanya sebuah server. Hal ini mengakibatkan kita tidak perlu repot-repot dalam pembuatan, pengelolaan, ataupun administrasi sebuah server. Tanpa adanya server kita juga tidak perlu melakukan pembiayaan tambahan untuk menyewanya. Tanpa adanya server, beban kerja dari Kode juga akan semakin ringan, sehingga bisa melayani lebih banyak request. AWS Lambda juga sangat highly scalable sehingga jika suatu saat GamingPI sudah berkembang, bisa dilakukan penambahan performa secara mudah dan cepat.
2. AWS API Gateway : Amazon API Gateway digunakan sebagai Endpoint untuk bisa dilakukan akses terhadap GamingPI. Disini Amazon API Gateway Memudahkan pengembang untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan API yang sangat highly scalable. Amazon API Gateway memudahkan user dalam melakukan akses endpoint, memberikan key untuk authentication, dan melakukan pengiriman response secara cepat terhadap semua request yang ada.
3. AWS DynamoDB : AWS DynamoDB digunakan sebagai penyimpanan database dalam GamingPI. AWS DynamoDB bisa memberikan reliabilitas penyimpanan data yang bisa diandalkan sehingga kami yakin datanya tidak akan hilang. Selain itu, AWS DynamoDB digunakan karena Database NoSQL lebih fleksibel didalam struktur datanya, sehingga cocok untuk GamingPI yang menyimpan dokumen JSON. AWS DynamoDB juga sangat highly scalable sehingga jika suatu saat GamingPI sudah berkembang, bisa dilakukan penambahan performa secara mudah dan cepat.
4. AWS IAM Role : AWS IAM Role digunakan untuk menambah 1 lapisan lagi keamanan dengan memberikan role atau privilege ke berbagai macam servis AWS

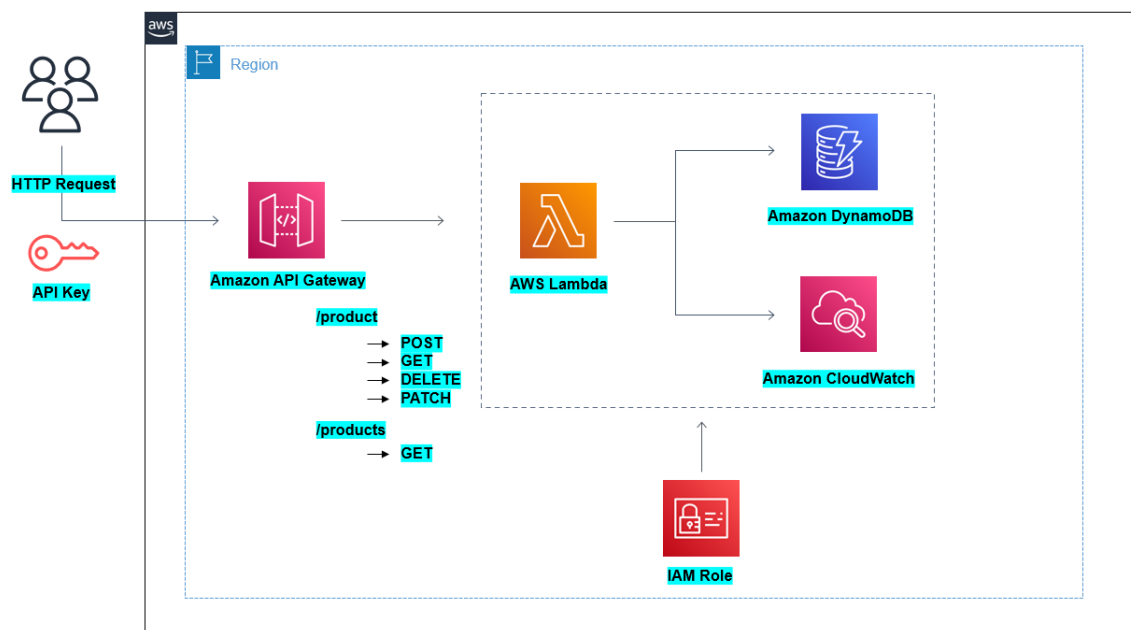
yang ada di dalam GamingPI sehingga suatu servis hanya bisa melakukan dan dilakukan akses sesuai dengan Role yang sudah diberikan.

5. AWS Cloudwatch : AWS Cloudwatch akan dilakukan pemasangan di dalam AWS Service sehingga setiap proses yang berjalan di dalamnya akan dilakukan pencatatan secara detail dan lengkap. Hal ini akan menambah 1 lagi lapisan keamanan di dalam GamingPI karena setiap proses yang berjalan akan bisa tercatat tanpa terkecuali sehingga tidak ada proses yang berjalan yang tidak kami ketahui.

Flow Aplikasi



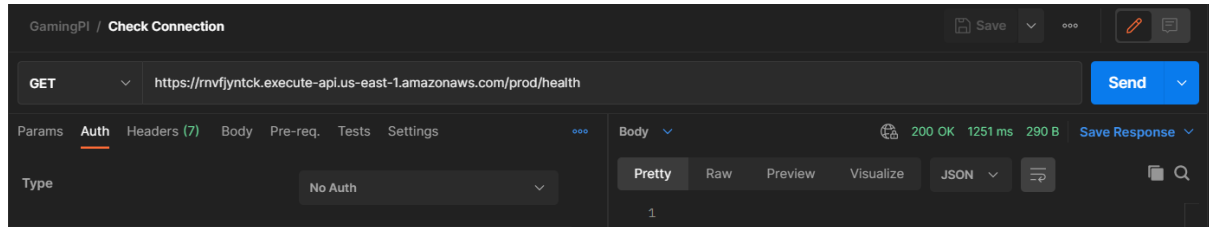
Arsitektur Aplikasi



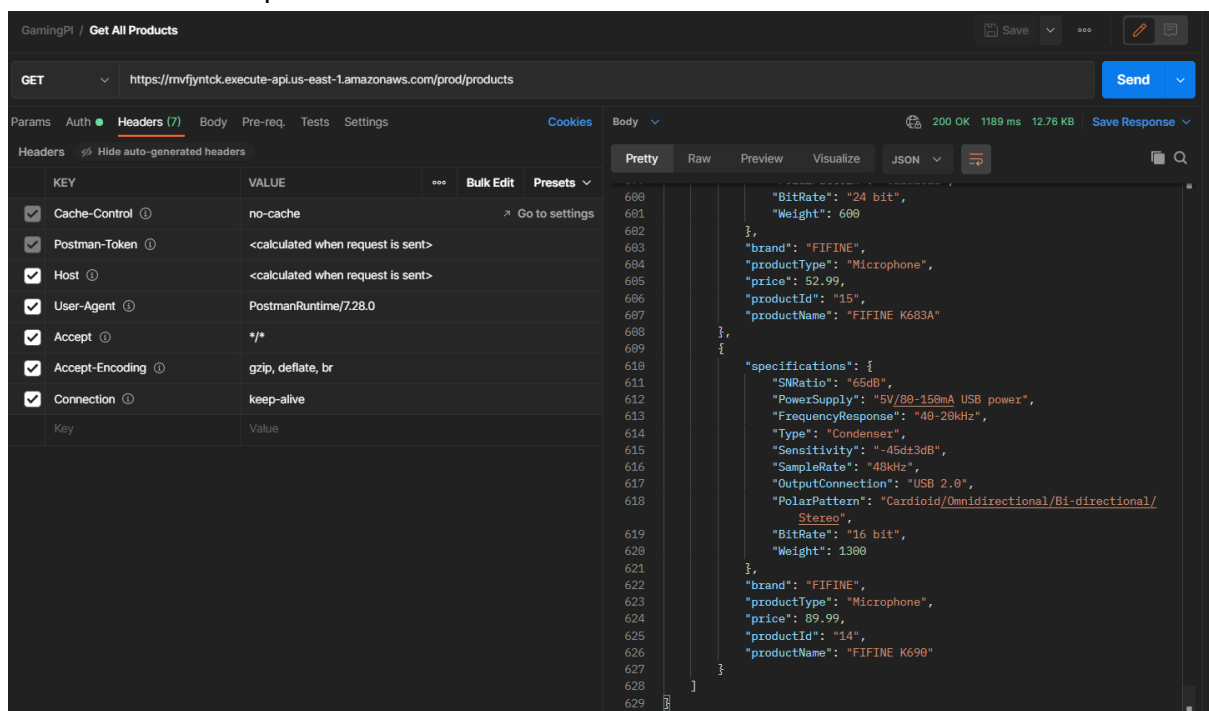
Dokumentasi Aplikasi

Link dokumentasi API -> [GamingPI \(getpostman.com\)](https://getpostman.com)

- Cek koneksi ke API



- GET semua produk



- GET produk berdasarkan Produk ID

The screenshot shows a REST client interface for a "GamingPI" project. The request is a GET to `https://mvfjyntck.execute-api.us-east-1.amazonaws.com/prod/product?productId=1`. The response is a JSON object with the following structure:

```
1 {
2   "specifications": {
3     "design": {
4       "Length": 436.7,
5       "Color": "Metallic Black",
6       "Illumination": "Dynamic Per Key RGB Illumination",
7       "TopMaterial": "Aircraft Grade Aluminum Alloy Frame",
8       "Width": 139.2,
9       "N-KeyRollOver": 104,
10      "Weight": 2.14,
11      "Depth": 40.3,
12      "Anti-Ghosting": "100%"
13    },
14    "switch": {
15      "ActuationPoint": "0.4mm to 3.6mm",
16      "TypeName": "OmniPoint Adjustable Mechanical Switch (Analog Hall Effect Magnetic Sensor)",
17      "Lifetime": "100 Million Keypresses",
18      "responseTime": "0.7ms"
19    }
20  },
21   "brand": "Steelseries",
22   "productType": "Mechanical Keyboard",
23   "price": 289.99,
24   "productId": "1",
25   "productName": "APEX PRO"
26 }
```

- Menambahkan produk menggunakan POST

The screenshot shows a REST client interface for a "GamingPI" project. The request is a POST to `https://mvfjyntck.execute-api.us-east-1.amazonaws.com/prod/product`. The request body is a JSON object representing a new product. The response is a JSON object indicating the operation was successful.

```
1 {
2   "productId": "31",
3   "productName": "G311 GAMING HEADSET",
4   "brand": "Logitech",
5   "productType": "Gaming Headset",
6   "price": 33.99,
7   "specifications": {
8     "Driver": "50mm",
9     "FrequencyResponse": "100Hz-10,000Hz",
10    "Impedance": "35 Ω",
11    "SoundPressure": "116dBSPL/mW at 1kHz",
12    "Weight": 240,
13    "Microphone": "Cardioid"
14  }
15 }
```

```
1 {
2   "Operation": "SAVE",
3   "Message": "SUCCESS",
4   "Item": {
5     "productId": "31",
6     "productName": "G311 GAMING HEADSET",
7     "brand": "Logitech",
8     "productType": "Gaming Headset",
9     "price": 33.99,
10    "specifications": {
11      "Driver": "50mm",
12      "FrequencyResponse": "100Hz-10,000Hz",
13      "Impedance": "35 Ω",
14      "SoundPressure": "116dBSPL/mW at 1kHz",
15      "Weight": 240,
16      "Microphone": "Cardioid"
17    }
18  }
19 }
```

- Update produk menggunakan PATCH

The screenshot shows a REST client interface for a "GamingPI" project. The request is a PATCH to `https://mvfjyntck.execute-api.us-east-1.amazonaws.com/prod/product`. The request body is a JSON object specifying the update operation. The response is a JSON object indicating the update was successful.

```
1 {
2   "productId": "31",
3   "updateKey": "productName",
4   "updateValue": "GG GAMING HEADSET"
5 }
```

```
1 {
2   "Operation": "UPDATE",
3   "Message": "SUCCESS",
4   "UpdatedAttributes": {
5     "Attributes": {
6       "productName": "GG GAMING HEADSET"
7     }
8   }
9 }
```

- Menghapus produk menggunakan DELETE

The screenshot shows a REST client interface with a DELETE request to `https://mvfjyntck.execute-api.us-east-1.amazonaws.com/prod/product`. The request body is a JSON object with `"productId": "31"`. The response is a 200 OK status with a JSON body indicating a successful deletion operation.

Request:

```
1 {
2   "productId": "31"
3 }
```

Response:

```
1 {
2   "Operation": "DELETE",
3   "Message": "SUCCESS",
4   "Item": {
5     "Attributes": {
6       "specifications": {
7         "Driver": "50mm",
8         "FrequencyResponse": "100Hz-10,000Hz",
9         "Microphone": "Cardioid",
10        "Impedance": "35 Ω",
11        "SoundPressure": "116dBSPL/mW at 1kHz",
12        "Weight": 240
13      },
14      "brand": "Logitech",
15      "productType": "Gaming Headset",
16      "price": 33.99,
17      "productId": "31",
18      "productName": "GG GAMING HEADSET"
19    }
20  }
21 }
```