

Nama : Nicholaus Danispadmanaba Y.

NIM : 18217028

API LinkedIn for Essential Information

A. Tujuan

Application programming interface (API) untuk LinkedIn *Essential Information* ditujukan bagi pengembang yang ingin mengintegrasikan layanan yang ada pada LinkedIn dalam aplikasi mereka. Manfaat yang dapat diperoleh oleh pengembang melalui integrasi tersebut adalah adanya fitur tambahan pada aplikasi yang dikembangkan yang membuat *user* dapat menjalin koneksi bisnis secara lebih mudah. Oleh karena itu, integrasi yang kami tawarkan berupa layanan untuk menampilkan detail akun dan melakukan aksi “*Connect*” dengan akun LinkedIn tertentu. Berikut ini merupakan beberapa *use case* yang mungkin dapat menjadi referensi dari penggunaan API ini:

1. Pembuatan aplikasi perusahaan/unit bisnis/divisi yang menampilkan profil akun LinkedIn dari masing-masing karyawan dalam perusahaan/unit bisnis/divisi tersebut. Integrasi API akan memudahkan karyawan untuk saling mengetahui profil kerja satu sama lain secara langsung tanpa harus membuka aplikasi LinkedIn untuk melakukan pencarian akun.
2. Pembuatan *website* agar perusahaan dapat mengenalkan produk dan jasa kepada konsumen. Integrasi API dengan *website* tersebut akan memungkinkan konsumen untuk melihat secara langsung profil, pendidikan, dan tempat kerja dari masing-masing karyawan yang berinteraksi langsung dengan konsumen. Konsumen yang memiliki akun LinkedIn juga dapat menjalin koneksi secara langsung dengan karyawan yang bersangkutan.

a. Business Requirement

Berikut ini merupakan beberapa kebutuhan bisnis yang dapat dipenuhi dengan adanya API LinkedIn:

1. Memberikan informasi profil akun LinkedIn.
2. Memberikan informasi pendidikan dari akun LinkedIn.

3. Memperoleh informasi tempat kerja dari akun LinkedIn.

b. Functional Requirement

Berikut ini merupakan beberapa kebutuhan fungsional yang diturunkan dari kebutuhan bisnis yang ditawarkan oleh API LinkedIn:

1. API mampu memperoleh informasi profil dari akun LinkedIn.
2. API mampu memperoleh informasi pendidikan dari akun LinkedIn.
3. API mampu memperoleh informasi tempat kerja dari akun LinkedIn.

c. Technical Requirement

Berikut ini merupakan beberapa kebutuhan teknis terkait *environment* yang harus dipenuhi oleh aplikasi *client* yang menggunakan layanan API untuk LinkedIn:

1. Aplikasi *client* mempunyai *library* yang mampu melakukan *parsing* JSON.
2. Aplikasi *client* merupakan aplikasi berbasis *web* dan mendukung HTTP v1.1.
3. API diimplementasikan dengan menggunakan Python versi 3.xx pada *server side*.

B. Methods and API's Base URL

Berikut ini merupakan beberapa *endpoint* yang tersedia dari LinkedIn API beserta dengan metode dan fungsinya masing-masing:

No	Metode	<i>Endpoint</i>	Fungsi
1	GET	/users/accounts/general/{account_id}	Mendapatkan informasi profil dari akun dengan ID tertentu.
2	GET	/users/accounts/education/{account_id}	Mendapatkan informasi pendidikan dari akun dengan ID tertentu.

3	GET	/users/accounts/workplace/{account_id}	Mendapatkan informasi tempat kerja dari akun dengan ID tertentu.
---	-----	--	--

C. Technology to Use

Berikut ini merupakan beberapa teknologi yang akan digunakan pada penggunaan dan implementasi dari LinkedIn API.

a. HTTP

Hypertext Transfer Protocol (HTTP) merupakan sebuah protokol pada *layer* aplikasi (*layer 7*) dari *OSI Model*. Protokol ini memungkinkan terjadinya *transfer* data di antara aplikasi *client* dengan aplikasi *server*. Adapun, beragam format data didukung oleh protokol ini, seperti: *text*, JSON, dan XML.

b. Selenium

Selenium merupakan sebuah *package* yang tersedia untuk Python 3 yang digunakan untuk otomasi *web browser*. *Package* ini digunakan untuk melakukan otomasi kueri pada *web browser* dan *login* pada akun LinkedIn. Tanpa dilakukannya otomasi *login*, maka informasi yang tersedia untuk di-*scrape* nantinya akan menjadi terbatas.

c. Parsel

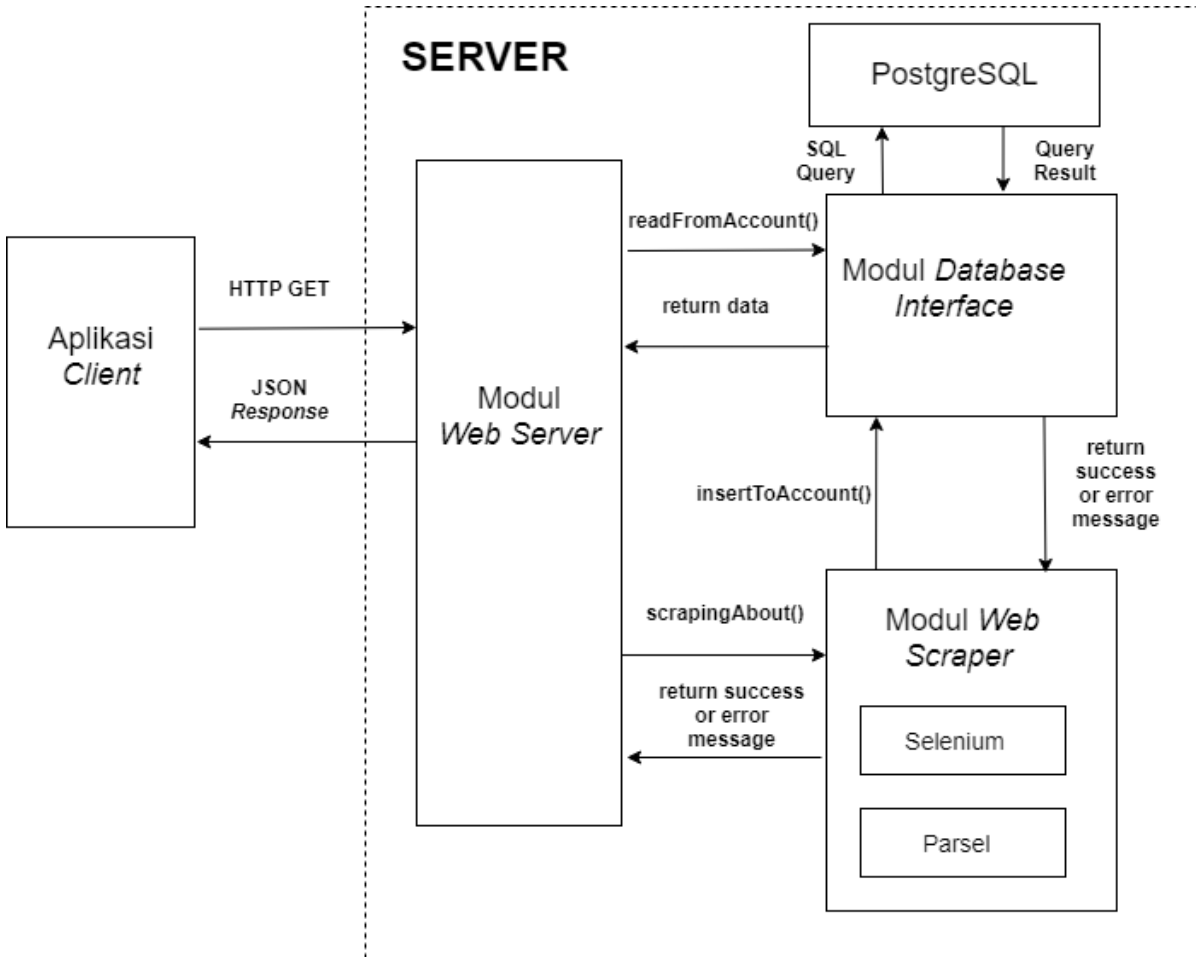
Parsel merupakan sebuah *library* yang tersedia untuk Python 3 yang digunakan untuk melakukan *scraping* pada *web page*. Adapun, proses *scraping* dilakukan dengan mengidentifikasi kelas CSS tertentu pada laman tersebut, lalu mengekstraksi konten teks yang ada di dalam kelas yang bersangkutan.

d. PostgreSQL

PostgreSQL merupakan *relational database management system* (RDMS) yang bertanggung jawab atas pengelolaan basis data beserta tabel di dalamnya dengan format relasional. Bahasa kueri yang digunakan merupakan *Structured Query Language* yang memungkinkan pemanggilan elemen dalam tabel basis data secara cepat dan mudah.

Penggunaan basis data menyebabkan pengaksesan data menjadi semakin cepat karena data hasil *scraping* sudah tersimpan secara lokal dalam *server*.

D. Mechanism



Berikut ini merupakan cara kerja dari LinkedIn API yang akan dibuat:

1. Aplikasi *client* melakukan *request* HTTP GET dengan salah satu *endpoint* (contoh: `/users/accounts/general/15000`)
2. Modul *web server* menerima *request*, lalu memanggil modul *database interface* melalui *method* `readFromAccount()`.
3. Modul *database interface* lalu melakukan SQL *query* pada RDBMS PostgreSQL.

4. PostgreSQL kemudian akan mengembalikan *query result* kepada modul *database interface* yang dapat berupa 0 row (ID akun tidak ditemukan) atau seluruh *field* pada tabel *account* dengan ID yang sesuai.
5. Jika ID ditemukan, maka data akan dikirimkan oleh modul *database interface* ke modul *web server*. Modul *web server* lalu mengkonversi data menjadi JSON, lalu mengirimkannya ke aplikasi *client* dalam bentuk HTTP *response*.
6. Jika ID tidak ditemukan pada tabel, maka modul *database interface* akan mengirimkan pesan “*Invalid AccountID*” ke modul *web server*.
7. Modul *web server* kemudian akan memanggil modul *web scraper* melalui *method* *scrapingAbout()*.
8. Modul *web scraper* kemudian akan melakukan *scraping* dengan memanfaatkan *method-method* pada *package* Selenium dan *library* Parsel. Hasil *scraping* kemudian akan dimasukkan ke dalam tabel *account* pada basis data melalui modul *database interface* sebagai perantara dengan *method* *insertToAccount()*.
9. Modul *database interface* kemudian akan mengirimkan pesan sukses / gagal *insert* ke modul *web scraper*.
10. Modul *web scraper* kemudian akan meneruskan pesan sukses / gagal tersebut ke modul *web server*.
11. Jika modul *web server* menerima pesan sukses, maka *web server* akan memanggil *method* *readFromAccount()* lagi. Data yang diterima dari metode tersebut, lalu dibungkus dalam JSON dan dikirimkan ke aplikasi *client* dalam bentuk HTTP *response*.
12. Jika modul *web server* menerima pesan gagal, maka *web server* akan meneruskannya ke aplikasi *client*.

E. Data Format and Error Handling

Berikut ini merupakan format data JSON sebagai *response body* yang dikirimkan *server*.

a. GET /users/accounts/general/{account_id}

```
{
  "AccountID": "08222a18b",
  "AccountName": "Nicholaus Yosodipuro",
  "AccountTitle": "Information System and Technology Undergraduate
    Student at Bandung Institute of Technology (ITB)",
  "AccountRegion": "Greater Jakarta Area, Indonesia"
}
```

b. GET /users/accounts/education/{account_id}

```
{
  "AccountID": "08222a18b",
  "EducationInstitution": "Institut Teknologi Bandung (ITB)",
  "EducationTitle": "Bachelor's degree"
}
```

c. GET /users/accounts/workplace/{account_id}

```
{
  "AccountID": "805965120",
  "Workplace1": "Packet Systems Indonesia",
  "Workplace2": "Telkom University"
}
```

d. Pesan menunggu proses *scraping*

```
{
  "Feedback": "Account ID not found on database",
  "Message": "Please wait until scraping is finished (2-3 minutes), then reload the page"
}
```

F. Programming Languages

Bahasa pemrograman yang akan digunakan untuk implementasi LinkedIn API adalah Python 3. Hal tersebut dikarenakan Python memiliki *library* dan *package* yang sangat banyak sehingga memudahkan proses pengembangan. Selain itu, banyaknya dukungan dari komunitas membuat proses *troubleshooting* dan *debugging* menjadi cepat.

G. Development Method and Version Control

Berikut ini merupakan langkah-langkah pengembangan yang digunakan dalam proses implementasi LinkedIn API:

1. Membuat desain LinkedIn API.
2. Membuat modul *web scraper*.
3. *Testing* modul *web scraper*.
4. *Push* modul *web scraper* ke *local repository*.
5. Menyiapkan basis data dan tabel menggunakan PostgreSQL.
6. Membuat modul *database interface*.
7. *Testing* modul *database interface*.
8. *Push* modul *database interface* ke *local repository*.
9. Membuat modul *web server* dengan mengintegrasikan modul *web scraper* dan *database interface*.
10. *Testing* modul *web server*.
11. *Push* modul *web server* ke *local repository*.
12. *Push* semua modul ke dalam *remote repository*.

Berikut ini merupakan *version control* yang digunakan dalam pengembangan LinkedIn API:

1. *Local Repository*: Git
2. *Remote Repository*: GitHub (<https://github.com/nicholausdy/LinkedInAPI>)

