

Rancang Bangun *Website Fund Myedu* Untuk Mencari Beasiswa Dengan Menggunakan *Crawling Api*

1st Michael Christensen Bonar Kasparov
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
michaelchristensen@student.telkomuniversity.ac.id

2nd Irfan Darmawan
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
irfandarmawan@telkomuniversity.ac.id

3rd Oktariani Nurul Pratiwi
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
onurulp@telkomuniversity.ac.id

Abstrak — Teknik pengambilan data melalui proses *crawling* merupakan suatu metode otomatis untuk mengambil informasi dari sebuah situs *web* dengan bantuan program komputer. Pendekatan ini memungkinkan pengambilan informasi dari beragam sumber secara efektif dan terorganisir. Contoh penggunaan *Data Crawling* adalah mengetahui kriteria suatu kampus untuk menerima seseorang melalui jalur beasiswa. Banyak kampus mempublikasikan informasi tentang kriteria yang harus dipenuhi oleh calon penerima beasiswa di situs *web* mereka. Namun, mencari dan mengumpulkan informasi secara manual menjadi sangat memakan waktu. Metode yang digunakan pada penelitian ini adalah *Web Crawler API*. Terbukti bahwa metode *web crawling API* dapat secara otomatis mengumpulkan informasi tentang kriteria suatu kampus untuk menerima seseorang melalui jalur beasiswa menggunakan *Data Crawling* yaitu dengan membuat *website*, data *dummy*, dan API. Cara menyajikan data yang sudah diperoleh agar calon mahasiswa bisa mendapatkan rekomendasi beasiswa dari formulir data yang telah diisi sesuai dengan jurusan yang telah diambil di bangku SMA/SMK yaitu dengan dibuatnya *website Fund MyEdu* sebagai tempat menampilkan informasi beasiswa bagi calon penerima beasiswa. Dimana terdapat fitur *Dashboard Admin*, *Lihat Beasiswa*, *Kelola Users*, *Verifikasi Users*, *Beranda*, *Beasiswa*, *Rekomendasi Beasiswa*, dan *Tentang Kami*. *Fund MyEdu* memiliki dua *role* yaitu *admin* dan *users*, *admin* berperan sebagai pengelola dan verifikator data *users* serta data beasiswa. Sedangkan *users* merupakan calon penerima beasiswa.

Kata kunci— *Crawler API*, *Beasiswa*, *Data Crawling*, *Website*

I. PENDAHULUAN

Teknik pengambilan data melalui proses *crawling* merupakan suatu metode otomatis untuk mengambil informasi dari sebuah situs *web* dengan bantuan program komputer. Pendekatan ini memungkinkan pengambilan informasi dari beragam sumber secara efektif dan terorganisir [1].

Salah satu contoh penggunaan *Data Crawling* adalah dalam mengetahui kriteria suatu kampus untuk menerima seseorang melalui jalur beasiswa. Banyak kampus mempublikasikan informasi tentang kriteria yang harus

dipenuhi oleh calon penerima beasiswa di situs *web* mereka. Namun, mencari dan mengumpulkan informasi ini secara manual dapat menjadi sangat memakan waktu [2].

Sebagai contoh, kita dapat menggunakan *Data Crawling* untuk mengumpulkan informasi tentang kriteria yang harus dipenuhi oleh calon penerima beasiswa di Universitas XYZ. Kita dapat mencari dan mengumpulkan informasi tentang persyaratan akademik, pengalaman *volunteering*, dan prestasi yang diperlukan oleh Universitas XYZ untuk menerima seseorang melalui jalur beasiswa. Dengan mengetahui informasi ini, kita dapat lebih terfokus dalam menyiapkan diri untuk mendaftar beasiswa di Universitas XYZ.

Dengan demikian, *Data Crawling* merupakan teknik yang sangat berguna dalam mengetahui kriteria suatu kampus untuk menerima seseorang melalui jalur beasiswa. Tujuan utama dari *crawler* adalah untuk mengumpulkan sejumlah besar informasi yang bermanfaat dari berbagai halaman *web* dengan cepat dan efisien, termasuk mengidentifikasi struktur *link* yang terhubung dengan halaman-halaman tersebut, sehingga kita dapat lebih terfokus dalam menyiapkan diri untuk mendaftar beasiswa di kampus yang diinginkan [3].

II. KAJIAN PUSTAKA

A. *Data Crawling*

Data Crawling adalah proses mengumpulkan data dari internet menggunakan komputer. Data yang dikumpulkan berupa teks, gambar, video, atau jenis data lainnya yang tersedia di internet. *Data Crawling* dilakukan menggunakan program khusus "*crawler*" atau "*spider*" yang dapat menjelajah seluruh *website* dan mengumpulkan data yang diinginkan. Data yang terkumpul digunakan untuk berbagai keperluan, seperti analisis data, pembuatan indeks pencarian, atau pembuatan aplikasi yang memerlukan data dari internet [4].

B. *Web Crawler API*

Data crawling bisa dilakukan dengan metode *Web Crawler API*. Implementasi *Web Crawler* menggunakan bahasa *scripting PHP* dan *PHP Simple HTML DOM Parser* sebagai *Application Programming Interface (API)*, yang

berfungsi untuk menjelajahi dan mengekstrak struktur HTML dari setiap situs yang dikunjungi [5].

C. Bahasa Pemrograman PHP

PHP atau *Hypertext Preprocessor*, adalah bahasa pemrograman *open-source* yang dirancang khusus untuk pengembangan *web* dan dapat disisipkan ke dalam dokumen HTML. PHP memiliki karakteristik mirip bahasa pemrograman C, Java, dan Perl, serta memiliki tingkat pembelajaran yang mudah. PHP berfungsi sebagai bahasa skrip sisi server, yang berarti proses pemrosesan data terjadi di server. Dengan kata lain, server bertanggung jawab menerjemahkan skrip program, dan hasilnya dikirimkan ke klien yang meminta. PHP berperan dalam mengolah data dan mengirimkannya sebagai kode HTML kembali ke peramban *web* [6].

D. Framework Laravel

Laravel adalah *framework* PHP *open-source* untuk pengembangan aplikasi *website*. Laravel menyediakan seperangkat *tools*, *library*, dan konvensi yang membantu pengembangan dalam membangun aplikasi *website* dengan cepat dan efisien.

Fitur utama *framework* Laravel yaitu sistem *routing*, manajemen *database*, sistem *templating*, otentikasi pengguna, serta dukungan pengembangan *website* seperti validasi data, penanganan *session*, dan banyak lagi.

Penggunaan Laravel memberi kemudahan dalam melakukan operasi dasar penciptaan, pembacaan, pembaruan, dan penghapusan data dengan model MVC yang lebih teratur. Juga menghasilkan kinerja yang lebih cepat, stabilitas yang baik saat memuat data, pengamanan data yang baik, serta pemanfaatan fitur-fitur canggih seperti *Blade* dengan konsep HMVC (*Hierarchical Model View Controller*). Terdapat pustaka-pustaka siap pakai, dan tersedia fitur pengelolaan migrasi untuk menyusun skema tabel di *database* [6].

E. Application Programming Interface (API)

API adalah teknologi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. Merupakan antarmuka virtual yang memungkinkan kerja sama antara dua fungsi perangkat lunak, seperti antara pengolah kata dan *spreadsheet*. API menentukan cara *programmer* menggunakan fitur tertentu dari komputer. Berbagai jenis API tersedia untuk berbagai sistem, termasuk sistem *windowing*, sistem file, sistem *database*, dan sistem jaringan [7].

Penggunaan API bertujuan untuk memfasilitasi pertukaran data antara aplikasi yang berbeda, mempercepat proses pengembangan aplikasi dengan menyediakan fungsi-fungsi terpisah, dan menghilangkan kebutuhan untuk merancang fitur yang serupa. API yang beroperasi di tingkat sistem operasi membantu aplikasi dalam berkomunikasi dengan lapisan dasar dan antar aplikasi sesuai dengan serangkaian protokol dan spesifikasi yang telah ditetapkan [8].

F. Postman

Postman adalah aplikasi pada *browser Chrome* yang berperan sebagai *plugin REST Client* untuk melakukan uji coba pada *REST API*. Postman merupakan *platform* GUI untuk mempercepat pengembangan API dengan menyediakan kemudahan dalam pembangunan API, pengujian, dokumentasi, dan berbagi. Postman direkomendasikan untuk digunakan pada sistem operasi Mac, Windows, dan Linux [9].

G. Database phpMyAdmin

PhpMyAdmin adalah perangkat lunak gratis yang dibuat dengan bahasa pemrograman PHP, yang bertujuan melakukan administrasi *database* MySQL melalui antarmuka *web*. PhpMyAdmin mendukung berbagai tugas di dalam *database* MySQL dan MariaDB. Tugas-tugas yang paling umum seperti mengelola *database*, tabel, kolom, relasi, indeks, pengguna, izin, dan lainnya dapat dilakukan melalui antarmuka pengguna, dan memiliki kemampuan untuk menulis perintah SQL secara langsung untuk mengelola *database*.

Pengembangan PhpMyAdmin dimulai pada tahun 1998 oleh Tobias Ratschiller, seorang konsultan IT. Awalnya, Ratschiller bekerja pada proyek MySQL-Webadmin, hasil karya Petrus Kuppelwieser. Ratschiller kemudian menulis ulang kode dari awal untuk PhpMyAdmin dan mengembangkannya [10].

H. Beasiswa Kuliah

Beasiswa kuliah adalah bantuan keuangan yang diberikan kepada seseorang untuk mendukung kelangsungan pendidikan mereka. Bantuan ini bisa disediakan oleh entitas pemerintah, perusahaan, atau yayasan. Beasiswa kuliah bisa bersifat cuma-cuma atau dengan persyaratan ikatan kerja (yang sering disebut sebagai ikatan dinas) setelah menyelesaikan Pendidikan [11].

Badan Pusat Statistik (BPS) menunjukkan bahwa selama satu dekade terakhir, penerima beasiswa atau bantuan pendidikan tercatat mengalami peningkatan dari 3,89% pada 2009 menjadi 20,14% pada 2021. Peningkatan terbesar terjadi pada periode 2015 sampai 2018, di mana persentase pelajar yang menerima beasiswa pada tahun 2018 mencapai 20,28%, naik dua kali lipat dibandingkan tahun 2015 yang sebesar 10,95%. Beasiswa tersebut termasuk Program Indonesia Pintar (PIP), beasiswa dari pemerintah pusat, beasiswa dari pemerintah daerah, beasiswa atau bantuan dari lembaga non pemerintah, dan lainnya [12].



GAMBAR 1
Peningkatan Penerima Beasiswa dari Tahun 2009 – 2021

I. Evaluasi Performansi

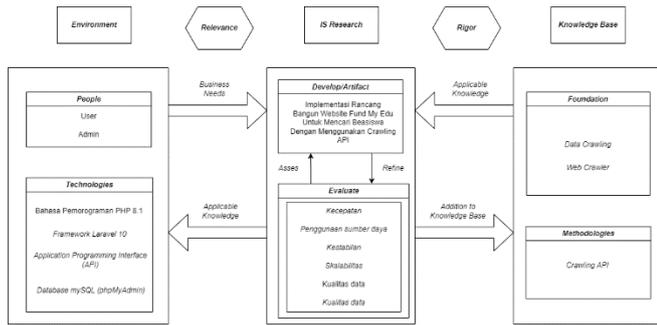
Evaluasi sistem *web crawling* merupakan proses pengecekan dan penilaian terhadap bagaimana sebuah sistem *web crawling* bekerja dalam mengumpulkan data dari sebuah situs *web*. Pada dasarnya, tujuan evaluasi ini adalah untuk mengidentifikasi kelebihan dan kekurangan sistem tersebut, serta memberikan rekomendasi perbaikan yang diperlukan [13].

III. METODE

A. Model Konseptual

Model konseptual atau kerangka konseptual penelitian adalah model yang menunjukkan hubungan logis antara

faktor atau variabel yang telah diidentifikasi. Berikut model yang digunakan dalam penelitian ini seperti pada Gambar 2.

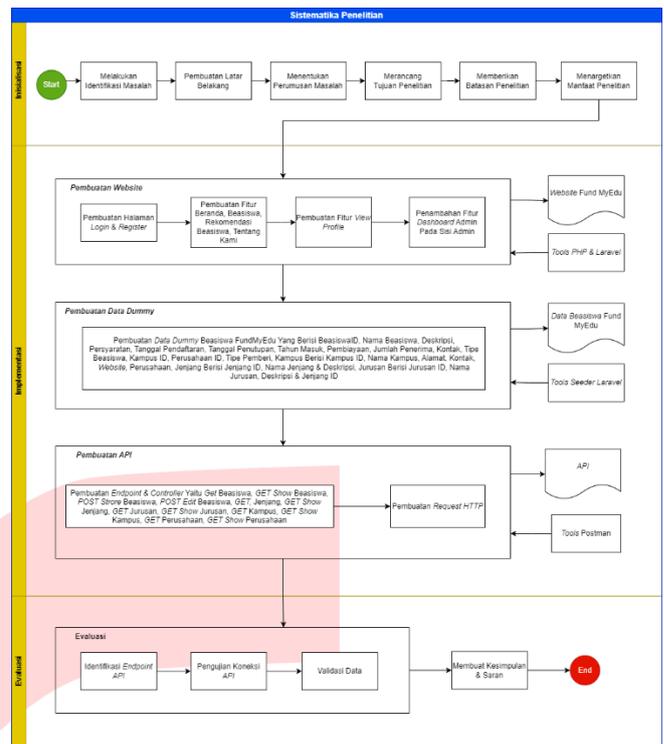


GAMBAR 2 Model Konseptual

Gambar 2 menjelaskan model konseptual yang digunakan pada penelitian ini. Hal pertama yang dilakukan adalah identifikasi objek penelitian. Model konseptual terbagi menjadi tiga bagian diantaranya yakni *Environment* (lingkungan penelitian), *IS Research* (penelitian bidang Sistem Informasi yang dilakukan), dan *Knowledge Base* (basis pengetahuan atau dasar ilmu yang digunakan). *Environment* terbagi menjadi dua bagian yaitu *People* (orang atau aktor yang terlibat dalam penelitian) dan *Technologies* (teknologi yang digunakan dalam penelitian). Dalam penelitian ini *People* mencakup pengguna aplikasi Fund MyEdu yaitu *User* dan *Admin*, sedangkan *Technologies* yang digunakan adalah Bahasa Pemrograman PHP 8.1 *Framework* Laravel 10, *Application Programming Interface* (API), dan *Database* phpMyAdmin. Penelitian ini akan menghasilkan penerapan Implementasi Rancang Bangun *Website* Fund MyEdu Untuk Mencari Beasiswa Dengan Menggunakan *Crawling* API, dimana pada penyusunannya menerapkan metode *Web Crawler* API. Tujuan akhir dari penelitian ini adalah untuk menyelesaikan masalah terkait cara terbaik dalam mengumpulkan informasi tentang kriteria suatu kampus untuk menerima calon mahasiswa menggunakan jalur beasiswa dan bertujuan mengetahui penyajian data yang dapat dipahami oleh calon mahasiswa.

B. Sistematika Penelitian

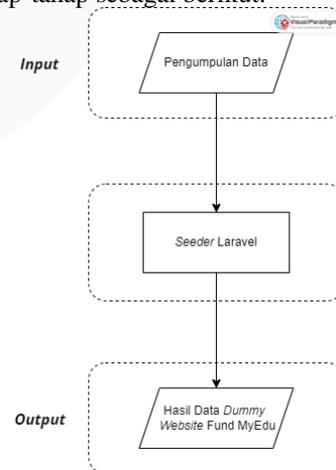
Sistematika penelitian merujuk pada serangkaian tahapan dalam penelitian yang dirancang untuk memahami proses yang akan dijalankan selama penelitian guna mencapai solusi dari setiap permasalahan yang muncul. Dalam penelitian ini, sistematika penelitian direpresentasikan dalam bentuk diagram alur (*flowchart*) yang terdiri dari tiga tahapan utama, yaitu tahap inisialisasi, tahap implementasi, dan tahap evaluasi. Diagram ini dibuat dengan tujuan untuk memberi gambaran visual yang jelas terkait alur kerja penelitian, serta membantu menciptakan struktur dan perencanaan yang terorganisir. Gambar 3 menunjukkan diagram sistematika penelitian yang mencakup tahap-tahap tersebut, yakni dipaparkan sebagai berikut:



GAMBAR 3 Sistematika Penelitian

C. Pengumpulan Data

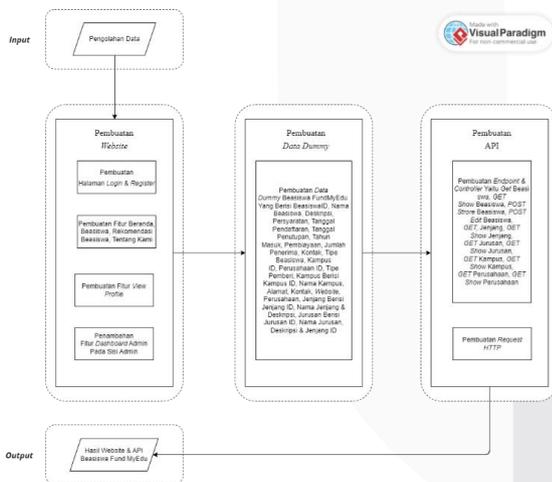
Pengumpulan data yang dilakukan pada penelitian ini menggunakan metode *Seeder* pada Laravel dimana *Seeder* adalah mekanisme yang digunakan untuk mengisi basis data dengan data *dummy* atau awal yang diperlukan untuk pengujian atau pengembangan *Website* Fund MyEdu. *Seeder* dalam Laravel dapat digunakan untuk mengisi data ke dalam tabel-tabel *database* dengan cara yang terstruktur. *Database Seeder* digunakan untuk memasukkan data awal ke dalam tabel *database*, sehingga dapat menguji aplikasi dengan *dataset dummy* atau mengisi basis data dengan data awal yang dibutuhkan. *Dataset dummy* yang dibuat pada awal perancangan berjumlah 1000 data, dimana data tersebut akan terus bertambah pada saat dilakukannya proses *crawling*. Gambar 4 menunjukkan diagram pengumpulan data yang mencakup tahap-tahap sebagai berikut:



GAMBAR 4 Pengumpulan Data

D. Pengolahan Data

Pengolahan data yang dilakukan pada penelitian ini dengan tiga tahap yaitu pembuatan *website*, data *dummy*, dan API. Pada tahap pembuatan *website* dilakukan Pembuatan Halaman *Login & Register*, Pembuatan Fitur Beranda, Beasiswa, Rekomendasi Beasiswa, Tentang Kami, Pembuatan Fitur *View Profile*, Penambahan Fitur *Dashboard Admin* Pada Sisi *Admin*, dimana menggunakan *Tools PHP & Laravel* dan *output* yang dihasilkan adalah *Website Fund MyEdu*. Selanjutnya pada tahap pembuatan data *dummy* dilakukan dengan cara membuat *Data Dummy* Beasiswa *FundMyEdu* Yang Berisi *BeasiswaID*, Nama Beasiswa, Deskripsi, Persyaratan, Tanggal Pendaftaran, Tanggal Penutupan, Tahun Masuk, Pembiayaan, Jumlah Penerima, Kontak, Tipe Beasiswa, Kampus ID, Perusahaan ID, Tipe Pemberi, Kampus Berisi *Kampus ID*, Nama Kampus, Alamat, Kontak, *Website*, Perusahaan, Jenjang Berisi *Jenjang ID*, Nama Jenjang & Deskripsi, Jurusan Berisi *Jurusan ID*, Nama Jurusan, Deskripsi & *Jenjang ID* yang menggunakan *Tools Seeder Laravel* yang nantinya akan menghasilkan *output* Data Beasiswa *Fund MyEdu*. Kemudian pada tahap pembuatan API dilakukan dengan membuat *Endpoint & Controller* yaitu *Get Beasiswa*, *GET Show Beasiswa*, *POST Store Beasiswa*, *POST Edit Beasiswa*, *GET*, *Jenjang*, *GET Show Jenjang*, *GET Jurusan*, *GET Show Jurusan*, *GET Kampus*, *GET Show Kampus*, *GET Perusahaan*, *GET Show Perusahaan* dan Pembuatan *Request HTTP* yang menggunakan *Tools Postman* yang nantinya akan menghasilkan *output* API. Gambar 5 menunjukkan diagram pengolahan data yang mencakup tahap-tahap sebagai berikut:

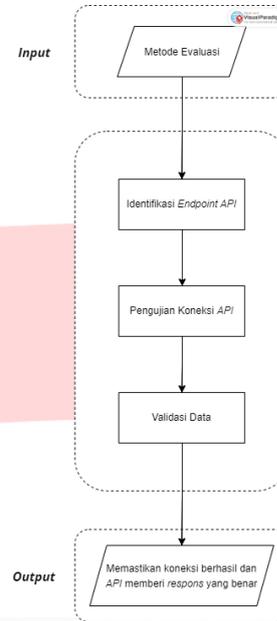


GAMBAR 5
Pengolahan Data

E. Metode Evaluasi

Metode evaluasi adalah strategi perencanaan yang digunakan untuk menilai validitas (keketatan, keandalan, atau keabsahan) dari proses dan hasil penelitian. Pada penelitian ini, tahapan evaluasi data yang dilakukan yaitu menerapkan Identifikasi *Endpoint* API, Pengujian Koneksi API, dan Validasi Data. Dimana Identifikasi *Endpoint* API dengan menemukan *Endpoint* API yang akan digunakan untuk mengakses data dari *website* serta memastikan *endpoint* memberikan data yang diperlukan. Kemudian pengujian koneksi API dengan melakukan pengujian koneksi ke *Endpoint* API untuk memastikan koneksi berhasil dan API dapat memberikan *respons* yang benar. Selain itu

menggunakan perangkat lunak seperti Postman untuk melakukan pengujian ini. Selanjutnya Validasi Data dengan mengambil data dari API untuk memastikan keakuratannya serta konsistensinya dan untuk memastikan data yang diterima sesuai format yang diharapkan. Gambar 6 menunjukkan diagram metode evaluasi yang mencakup tahap-tahap sebagai berikut:



GAMBAR 6
Metode Evaluasi

IV. HASIL DAN PEMBAHASAN

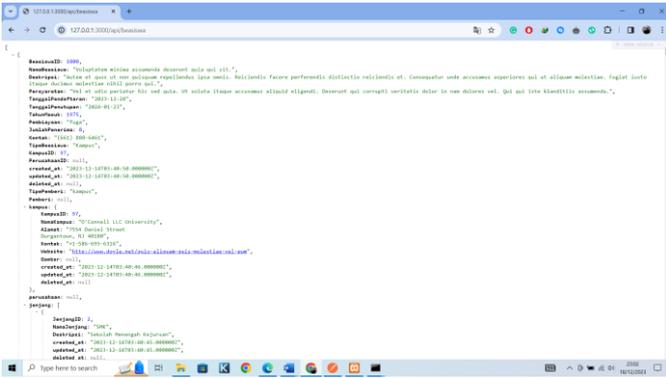
A. Pengujian Aplikasi

Pengujian aplikasi menggunakan *web crawling* API yang melibatkan penggunaan otomatisasi untuk menjelajahi dan mengumpulkan informasi dari *website* yang telah dibuat. *Web crawling* API memungkinkan pengujian aplikasi untuk mendapatkan data dengan lebih efisien dan terstruktur. Berikut langkah-langkah pengujian aplikasi menggunakan *web crawling* API:

Persiapan lingkungan uji dengan memastikan lingkungan pengujian telah disiapkan dengan baik, termasuk akses ke *web crawling* API yang akan digunakan, yaitu dengan *command* <http://127.0.0.1:3000/api/beasiswa>.

Menerapkan integrasi *web crawling* API ke dalam aplikasi yang akan diuji serta memastikan koneksi antara aplikasi dan API berjalan dengan baik. Dimana pada *website* *Laravel* yang dibangun dilakukan perintah untuk menjalankan server pengembangan lokal dengan *command* `php artisan serve`, lalu untuk memproses pekerjaan yang ada dalam antrian data pada *Laravel* dengan *command* `php artisan queue:work` dan untuk menjalankan server API dalam pengembangan lokal pada penelitian ini menggunakan *port* khusus yaitu *port* 3000, dengan *command* `php artisan serve --port 3000`.

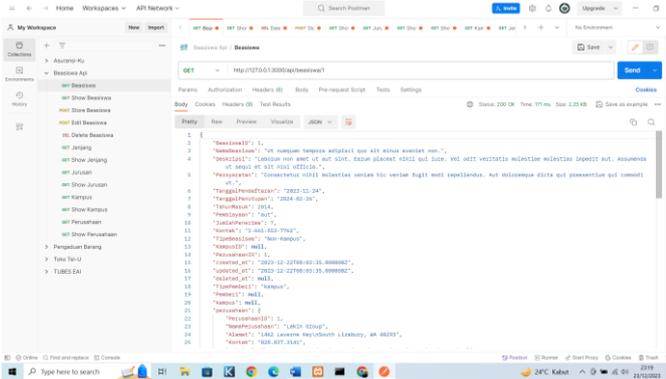
Membuat data yang perlu dikumpulkan dari situs *web* target. Pada penelitian ini data *dummy* yang berhasil dibuat yaitu sebanyak 1000 data dalam format *JSON*. Berikut contoh data yang berhasil dilakukan *crawling*:



GAMBAR 7
Hasil *Crawling* Data Dalam Format JSON

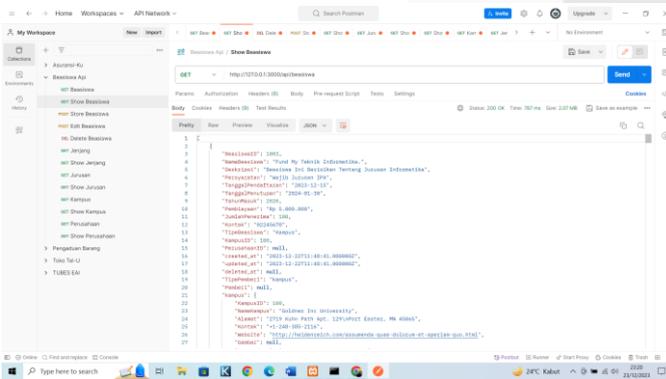
Dengan menyesuaikan *endpoint* beserta parameter *web crawling* API sesuai kebutuhan pengujian. Mencakup konfigurasi kedalaman pencarian, batasan kecepatan, dan filter lainnya. Pada penelitian ini menggunakan 4 *endpoint* yaitu:

GET Basiswa, untuk menampilkan 1 data basiswa berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/basiswa/id>.



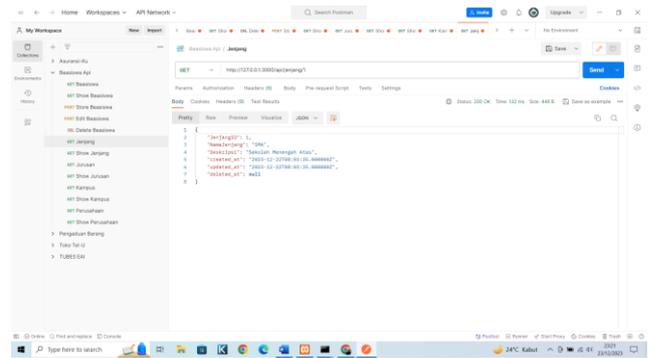
GAMBAR 8
Data *GET* Basiswa

GET Show Basiswa, untuk menampilkan seluruh data basiswa dengan URL <http://127.0.0.1:3000/api/basiswa>.



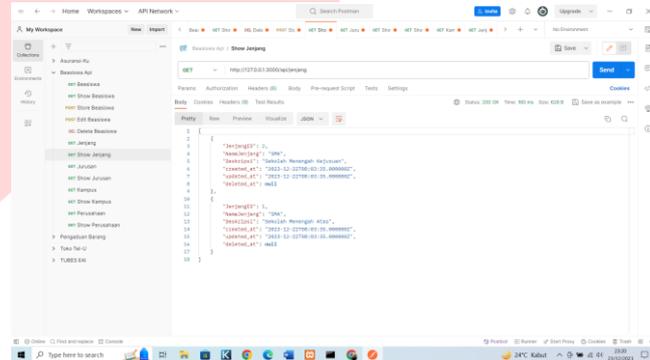
GAMBAR 9
Data *Show* Basiswa

GET Jenjang, untuk menampilkan 1 data jenjang berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/jenjang/id>.



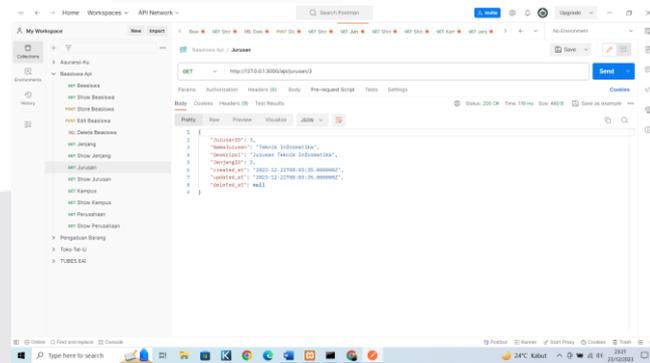
GAMBAR 10
Data *GET* Jenjang

GET Show Jenjang, untuk menampilkan seluruh data jenjang dengan URL <http://127.0.0.1:3000/api/jenjang>.



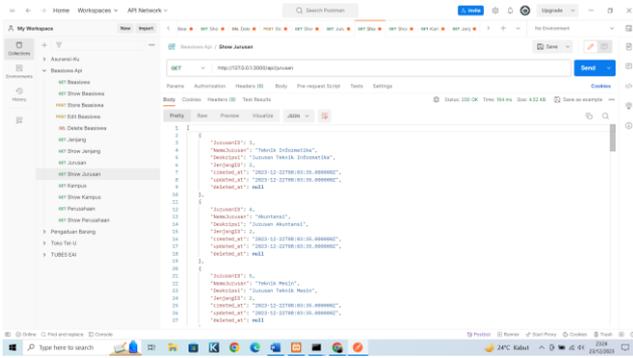
GAMBAR 11
Data *GET Show* Jenjang

GET Jurusan, untuk menampilkan 1 data jurusan berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/jurusan/id>.



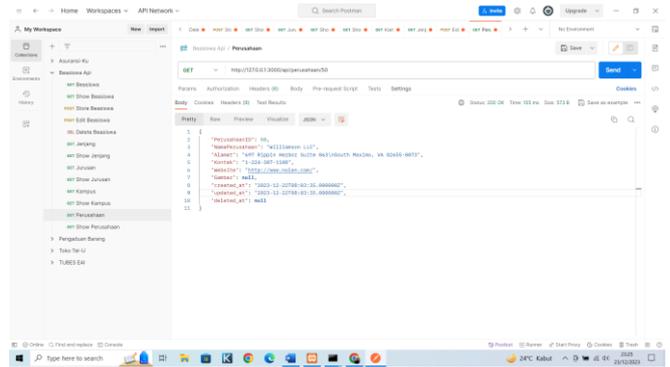
GAMBAR 12
Data *GET* Jurusan

GET Show Jurusan, untuk menampilkan seluruh data jurusan dengan URL <http://127.0.0.1:3000/api/jurusan>.



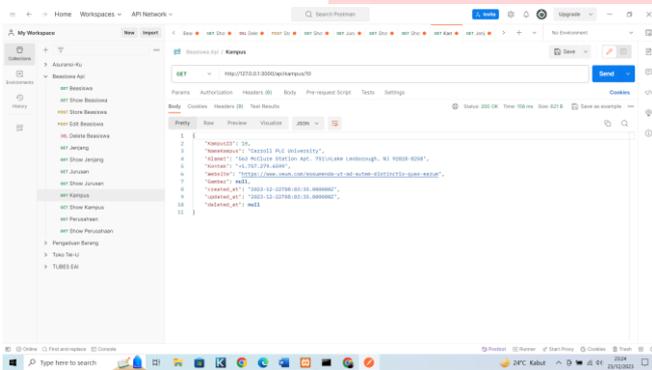
GAMBAR 13
Data GET Show Jurusan

GET Kampus, untuk menampilkan 1 data kampus berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/kampus/id>.



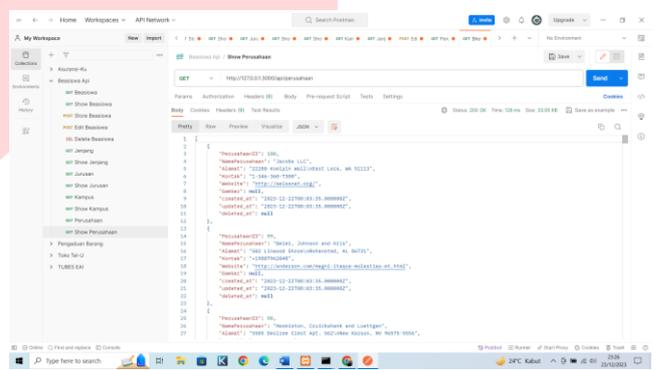
GAMBAR 16
Data GET Perusahaan

GET Show Perusahaan, untuk menampilkan seluruh data perusahaan dengan URL <http://127.0.0.1:3000/api/perusahaan>.



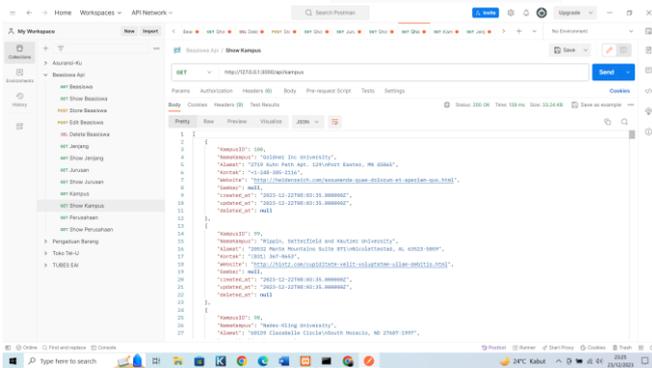
GAMBAR 14
Data GET Kampus

GET Show Kampus, untuk menampilkan seluruh data kampus dengan URL <http://127.0.0.1:3000/api/kampus>.



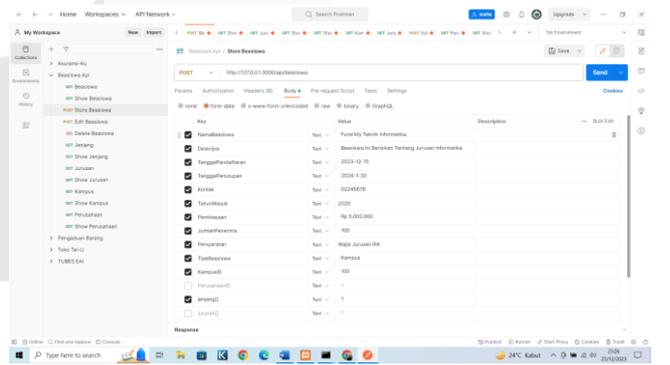
GAMBAR 17
Data GET Show Perusahaan

POST Store Basiswa untuk meng-input-kan data basiswa yang telah dibuat dengan URL <http://127.0.0.1:3000/api/basiswa> serta parameter NamaBasiswa, Deskripsi, TanggalPendaftaran, TanggalPenutupan, Kontak, TahunMasuk, Pembiayaan, JumlahPenerima, Persyaratan, TipeBasiswa, KampusID, jenjang[], jurusan[].



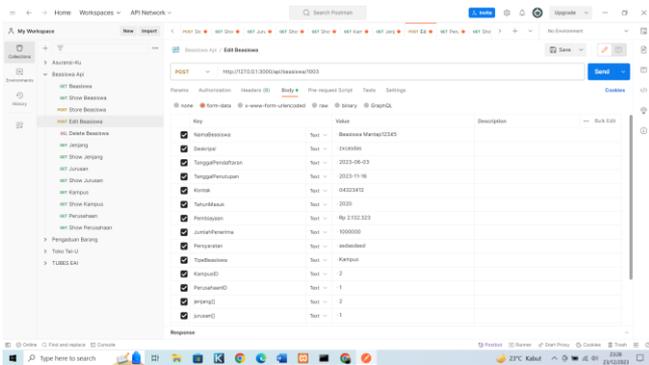
GAMBAR 15
Data GET Show Kampus

GET Perusahaan, untuk menampilkan 1 data perusahaan berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/perusahaan/id>.

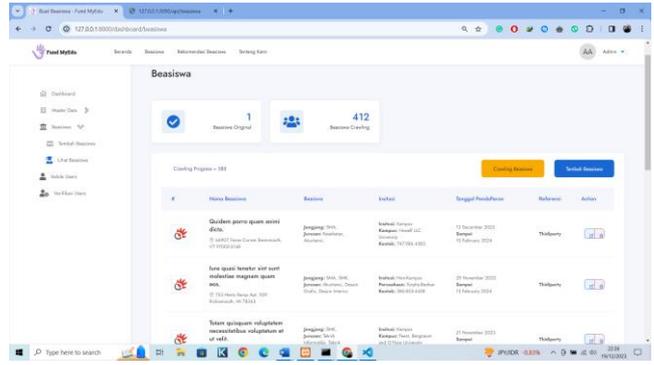


GAMBAR 18
Data POST Store Basiswa

Post Edit Basiswa untuk merubah data basiswa yang telah dibuat dengan URL <http://127.0.0.1:3000/api/basiswa> serta parameter NamaBasiswa, Deskripsi, TanggalPendaftaran, TanggalPenutupan, Kontak, TahunMasuk, Pembiayaan, JumlahPenerima, Persyaratan, TipeBasiswa, KampusID, PerusahaanID, jenjang[], jurusan[].

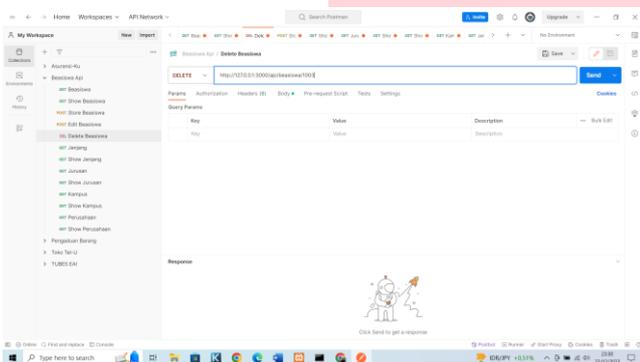


GAMBAR 19
Data Post Edit Beasiswa



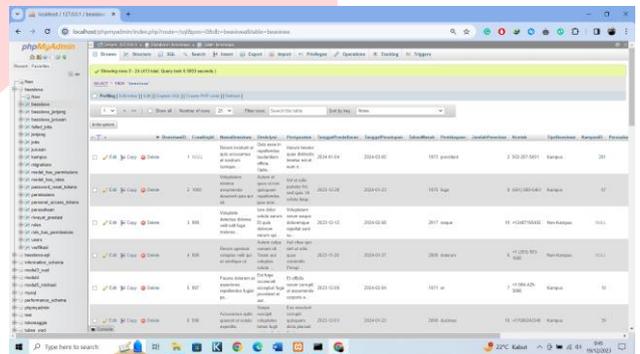
GAMBAR 22
Contoh Crawling Pada Website Fund MyEdu

DELETE Beasiswa untuk menghapus data beasiswa yang telah dibuat berdasarkan id yang dimasukkan dengan URL <http://127.0.0.1:3000/api/beasiswa/id>.



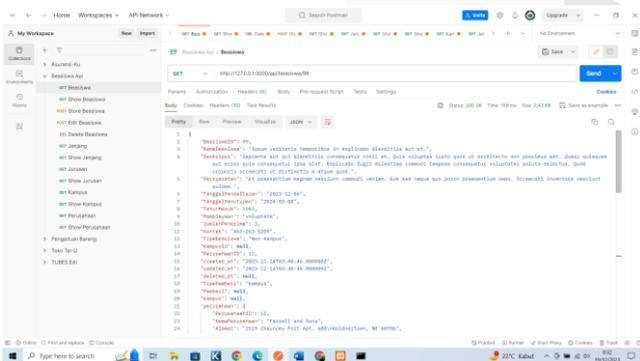
GAMBAR 20
Data DELETE Beasiswa

Dokumentasikan hasil pengujian, termasuk data yang berhasil dikumpulkan waktu eksekusi. Pendokumentasian dilakukan dengan menyimpan seluruh data ke dalam *database* beasiswa. Berikut *database* beasiswa pada *website* Fund MyEdu:



GAMBAR 23
Database Beasiswa

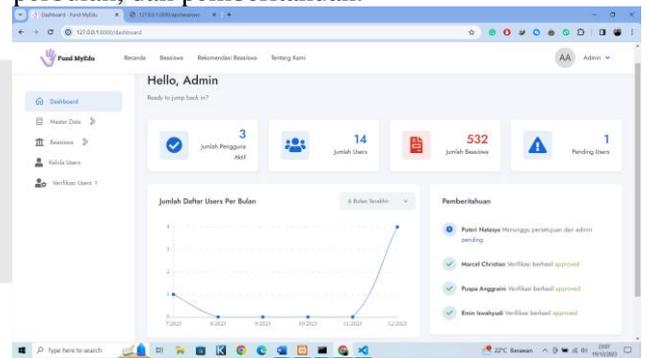
Eksekusi Pencarian dengan memulai eksekusi *web crawling* API untuk mengumpulkan data dari *web* Fund MyEdu serta memastikan API berjalan sesuai dengan skenario pengujian yang telah ditentukan. Berikut merupakan contoh ketika dilakukan pemanggilan *endpoint* serta URL pada API:



GAMBAR 21
Contoh Pemanggilan Endpoint Serta URL Pada API

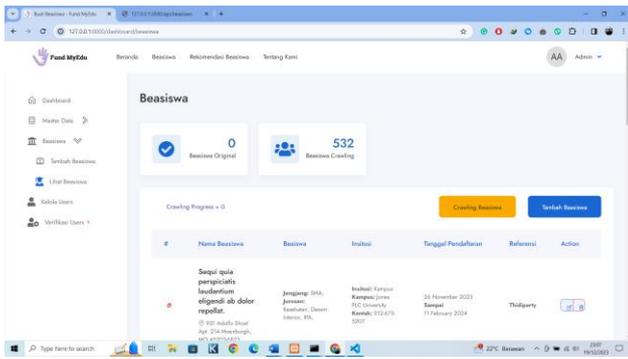
Kemudian berikut merupakan contoh ketika dilakukan *crawling* pada *website* Fund MyEdu:

B. Hasil Aplikasi
Dashboard Admin merupakan tampilan untuk mengelola informasi jumlah pengguna aktif, jumlah *user*, jumlah beasiswa, *pending user*, jumlah daftar *users* perbulan, dan pemberitahuan.

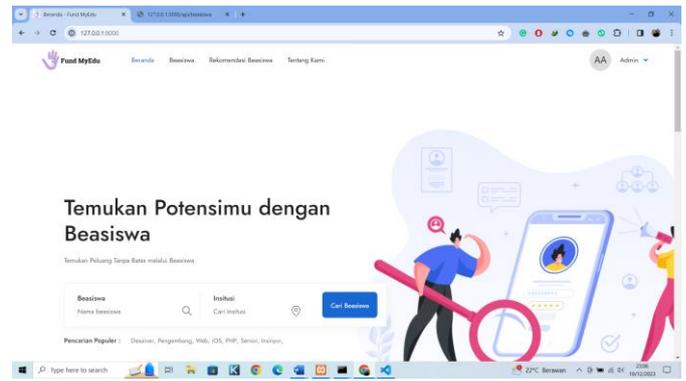


GAMBAR 24
Dashboard Admin

Lihat Beasiswa merupakan tampilan untuk mengambil data beasiswa dari berbagai *website* penyedia beasiswa.



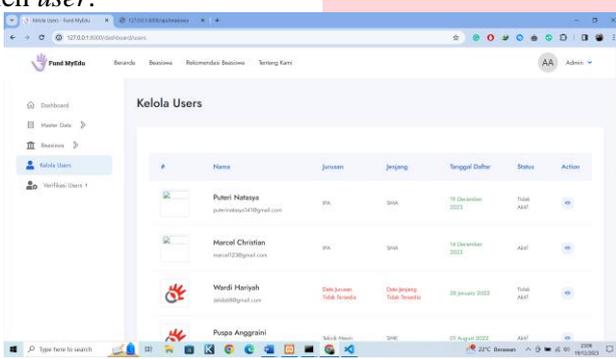
GAMBAR 25
Tampilan Fitur Lihat Beasiswa



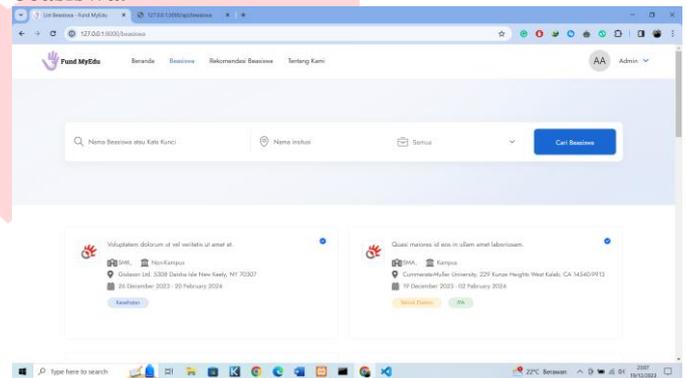
GAMBAR 28
Tampilan Fitur Beranda

Kelola *Users* merupakan tampilan untuk menampilkan informasi data-data *user* yang telah dibuat oleh *user*.

Beasiswa adalah tampilan untuk melihat informasi beasiswa.



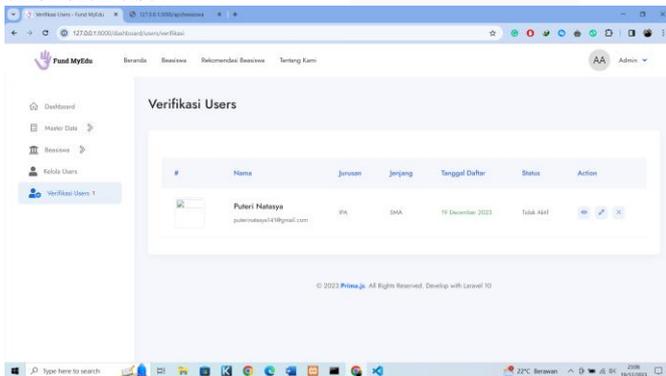
GAMBAR 26
Tampilan Fitur Kelola *Users*



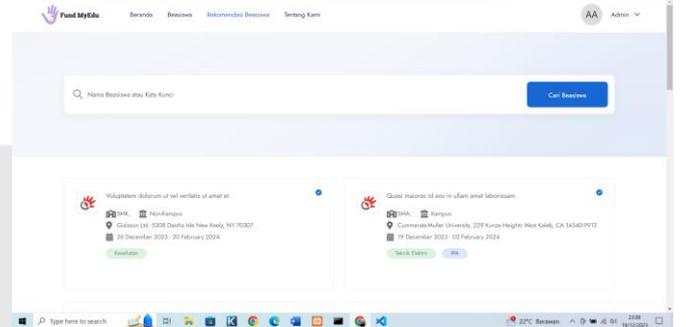
GAMBAR 29
Tampilan Fitur Beasiswa

Verifikasi *Users* merupakan tampilan untuk menampilkan data *user* yang telah dibuat oleh *user* yang nantinya akan diverifikasi oleh *admin* untuk bisa melihat dan menerima rekomendasi beasiswa dari data yang telah diisi oleh *user*.

Rekomendasi Beasiswa merupakan tampilan untuk memberikan rekomendasi beasiswa dari data-data yang telah diisi *user* saat melakukan pengisian formulir.



GAMBAR 27
Tampilan Fitur Verifikasi *Users*



GAMBAR 30
Tampilan Fitur Rekomendasi Beasiswa

Beranda merupakan tampilan untuk mencari dan menampilkan data beasiswa yang ada di dalam *website* ini.

Tentang Kami adalah tampilan informasi *website* Fund MyEdu.



GAMBAR 31
Tampilan Fitur Tentang Kami

C. Hasil Pengujian

Hasil yang didapat pada persiapan lingkungan uji yaitu akses ke *web crawling* API dapat berjalan dengan baik tanpa ada kendala sistem atau *error*.

Hasil yang didapat pada integrasi API yaitu koneksi antara aplikasi dan API berjalan dengan baik, *website* dapat ditampilkan dengan lancar, antrian data dapat berjalan sesuai dengan perintah yang telah dijalankan, serta API dapat menghasilkan informasi sesuai dengan keinginan *user*.

Hasil yang didapat pada pengumpulan data yaitu data *dummy* berhasil dibuat sebanyak 1000 data, Dimana dapat bertambah seiring dilakukannya *crawling* data.

Hasil yang didapat pada pengaturan *endpoint* dan parameter yaitu API dapat menampilkan seluruh data beasiswa sesuai *endpoint* dan parameter yang di-*input* oleh *user* melalui *platform* Postman.

Hasil yang didapat pada eksekusi pencarian yaitu API dapat mengumpulkan data dari *web* Fund MyEdu serta API berjalan sesuai skenario pengujian.

Hasil yang didapat dari pelaporan hasil yaitu seluruh data beasiswa yang berhasil dikumpulkan saat proses eksekusi telah didokumentasikan dengan baik. Pendokumentasian dilakukan dengan menyimpan seluruh data ke dalam *database* beasiswa.

V. KESIMPULAN

Cara mengumpulkan informasi tentang kriteria suatu kampus untuk menerima seseorang melalui jalur beasiswa menggunakan *Data Crawling* yaitu dengan membuat *website*, data *dummy*, dan API menggunakan metode *web crawling* API secara otomatis untuk menjelajahi dan mengumpulkan informasi dari *website* yang telah dibuat. Langkah-langkah pengujian aplikasi menggunakan *web crawling* API dengan Persiapan Lingkungan Uji, Integrasi API, Integrasi API, Pengaturan *Endpoint* dan Parameter, Eksekusi Pencarian, serta Pelaporan Hasil.

Cara menyajikan data yang sudah diperoleh agar calon mahasiswa bisa mendapatkan rekomendasi beasiswa dari formulir data yang telah mereka isi sesuai jurusan yang telah diambil di bangku SMA/SMK yaitu dengan dibuatnya *website* Fund MyEdu sebagai tempat

menampilkan informasi beasiswa bagi calon penerima beasiswa. Dimana terdapat fitur *Dashboard Admin*, Lihat Beasiswa, Kelola *Users*, Verifikasi *Users*, Beranda, Beasiswa, Rekomendasi Beasiswa, dan Tentang Kami. Fund MyEdu memiliki dua *role* yaitu *admin* dan *user*, *admin* berperan sebagai pengelola dan verifikator data *user* serta data beasiswa. Sedangkan *user* merupakan calon penerima beasiswa.

REFERENSI

- [1] S. A. Rismawan, "Implementasi Website Berita Online Menggunakan Metode Crawling Data Dengan Bahasa Pemrograman Python," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 10, no. 3, pp. 167–178, 2023.
- [2] Salsabila Miftah Rezkia, "Metode Pengolahan Data: Tahapan Wajib yang Dilakukan Sebelum Analisis Data," DQLab. Accessed: Dec. 01, 2023. [Online]. Available: <https://dqlab.id/metode-pengolahan-data-tahapan-wajib-yang-dilakukan-sebelum-analisis-data>
- [3] J. Sasongko, "Aplikasi untuk Membangun Corpus dari Data Hasil Crawling dengan Berbagai Format Data Secara Otomatis," *Dinamik*, vol. 15, no. 1, 2010.
- [4] R. Mitchell, *Web scraping with Python: Collecting more data from the modern web*. "O'Reilly Media, Inc.," 2018.
- [5] E. G. Pradana, "IMPLEMENTASI WEB CRAWLER UNTUK Mencari Harga BARANG TERMURAH DARI BERBAGAI SITUS E-COMMERCE INDONESIA," *Jurnal Teknologi Pintar*, vol. 2, no. 9, 2022.
- [6] D. Alpina and H. Witriyono, "Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web," *Jurnal Media Infotama*, vol. 18, no. 1, pp. 36–42, 2022.
- [7] A. Hanafi, I. M. Sukarsa, and A. Wiranatha, "Pertukaran data antar database dengan menggunakan teknologi API," *Lontar Komputer*, vol. 8, no. 1, pp. 22–30, 2017.
- [8] H. Asgar and B. Hartono, "Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan," *Jurnal Informatika Teknologi dan Sains (Jinteks)*, vol. 4, no. 1, pp. 8–14, 2022.
- [9] B. S. Arifin and M. Laya, "Web Service Processor sebagai Penghubung Sistem Kiosk Medicom dengan SIM RS Kanker Dharmais," *vol*, vol. 3, pp. 49–56, 2017.
- [10] E. Hartati, "SISTEM INFORMASI TRANSAKSI GUDANG BERBASIS WEBSITE PADA CV. ASYURA," *Klik-Jurnal Ilmu Komputer*, vol. 3, no. 1, pp. 12–18, 2022.
- [11] J. Jumadi, "Sistem Pendukung Keputusan untuk Menentukan Siswa Penerima Beasiswa," *JURNAL ISTEK*, vol. 6, no. 1–2, 2012.
- [12] Badan Pusat Statistik, "Statistik Penunjang Pendidikan 2021 (Hasil Susenas Modul Sosial

Budaya dan Pendidikan),” 2022, Accessed: Oct. 31, 2023. [Online]. Available: <https://www.bps.go.id/publication/2022/06/30/994083098999eacc4abc9207/statistik>

- [13] I. N. Husada, E. H. Fernando, H. Sagala, A. E. Budiman, and H. Toba, “Ekstraksi dan Analisis

Produk di Marketplace Secara Otomatis dengan Memanfaatkan Teknologi Web Crawling,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 5, no. 3, Jan. 2020, doi: 10.28932/jutisi.v5i3.1977.

