RANCANG BANGUN SISTEM KURIKULUM BERBASIS KELUARAN DENGAN ORIENTASI WEB SERVICE

Herlangga Yusuf Syailendra; Husni Thamrin
Informatika, Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta

Abstrak

Proses belajar mengajar pada lingkungan universitas, tidak terlepas kebutuhan atas suatu standarisasi yang sering disebut kurikulum. Desain kurikulum melibatkan banyak orang, dokumen dan juga waktu. Penelitian ini di buat untuk menguraikan kompleksitas dari proses pembuatan kurikulum. Pengembangan aplikasi ini menggunakan pemrograman django serta menggunakan metode pengembangan perangkat lunak tipe waterfall. Sekalipun begitu, aplikasi ini didesain agar program dapat dikembangkan lebih lanjut di masa mendatang. maka bagian-bagian program akan dipisah berlandaskan suatu paradigma pemrograman. Paradigma digunakan sebagai landasan bagaimana aplikasi mengelola dan bekerja. Paradigma yang digunakan adalah Active Record design pattern. Paradigma ini digunakan dalam proses pengerjaan aplikasi. Tidak lupa untuk memisahkan bagian logika bisnis dengan logika program agar aplikasi dapat digunakan walaupun endpoint dari aplikasi berubah. Pengujian aplikasi ini dilakukan dengan metode uji fungsionalitas dan menunjukan hasil bahwa seluruh fungsionalitas telah berfungsi dengan baik. Aplikasi ini dibuat untuk diaplikasikan pada PTMA.

Kata Kunci : Kurikulum, Kurikulum Berbasis Keluaran, PTMA, Sistem Informasi Manajemen

Abstract

On today teaching and learning in university, we must known if that need a standarization that called as curriculum. In that process to build that we need people documents and time. This research created based on premise to do the management of that complex process and make it easier to do. This application is created using an development live cycle that call waterfall. Though it created using waterfall this program created by considering that app will be managed / develop even more. So by using a paradigm when creating this program we can trace how to manage and works that app. This app developed using Active Record Design Pattern. And on doing this we separate the bussiness logic and the code logic so we can see how it works and the component can be reuseable even the endpoint was modified. This application was tested and outputing result that satisfying all requirement. The application was created for use in PTMA.

Keywords: Curriculums, Curriculum with Output Result, Management Informatic System, PTMA,

1. PENDAHULUAN

Setiap pendidikan tinggi di Indonesia memiliki kurikulum, kurikulum di Indonesia di standarisasi oleh pusat, maka tiap ada perubahan pada pusat maka perguruan tinggi diindonesia

harus mengikuti (Marsanto, A, 2017). Proses perubahan dan pembuatan kurikulum, meliputi membangun CPL, kemudian berkembang lagi menjadi acuan kompetensi berdasar mata kuliah, hingga proses penilaian. Jika dilihat dari prosesnya yang panjang dan juga melibatkan banyak orang maka dapat dipastikan proses tadi menjadi kompleks dan juga menguras banyak waktu karena pada proses ada yang harus dilakukan secara berkelompok. Hal ini juga mempengaruhi proses pencatatan yang cukup banyak komponennya.

Karena hal ini dibutuhkan suatu metode untuk mengurai kompleksitas tadi. Salah satu solusinya adalah dengan menggunakan sistem informasi manajemen. Sistem informasi manajemen adalah sistem yang bertujuan untuk mengumpulkan data, mengubah data menjadi informasi yang nantinya dapat digunakan oleh aplikasi lain sebagai data untuk diproses(Preston Mwiinga, 2023).

Dengan adanya sistem informasi manajemen diharap dapat meminimalkan kekurangan dari proses manual seperti dokumen hilang, pencarian dokumen yang sulit, tidak memungkinkan untuk mengolah data hasil karena format tabulasi yang berbeda tiap prosesnya. Sistem informasi yang dikembangkan diharap mampu menghadapi kesulitan yang dihadapi di atas.

Melihat penelitian dari Adilah, N. S. (2023). Aplikasi yang dihasilkan tidak didesain dengan mengutamakan *reusability, endpoint* belum terdokumentasi serta tidak menggunakan paradigma sehingga pengembangan selain penulis tidak bisa dipertahankan. Kemudian melihat penelitian oleh Cahyawardani dibuat sistem sejenis namun hal ini menangani tentang pemrosesan dari tahap CPL. Sehingga penelitian ini cukup patut untuk dilakukan (Cahyawardani, 2020).

Merujuk dari kegunaan aplikasi ini penulis menemui penggunaan aplikasi serupa di Universitas Muhammadiyah Surakarta (UMS). UMS telah memiliki beberapa servis yang dapat menangani hal tadi berikut seperti MyKurikulum. Sekalipun begitu ada problem yang cukup spesifik yaitu pada sistem yang dibangun UMS yaitu menggunakan *django* sebagai pusat *render*. Hal ini mengakibatkan *throughput* pada koneksi tinggi menurun drastis daripada yang dapat dihasilkan ketika hanya menyajikan data.

Pilihan Sistem yang bisa digunakan adalah sistem monolitik dengan modul atau menggunakan *microservice*. Pada sistem *microservice* maka akan terjadi *overhead* yang cukup besar karena hubungan satu data dengan data lain cukup tinggi sehingga tidak cukup efektif karena terjadi *coupling* yang tinggi (Li X, 2024). Sedangkan sistem monolitik dengan konsep *module* agar

dapat mengeliminasi proses memindahkan data menggunakan protokol yang berat dan juga kemudahan dalam manajemen server pada kemudian hari.

Tulisan ini mendeskripsikan hasil pengembangan sistem kurikulum berbasis keluaran dengan metode web *service*. Sistem ini merupakan bagian dari sebuah sistem yang sedang dikembangkan untuk pemenuhan kebutuhan PTMA. Tulisan ini hanya mencakup bagian *Backend.* Bagian komponen lain seperti Desain dan *Frontend* dikerjakan oleh tim lain sehingga penelitian ini bersifat komplementer dan juga dapat diterapkan. Aplikasi ini akan diterapkan pada Perguruan Tinggi Muhammadiyah dan Aisyiyah (PTMA).

2. METODE

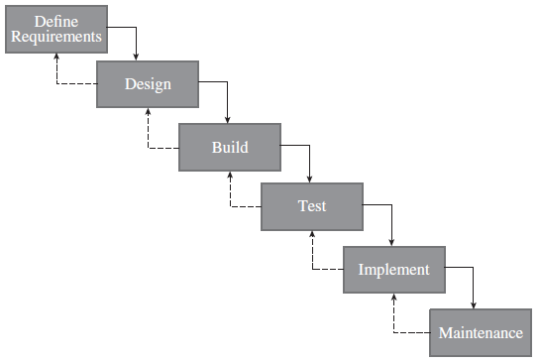
Pada penelitian ini penulis menggunakan salah satu model dari siklus hidup aplikasi yaitu metode *Waterfall*. Metode *Waterfall* dipilih karena berdasar beberapa jurnal menggolongkan beberapa ciri / model yang ada saat ini memiliki kemiripan yang cukup tinggi. Pada Jurnal yang ditulis Kumar M pada tahun 2018 memberikan ciri khas aplikasi yang dikembangkan oleh metode ini antara lain:

1. Dilakukan secara bertahap
2. Bersifat tidak berubah ketika proses dikerjakan
3. *Requirement* dari awal sudah ditentukan jelas
4. Testing dilakukan di akhir
5. Perkembangan pengerjaan bersifat linier
6. Proses terkontrol

Kemudian apa bila melihat karakteristik dari *waterfall* yang ditulis oleh Mohammad Ikbal Hosain memiliki tambahan karakteristik seperti :

1. Bersifat *incremental* dan berulang
2. Terbuatnya *prototype* pada proses pengerjaan
3. Kolaborasi tim pengembang sangat berpengaruh terhadap laju pengembangan aplikasi
4. Kebutuhan susah berkembang

Dengan mempertimbangkan hal tadi maka pada pengerjaan rancang bangun kali ini menggunakan metode *Waterfall*. Dengan berpegang metode *Waterfall* proyek rancang bangun ini dimulai. *Waterfall* sendiri memiliki struktur SDLC sebagimana diagram pada Gambar 1.



Gambar 1. Diagram *Waterfall* SDLC

2.1 Analisis Kebutuhan / *Requirement Analysis*

Mengacu pada dokumen contoh dari sistem berjalan di UMS serta sumber yang terjadi di lapangan maka diambil sebuah alur kerja kurikulum yang digunakan sebagai dasar pengembangan aplikasi ini. Berikut prosesnya.

1. Kepala Program Studi membuat kurikulum serta menunjuk tim pengembang kurikulum.
2. Kepala Program Studi dan Tim Pengembang mengisi CPL, Mata kuliah (berserta CPMK), dan Dokumen Kurikulum
3. Kepala Program Studi atau tim pengembang mengajukan Kurikulum.
4. Admin menunjuk *reviewer*
5. *Reviewer* melakukan *review* pada kurikulum
6. *Reviewer* menentukan Revisi kurikulum / menyetujui kurikulum
    a. Bila revisi maka Kepala Program Studi dan Tim Pengembang Kurikulum melakukan perubahan sesuai dengan Hasil Review. Dan mengulangi poin 5.
    b. Bila Disetujui *Review* akan meninggalkan informasi bahwa setuju
7. Admin mendapat notifikasi apa bila *reviewer* telah melakukan *review*

8. Admin mengkonfirmasi hasil semua *reviewer*
9. Admin melakukan persetujuan akhir
10. Kurikulum berhasil diajukan

Dari proses tadi maka bagian kurikulum ini dapat digambarkan pada diagram alir seperti pada Gambar 2.

Gambar 2. Diagram Alir Proses Pembuatan Kurikulum

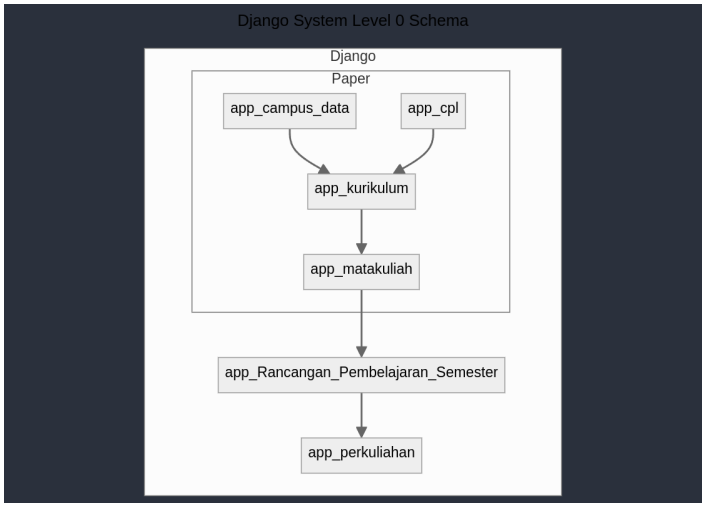
Berdasar alur aplikasi tersebut maka terciptalah sebuah spesifikasi / syarat kebutuhan pada aplikasi sebagai berikut.

1. *Login*
    a. Dapat memiliki beberapa *role* dalam satu akun
    b. Ke depan dapat di integrasikan dengan cukup mudah ke aplikasi lain
    c. *Permission role* dapat ditambahkan sewaktu waktu dengan mudah

2. Data Kampus. Data kampus yang meliputi data dosen, lembaga, dan mahasiswa dapat di impor dari aplikasi / API lain
3. Pembuatan Kurikulum.
    a. Kurikulum hanya bisa dimulai dan diedit namanya oleh Kepala Program Studi
    b. Kepala Program Studi dapat menunjuk dosen sebagai tim pengembang kurikulum
    c. Kepala Program Studi dan dosen terdaftar pada tim pengembang dapat melakukan perubahan data (menambah, mengubah, dan menghapus) data pada CPL, Mata kuliah, serta dokumen pendukung lainnya
4. Pengajuan Kurikulum. Kurikulum dapat diajukan apabila terpenuhi syarat berikut.
    a. Dokumen kurikulum terisi lengkap
    b. Tim pengembang kurikulum ada
5. Proses *Review* kurikulum
    a. *Role Admin* dapat menunjuk *reviewer* kurikulum
    b. *Reviewer* dapat melakukan *review*
    c. *Reviewer* mampu menyatakan setuju kurikulum
    d. *Reviewer* mampu menyatakan kurikulum direvisi
    e. *Role* Admin mampu memajukan status menjadi disetujui setelah melihat revisi kurikulum
6. Kurikulum Aktif Kurikulum bisa diaktifkan oleh admin setelah disetujui

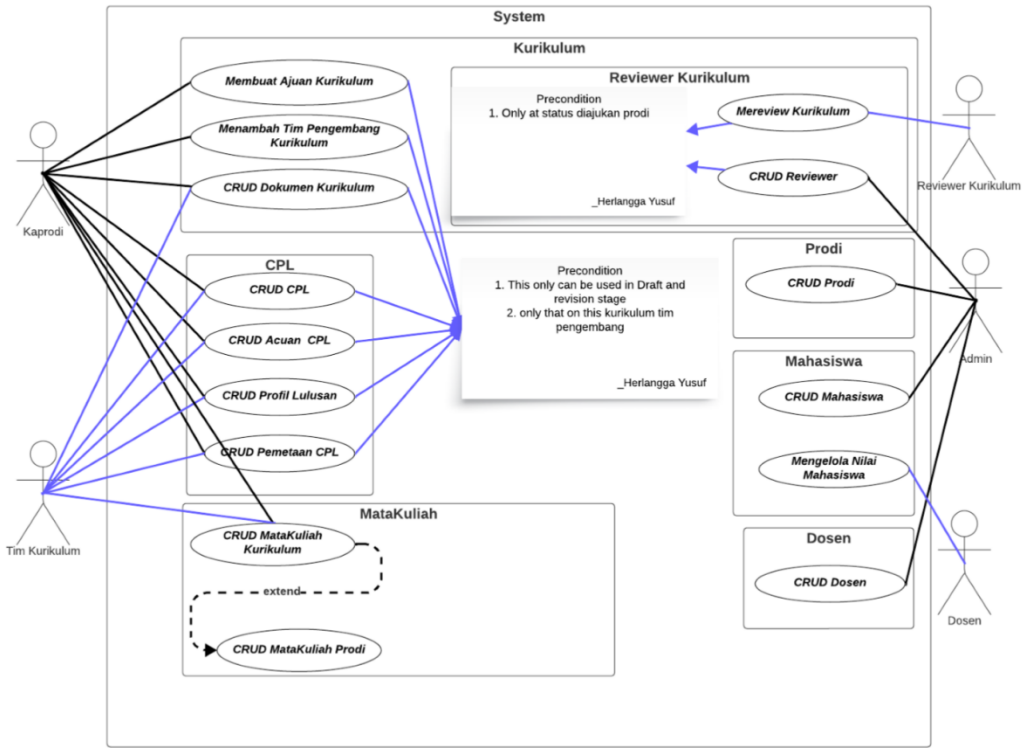
2.2 Design Aplikasi

Aplikasi ini merupakan sebuah bagian dari sebuah proyek pengabdian yang memiliki gambaran yang ditunjukkan pada Gambar 3. Paper ini mengerjakan pada proses yang di dalam kotak *subgraph "Paper"* pada Gambar 3.



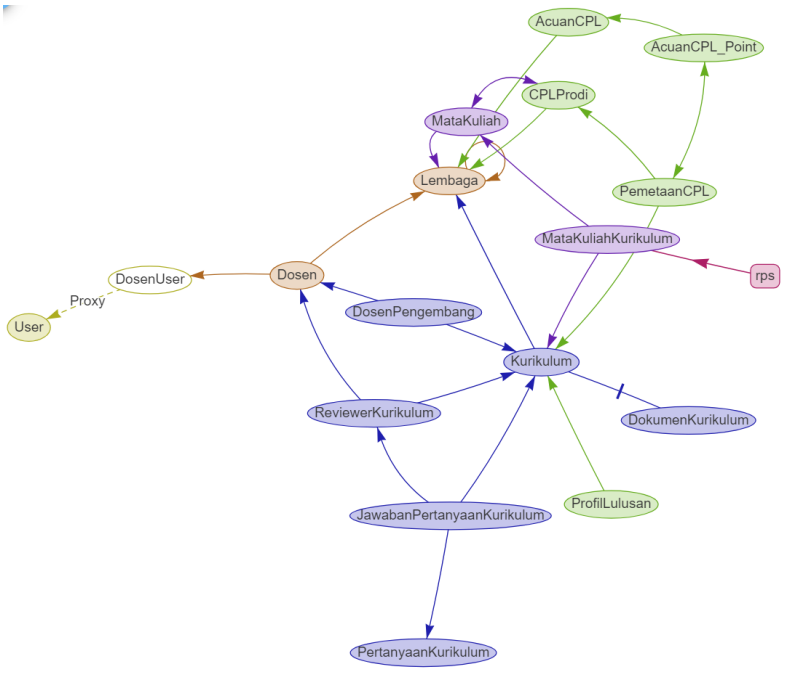
Gambar 3. Diagram Aplikasi Penuh / Level 0

Berdasar *requirement* pada subbab 2.2 maka diciptakanlah sebuah *Usecase* diagram yang mewakili semua tindakan di atas. Pada Gambar 4



Gambar 4. *Usecase* Diagram

Dari kedua diagram di atas, maka dapat di buatlah sebuah diagram relasi sebagaimana pada Gambar 5.



Gambar 5. Diagram Relasi Data

2.3 **Teknologi Yang Digunakan**

Pada tahap berikutnya menentukan teknologi yang dipakai. Teknologi *backend* yang dipakai adalah: *django, django-rest-framework (DRF), drf-standarized-error* dan *drf-spectacular*. Pemilihan teknologi ini bertujuan untuk memenuhi kebutuhan – kebutuhan aplikasi tersebut. Berikut teknologi yang digunakan.

1. *Django*

Django digunakan karena *support* platform dan memiliki stabilitas yang baik. *Django* juga merupakan ORM yang tersedia pada platform python yang cukup mudah untuk dibuat serta dirawat.

Django juga dipilih karena sering digunakan pada Universitas Muhammadiyah Surakarta terlihat dari publikasi Islami, M. H. (2020) dan juga

2. *django-rest-framework*

Karena membangun *Application Programming Interface* (API), maka ini adalah modul yang cukup standar dipakai di atas *django*. Memiliki *support* yang memadai. Serta dapat melakukan validasi tidak dengan cara *default*.

3. *drf-standarized-error*

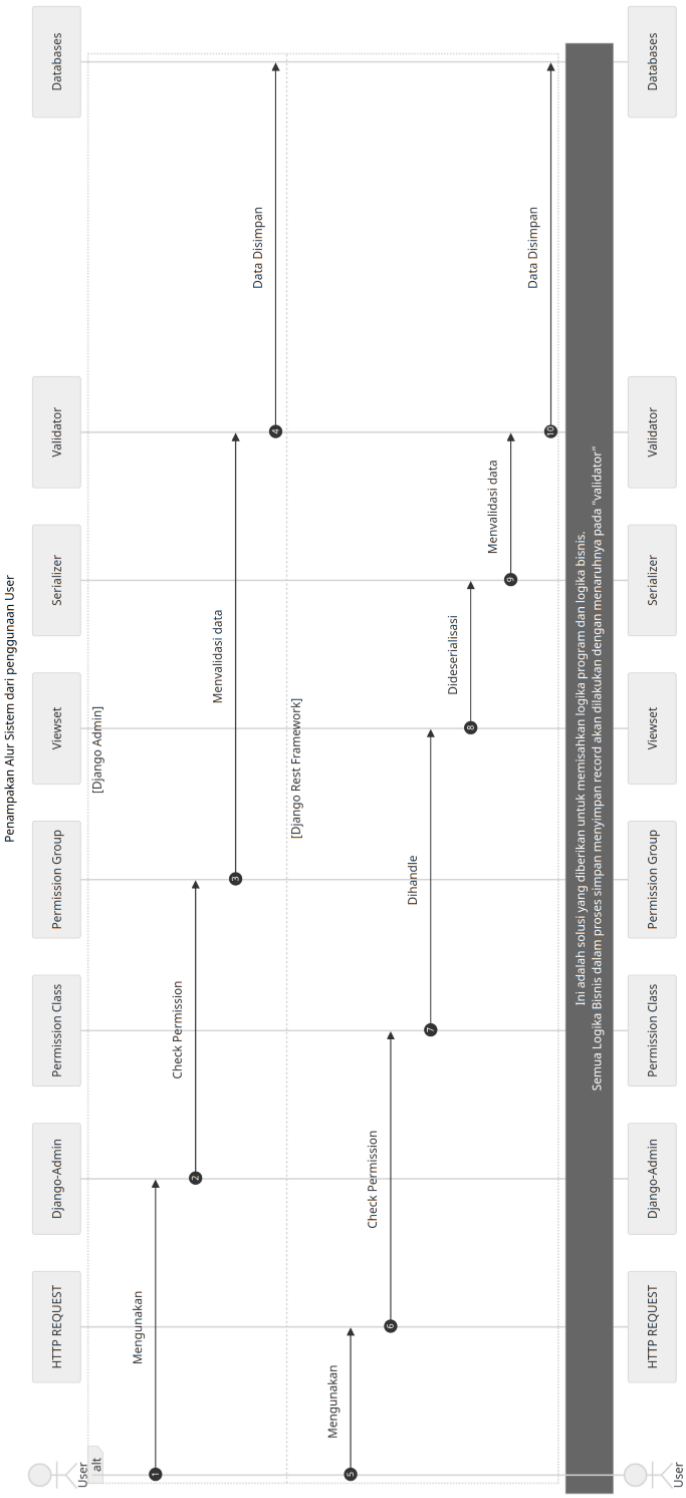
Memberikan sebuah model skema pada semua galat yang terjadi pada sistem *django* dan DRF. Hal ini perlu dilakukan agar pesan galat dapat di tangani dengan efisien pada aplikasi yang menggunakannya. Modul ini juga menyediakan kode HTTP *response* yang tepat sehingga tidak terjadi kesulitan ketika menangani data tersebut.

4. *drf-spectacular*

Ketika mengeluarkan API maka diperlukan manajemen pada semua *endpoint,* hal ini sangat diperlukan sebagai sebuah metode memberi tahu apa tujuan dan fungsi yang digunakan oleh suatu *endpoint* tersebut. Paket ini menyediakan sebuah *tool* sebagai tempat mengabstraksi kode menjadi sebuah skema yang dapat di proses oleh aplikasi lain. Skema yang dihasilkan berstandar Open API 3.

Setelah menentukan teknologi yang digunakan maka proses berikutnya adalah menentukan paradigma pemrograman yang digunakan. Paradigma pemrograman yang digunakan pada *django* adalah *Active Record design pattern*. Kemudian dengan mengabstraksi validasi dengan model dengan tujuan memisahkan logika bisnis dan juga logika untuk program mengakibatkan kemudahan ketika menambah suatu spesifikasi di kemudian hari. Hal ini juga menangani permasalahan yang dilalui oleh DRF di mana pada rilis versi 3 menghilangkan pemanggilan fungsi Model.clean() pada saat menggunakan ModelSerializer.

Oleh sebab itu diperlukan adanya perubahan dari sisi cara penerapannya. Penerapannya dapat dilihat pada skema pada Gambar 6. Dengan menggunakan proses pada Gambar 6. Maka secara tidak langsung juga mengakibatkan dampak pada proses pembuatan aplikasi terutama di bagian model. Dengan memisah *file* pada untuk fungsi validasi menjadikan validasi dapat dipanggil ketika menggunakan *serializer* maupun pada saat di *models* sendiri.



Gambar 6. Skema penyelesaian masalah validasi

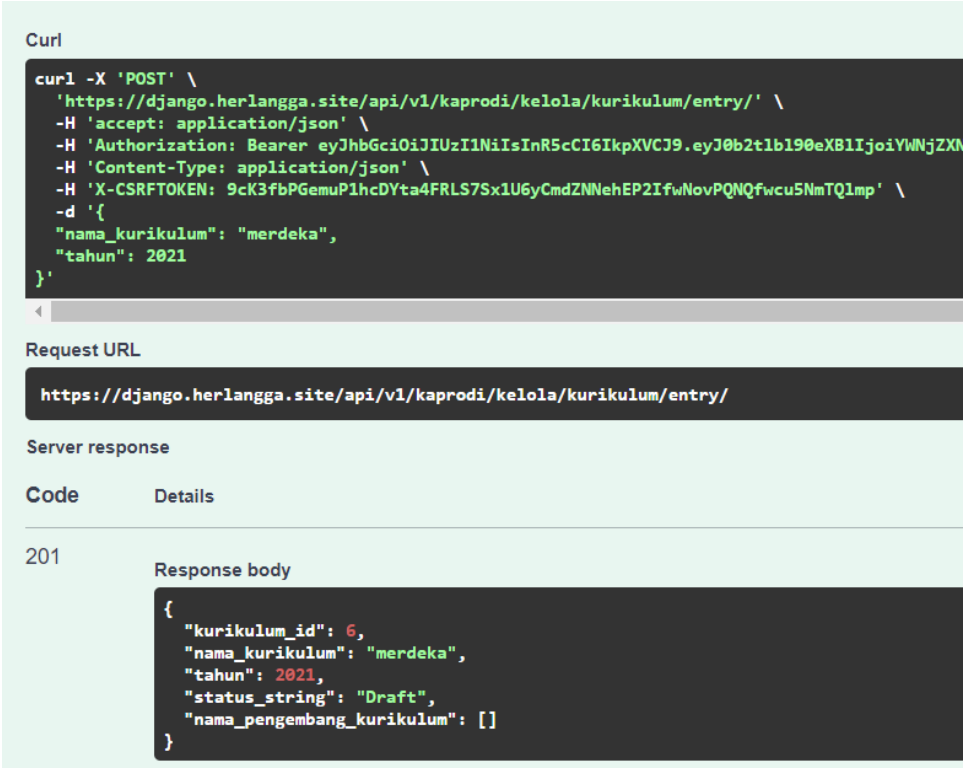
3. HASIL DAN PEMBAHASAN

3.1 HASIL

Tahapan ini tahapan terakhir dari naskah publikasi ini. Pada tahapan ini akan melakukan uji coba pada *endpoint* yang telah dibuat. Bagian ini akan meliputi pada proses pengujian yang dilakukan dengan menggunakan aplikasi *Swagger* yang di buat oleh *drf-spectacular*. Setiap *endpoint* yang berhubungan dengan model memiliki CRUD yang lengkap karena menggunakan DRF *ModelViewset*

3.1.1 *Endpoint* Kurikulum

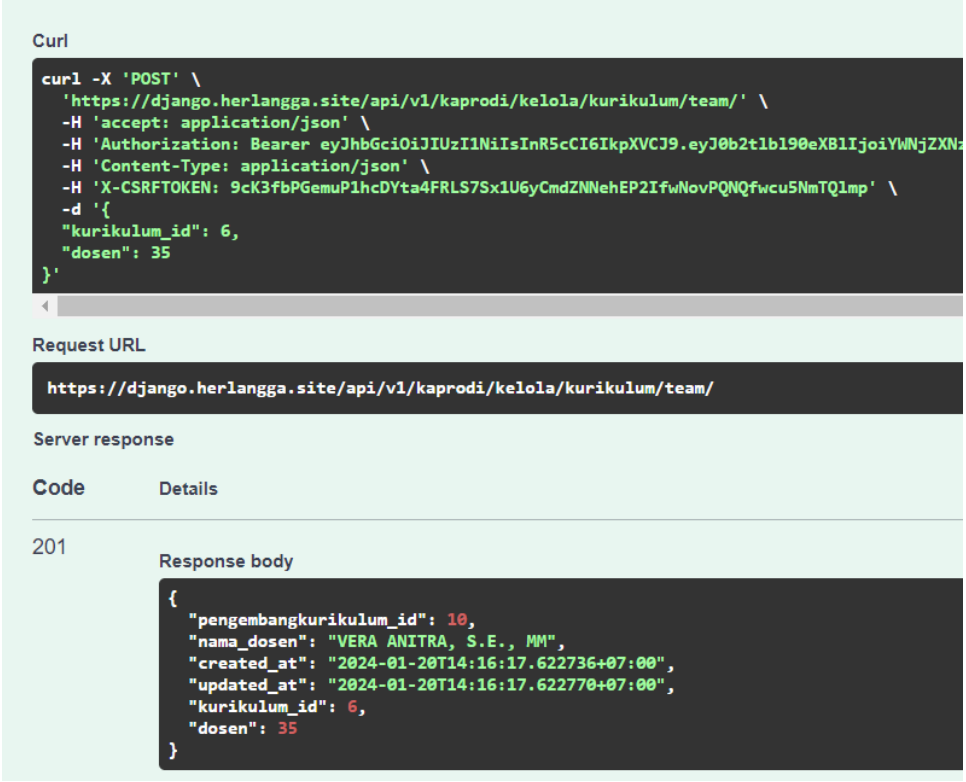
Endpoint ini digunakan untuk membuat kurikulum oleh kepala program studi dan mengedit nama dan tahun kurikulum dibuat. Seperti yang terlihat pada Gambar 7, yang menunjukkan proses penambahan kurikulum



Gambar 7. *Endpoint* buat Kurikulum

3.1.2 *Endpoint* Tim Kurikulum

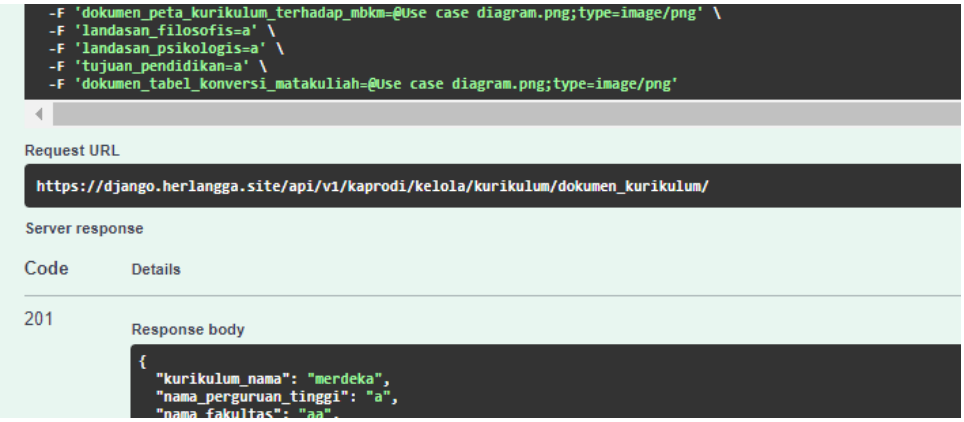
Endpoint ini untuk mengelola tim kurikulum. Pada *endpoint* ini dapat menggunakan proses CRUD dan sebagai contoh di tunjukan proses penambahan tim pengembang kurikulum pada Gambar 8.



Gambar 8. Menambahkan anggota tim pengembang kurikulum

3.1.3 *Endpoint* Dokumen Kurikulum

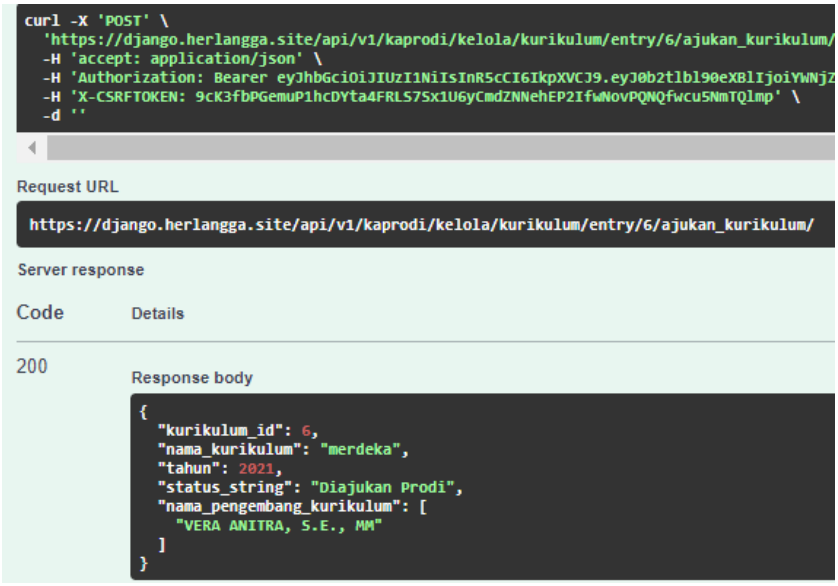
Endpoint ini menangani dalam manajemen data dokumen kurikulum. Data ini dapat diubah oleh Tim kurikulum dan juga kepala program studi. Sebagai pembuktian hasil aplikasi di tunjukan pada hasil memasukkan dokumen kurikulum pada Gambar 9.



Gambar 9. Memasukkan dokumen kurikulum.

3.1.4 *Endpoint* Ajuan Kurikulum

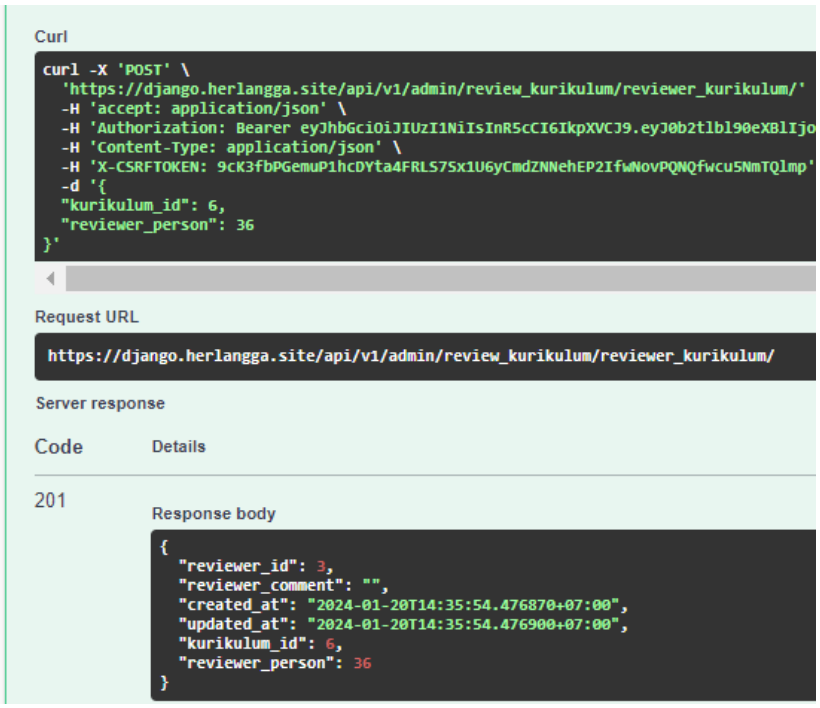
Endpoint ini digunakan dalam proses mengajukan kurikulum. Bila syarat terpenuhi maka kurikulum akan diubah menjadi status diajukan. Hasil *Endpoint* ini terlihat pada Gambar 10.



Gambar 10. Mengajukan kurikulum melalui *endpoint*

3.1.5 *Endpoint* Menambah *Reviewer*

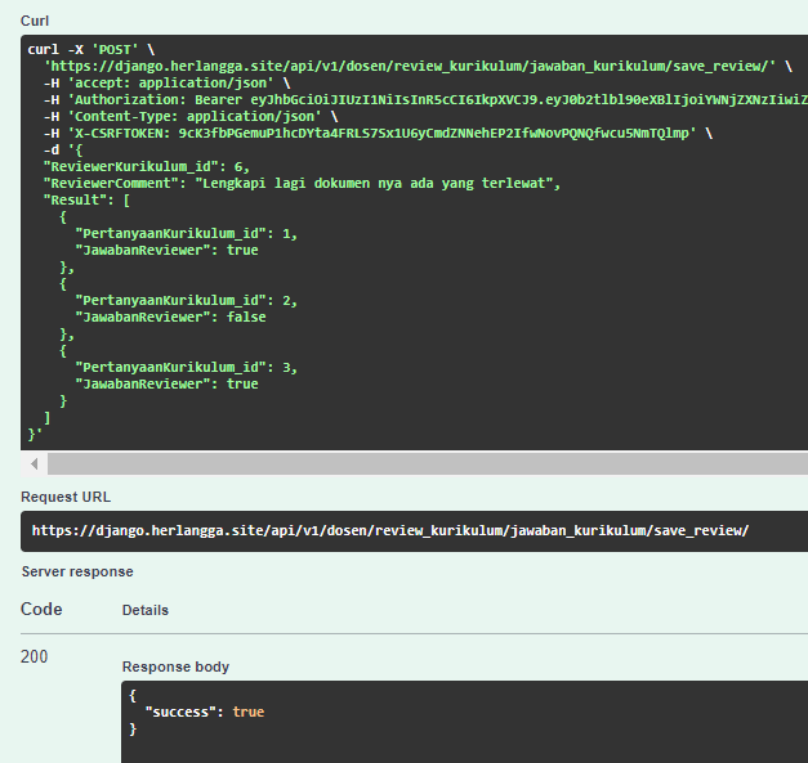
Endpoint ini digunakan untuk manajemen *reviewer* menambah *reviewer* yang me-*review* kurikulum yang telah masuk pada status diajukan. Melakukan hal ini akan mengubah status kurikulum menjadi di-*review*. Hasil dari endpoint ini di tunjukan oleh Gambar 11. Gambar 11 adalah proses menambah reviewer pada sebuah kurikulum yang telah diajukan.



Gambar 11. Menambah *reviewer* untuk *review* kurikulum

3.1.6 *Endpoint Review*

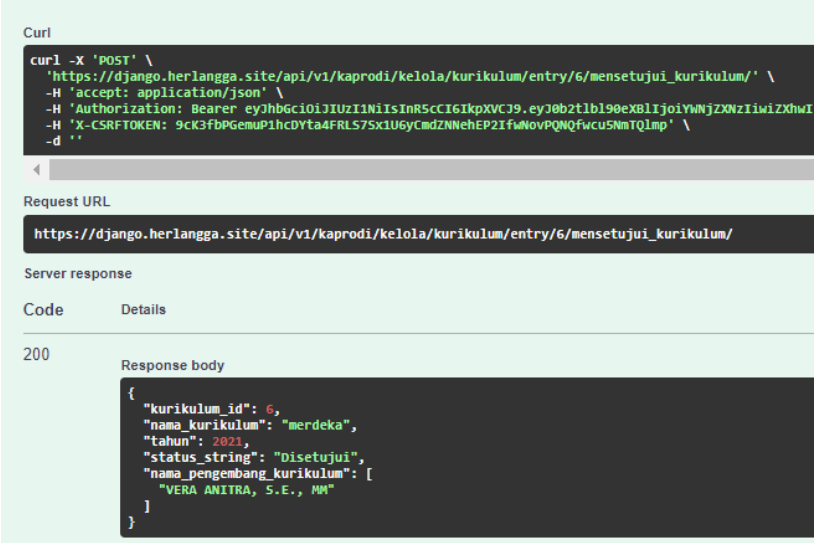
Endpoint ini digunakan sebagai manajemen ketika *reviewer* melakukan *review*. *Reviewer* memberikan jawaban atas kelengkapan yang di tampilkan oleh sistem. Berupa *checklist* dan juga meninggalkan sebuah komen juga untuk tiap *reviewer*. Hal ini mampu dilakukan dengan hasil seperti pada Gambar 12.



Gambar 12. *Reviewer* melakukan *review* dan menyimpan jawaban

3.1.7 *Endpoint* Menyetujui Kurikulum

Endpoint ini digunakan sebagai tempat untuk menyetujui kurikulum dari sisi admin. Dapat dilakukan apabila semua *reviewer* sudah mengatakan setuju atau admin telah meyakini semua komponen lengkap. Apabila dilakukan oleh admin maka kurikulum akan statusnya akan menjadi disetujui dan mengubah semua hasil *review* oleh *reviewer* menjadi setuju. Hasilnya dapat dilihat pada Gambar 13.



Gambar 13. Admin menyatakan kurikulum telah disetujui

Selain dengan *endpoint* yang telah terlihat ada banyak *endpoint* lainnya yang digunakan dalam memenuhi kebutuhan. Daftar kebutuhan yang sudah diuji coba lainnya adalah sebagai berikut.

1. Sinkronisasi Data Dosen dengan server lain.
2. Sinkronisasi Data Lembaga dengan server lain.
3. Autentikasi dengan model *JWT*
4. Kelola Mata kuliah Program Studi
5. Kelola Mata kuliah terhadap kurikulum
6. Pengelolaan CPL program studi
7. Pengelolaan Acuan CPL
8. Pengelolaan Pemetaan CPL program studi terhadap Acuan CPL
9. Pengelolaan Profil Lulusan

Sehingga dengan membandingkan fungsionalitas dan juga hasil dari aplikasinya maka aplikasi ini dapat dibuktikan memenuhi syarat dan dapat di implementasikan

Tabel 1. Uji Fungsionalitas

No	Fungsi	Berfungsi

1.	Sinkronisasi Data Lembaga dengan server lain.	v
2.	Sinkronisasi Data Dosen dengan server lain.	v
3.	Autentikasi dengan model JWT	v
4.	CRUD Mata kuliah Program Studi	v
5.	CRUD Mata kuliah terhadap kurikulum	v
6.	CRUD CPL program studi	v
7.	CRUD Acuan CPL program studi	v
8.	CRUD Profil Lulusan	v
9.	CRUD Pemetaan CPL program studi terhadap Acuan CPL	v
10.	CRU Dokumen Kurikulum	v
11.	CRUD Tim Kurikulum	v
12.	CRU *Review* Kurikulum	v
13.	CRU Kurikulum	v
14.	CRU *Reviewer* Kurikulum	v
15.	*Update* Menyetujui Kurikulum	v
16.	*Update* Mengaktifkan Kurikulum	v
17.	*Permission* Kepala Program Studi	v
18.	*Permission* Pengembang Kurikulum	v
19.	*Permission* Reviewer	v
20.	*Permission* Admin	v

4. PENUTUP

Dengan hal adanya pengembangan perangkat lunak ini telah dapat memenuhi bagian awal dari sebuah sistem yang dapat diintegrasi secara terstruktur ke depannya dan memiliki sebuah standar

yang cukup kuat karena masih memiliki sebuah struktur penulisan aplikasi yang dikenal. Hal ini dapat mengurangi kesulitan pada proses *maintenace* dan juga dapat di kembangkan lebih lanjut. Pada saat ini sistem ini memiliki telah memiliki 16 kelompok *endpoint* yang mengarah pada fungsional dan 4 fungsi yang mengarah kepada kemampuan membatasi akses. Pengujian dilakukan dengan menggunakan metode uji fungsionalitas dan menghasilkan hasil yang memenuhi standar dalam pemenuhan kebutuhan yang dibutuhkan.

Dengan adanya sistem ini diharap proses kurikulum tadi yang memerlukan waktu dan tempat yang cukup banyak (untuk rapat), maka dapat berkurang. Hal ini dapat membuat beban penentuan kurikulum dapat berkurang. Serta aman dalam proses pengembangan lebih lanjut.

DAFTAR PUSTAKA

Adilah, N. S., Hadjaratie, L., & Yusuf, R. (2022). Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis Progressive Web App. Diffusion: Journal of Systems and Information Technology, 2(2), 84–96. https://doi.org/10.37031/DIFFUSION.V2I2.15571

Kumar, M., & Rashid, E. (2018). An Efficient Software Development Life cycle Model for Developing Software Project. International Journal of Education and Management Engineering, 8(6), 59–68. https://doi.org/10.5815/IJEME.2018.06.06

Ikbal H., M. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management. IJFMR - International Journal For Multidisciplinary Research, 5(5). https://doi.org/10.36948/IJFMR.2023.V05I05.6223

Cahyawardani, P. D. (2020). Pengembangan Sistem Informasi Evaluasi Capaian Pembelajaran Lulusan Jurusan Informatika FTI UII. AUTOMATA, 1(1). https://journal.uii.ac.id/AUTOMATA/article/view/13873

Li, X., d'Aragona, D. A., & Taibi, D. (2024). Evaluating Microservice Organizational Coupling Based on Cross-Service Contribution. 435–450. https://doi.org/10.1007/978-3-031-49266-2_30

Mwiinga, P. (2023). Management Information Systems (MIS). Zenodo. https://doi.org/10.5281/zenodo.10406625

Islami, M. H. (2020). PENGEMBANGAN SISI BACK-END APLIKASI PENGELOLAAN SURAT MENGGUNAKAN FRAMEWORK DJANGO DI UNIVERSITAS MUHAMMADIYAH SURAKARTA. https://eprints.ums.ac.id/82503/

Ardani, D., & Aris Rakhmadi, S. T. (2023). Rancang Bangun Sistem Pembayaran Dengan Menerapkan Payment Gateway Pada Pembayaran Layanan Event Management System (Studi Kasus Eventeer). https://eprints.ums.ac.id/110189/