

**IMPLEMENTASI REST DALAM MEMBANGUN WEB SERVICE
MENGUNAKAN GOLANG (STUDI KASUS: MODUL
BERITA APLIKASI SATUDIkti)**

SIGIT PRIADI¹, MOH IDRIS²

^{1,2}Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam
Indonesia

Jl. Kaliurang KM 14.5, Sleman, Yogyakarta 55584

Email: 18523125@students.uui.ac.id¹, moh.idris@uui.ac.id²

ABSTRAK

Aplikasi Satudikti adalah aplikasi dikembangkan yang bertujuan untuk menggabungkan seluruh layanan yang ada pada Ditjen Diktiristek. Salah satu layanan yang akan dibutuhkan yaitu modul Berita. Modul tersebut dibutuhkan sebagai modul penyebaran informasi yang ada pada Ditjen Diktiristek. Untuk mendapatkan data tersebut, aplikasi Satudikti membutuhkan web *service* yang menyediakan kebutuhan data pada modul Berita. Penelitian ini dibuat untuk membangun web *service* aplikasi Satudikti dengan menggunakan metode REST karena kesederhanaannya dan terstandarisasi protokol HTTP. Pengembangan web *service* dibuat menggunakan bahasa pemrograman Golang dengan Echo Framework karena dapat memenuhi kebutuhan dalam membangun web *service* yang menerapkan arsitektur REST. Hasil akhir pada pengembangannya akan menjadi web *service* yang menyediakan API yang membawa data Berita sehingga aplikasi Satudikti dapat menggunakan API tersebut untuk menampilkan daftar berita.

Kata Kunci: Satudikti, REST, Web Service, API, Golang

I. PENDAHULUAN

Aplikasi Satudikti adalah salah satu aplikasi berbasis *mobile* dan web yang dikembangkan oleh Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi (Ditjen Diktiristek). Aplikasi ini merupakan solusi yang diberikan untuk mengatasi masalah yang ada pada Ditjen Diktiristek yaitu banyaknya layanan atau modul yang dikembangkan oleh Ditjen Diktiristek masih bersifat silo dan media penyebaran informasi yang ada menjadi tidak terpusat. Aplikasi Satudikti diharapkan dapat memenuhi kebutuhan tersebut yang mana kebutuhan yang diinginkan adalah menggabungkan semua modul, media penyebaran informasi, maupun pengembangan aplikasi menjadi terpusat, terpadu, dan terintegrasi dalam satu aplikasi. Salah satu kebutuhan modul yang ada pada aplikasi Satudikti adalah

modul berita. Berita adalah suatu pernyataan fakta, informasi, atau laporan dari kejadian suatu peristiwa yang sedang hangat (Restendy, 2017). Modul berita merupakan suatu modul yang menampilkan informasi seputar berita di Ditjen Diktiristek. Salah satu kebutuhan informasi pada modul berita yaitu menampilkan *list* atau daftar berita yang ada pada Ditjen Diktiristek.

Aplikasi Satudikti membutuhkan suatu wadah yang mengolah data maupun informasi agar aplikasi Satudikti hanya fokus menampilkan informasi saja tanpa berinteraksi langsung dengan *database*. *Database* adalah suatu wadah yang menampung kumpulan data yang terintegrasi dan terorganisir sehingga informasi dapat dicari, diakses, dan dikelola dengan cepat (Hesananda et al., 2017). Dengan demikian dibutuhkan web *service* sebagai wadah yang mengolah seluruh data dan interaksi terhadap *database*. Menurut Saputra dan Fathoni Aji (2018) web *service* merupakan layanan web yang disediakan untuk melakukan interoperabilitas antar sistem pada jaringan menggunakan format pertukaran data saat mengirim pesan atau informasi dan dibuat untuk berinteraksi atau berkomunikasi antar platform maupun aplikasi. Salah satu bentuk keluaran dari web *service* berupa API. API adalah antarmuka aplikasi yang berupa kumpulan instruksi yang disimpan di *library* yang menjelaskan bagaimana aplikasi dapat berkomunikasi dengan aplikasi lain untuk berbagi data (Firdaus et al., 2019).

Dilihat dari penelitian tersebut, dibutuhkan metode yang dapat membantu dalam membangun suatu web *service*. Metode yang akan digunakan adalah REST. Menurut Akbar (2018) REST atau singkatan dari *Representational State Transfer* merupakan penyederhanaan suatu metode untuk melakukan pengiriman suatu data atau informasi pada jaringan yang terstandarisasi protokol HTTP. Dalam pengembangannya, bahasa pemrograman yang akan dipakai untuk membangun web *service* yang mengimplementasikan arsitektur REST yaitu Golang (*Google Language*). Golang dipilih karena bahasa pemrograman yang diciptakan Google tahun 2009 oleh Ken Thompson, Robert Griesemer dan Rob Pike memiliki keunggulan dalam kecepatannya, kehandalan, skalabilitas, dan kesederhanaannya (Sari & Hidayat, 2022).

Dengan adanya web *service*, harapannya dapat mempermudah aplikasi Satudikti dalam mendapatkan data dan informasi yang akan ditampilkan kepada pengguna aplikasi tersebut. Dari API yang tersedia pada web *service* yang dikembangkan tidak perlu diolah oleh aplikasi Satudikti karena data yang ada pada API tersebut sudah siap digunakan. Tidak hanya itu, web *service* tersebut bisa digunakan oleh aplikasi maupun *platform* lainnya yang membutuhkan.

II. METODE PENELITIAN

Dalam proses pengembangan web *service* akan menggunakan arsitektur REST. Ada beberapa komponen yang diperhatikan pada arsitektur REST sebagai berikut (Perdana, 2018):

- a. URL *design*, dalam mengakses *resource* pada suatu API, dibutuhkan *Uniform Resource Locator* (URL) atau disebut *endpoint*. Contohnya yaitu *customers, customers/123*, dan lainnya.
- b. HTTP *verbs*, metode HTTP yang digunakan ketika *client* melakukan *request* ke *server* seperti GET, POST, PUT, dan DELETE.
- c. HTTP *response code*, terdapat 3 jenis kode pada suatu REST API ketika *client* melakukan *request* yaitu 2XX menandakan *request* berhasil diminta, 4XX menandakan terjadinya kesalahan pada *client* ketika melakukan *request*, dan 5XX menandakan terjadinya kesalahan pada *server* ketika memproses *request* yang diminta.
- d. Format *response*, *server* akan mengembalikan suatu response dan format yang akan digunakan yaitu *JavaScript Object Notation* (JSON).

Ada tiga tahapan utama dalam menerapkan REST API pada web *service* yang akan dibangun yaitu perancangan, implementasi, dan pengujian.

a. Perancangan

Tahap perancangan merupakan tahap penting yang dilakukan pada tahap awal sebelum sistem dikembangkan. Tahap ini digunakan sebagai pedoman atau panduan awal sebelum melakukan implementasi. Hal yang akan dilakukan

pada tahap ini yaitu merancang basis data untuk kebutuhan modul Berita pada aplikasi Satudikti dan membuat tabel kebutuhan suatu API pada modul Berita.

b. Implementasi

Tahap ini mengimplementasikan rancangan yang dibuat pada tahap sebelumnya. Bahasa pemrograman Golang akan digunakan dalam membangun web *service* dan memanfaatkan Echo Framework yang dapat mempermudah penerapan arsitektur REST dengan Golang.

c. Pengujian

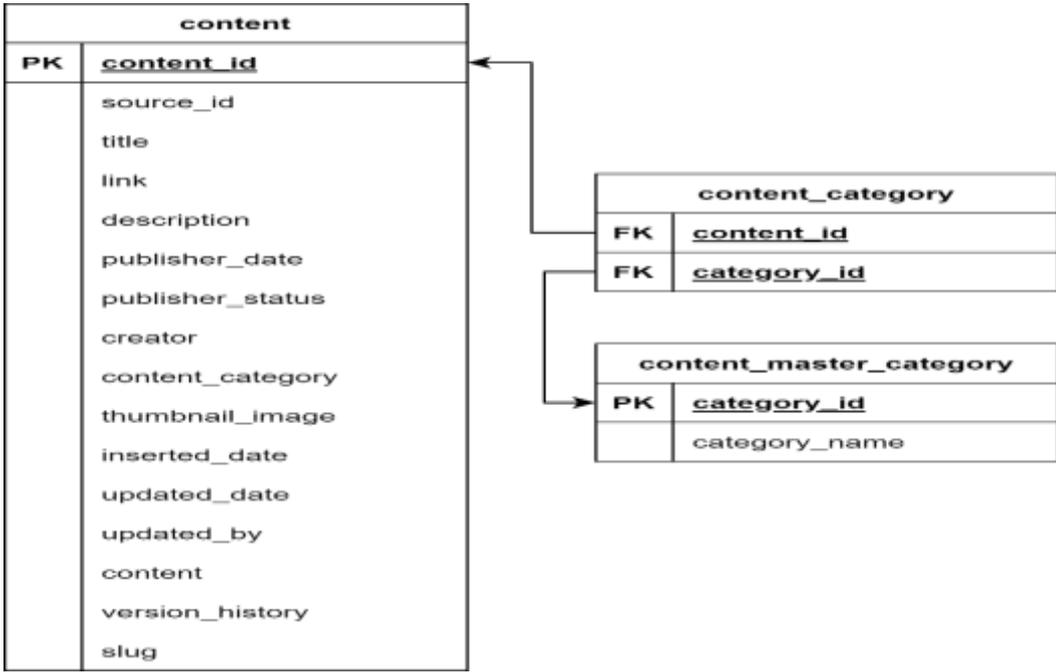
Tahap ini merupakan tahap akhir yang akan dilakukan. Ketika web *service* yang dibuat telah menyediakan API yang dibutuhkan maka API tersebut harus diujikan untuk memastikan apakah API telah berfungsi dengan baik dan sesuai rancangan yang telah dibuat. Tahap pengujian API menggunakan *tools* atau aplikasi yaitu Postman.

III. HASIL DAN PEMBAHASAN

Dalam bab ini akan dibahas hasil dari setiap tahap yang sudah dijelaskan pada bab sebelumnya.

A. Perancangan

Pada modul Berita dibutuhkan entitas atau tabel basis data yang dapat menampung data yang dibutuhkan. Gambar 3.1 menunjukkan ada 3 tabel yang telah dirancang yaitu *content*, *content_category*, dan *content_master_category*. Dengan adanya tabel tersebut harapannya API yang dibuat dapat memenuhi kebutuhan pada aplikasi Satudikti dalam menampilkan data atau informasi.



Gambar 3.1. Rancangan Tabel Modul Berita

Setelah rancangan tabel basis data telah dibuat, selanjutnya membuat rancangan kebutuhan suatu API pada modul Berita. Kebutuhan tersebut akan disesuaikan dengan komponen yang ada pada arsitektur REST. Tabel menunjukkan kebutuhan API pada modul Berita.

Tabel 1. Kebutuhan API pada Modul Berita

URL	/berita
Method	GET
URL Params	{ slug: "required string" }
Data Params	None
Headers	{ Content-Type: "application/json" }
Success Response	Code 200: { current_page: integer, data: [{

	<pre> news_id: integer, slug: string, title: text, link: text, description: text, publisher: string, publisher_date: datetime, publisher_status: string, creator: string, thumbnail_image: string, inserted_date: datetime, categories: [{ category_id: integer, category_name: string }],], last_page: integer, total_item: integer, total_item_per_page: integer } </pre>
Error Response	<pre> Code 404: { message: "Not Found" } Code 500: { message: "Terjadi Error". error: "{Pesan Error}" } </pre>

B. Implementasi

Pengembangan web *service* berbasis REST menggunakan bahasa pemrograman Golang dengan Echo Framework pada modul Berita. Dalam pengembangannya, terdapat tahapan dalam membuat blok kode program yaitu sebagai berikut:

1. *controller*, berisi fungsi atau *method* yang berkaitan dengan penanganan *response* suatu *request* oleh *client*. Sesuai dengan kebutuhan API, pada

Gambar 3.2 menunjukkan kode program yang menangani URL Params dan Gambar 3.3 menunjukkan kode program yang menangani *Return Response* dan *Format Response*.

```

22 func (controller *NewsControllerImpl) GetListBerita(c echo.Context) error {
23     var (
24         count int
25         result []web.ListContentJSON
26         modul = "news"
27     )
28
29     category := c.QueryParam("category_id")
30     title := c.QueryParam("title")
31     page, err := strconv.Atoi(c.QueryParam("page"))
32     if err != nil {
33         return c.JSON(http.StatusBadRequest, map[string]string{
34             "message": "Page harus angka.",
35             "error":   err.Error(),
36         })
37     }
38     limitParam := c.QueryParam("limit")
39     if limitParam == "" {
40         limitParam = "10"
41     }
42     limit, err := strconv.Atoi(limitParam)
43     if err != nil {
44         return c.JSON(http.StatusBadRequest, map[string]string{
45             "message": "Limit harus angka.",
46             "error":   err.Error(),
47         })
48     }
49 }

```

Gambar 3.2 Kode Program bagian URL Params

```

50     switch {
51     case len(category) > 0 && len(title) > 0:
52         count, result, err = controller.NewsService.GetListContentByCategoryTitle(
53             c.Request().Context(), modul, category, title, page, limit)
54     case len(category) > 0:
55         count, result, err = controller.NewsService.GetListContentByCategory(c.
56             Request().Context(), modul, category, page, limit)
57     case len(title) > 0:
58         count, result, err = controller.NewsService.GetListContentByTitle(c.
59             Request().Context(), modul, title, page, limit)
60     default:
61         count, result, err = controller.NewsService.GetListContentByOrder(c.
62             Request().Context(), modul, page, limit)
63     }
64
65     if err != nil {
66         return c.JSON(http.StatusInternalServerError, map[string]string{
67             "message": "Terjadi Error",
68             "error":   err.Error(),
69         })
70     }
71
72     if count == 0 {
73         return c.JSON(http.StatusNotFound, map[string]string{
74             "message": "Berita tidak ditemukan",
75         })
76     }
77
78     return c.JSON(http.StatusOK, helper.ToWebResponseListContent(page, count,
79         limit, result))
80 }

```

Gambar 3.3 Kode Program bagian Return dan Format Response

2. *handler*, berisi fungsi atau *method* yang berkaitan terhadap pendefinisian suatu alamat *endpoint*. Sesuai dengan kebutuhan API, pada Gambar 3.4 menunjukkan kode program yang menangani komponen *endpoint* yang

didefinisikan yaitu *endpoint* “/berita” dan Method yang didefinisikan yaitu GET.

```

12 func BeritaHandler(g *echo.Group, db *sql.DB) {
13     var (
14         //repo
15         newsRepo = repository.NewNewsRepository(db)
16         //service
17         newsService = service.NewNewsService(newsRepo)
18         //controller
19         newsController = controller.NewNewsController(newsService)
20     )
21     g.GET("/berita", newsController.GetListBerita)
22 }

```

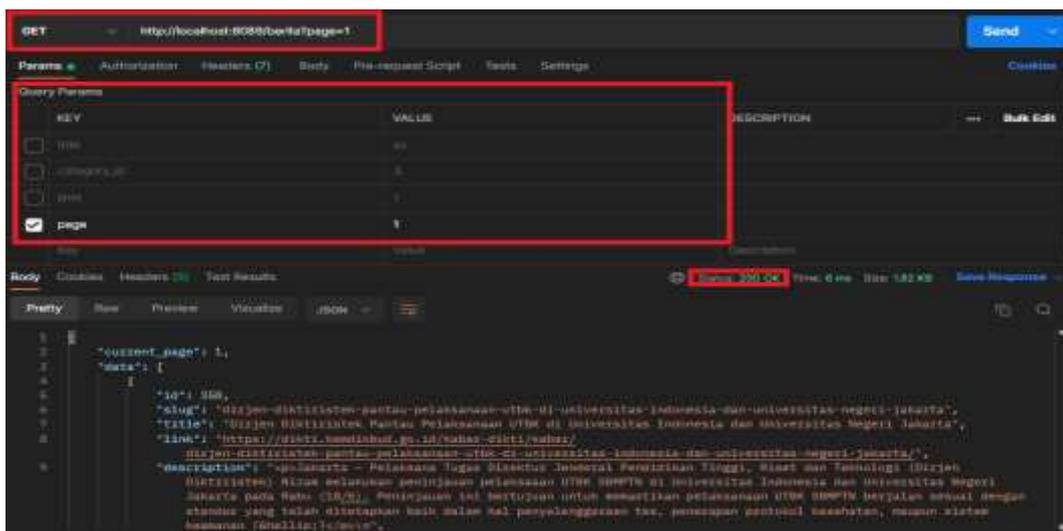
Gambar 3.4 Kode Program bagian *Endpoint* dan Method

3. *helper*, berisi fungsi atau *method* bantuan yang dapat membantu pembuatan kode program.
4. *model*, berisi fungsi atau *method* yang berkaitan terhadap mendefinisikan suatu struktur data pada *response* dan *request*.
5. *repository*, berisi fungsi atau *method* yang berkaitan terhadap *query logic* pada basis data.
6. *service*, berisi fungsi atau *method* yang berkaitan terhadap *business logic* atau semua operasional yang dilakukan yang menggunakan logika pemrograman maupun pengolahan data.
7. *route*, berisi fungsi atau *method* yang bertujuan untuk mendaftarkan *endpoint* yang ada pada *handler* agar bisa diakses oleh *client*.
8. *server*, berisi fungsi atau *method* yang berkaitan terhadap pendefinisian alamat URL yaitu *Host* dan *Port*. *Host* yang digunakan yaitu *localhost* dengan *Port* 8089.

C. Pengujian

Setelah implementasi selesai dan diperoleh hasil pengujian. Hasil pengujian ini menunjukkan kegunaan dan kompatibilitas dari REST API yang dibangun. *Endpoint* /berita untuk melihat seluruh *list* berita yang disediakan oleh server. Pengujian dilakukan menggunakan Postman dengan cara mengirimkan *request* data ke *endpoint* tersebut dengan *method* GET dan memasukkan URL

`http://localhost:8089/berita`. Untuk membaca seluruh data berita, tersedia 4 URL Params yaitu `title`, `category_id`, `limit`, dan `page`. Kemudian, server memberikan response seperti pada Gambar 3.5 yang menunjukkan response berhasil dengan status code 200 yang menandakan `success` dan data yang diberikan memiliki format berupa JSON.



Gambar 3.5 Response Success Membaca List Berita

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dapat disimpulkan sebagai berikut:

- a. Terdapat 3 tabel basis data yang dibutuhkan dalam menyediakan data untuk modul Berita yaitu `content`, `content_category`, dan `content_master_category`.
- b. Dalam membuat kebutuhan pada suatu API, ada 7 komponen yang perlu diperhatikan yaitu URL, Method, URL Params, Data Params, Headers, Success Response, dan Error Response.
- c. Metode REST dapat diimplementasikan dalam membangun *Web Service* yang berjalan pada protokol HTTP dalam berkomunikasi sehingga kebutuhan data pada modul Berita yang ada pada aplikasi Satudikti dapat terpenuhi.
- d. Bahasa pemrograman Golang dengan Echo Framework dapat mempermudah pembangunan *Web Service* yang menerapkan metode REST.

- e. Berdasarkan hasil penelitian yang telah dilakukan, kebutuhan pada aplikasi Satudikti dalam mendapatkan data dan informasi pada modul Berita dapat diselesaikan dengan membangun Web Service yang menyediakan API yang dibutuhkan menggunakan metode REST yang berjalan pada protokol HTTP dalam berkomunikasi.

DAFTAR PUSTAKA

- Akbar, M. (2018). *Pengembangan Restful Api Untuk Application Specific High Level Location Service*. <https://dspace.uin.ac.id/handle/123456789/9836>
- Firdaus, A., Widodo, S., Sutrisman, A., Gading, S., Nasution, F., Mardiana, R., Komputer, J. T., Negeri, P., & Palembang, S. (2019). Rancang bangun sistem informasi perpustakaan menggunakan web service pada jurusan teknik komputer polsri. *Jurnal Informanika*, 5(2).
- Hesananda, R., Leslie, H., Spits Warnars, H., & Sianipar, N. F. (2017). Supervised classification karakter morfologi tanaman keladi tikus (*typhonium flagelliforme*) menggunakan database management system. *Jurnal Sistem Komputer*, 7(2), 50–58.
- Perdana, M. A. K. (2018). *Pengembangan rest api layanan penyimpanan menggunakan metode rapid application development (studi kasus: PT. XYZ)*.
- Restendy, M. S. (2017). Daya Tarik Jurnalistik, Pers, Berita Dan Perbedaan Peran Dalam News Casting. *Al-Hikmah: Jurnal Pendidikan Dan Studi Islam*, 4(2), 1–12.
- Saputra, D., & Fathoni Aji, R. (2018). Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus STT Indonesia Tanjungpinang). *Bangkit Indonesia*, 2.
- Sari, A. S., & Hidayat, R. (2022). Designing website vaccine booking system using golang programming language and framework react JS. *Journal of Information System, Informatics and Computing Issue Period*, 6(1), 22–39. <https://doi.org/10.52362/jisicom.v6i1.760>