Analysis and Design of Headless CMS and Graphql in Back-End Development

Yudha Pandu Dewanata[*1], Hadiq[2]
[1,2]Informatics Engineering, STIKOM PGRI Banyuwangi
E-mail: [*1]yudhapand002@gmail.com, [2]hadiq@stikombanyuwangi.ac.id

Abstract

The application of the website in Village Government can also be useful for informing news or services the community. Therefore, here the author creates an information website for Pondoknongko Village Government. In Pondoknongko Village itself, it does not yet have an information website that is easy to access by admins and users. Indeed, there is already a built-in website from the government, but the admin is still having a hard time adapting and accessing it. There were many deficiencies in the previous website, such as the difficulty of accessing data, and adding and editing data. Therefore, the solution to overcome this problem is to create a website that uses Strapi as a database manager so that it is easy for the admin to manage the data that will be loaded on the information website. The author uses Strapi as a data manager and Graphql as a liaison between the Back end and Front End. This research is also based on a review of the information system used by Pondoknongko Village, the website created has been successfully developed and put to good use by Pondoknongko Village and the community.

Keywords *— Graphql, Headles CMS, Pondoknongko Village, Data, Strapi, Website*

1. INTRODUCTION

The rapid development of technology today has provided many benefits in progress in various social aspects[1]. The use of technology by humans in helping to complete work is a necessity in life. The rapid development of information technology has brought the world into the digital era[2]. Fast and accurate data processing/management is very important to produce the required data, especially if the data being processed is financial data, it also requires high accuracy/speed of processing[3].

Websites are very important nowadays, because various authorities such as schools need websites to carry out campaigns or disseminate information. Websites are a widely used source of information. Various web applications are made with the aim that users can easily and quickly interact with information providers through online media. A website is also an implementation of a web programming language (web programming) which consists of a collection of information pages, text data, static or moving images, animated data, sound, video or a combination of all, both static and dynamic, forming a group of buildings. related to each other, each connected to the web network (hyperlink)[4].

Also inseparable from the government system in the village, village administrations now also often use computers to complete their work. The use of information technology that is developing in aspects of government life is expected to bring benefits in strengthening

society through improving information and services, as well as increasing more efficient governance[5]. It's not surprising that everyone can see that many village administrations have social media and websites, which are useful as information and communication facilities for the community. Communication is one of the important things in describing something, such as being an intermediary, to present something to the public.

Therefore, the task that the author is working on is related to creating a website for village government, more precisely the Government of Pondoknongko Village, Kabat District, Banyuwangi Regency. Here the author will implement the back end on the Pondoknongko Village website using strapi.

At first, the website in Pondoknongko Village was only inherited from the central government. So here the author will create a new website with the method that will be applied, in order to help admin ease in managing the data to be published.

The problems discussed in this study are how to design a Headless CMS strapi according to what is needed, how to configure a Headless CMS strapi so that the Front-End can make requests according to needs, how to design a graphql query structure so that it can respond to the Front-End according to with what is needed.

According to Suwarno & Teddy Sanjaya in their research entitled "Design and Implementation of a website-based Company Profile as Promotional Media at PT. Hassani Can Packaging" said that in the current era, the development of information technology is growing rapidly, a lot of information that we can access through the website-based digital world. One example of the impact of technological developments is that companies begin to create company profiles. The purpose of this research is to create a company profile for Hassani Can Packaging as a tool and media to be able to reach more and wider areas as well as a means to be able to overcome difficulties in finding new customers. In this study, the waterfall method will be used to design the company profile website, where the website has a homepage, about us, product, career and contact us. And for the conclusion of the research designed, this company profile website has been distributed online so that the information needed can be accessed anytime and anywhere with a network connection, and the project results in the form of a company profile website are able to help PT. Hassani Can Packaging to introduce its products to the wider community[6] [7].

And according to Nur Ayuni Nor Sobri and friends in her research entitled "Comparison between Headless CMS and Backend-as-a-Service Products for E-Suripreneur Backend" said that many businesses have entered the business market in recent years, each focuses on a specific sector or industry. They often have traits that set them apart from the competition. A startup's main goal is to get their product to market quickly and build user momentum for their product and system. However, there may be various problems and obstacles to bringing the product to market, one of which is product software development. Fortunately, there are a variety of development tools available today to help bootstrap early development complexities, which is beneficial for companies trying to quickly get their product to market. Backend-as-a-service (BaaS) and headless content management system (CMS) are two solutions that help simplify and increase the pace of software development,

enabling faster time to market. This article compares Strapi, Directus, and Supabase as possible backend systems for e-Suripreneur systems. Directus appears to be the best fit for an e-Suripreneur system, as it meets the baseline system requirements and provides additional benefits over Strapi and Supabase. And the conclusions drawn from this research are comparisons between the two Headless CMS Strapi and Directus, and one BaaS application, Supabase, which will serve as the backend for the e-Suripreneur system. Directus seems to be the most suitable solution for our system based on the comparison results, as it fits all the requirements of an e-Suripreneur project. In future work, we will investigate the optimal option for integrating the selected Directus system for the e-Suripreneur backend with third-party services to meet the needs of more sophisticated systems[8].

Meanwhile, according to Rahayu Pangestika & Raden Teguh Dirgahayu in their research entitled "Back-End Development of Village School Data Collection Information Systems for the Pendar Foundation Yogyakarta Community" said that the Pendar Foundation Community is a social organization engaged in philanthropy and education aimed at the children of farmers and workers in rural areas of Yogyakarta. Pendar Foundation wants rural children to get sufficient knowledge and there is no gap in knowledge between villages and cities, so they hold "Village School" activities. “Village Schools” are held in several villages with a large number of children and with a variety of materials, but the method of recording activity data currently used is still manual, namely using Microsoft Word. The Pendar Foundation community feels that this method is not effective because they often find it difficult to synchronize data. For this reason, the Pendar Foundation Community needs an information system that can record every “Village School” activity. The purpose of developing an information system for the Pendar Foundation Community data collection is to make it easier to record and classify “Village School” activity data for the Pendar Foundation Community. In addition, the system is also designed to be accessible to all community members so that all members can view and manage data in detail. Information system development is carried out using the CodeIgniter framework with a MySQL database. The results of this research are database implementation and system logic. For its own conclusions Based on the results of black-box testing, where all the features for data collection needs for the activities of the Pendar Foundation community village school have been able to run properly, this research has succeeded in developing an information system for data collection for the Pendar Foundation Yogyakarta community village school using the PHP programming language , Javascript and HTML, and MySQL as the system database. The system is implemented by conducting trials on members of the Pendar Foundation community with stages such as Admin registering an account for the Pendar Foundation community staff, Staff managing village school information, including adding the names of village school students , adding village school attendance, adding lesson plans, adding materials and adding photos to galleries, Volunteers manage lesson plans and materials, Volunteers and other staff get information based on village school data that has been uploaded. Then, system testing is carried out using the black box method to determine system functionality. From the testing stage it is known that the system functionality can run according to user needs. Based on feedback from users, namely members of the Pendar Foundation community, users feel that the system has met their needs and is as expected[9].

These studies have not explained how to develop Headless CMS using Strapi and Graphql. Therefore, this research will discuss further how to develop the back-end using strapi and graphql.

2. RESEARCH METHOD

Research Methods is a scientific way to obtain data with the aim of being able to describe, prove, develop and discover knowledge, theory, to understand, solve, and anticipate problems in human life. This study has several research methods such as data collection, requirements analysis and system design.

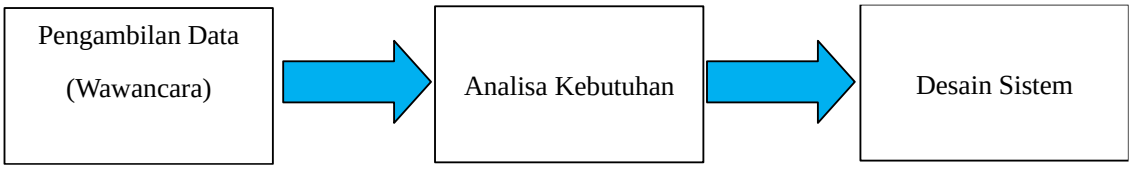

Figure 1. Research Methods

2.1. Data Collection (Interview)

In this research data collection method using the interview method, the researchers asked about the wants and needs of what the informants wanted in the program. In the interview, the researcher found out what needs the resource person wanted. And the results of these needs are then written into a needs analysis by the researcher.

2.2. Needs Analysis

The development of a learning media is carried out in several stages, according to the instructional design model used. In general, the first stage in developing instructional media is conducting a needs analysis [10]. Needs Analysis is a systematic process to determine goals, identify the actual conditions expected, and set priorities for action. The needs analysis here contains the expected features in an information system that is made according to what the Pondoknongko Village wants. This needs analysis is described using a table writing system as follows:

Tabel 1. Needs Analysis

Village Needs	Information
Profil	Filled with information data of Pondoknongko Village
Berita	Contains the Latest News information data and has a sub menu in the form of News Category which contains Activity News, MSME News, and Articles
User	Filled with User Permission Form to Sign Up
Kontributor	Fill in the Log In or Sign In Form
Service	Filled with Community Service information data
Footer	Filled with contact person information data for the village

Information:

Profil contains information data related to Pondoknongko Village, such as the name of the Village Head, area area, population number etc. Berita contains information data related to the Latest News which is the latest content. Kategori Berita contains selected data to be able to choose the desired news content such as Activity News, MSME News or Articles. Berita Kegiatan contains data related information about activities carried out by the Village Party and the Village Community. Berita UMKM contains data related information about MSME activities carried out by the community. Artikel contains data related information about incident reports and articles written by the community in Pondoknongko Village. User contains a form for registering an account for people who want to upload the report or essay they want. Kontributor contains a form to log in or enter into an account that was previously created. Layanan contains information data related to community service procedures in Pondoknongko Village. Footer contains information data related to the contact person from Pondoknongko village.

2.3. System Design

System design is the process of identifying system components. System design is intended for programmers and other engineers who implement systems. The system design phase is usually carried out after the analysis phase has been completed. In this research there are three system designs namely Collection, Json Structure and Query.

Content-type Builder is a core plugin from Strapi. This is a feature that is always enabled by default and cannot be disabled. However the Content-type Builder can only be accessed when the app is in a development environment. Collection is a description and description of the database and the tables used. After creating a collection, the admin will be redirected to the content manager to fill in each field with the required data. If you want to add data or change data, the admin can do it in the content manager, but if you want to add a field the admin is required to add it to the content type-builder and choose which collection you want to add the field to.

JSON (JavaScript Object Notation) is a very lightweight data exchange format that is easier for humans to read and write, making it easier for computers to translate and generate. JSON is made up of two structures: A collection of name/value pairs. In some languages, this is represented as an object, record, struct, dictionary, hash table, keyed list, or associative array.

Query is a command that has the ability to set the data that must be displayed or the data that is displayed as desired. In addition, the query is the syntax or command that is used so that we can interact with each other. The meaning of query in English is a request for information.

2.3.1. Collection

The table below describes the profile collection, which contains several fields in the profile collection, such as: judulProfil, isiProfil, UrlSlugProfil, dan gambarprofil dengan menggunakan type data text, rich text, UID dan Media.

Tabel 2. Collection Profil

Field	Type Data
judulProfil	Text
isiProfil	Rich Text
UrlSlugProfil	UID
GambarProfil	Media

The table below describes the news collection in which the collection contains fields judulBerita, isiBerita, IsiBerita2, UrlSlugBerita, GambarBerita, Konfirmasi, and Kategori_berita. Which here to use type data text, rich text, UID, Media, Boolean, and Relation for kategoriBerita, relation here serves to connect collection berita with collection kategori berita.

Tabel 3. Collection News

Field	Type Data
judulBerita	Text
isiBerita	Rich Text
isiBerita2	Rich Text
UrlSlugBerita	UID
GambarBerita	Media
Kategori_berita	Relation with KategoriBerita
Konfirmasi	Boolean

The table below describes about collection Kategori Berita which one is inside the collection exists Title, UrlSlugKtgBerita, beritas, kegiatans, umkms and artikels. And it only has 3 kinds of data types namely text, UID, and Relations, this relationship is to connect between news categories with news, activities, umkm and articles.

Tabel 4. Collection News Category

Field	Type Data
Title	Text
UrlSlugKtgBerita	UID
beritas	Relation with Berita
kegiatans	Relation with Kegiatan
umkms	Relation with UMKM
artikels	Relation with Artikel

The table below is a table collection kegiatan, umkm and artikel. Here the contents of each field are the same, that is, there are judul, isi, isi2,urlslug, kategori_berita and gambar. There is also a data type in the form text, rich text, UID, relation with kategoriberita and media. However, within each field there is the name of its own collection for example

judulkegiatan, judulumkm, judulartikel etc. Here only will be displayed collection kegiatan For example.

Tabel 5. Collection Activity

Field	Type Data
judulKegiatan	Text
isiKegiatan	Rich Text
isiKegiatan2	Rich Text
UrlSlugKegiatan	UID
Kategori_berita	Relation with KategoriBerita
GambarKegiatan	Media

The table below describes about Collection User which one is deep This User Collection has a field Username, Email, Provider, Password, resetPasswordToken, confirmtionToken, confirmed, blocked, role. And have a data type like Text, Email, Text, Password, Bolean, and Relation for role, which relation with this role to connect the user to the role of users-permission.

Tabel 6. Collection User

Field	Type Data
Username	Text
Email	Email
Provider	Text
Password	Password
resetPasswordToken	Text
confirmationToken	Text
confirmed	Boolean
Blocked	Boolean
role	Relation with role (from: users-permissions)

The table below describes about Collection Kontributor which in this Contributor Collection there are several fields such as username, email, password, tempatlahir, tanggallahir, alamat, nomortelepon, NIK, gambarKTP. And it has several data types viz text, email, password, date, rich text, dan number.

Tabel 7. Collection Contributor

Field	Type Data
Username	Text
Email	Email
Password	Password
TempatLahir	Text
TanggalLahir	Date
Alamat	Rich Text
NomorTelepon	Number
NIK	Text
gambarKTP	Text

The table below describes about Collection Layanan which in this Service Collection there are only a few fields such as judulLayanan, gambarLayanan and UrlSlugLayanan. And has several data types namely text, media and UID.

Tabel 8. Collection Service

Field	Type Data
judulLayanan	Text
gambarLayanan	Media
UrlSlugLayanan	UID

The table below describes about Collection Footer which in the Collection Footer there are only a few fields such as alamat, jam_kerja dan kontak. Which only has one data type in the form of rich text.

Tabel 9. Collection Footer

Field	Type Data
Alamat	Rich text
Jam_kerja	Rich text
kontak	Rich text

2.3.2. Json Structure

Tabel 10. Segmen Program 1 Struktur Json Profil

```
{"data": {"profils": {"data": [{"id": "1","attributes": {
            "gambarProfil": {
             "data": [{"id": "1","attributes": {"url": "/"}}]},
            "judulProfil": "","isiProfil": "/",
            "UrlSlugProfil": ""}}]}}}
```

The program segment above explains the profile design written using the json structure. Where in the profile data there are attribute fields gambar, judul, isi and url each of which will bring up data according to the content being worked on.

Tabel 11. Segmen Program 2 Struktur Json News

```
{"data": {"beritas": {"data": [{"id": "1","attributes": {
           "gambarBerita":
             {"data": [{"id": "2","attributes": {"url": "/"}}]},
           "judulBerita": "","isiBerita": "",
           "isiBerita2": "","Konfirmasi": true or false,
           "UrlSlugBerita": ""
          "kategori_beritas": {"data": [{"attributes":
            {"Title": "","UrlSlugKtgBerita": ""}}]}}}]}}}
```

In the program segment above, it explains the news design written using the json structure, in the news data there are attributes in the form of fields such as judulberita, isiberita, isiberita2, konfirmasi, urlslugberita, gambarberita and kategoriberita. for gambarberita and kategoriberita it still has the attribute again, gambarberita has a url attribute whose function is to call the link from the uploaded image, and for kategoriberita has an attribute of title and urlslugberita which is its function to categorize news created and related to collection Kategori Berita.

Tabel 12. Segmen Program 3 Struktur Json News Category

```
{"data": {"kategoriberitas": {"data": [{"attributes": {
        "Title": "","UrlSlugKtgBerita": "",
        "beritas": {"data": [{"id": "1","attributes": {
          "gambarBerita": {"data": [{"id": "2","attributes": {
            "url": "/"}}]},
          "judulBerita": "","isiBerita": "",
          "isiBerita2": "","UrlSlugBerita": ""}}]},
        "kegiatans": {"data": [{"attributes": {
          "judulKegiatan": "","isiKegiatan": "",
          "isiKegiatan2": "","UrlSlugKegiatan": "",
          "gambarKegiatan": {"data": [{"attributes": {
                    "url": "/"}}]}}}]}
```

Tabel 13. Advanced Segmen Program 3 Struktur Json Kategori Berita

```
        "artikels": {"data": [{"attributes": {
          "judulArtikel": "","isiArtikel": "",
          "isiArtikel2": "","UrlSlugArtikel": "",
          "gambarArtikel": {"data": [{"attributes": {
                    "url": "/"}}]}}}]}
        "umkms": {"data": [{"attributes": {
          "judulUMKM": "","isiUMKM": "",
          "isiUMKM2": "","UrlSlugUMKM": "",
          "gambarUMKM": {"data": [{"attributes": {
                    "url": "/"}}]}}}]}}}}}}}}
```

In the above program describes the design news category written using json structure. In this design there are attributes from the news category namely tittle, urlslugkategoriberita, beritas, kegiatans, artikels, umkms. Fields beritas, kegiatans, artikels and umkms This is a relation between the news category collection and the four collections, therefore in the news category collection there are attribute fields from the news collection, activities, articles and umkm. This relationship is useful to make it easier for the front-end to retrieve data and there is no need to arrange or rearrange it but only need to enter queries to display data that has been created by the back-end. And in the news, activities, articles and umkm collections it appears that they have the same attribute fields because the layout and appearance are the same, only what differs is the content of the content displayed. Therefore, the collection itself is created to make it easier for the front-end to sort and categorize the news they want to produce, it is also useful so that later readers do not have trouble finding the desired news according to the existing categories.

Tabel 14. Segmen Program 4 Struktur Json Activity

```
{"data": {"kegiatans": {"data": [{"id": "1","attributes": {
           "gambarKegiatan":
             {"data": [{"id": "2","attributes": {"url": "/"}}]},
           "judulKegiatan": "","isiKegiatan": "",
           "isiKegiatan2": "","UrlSlugKegiatan": ""
           "kategori_beritas": {"data": [{"attributes":
             {"Title": "","UrlSlugKtgBerita": ""}}]}}}]}}}
```

In the above program segment describes the design of collection kegiatan which is written using the json structure. In the design there are attribute fields such as judulkegiatan, isikegiatan, isikegiatan2, urlslugkegiatan, gambarkegiatan dan kategori_beritas. Same as the

previous explanation in the attribute field gambarkegiatan there is another attribute in the form of a url which is useful for displaying uploaded images. And in the attribute field kategori_beritas there is another attribute in the form of title and urlslugktgberita, attribute kategori_beritas this is an intermediate relationship collection kegiatan with collection kategori berita.

Tabel 15. Segmen Program 5 Struktur Json Article

```
{"data": {"artikels": {"data": [{"id": "1","attributes": {
           "gambarArtikel":
             {"data": [{"id": "2","attributes": {"url": "/"}}]},
           "judulArtikel": "","isiArtikel": "",
           "isiArtikel2": "","UrlSlugArtikel": ""
          "kategori_beritas": {"data": [{"attributes":
            {"Title": "","UrlSlugKtgBerita": ""}}]}}}]}}}
```

In the above program segment describes the design of collection articles written using json structure. In the design there are attribute fields such as judulartikel, isiartikel, isiartikel2, urlslugartikel, gambarartikel and kategori_beritas. Same as the previous explanation in the attribute field gambarartikel there is another attribute in the form of a url which is useful for displaying uploaded images. And in the attribute field kategori_beritas there is another attribute in the form of title and urlslugktgberita, attribute kategori_beritas this is an intermediate relationship collection artikel with collection kategori berita.

Tabel 16. Segmen Program 6 Struktur Json UMKM

```
{"data": {"umkms": {"data": [{"id": "1","attributes": {
           "gambarUMKM":
             {"data": [{"id": "2","attributes": {"url": "/"}}]},
           "judulUMKM": "","isiUMKM": "",
           "isiUMKM2": "","UrlSlugUMKM": ""
          "kategori_beritas": {"data": [{"attributes":
            {"Title": "","UrlSlugKtgBerita": ""}}]}}}]}}}
```

In the above program segment describes the design of collection UMKM which is written using the json structure. In the design there are attribute fields such as judulUMKM, isiUMKM, isiUMKM2, urlslugUMKM, gambarUMKM and kategori_beritas. Same as the previous explanation in the attribute field gambarUMKM there is another attribute in the form of a url which is useful for displaying uploaded images. And in the attribute field kategori_beritas there is another attribute in the form of title and urlslugktgberita, attribute kategori_beritas this is an intermediate relationship collection UMKM with collection kategori berita.

Tabel 17. Segmen Program 7 Struktur Json User Permissions

```
{"data": {"usersPermissionsRoles": {"data": [{"attributes":
        {"users": {"data": []}}},{"attributes":
        {"users": {"data": [{"attributes": {
               "email": "","provider": "local","password": "***",
               "username": "yudhapand","confirmed": true or false,
               "role": "users-permissions"}}]}}}]}}}
```

The program segment above explains the design of the user collection which is written using the json structure. In this design, it displays data from users' permissions roles in which there are attribute users, then in the user attribute there are still other attributes in the form of email, provider, password, username, confirmed and role.

Tabel 18. Segmen Program 8 Struktur Json Contributor

```
{"data": {"kontributors": {"data": [{"attributes": {
            "Email": "","Username": "vyan","Password": "***",
            "Alamat": "","TempatLahir": "",
            "TanggalLahir": "","NomorTelepon": ,
            "NIK": "","gambarKTP": ""}}]}}}
```

In the above program segment describes the design of collection kontributor which is written using the json structure. In the design there are attribute fields such as email, username, password, alamat, tempatlahir, tanggallahir, nomortelepon, NIK and gambarKTP. Unlike the collection user, here the contributor collection functions to register accounts for users who want to register, while the collection user is only for logging in to accounts that have been created beforehand.

Tabel 19. Segmen Program 9 Struktur Json Service

```
{"data": {"layanans": {"data": [{"attributes": {
            "judulLayanan": "","UrlSlugLayanan": "",
            "gambarLayanan": {"data": [{"attributes": {
                    "url": "/"}}]}}}]}}}
```

In the above program segment describes the design of collection Layanan which is written using the json structure. In the design there is an attribute field in the form of judulLayanan, urlsluglayanan and gambarlayanan. Inside the service image there is another attribute in the form of a url whose function is to display uploaded images.

Tabel 20. Segmen Program 10 Struktur Json Footer

```
{"data": {"footers": {"data": [{"attributes": {
            "alamat": "","jam_kerja": "",
            "kontak": ""}}]}}}
```

The program segment above explains the design of the collection footer which is written using the json structure. In the design there is an attribute field in the form of alamat, jam_kerja dan kontak.

3. RESEARCH RESULTS AND DISCUSSION

The results of this research can be seen by running the playground, its function is to find out whether the program structure design can run or not. In this study using the json structure design as a reference, and query as the caller.

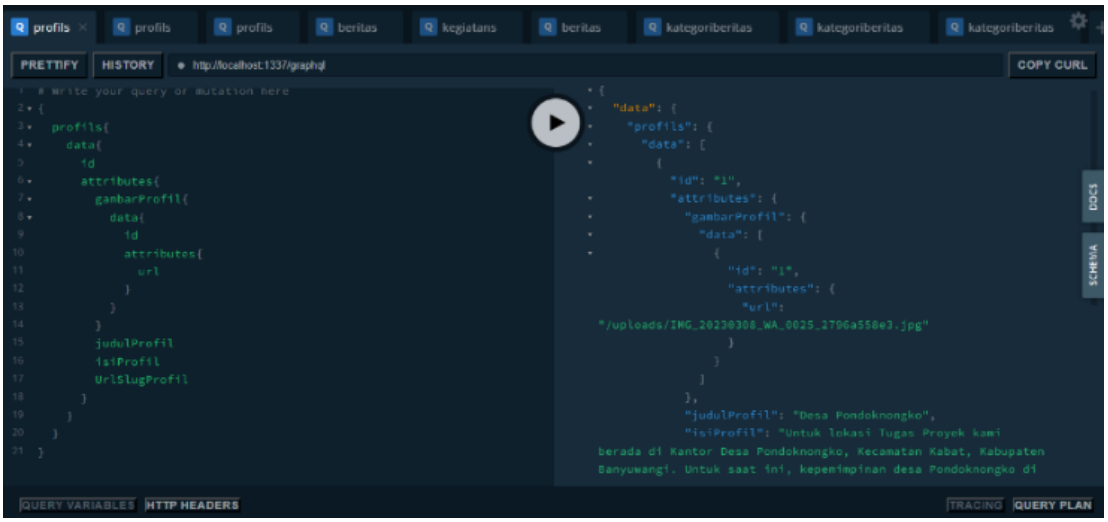

Figure 2. Playground Profil

The image above is the result of calling a query that brings up the json structure design from collection profil which on the left is a query to display the json structure as shown on the right. In the query there are attributes from collection profil as previously mentioned ie judulprofil, isiprofil, urlslugprofil and gambarprofil. In the gambarprofil there is also an attribute in the form of a url whose function is to call the uploaded image. It can be seen in the picture that the query and the json structure have been successfully executed and according to the design that was made in the previous section, if the query that is entered with the wrong json structure will display error data which is indeed an error in calling it.

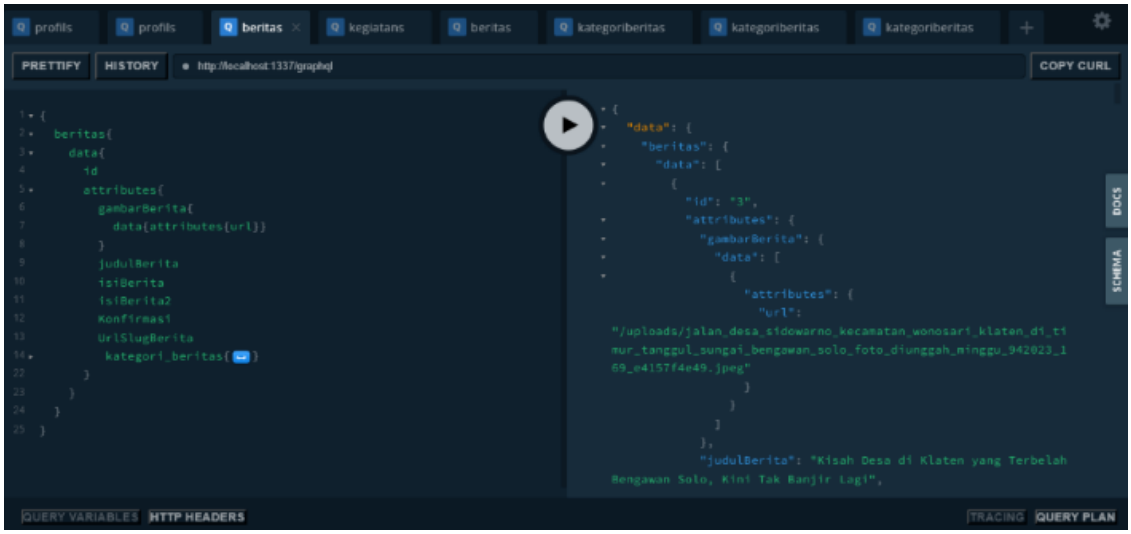

Figure 3. Playground Berita

The image above is a playground berita where the results of calling the query succeeded in bringing up the appropriate json structure. In the query there are several attributes such as: judulberita, isiberita, isiberita2, urlslugberita, gambarberita, konfirmasi and kategori_beritas. Inside the gambarberitas there are more attributes in the form of url and inside kategori_ beritas also has attributes in the form of title and the beritas itself. As with the previous explanation that if the query does not match then the json structure will display error data.

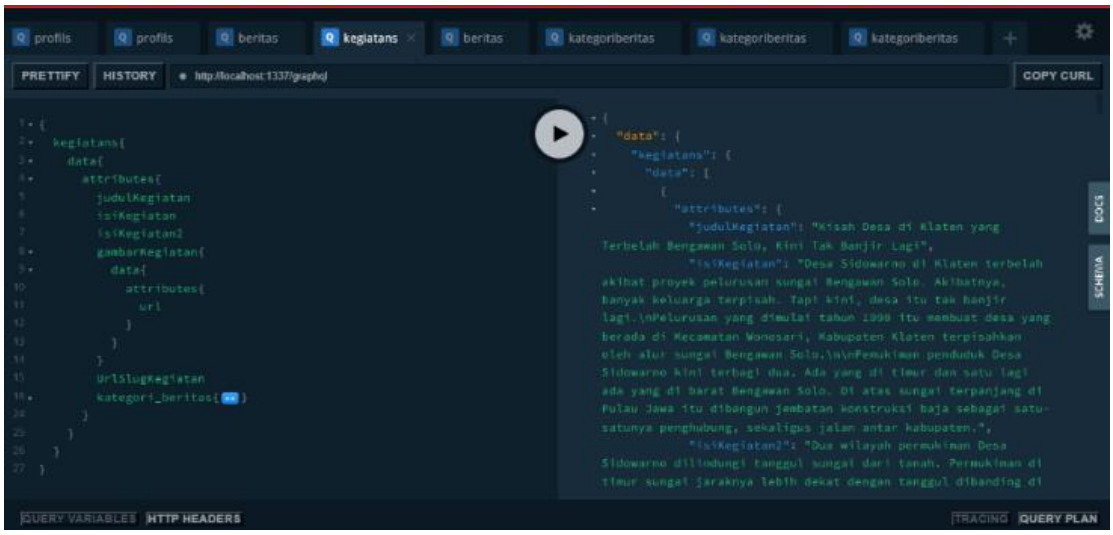

Figure 4. Playground Kegiatan

The image above is an playground Kegiatan where the results of calling the query succeeded in bringing up the appropriate json structure. In the query there are several attributes such as: judulkegiatan, isikegiatan, isikegiatan2, urlslugkegiatan, gambarkegiatan and kategori_beritas. In the gambarkegiatan there are more attributes in the form of url and inside kategori_ beritas also has attributes in the form of title and the beritas itself. As with the previous explanation that if the query does not match then the json structure will display error data.

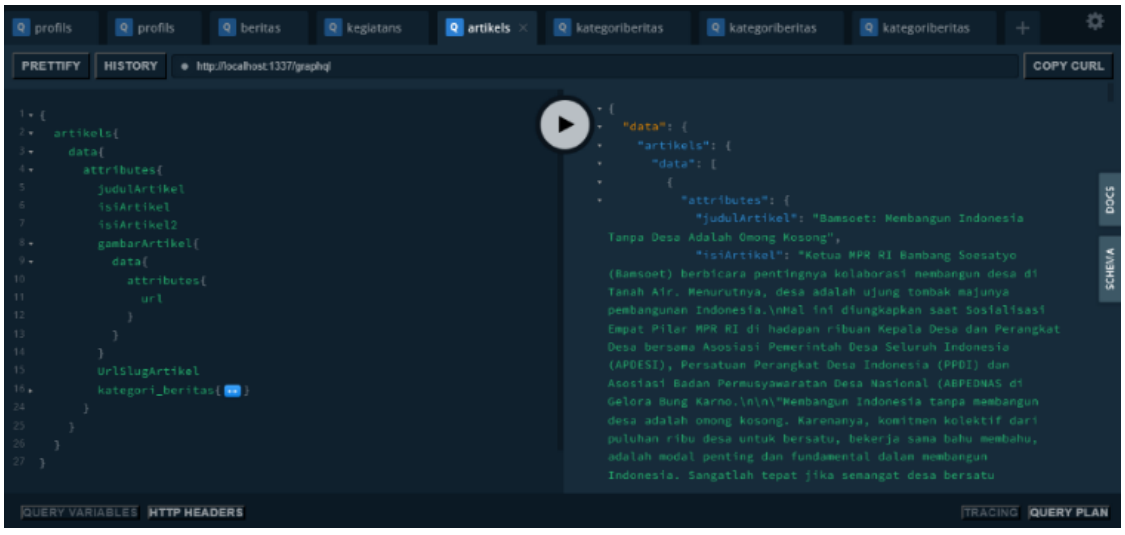

Figure 5. Playground Artikel

The image above is an playground Artikel where the results of calling the query succeeded in bringing up the appropriate json structure. In the query there are several attributes such as: judulartikel, isiartikel, isiartikel2, urlslugartikel, gambarartikel and kategori_beritas. Inside the gambarartikel there are more attributes in the form of url and inside kategori_ beritas also has attributes in the form of title and the beritas itself. As with the previous explanation that if the query does not match then the json structure will display error data.

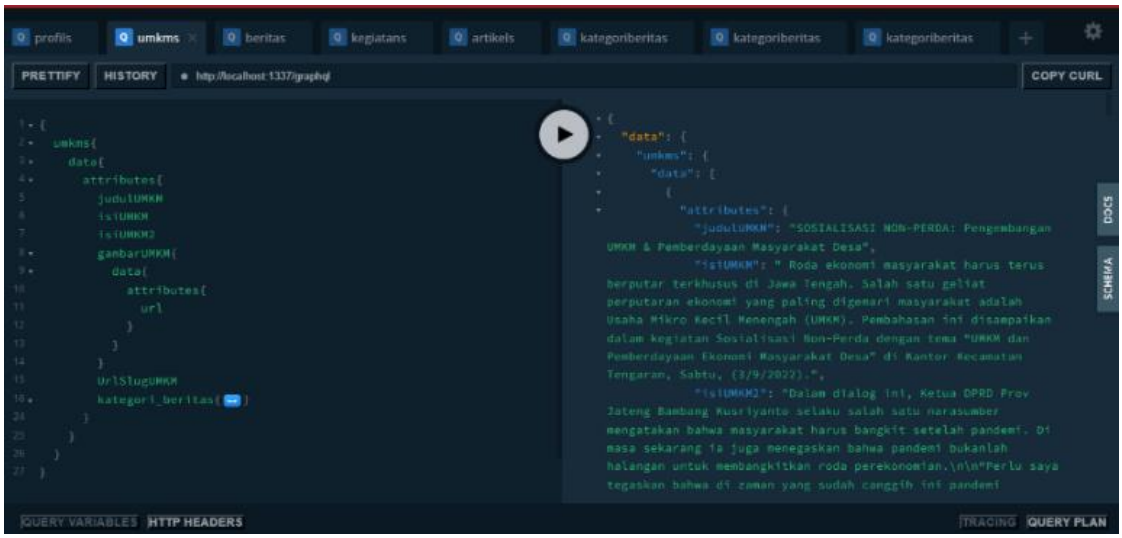

Figure 6. Playground UMKM

The image above is an playground UMKM where the results of calling the query succeeded in bringing up the appropriate json structure. In the query there are several attributes such as: judulumkm, isiumkm, isiumkm2, urlslugumkm, gambarumkm and kategori_beritas. Inside the gambarumkm there are more attributes in the form of url and inside kategori_ beritas also has attributes in the form of title and the beritas itself. As with the previous explanation that if the query does not match then the json structure will display error data.

Tabel 21. Results from Research

Query	Compatibility with Needs
Profil	Appropriate
Berita	Appropriate
Kategori Berita	Appropriate
Kegiatan	Appropriate
Artikel	Appropriate
UMKM	Appropriate
User	Appropriate
Kontributor	Appropriate
Layanan	Appropriate
Footer	Appropriate

4. CONCLUSION

1. The conclusion from the results of this study is that it is successful in utilizing headless cms to build back-end web applications
2. This research was also successful in compiling a graphql query structure that fits the needs analysis

5. REFERENCES

[1] M. T. A. D. H. R. Ramdhani Samboga, Pengembangan Website Desa Sebagai Informasi Pengenalan Potensi Desa Patokpicis Kecamatan Wajak Kabupaten Malang, Jurnal Graha Pengabdian, vol. 3, no. 4, pp. 345-351, 2021.

[2] R. I. F. F. K. P. S. M. I. M. Budi Usmanto, Implementasi Web Mobile Sebagai Media Informasi Pemberdayaan Masyarakat Di Desa Pirngadi, vol. 1, no. 1, pp. 32-40, 2018.
[3] S. Muthoharoh dan Nugraheni A. P., Evaluasi Penerapan Simda Keuangan Pada Dinas Kepemudaan Olahraga Dan Pariwisata Kota Magelang., vol. 01 No. 02, 2020.
[4] S. A. Siswanto, Aplikasi Pendaftaran Siswa Online di SMK PGRI 1 Giri Dengan Menggunakan PHP dan MySQL., 2019.
[5] S. V. Z. G. P. W. Desy Apriani, Application for Reporting Damage to Roads & Bridges at UPT DBMSDA Tangerang Regency, Universitas Raharja, CCIT Journal, vol. 16, no. 1, 2023.
[6] T. S. Suwarno, Perancangan Dan Implementasi Company Profile Berbasis Website Sebagai Media, UIB (Universitas Internasional Batam), vol. 1 No. 1, pp. 153-162, 2021.
[7] H. N. P. J. A. James Riady, Aplikasi E-Learning Berbasis Progressive Web App Pada Apologetika Indonesia,Jurnal Infra Petra, pp. 1-5, 2019.
[8] M. A. H. A. A. I. M. Y. Nur Ayuni Nor Sobri, Comparison between Headless CMS and Backend-as-a-Service, MSEA (Mathematical Statistican and Engineering Applications), vol. 71, no. 3s2, Special Issue on Mathematics Theory and its Contribution in Robotics and Computer Engineering, 2022.
[9] R. T. D. Rahayu Pangestika, Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta, UII (Universitas Islam Indonesia), vol. 01 No. 02, 2019.
[10] Kusuma Negara, R. I. T., Iriani, T., & Lenggogeni, Pengembangan Multimedia Interaktif Pada Mata Kuliah Manajemen Kontruksi di S1 Pendidikan Teknik Bangunan Universitas Negeri Jakarta, vol. 8(2), pp. 97-104, 2019.