

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Graphic Designer adalah jabatan yang bekerja secara langsung dengan graphic design teruntuk promosi dan juga konten, yang akan disajikan oleh Skystar melalui media platform yang dimiliki oleh organisasi tersebut

Dalam proses kerja magang yang dilakukan, Totally Bizarre Game Studio menugaskan tim *Web Developer* untuk menjadi *Full Stack Developer*. Hal ini berarti bahwa bagian *Frontend* dan *Backend* dari *website* akan dikerjakan secara terpisah. Proses Kerja magang memiliki tujuan, yaitu untuk melakukan perancangan dan pembangunan terhadap *website* Totally Bizarre. *Website* ini akan digunakan oleh perusahaan dalam proses pengembangan bisnis sekaligus sebagai media informasi tentang perusahaan. Proses kerja magang ini akan memiliki seorang supervisi, yaitu Bapak Renaldo Sutjiady selaku *Program Coordinator* Skystar Ventures.

Koordinasi pekerjaan antar divisi dilakukan secara *remote* dengan menggunakan *platform* Discord. Melalui *platform* tersebut, komunikasi terus dilakukan agar tidak terjadi kesalahpahaman dalam proses pengembangan modul. Sebagai bagian dari divisi *Web Developer*, salah satu tanggung jawab yang harus dihadapi adalah melakukan *request* terhadap isi konten atau asset pada *website* seperti materi dan gambar kepada divisi *Art* dan *Story*.

Untuk menyampaikan laporan mengenai perkembangan mingguan setiap divisi kepada supervisi Skystar Ventures, platform Google Meet digunakan dan kegiatan tersebut berlangsung setiap hari Jum'at pada pukul 11:30. Setelah semua

divisi menyampaikan perkembangan mingguan, supervisi akan memberikan *feedback* terhadap setiap poin-poin yang telah disampaikan sebelumnya.

3.2 Tugas dan Uraian Kerja Magang

3.2.1 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, terdapat beberapa tugas dan tanggung jawab yang harus diselesaikan, yaitu sebagai berikut:

1. Melakukan perancangan *website* berdasarkan *mockup* yang telah disesuaikan dengan keinginan *client*, dalam konteks ini adalah Totally Bizarre Game Studio.
2. Melakukan pembangunan *website* dengan menggunakan *framework* ReactJS sebagai *frontend* dan *framework* Laravel berbasis MySQL sebagai *backend*. Pemilihan *framework* dan basis data merupakan rekomendasi dari CTO secara langsung.
3. Melakukan *styling* terhadap setiap halaman pada *website* agar tampilan dapat terlihat lebih menarik.
4. Melakukan *testing* terhadap setiap fungsionalitas yang telah dibuat berdasarkan modul yang telah ditentukan untuk memastikan fungsi tersebut dapat berjalan lancar.

3.2.2 Uraian Kerja Magang

Proses kerja magang dilakukan selama 9 (Sembilan) minggu pada Program Inkubasi Bisnis Totally Bizarre Game Studio dengan rincian pekerjaan sebagai berikut pada Tabel 3.1:

Tabel 3.1 Kegiatan Mingguan Kerja Magang

Minggu ke-	Deskripsi Pekerjaan
1	Perancangan <i>Mockup Website</i> Pengenalan dan Instalasi <i>Framework</i>
2	Pembelajaran Laravel API, Implementasi <i>CRUD</i> , dan <i>Debugging</i>
3	Implementasi dan <i>Debugging</i> fungsi Messenger API, Instagram <i>Embed</i> , dan Pembuatan Komponen Artikel
4	Pembuatan Halaman <i>CRUD</i> Artikel dan <i>Debugging</i>
5	Riset, Pembelajaran, dan Implementasi <i>Rich Text Editor</i> dan <i>Debugging</i> Instagram <i>Embed</i>
6	<i>Debugging</i> Draft.js (Show Text dan Add & Show Image) dan penerapan fungsi Imgur API
7	<i>Debugging</i> Draft.js (Add & Show Youtube <i>Embed Link</i>), Pembuatan Komponen Manajemen Konten, menghubungkan dengan halaman Login, dan <i>Styling</i>
8	<i>Styling</i> halaman Artikel dan Home
9	Riset <i>provider</i> untuk deploy <i>Website</i> dan <i>Styling</i> halaman About

Minggu pertama kerja magang diawali dengan melakukan perancangan *mockup website* dengan menggunakan *tool* balsamiq dan menyesuaikannya dengan *request* yang telah diberikan. Setelah perancangan *mockup* selesai, tugas selanjutnya yang dilakukan adalah melakukan pengenalan dan instalasi dua *framework*, yaitu ReactJS dan Laravel untuk membiasakan diri dengan struktur dan cara kerja kedua *framework* tersebut. Untuk minggu pertama lebih fokus pada bagaimana cara menggunakan ReactJS, seperti membuat *Class* dan *Component*, melakukan *Routing* dan *Page Navigation*, serta mengimplementasikan fitur *State*, *Props*, *Fetch*, dan *Axios* dan proses pembelajaran ini hingga minggu pertama

berakhir.

Minggu kedua digunakan untuk mempelajari dasar-dasar Laravel dengan lebih fokus. Dimulai dengan mempelajari struktur, cara pembuatan, dan cara kerja daripada *Model, View, dan Controller*. Dilanjutkan dengan mempelajari bagaimana menggunakan dan mengimplementasikan Laravel sebagai API untuk mendukung dan menyediakan *resource* MySQL pada *frontend* dengan data *dummy*. Setelah API siap digunakan, kemudian dilakukan pembelajaran dan implementasi tentang bagaimana melakukan *CRUD* pada *backend* melalui fungsi yang telah dibuat pada *frontend*. Minggu kedua diakhiri dengan melakukan pertemuan mingguan dengan Skystar Ventures dan *test demo* dengan *internal* Totally Bizarre.

Minggu ketiga digunakan untuk melanjutkan percobaan dalam mengimplementasikan fungsi *CRUD* dengan menggunakan data *dummy*. Kemudian, oleh karena akan dibuatnya Facebook *Fan Page*, maka dilakukan percobaan untuk mengimplementasikan fungsi Facebook Messenger API pada *website*, namun untuk menggunakan fungsi ini, dibutuhkan *website* dengan *link* yang telah terdaftar dengan *domain* resmi dan harus terhubung dengan Facebook *Fan Page*, oleh karena itu percobaan dialihkan pada *website* lain berbasis PHP yang telah terdaftar pada *domain* resmi, dan fungsi tersebut telah berhasil diaplikasikan. Setelah itu, dilakukan pembelajaran terhadap bagaimana cara melakukan Instagram *Embed*. Minggu ketiga diakhiri dengan melakukan pembuatan komponen *Article View*.

Minggu keempat digunakan untuk melakukan pengembangan fungsi modul artikel. Dimulai dengan membuat *model* dan *controller* artikel pada Laravel dan melakukan percobaan *fetch data*. Setelah percobaan *fetch* telah berhasil

diimplementasikan, maka dilakukan pembuatan fungsi *Add*, *Edit* dan *Delete* untuk artikel pada *frontend* dalam bentuk form dan backend pada bagian *controller*. Untuk membuat tampilan lebih menarik, maka ditambahkan fitur *pagination* pada halaman artikel dan penerapan *layout* berupa *card* pada tiap artikel yang ada serta melakukan *debugging* terhadap fitur-fitur tersebut. Hal ini dilakukan hingga akhir minggu selesai.

Minggu kelima digunakan untuk melakukan riset, pembelajaran, dan implementasi beberapa *Rich Text Editor* serta melakukan *debugging* terhadap fungsi *Instagram Embed*. Dari beberapa *Rich Text Editor* yang telah dicoba, *Draft.js* menjadi pilihan akhir untuk digunakan sebagai fitur pendukung modul artikel. Setelah mencoba melakukan implementasi dasar *Draft.js*, dilakukan proses *debugging* terhadap fungsi *Instagram Embed* karena sebelumnya belum berhasil diimplementasikan. Kegiatan *debugging* *Instagram* dilakukan hingga minggu kelima selesai.

Minggu keenam digunakan untuk melakukan *debugging* terhadap *Rich Text Editor*, yaitu *Draft.js*. Secara spesifik, hal ini bermaksud untuk memperbaiki fungsi untuk menampilkan teks yang dikirim API pada halaman *website*. Setelah perbaikan fungsi teks, ditemukan bahwa fungsi untuk melakukan *upload* gambar pada *Rich Text Editor* tidak dapat berjalan dengan benar, oleh karena itu dilakukan perbaikan untuk fungsi tersebut. Setelah melakukan pembahasan *internal* dengan tim *Web Developer*, akhirnya diputuskan bahwa fungsi *upload* gambar pada *Draft.js* akan menggunakan fitur *Imgur API*, dimana gambar yang dimasukkan pada *Rich Text Editor* akan dikembalikan dalam bentuk *Imgur URL*. Kegiatan *debugging* dan implementasi *Imgur API* dilakukan hingga akhir minggu keenam selesai.

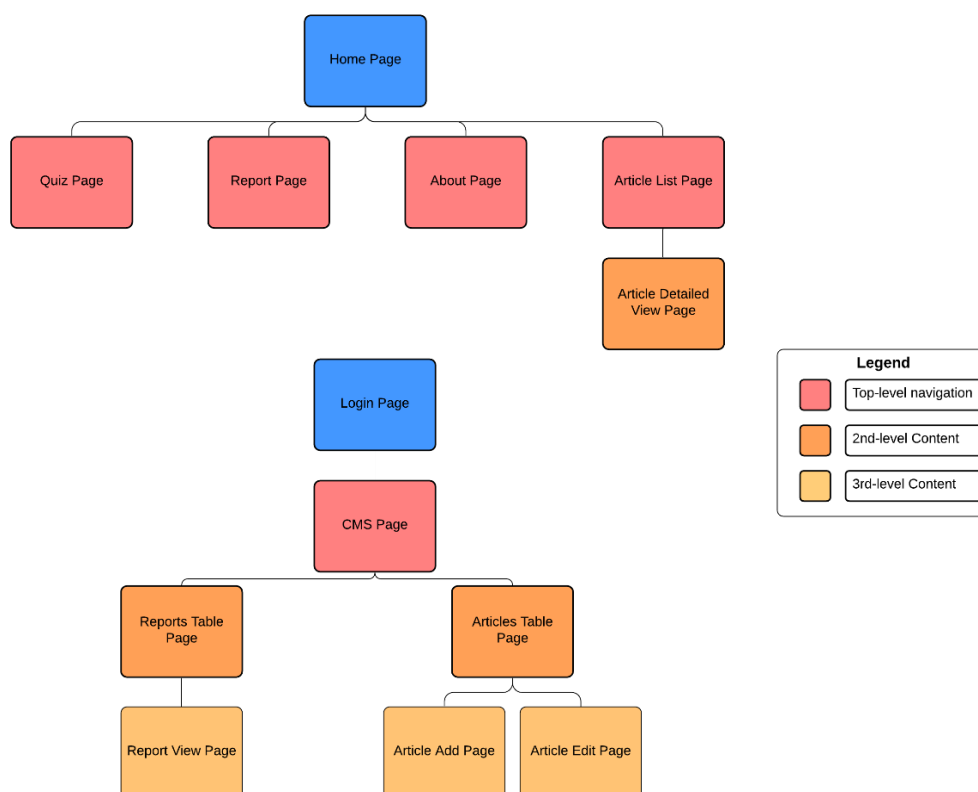
Minggu ketujuh digunakan untuk melakukan *debugging* terhadap *Rich Text Editor* kembali, dimana fungsi *Add* dan *Show* video dari Youtube tidak berjalan dengan baik. Setelah memperbaiki kesalahan tersebut, dilakukan pembuatan komponen manajemen konten dan menghubungkannya dengan halaman Login serta melakukan sedikit *styling*. Halaman manajemen konten tersebut berisi tabel untuk mengontrol data artikel dan *report* dan untuk mengakses halaman ini, *user* harus melakukan otentikasi terlebih dahulu melalui halaman Login.

Minggu kedelapan digunakan untuk melakukan *styling* terhadap halaman Article dan Home. Terjadi perubahan yang cukup besar terhadap desain halaman. Hal ini bermaksud agar tampilan dapat lebih menarik dari sebelumnya. Minggu kedelapan diakhiri dengan melakukan pertemuan mingguan dengan Skystar Ventures dan test demo dengan internal Totally Bizarre kembali.

Minggu kesembilan digunakan untuk melakukan riset terhadap penyedia layanan *deploy website* dan melakukan *styling* terhadap halaman About. Kegiatan ini dilakukan hingga masa magang berakhir.

3.2.3 Sitemap

Berikut merupakan *sitemap* yang digunakan dalam proses rancang bangun *website* Totally Bizarre:



Gambar 3.1 *Web Sitemap*

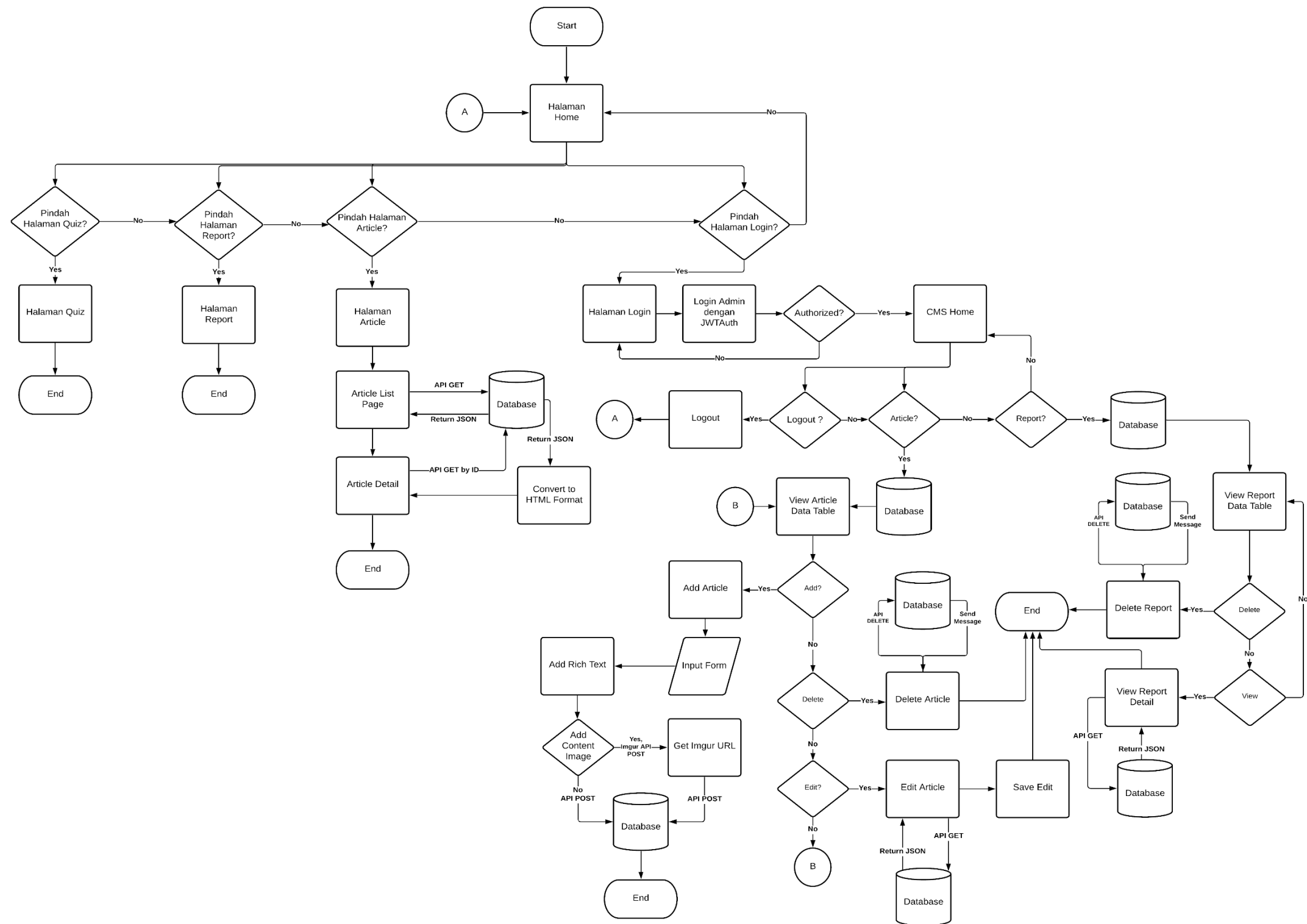
Pada Gambar 3.1 dijelaskan bagaimana alur *website* berjalan. Alur tersebut dimulai dari halaman Home, dan halaman Home terhubung dengan empat halaman lainnya yaitu halaman Quiz, halaman Report, halaman About, dan halaman Article List. Untuk halaman Article List, jika salah satu artikel dipilih, maka *website* akan melakukan navigasi kepada halaman Article Detailed View Page dimana halaman tersebut berisi konten detil dari artikel tersebut.

Selain halaman Home, terdapat halaman Login yang digunakan sebagai gerbang sebelum masuk pada halaman konten manajemen. Untuk masuk pada halaman login ini, *user* harus menuliskan *link* secara manual karena pada halaman Home tidak tersedia pilihan untuk melakukan navigasi kepada halaman Login. Setelah *user* berhasil masuk ke halaman konten manajemen, terdapat dua halaman yang tersedia, yaitu halaman untuk menampilkan tabel *articles* dan tabel *reports*.

Pada tabel *articles* terdapat tombol *add* dan *edit* yang akan mengarahkan *user* pada halaman yang berisi *form* untuk melakukan aksi menambah artikel atau mengubah konten dari artikel. Pada tabel *reports* terdapat tombol *view* yang akan mengarahkan *user* pada halaman yang berisi *form* untuk melihat *report* yang telah tersimpan pada *database*.

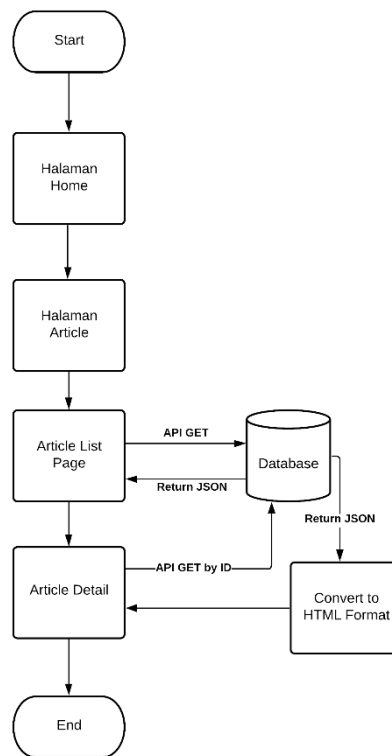
3.2.4 Flowchart

Berikut merupakan beberapa flowchart yang digunakan dalam proses rancang bangun *website* Totally Bizarre:



Gambar 3.2 Flowchart Utama Website

Gambar 3.2 menjelaskan fungsi utama dari *website* yang dikerjakan. Fungsi-fungsi tersebut meliputi fungsi Quiz, Report, Article List dan Article Detail, dan Konten Manajemen dimana semua fungsi tersebut kecuali halaman konten manajemen dapat diakses melalui *Navigation Bar* yang terdapat pada halaman Home. Pada *flowchart* tersebut, objek yang diwarnai biru muda merupakan modul utama yang dikerjakan, sedangkan objek yang diwarnai oranye merupakan modul yang dikerjakan oleh rekan satu tim.

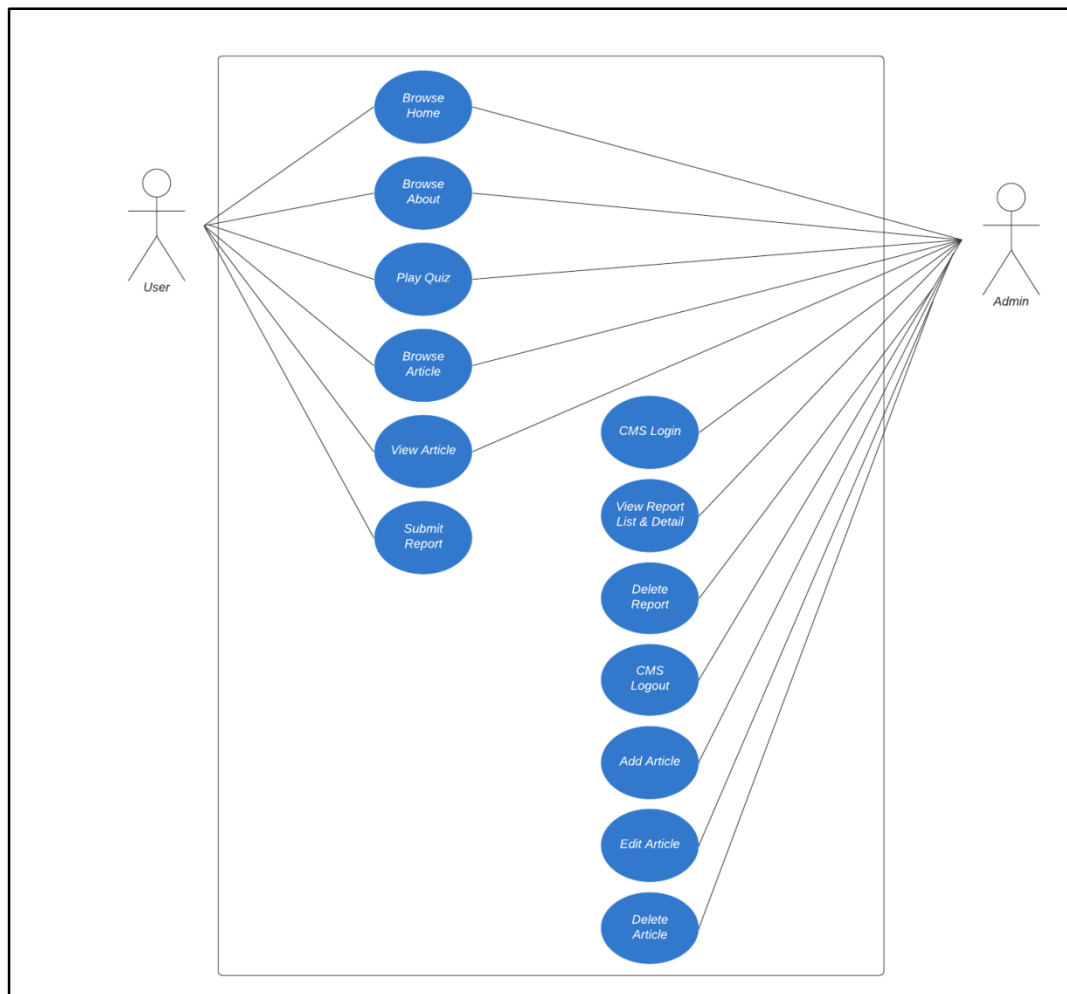


Gambar 3.3 Flowchart Modul Artikel

Gambar 3.3 menjelaskan garis besar dari alur kerja halaman artikel. Halaman Article List berisi mengenai artikel apa saja yang ada pada tabel artikel dan ketika salah satu artikel dipilih atau ditekan oleh *user*, maka halaman akan berpindah kepada halaman detil artikel, dimana halaman tersebut akan memuat konten berupa gambar dan tulisan dari *Rich Text Editor*.

report dimana tabel tersebut digunakan untuk melakukan kontrol terhadap konten didalamnya. Semua aksi kontrol terhubung dengan *backend* sehingga semua aksi yang dilakukan akan mempengaruhi data aslinya. Pada bagian terbawah *side bar*, terdapat tombol *logout* yang berfungsi untuk keluar dari halaman konten manajemen, dan akan mengarahkan *user* ke halaman Home.

3.2.5 Use Case Diagram

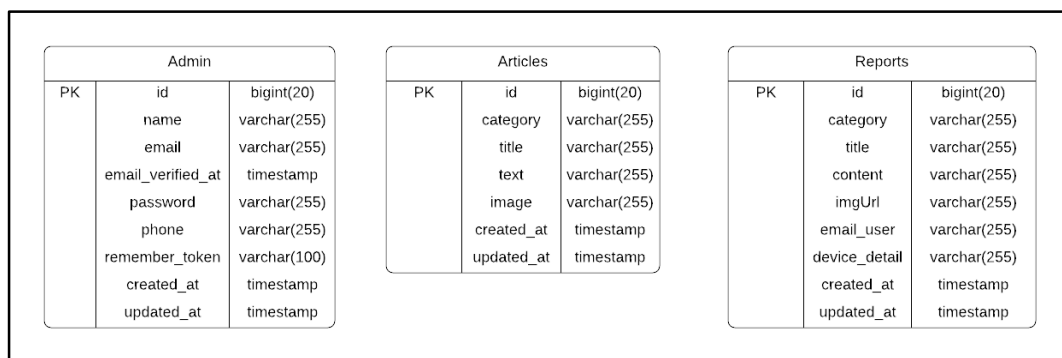


Gambar 3.5 Use Case Diagram

Gambar 3.5 merupakan gambar *use case diagram* dari *website* Totally Bizarre. Semua *user* dapat melakukan browsing pada halaman Home, About, dan Article. Semua *user* juga dapat memainkan Quiz pada halaman Quiz. *User* diluar

admin dapat menggunakan fitur Report dengan cara mengisi formulir laporan yang berisi saran, kritik, ataupun laporan tentang *bug* atau kerusakan yang ditemukan pada game perusahaan. *Admin* dapat melakukan login untuk masuk ke halaman manajemen konten. Pada halaman manajemen konten, *admin* dapat melihat daftar laporan yang masuk, melihat detail tiap laporan, dan menghapus laporan yang sudah ditanggapi. *Admin* juga dapat menambahkan artikel yang dapat berupa berita untuk nantinya ditampilkan pada halaman Artikel oleh *user* lain. *Admin* juga dapat melakukan edit dan menghapus pada artikel-artikel yang sudah di publikasi.

3.2.6 Entity Relational Database



Gambar 3.6 Entity Relational Database

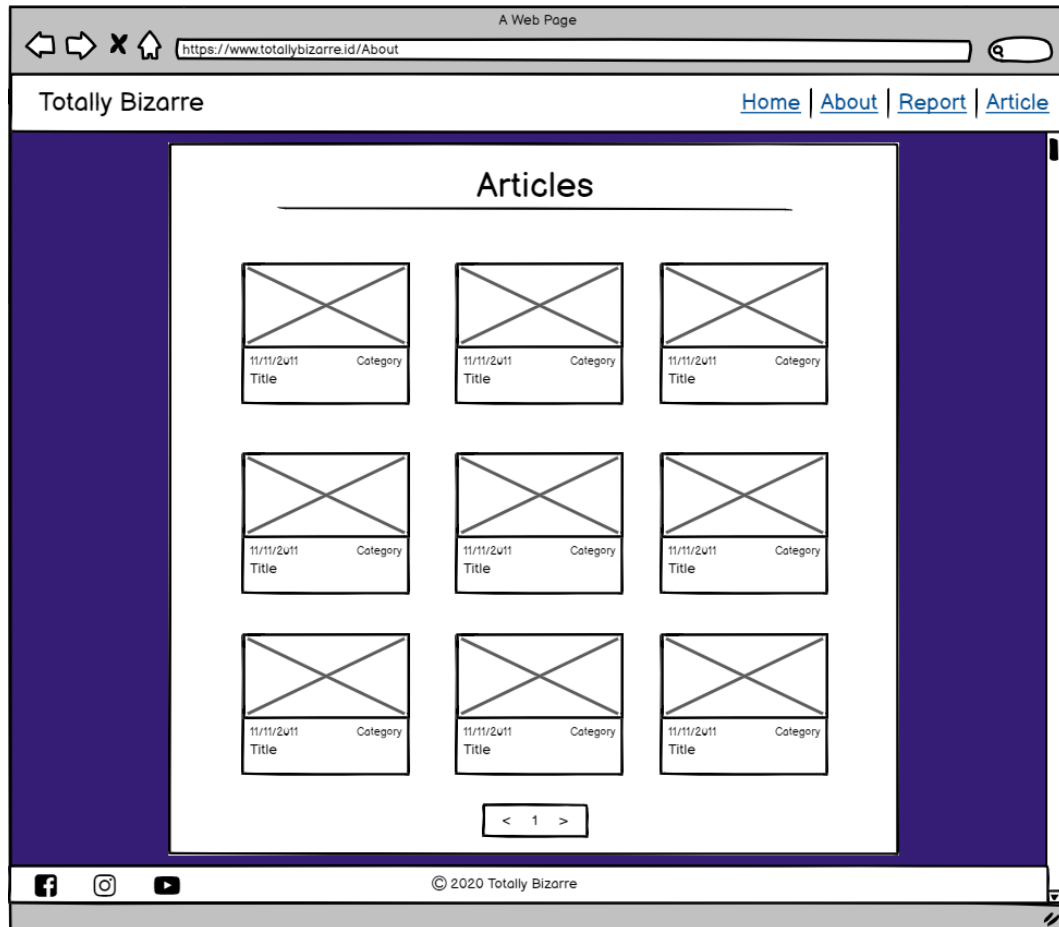
Gambar 3.6 merupakan gambar dari *Entity Relational Database* (ERD) dari *website* Totally Bizarre. Oleh karena *website* ini merupakan *landing page*, berarti *user* tidak perlu melakukan login sehingga tidak ada data yang sensitif yang akan disimpan. *Database* yang digunakan adalah MySQL. Jumlah tabel yang digunakan untuk membangun *website* ada 3 tabel yaitu tabel *Admin* untuk data setiap administrator, tabel *Articles* untuk data setiap artikel yang ada, dan tabel *Reports* untuk data setiap laporan yang ada.

3.2.7 Perancangan Antarmuka Sistem

Rancangan antarmuka dari *website* dapat dilihat pada gambar berikut yang memperlihatkan rancangan antarmuka yang telah disesuaikan dengan permintaan dari *client*. Berikut adalah daftar gambar antarmuka yang digunakan sebagai pendukung pengerjaan modul:

A. Halaman Article View

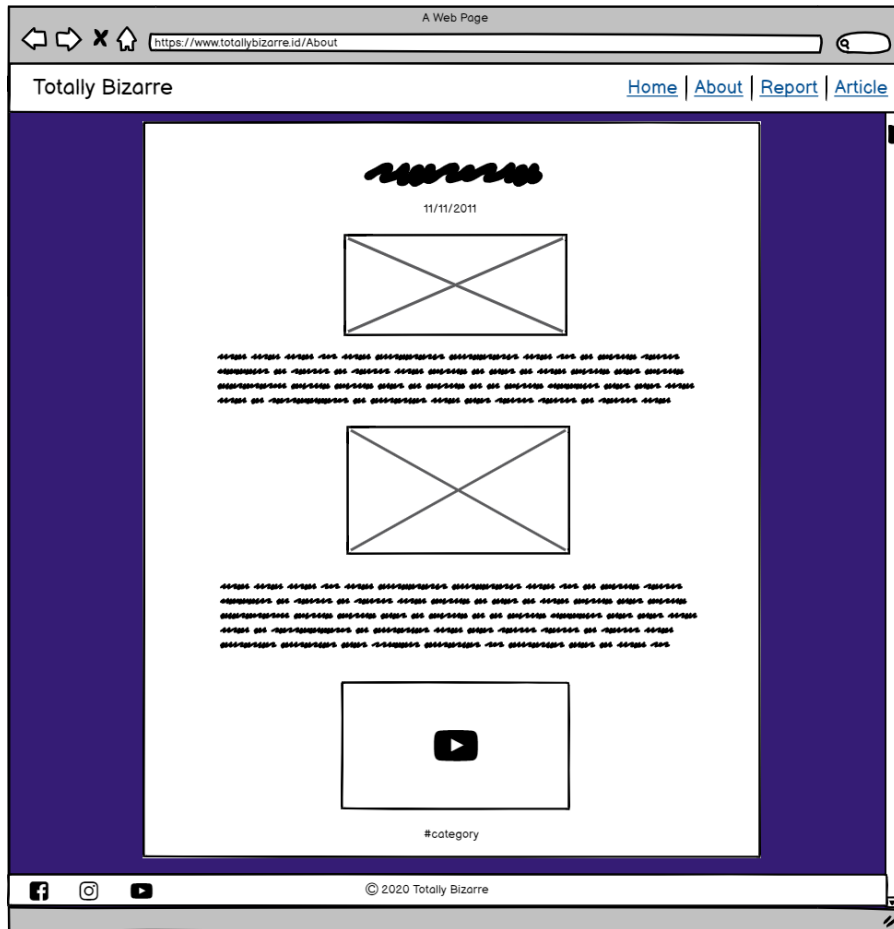
Halaman ini berisi *list* dari artikel-artikel yang telah dibuat dan disimpan pada tabel artikel. Halaman ini memiliki desain antarmuka *Card Grid* yang akan terbagi setiap tiga *item* per baris dan memiliki jumlah maksimal per halaman sebanyak tiga kali tiga artikel, atau 9 artikel dalam satu halaman. Halaman ini juga akan memuat fungsi *pagination* yang berfungsi untuk mengganti indeks artikel sehingga artikel yang dimuat pada halaman akan berubah. *Card* tiap artikel ini akan memuat informasi seperti gambar, tanggal, kategori, dan judul. Tampilan antarmuka halaman Article View dapat dilihat pada Gambar 3.7 berikut:



Gambar 3.7 Antarmuka Halaman Article View

B. Halaman Article Detail View

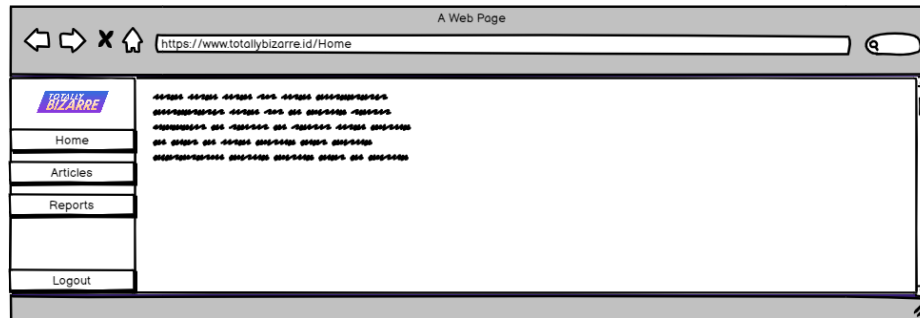
Halaman ini berisi tampilan detail artikel yang sebelumnya telah dipilih *user*. Halaman ini akan berisi judul, gambar, dan deskripsi konten. Secara detail, deskripsi konten tersebut dapat memuat teks, gambar, dan embed video. Hal itu bermaksud untuk membuat halaman artikel menjadi lebih menarik. Tampilan antarmuka halaman Article Detail dapat dilihat pada Gambar 3.8 berikut:



Gambar 3.8 Antarmuka Halaman Article Detail

C. Halaman Home Konten Manajemen

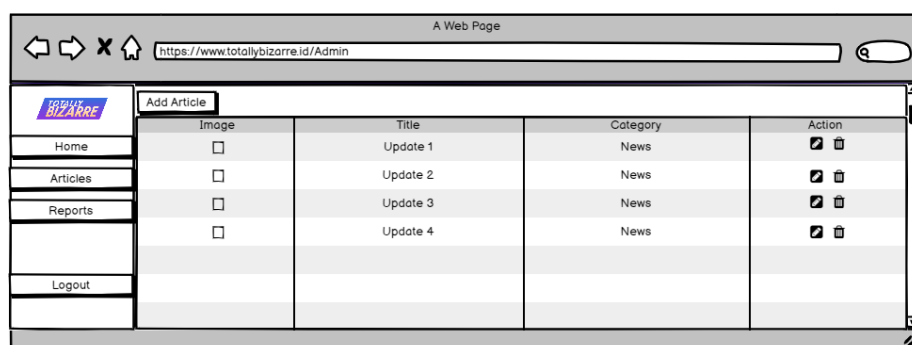
Halaman Home pada konten manajemen ini akan berisi notes yang ditujukan kepada *admin* tentang tugas apa saja yang harus dilakukan. Halaman Home ini memiliki *side bar* yang berisi tombol untuk mengarahkan *admin* ke halaman lain. Tombol *logout* digunakan ketika *admin* telah selesai mengubah, mengubah, atau menghapus data artikel atau *report*. Tampilan antarmuka halaman Home Manajemen Konten dapat dilihat pada Gambar 3.9 berikut:



Gambar 3.9 Halaman Home Manajemen Konten

D. Halaman Manajemen Konten Bagian Articles Table

Halaman ini akan menampilkan *list* data artikel pada *database* dalam bentuk tabel, dimana tabel tersebut terdiri atas kolom gambar, judul, kategori, dan aksi. Pada kolom aksi, terdapat dua tombol yang masing-masing akan menjalankan fungsi yang berbeda, yaitu Edit Article dan Delete Article. Pada bagian atas tabel, terdapat tombol Add Article yang berfungsi untuk menjalankan fungsi untuk menambah data artikel ke *database*. Baik Add Article maupun Edit Article, keduanya sama-sama mengarahkan *admin* ke halaman lain. Tampilan antarmuka halaman Manajemen Konten Bagian Articles Table dapat dilihat pada Gambar 3.10 berikut:

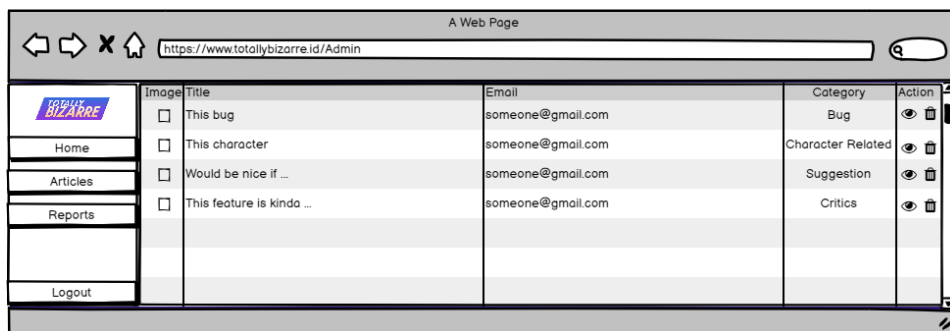


Gambar 3.10 Halaman Manajemen Konten Bagian Articles Table

E. Halaman Manajemen Konten Bagian Reports Table

Halaman ini akan menampilkan *list* data *reports* pada *database* dalam

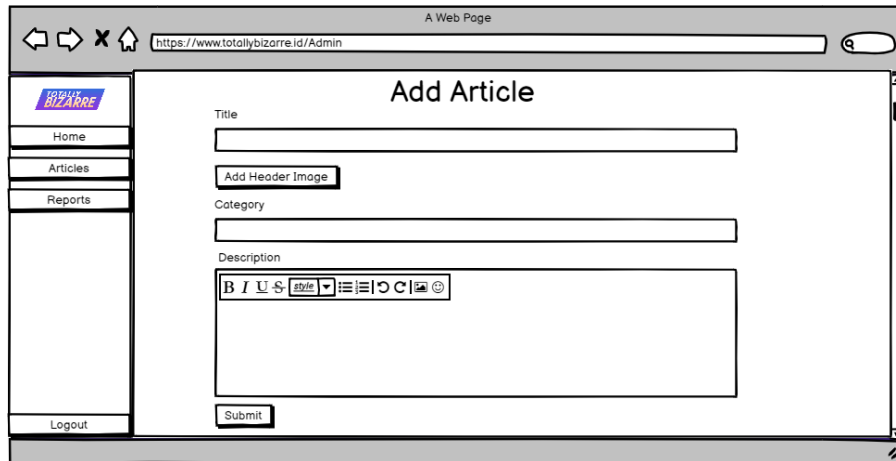
bentuk tabel, dimana tabel tersebut terdiri atas kolom gambar, judul, email, dan kategori *report*, dan aksi. Pada kolom aksi, terdapat dua tombol yang masing-masing akan menjalankan fungsi yang berbeda, yaitu View Report dan Delete Report. Ketika tombol *view* ditekan, maka *user* akan diarahkan ke halaman lain, sedangkan ketika tombol *delete* ditekan, maka data *report* akan dihapus dari *database*. Tampilan antarmuka halaman Manajemen Konten Bagian Reports Table dapat dilihat pada Gambar 3.11 berikut:



Gambar 3.11 Halaman Manajemen Konten Bagian Reports Table

F. Halaman Article View Halaman Manajemen Konten Bagian Add Article

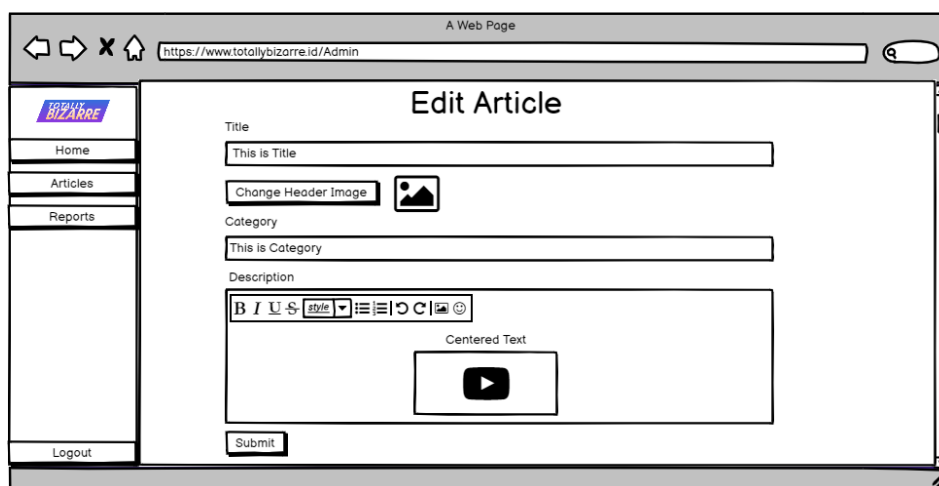
Halaman ini akan menampilkan form yang akan digunakan untuk menambah data artikel baru pada *database*. Form ini akan menampilkan kolom *input* judul, kategori dengan format teks, tombol untuk mengunggah gambar *header*, dan kolom deskripsi dengan format *rich text*. Tampilan antarmuka halaman Manajemen Konten Bagian Add Article dapat dilihat pada Gambar 3.12 berikut:



Gambar 3.12 Halaman Manajemen Konten Bagian Add Article

G. Halaman Manajemen Konten Bagian Edit Article

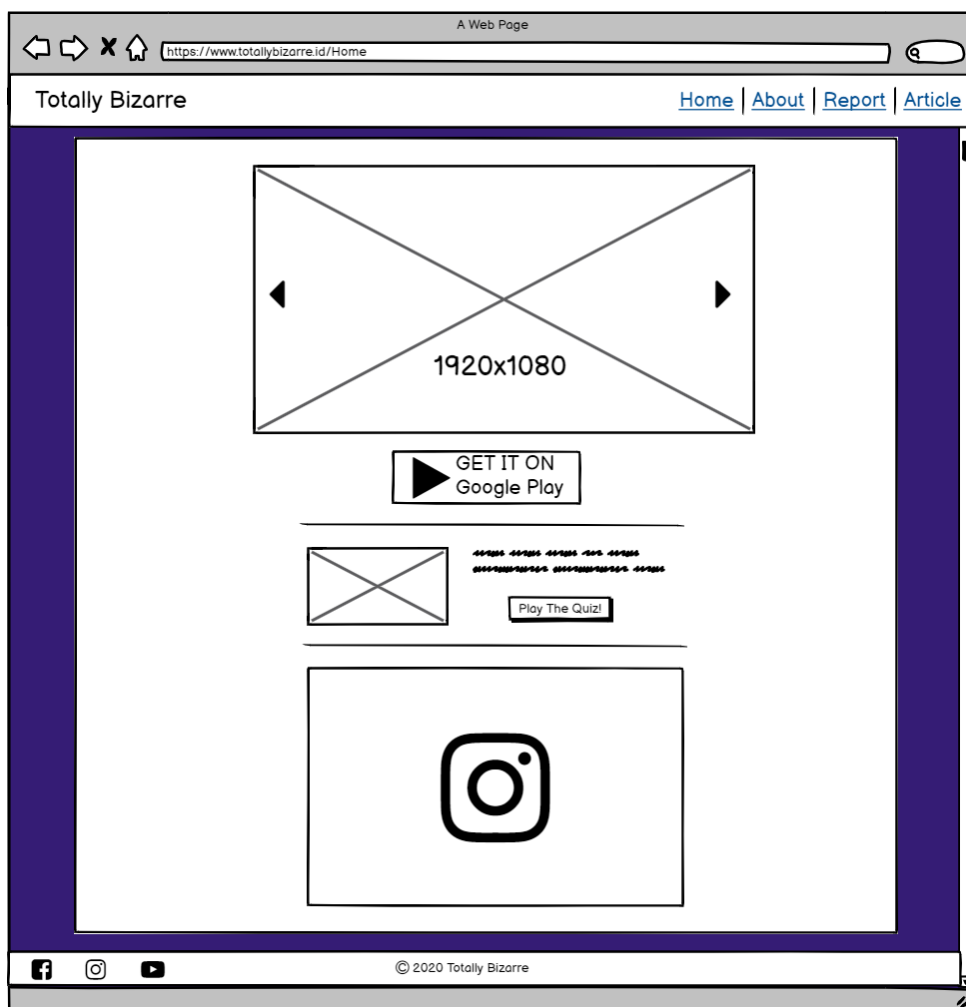
Halaman ini akan menampilkan form yang akan digunakan untuk mengubah data artikel pada *database* berdasarkan *item* pada tabel *list* artikel. Form ini akan menampilkan kolom *input* judul, kategori dan deskripsi yang telah terisi dengan data dari *database*. Gambar yang sebelumnya digunakan juga ditampilkan agar mempermudah *admin* dalam mengelola gambar konten. Tampilan antarmuka halaman Manajemen Konten Bagian Edit Article dapat dilihat pada Gambar 3.13 berikut:



Gambar 3.13 Halaman Manajemen Konten Bagian Edit Article

H. Halaman Home

Halaman ini merupakan tampilan pertama yang akan ditemui ketika *user* membuka *website* Totally Bizarre. Halaman ini akan memuat konten seperti *carousel*, yang akan menampilkan beberapa gambar dalam bentuk *slide*, tombol Google Play Store yang akan mengarahkan *user* kepada halaman untuk mengunduh game pada Google Play Store, tombol yang akan mengarahkan user pada halaman *minigame* berupa kuis, dan *embed post* milik akun Instagram Totally Bizarre. Tampilan antarmuka halaman Home dapat dilihat pada Gambar 3.14 berikut:

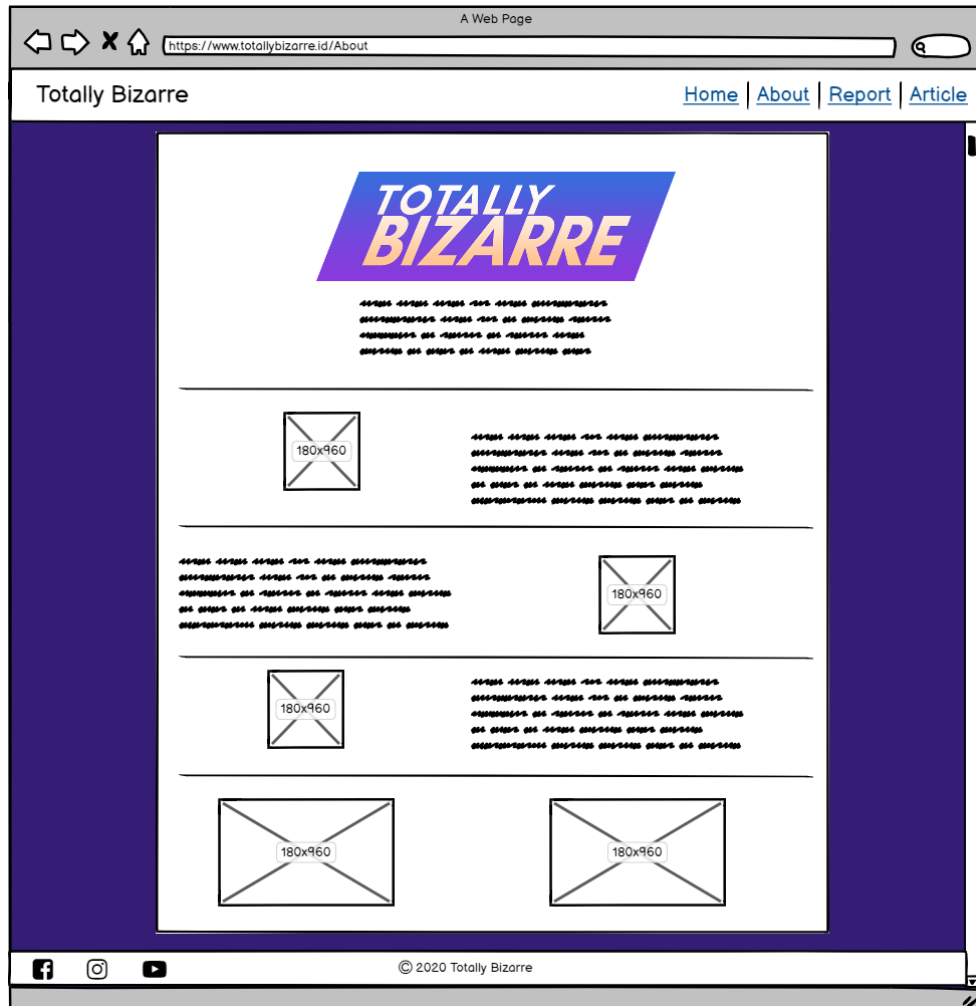


Gambar 3.14 Halaman Home

I. Halaman About

Halaman ini akan menampilkan profil dari perusahaan beserta pendiri-

pendiri dari Totally Bizarre yang meliputi CEO, CFO, dan CTO. Logo dari Skystar Ventures juga akan ditampilkan pada halaman ini. Tampilan antarmuka halaman About dapat dilihat pada Gambar 3.15 berikut:

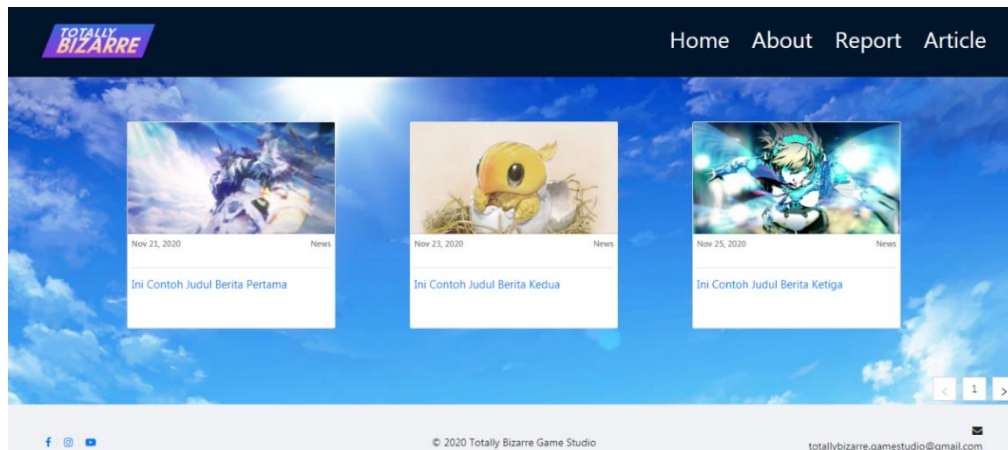


Gambar 3.15 Halaman About

3.2.8 Hasil Implementasi Sistem

Dari tampilan *mockup* yang telah dirancang, kemudian direalisasikanlah tampilan tersebut kedalam bentuk halaman *website*. Berikut daftar gambar halaman tersebut:

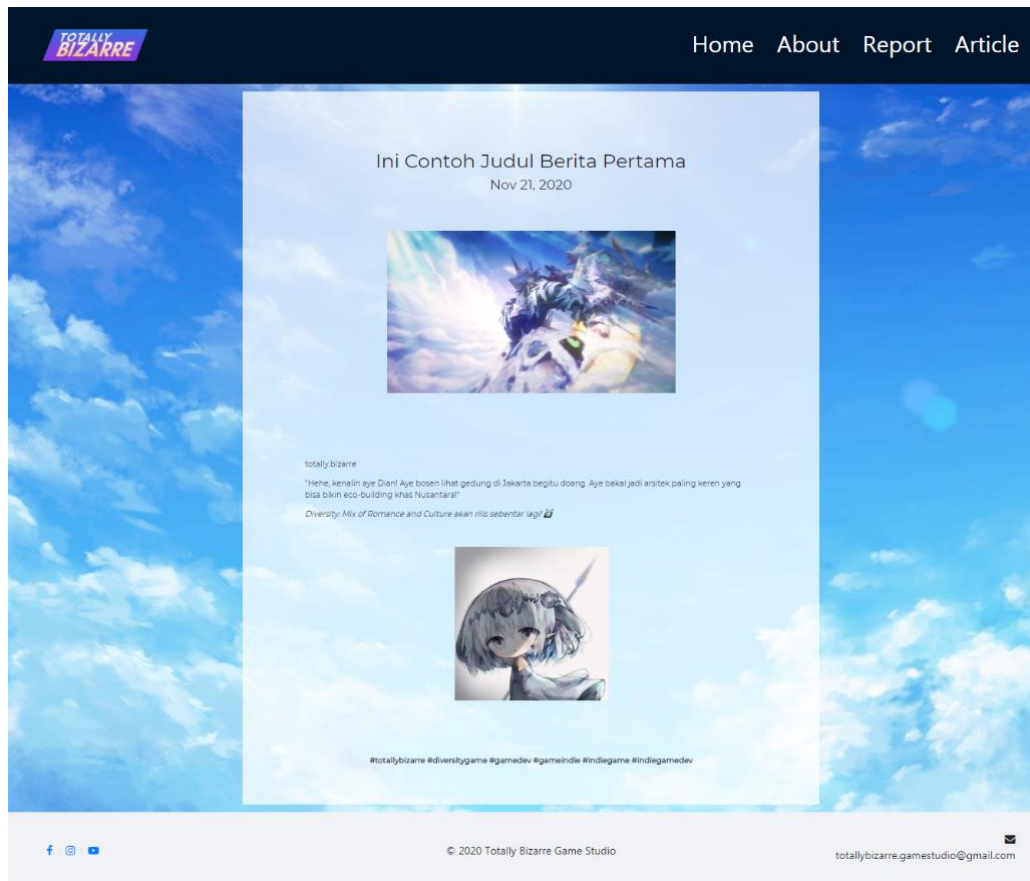
A. Halaman Article List



Gambar 3.16 Halaman Article List

Gambar 3.16 merupakan tampilan hasil realisasi dari halaman Article List. Sesuai dengan *mockup*, *grid* yang digunakan adalah kolom berukuran tiga kali tiga. Halaman ini juga mengimplementasikan fungsi *pagination* yang berfungsi untuk mengganti halaman artikel ke indeks selanjutnya. *Pagination* ini baru dapat berfungsi apabila terdapat lebih dari 9 indeks data pada *database*.

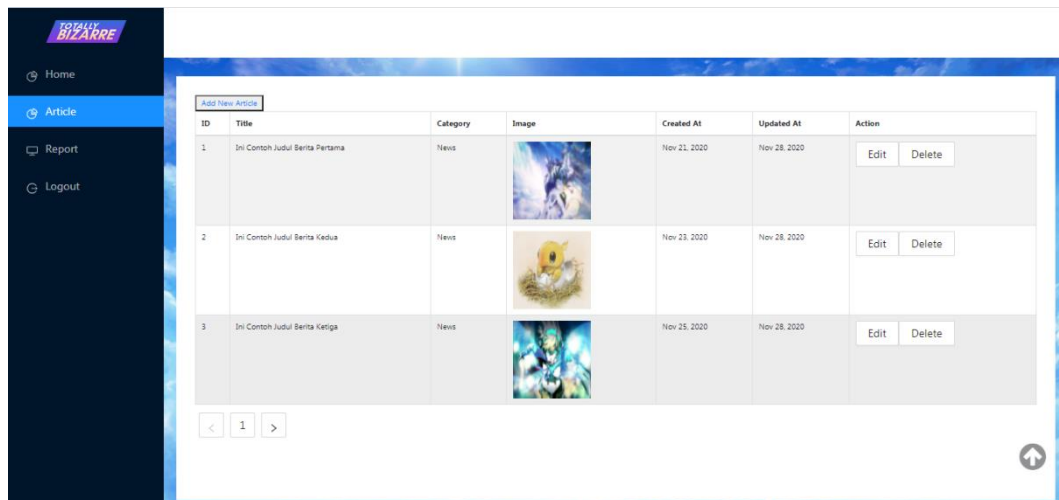
B. Halaman Article Detail






Gambar 3.17 Halaman Article Detail

Gambar 3.17 merupakan tampilan hasil realisasi dari halaman Article Detail. Halaman ini akan memuat konten dari *database* artikel seperti judul, tanggal dibuat, gambar konten, dan isi konten. Gambar dibawah judul disimpan secara lokal pada *backend*. Untuk bagian *body* konten, digunakan fitur Draft.js untuk melakukan konversi data yang disimpan pada *database* dimana data tersebut memiliki format *Rich Text* sehingga dibutuhkan *handler* khusus untuk membuatnya dapat terlihat secara *HTML*. Gambar yang dimuat pada *body* konten disimpan menggunakan fungsi Imgur API, dimana fungsi tersebut akan mengunggah gambar yang dimasukan Draft.js melalui fitur *Add Image* pada Draft.js. Imgur API ini akan mengembalikan *output* berupa *link* Imgur dimana *link* tersebut dapat bersinergi dengan *tag HTML img* sehingga Draft.js dapat menampilkan gambar tersebut.

C. Halaman Konten Manajemen Article Table

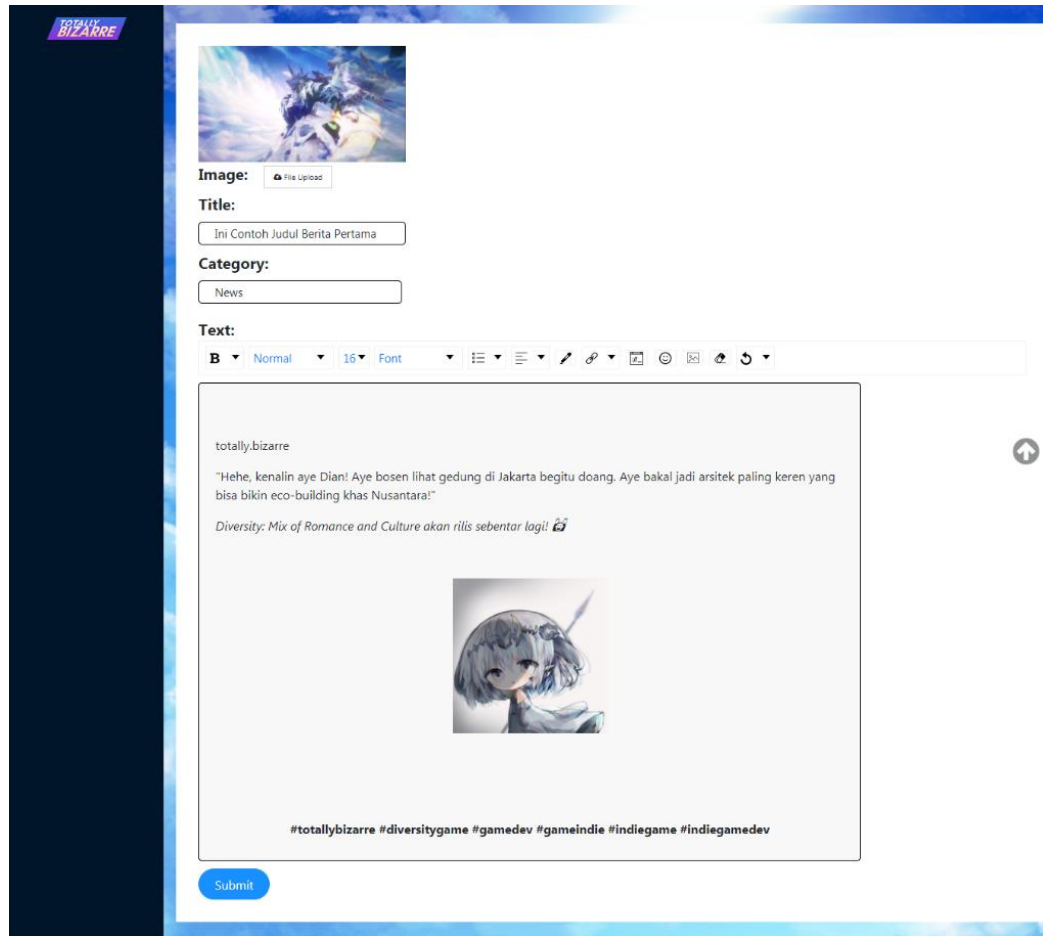


ID	Title	Category	Image	Created At	Updated At	Action
1	Ini Contoh Judul Berita Pertama	News		Nov 21, 2020	Nov 28, 2020	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Ini Contoh Judul Berita Kedua	News		Nov 23, 2020	Nov 28, 2020	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ini Contoh Judul Berita Ketiga	News		Nov 25, 2020	Nov 28, 2020	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Gambar 3.18 Halaman Konten Manajemen Article Table

Gambar 3.18 merupakan tampilan hasil realisasi dari halaman Konten Manajemen Article Table. Halaman ini menampilkan tabel yang berisi data *artikel* yang diambil dari *database*. Pada kolom Action, terdapat tombol Edit dan Delete, dimana ketika tombol Edit ditekan, maka *admin* akan diarahkan kepada halaman Edit Article, dan ketika tombol Delete ditekan, maka data akan dihapus. Pada bagian atas tabel terdapat tombol add yang akan mengarahkan *admin* ke halaman Add Article.

D. Halaman Konten Manajemen Article Edit



Gambar 3.19 Halaman Konten Manajemen Article Edit

Gambar 3.19 merupakan tampilan hasil realisasi dari halaman Konten Manajemen Article Edit. Halaman ini berisi form yang tiap tiap kolomnya telah diisi dengan data dari *database*. Gambar yang sebelumnya menjadi *header* dari artikel dapat diganti dimana data gambar yang lama akan dihapus dan diganti dengan data gambar baru sehingga tidak terjadi penumpukan data.

E. Halaman Konten Manajemen Report Table

ID	Title	Category	Created At	Updated At	Action
1	sdada	Save and Load Data	Nov 20, 2020	Nov 20, 2020	View Delete
2	asddasda	Edit	Nov 28, 2020	Nov 28, 2020	View Delete
3	dasdasd	Setting	Nov 28, 2020	Nov 28, 2020	View Delete

Gambar 3.20 Halaman Konten Manajemen Report Table

Gambar 3.20 merupakan tampilan hasil realisasi dari halaman Konten Manajemen Report Table. Halaman ini menampilkan tabel yang berisi data *report* yang diambil dari *database*. Pada kolom Action, terdapat tombol View dan Delete, dimana ketika tombol View ditekan, maka *admin* akan diarahkan kepada halaman View Report, dan ketika tombol Delete ditekan, maka data *report* akan dihapus.

F. Halaman Konten Manajemen View Report

Title:

Category:

Description:

Email User:

Device Detail:

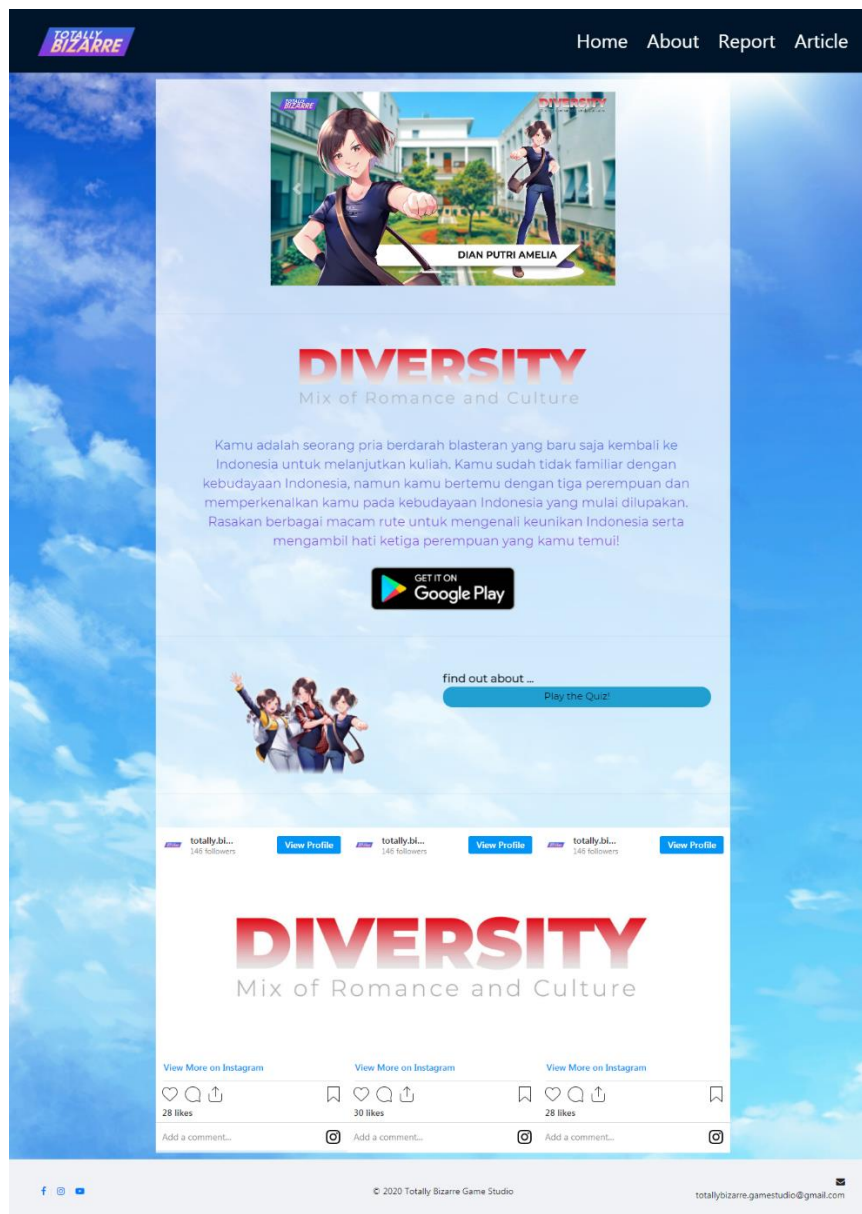
Screenshot :
No Screenshot

Gambar 3.21 Halaman Konten Manajemen View Report

Gambar 3.21 merupakan tampilan hasil realisasi dari halaman Konten

Manajemen View Report. Halaman ini memuat data masukan dari *user* yang mengirimkan *report* melalui halaman Report. Oleh karena merupakan halaman View, maka seluruh kolom input dibuat tidak bisa diubah, sehingga *admin* hanya bisa melihat tanpa mengubah informasi apapun.

G. Halaman Home



Gambar 3.22 Halaman Home

Gambar 3.22 merupakan tampilan hasil realisasi dari halaman Home.

Halaman ini memuat konten *carousel*, tombol Google Play Store yang akan mengarahkan *user* kepada halaman untuk mengunduh game pada Google Play Store, tombol yang akan mengarahkan user pada halaman *minigame* berupa kuis, dan *embed post* milik akun Instagram Totally Bizarre.

H. Halaman About



Gambar 3.23 Halaman About

Gambar 3.23 merupakan tampilan hasil realisasi dari halaman About. Halaman ini memuat konten berupa profil dari perusahaan beserta pendiri-pendiri

dari Totally Bizarre yang meliputi CEO, CFO, dan CTO. Logo dari Skystar Ventures telah ditampilkan pada halaman ini.

3.3 Kendala yang Ditemukan

Dalam proses rancang bangun *website landing page* Totally Bizarre, ditemukan berbagai kendala baik secara teknis maupun nonteknis. Hal ini dapat terjadi karena kurangnya pemahaman dan pengalaman terhadap *framework* yang digunakan, tidak adanya *mentor* yang membimbing selama proses rancang bangun *website*, serta adanya miskomunikasi yang terjadi pada saat berkoordinasi dengan divisi lain, seperti saat melakukan *request* aset atau konten untuk *website*.

3.4 Solusi atas Kendala yang Ditemukan

Berdasarkan kendala yang telah ditemukan selama proses kerja magang, solusi yang telah diambil adalah dengan mempelajari dokumentasi kedua *framework* yang digunakan, mencari tutorial secara mandiri di *internet* dan mencoba mengimplementasikannya, selalu berinisiatif dalam berkomunikasi dengan divisi lain dan mencoba untuk menjelaskan dengan lebih detil agar tidak terjadi kesalahpahaman yang dapat menghambat pekerjaan pribadi divisi tersebut.