

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang di PT Berlian Sistem Informasi sebagai *Fullstack Web Developer* pada divisi *General Affairs* diawali dengan melakukan pertemuan dengan *supervisor* yaitu Bapak Tovan Octa Ferdinan. Pertemuan tersebut membahas mengenai proyek apa yang akan di buat dan *flow* dari situs. Pengerjaan tugas diberikan melalui Azure DevOps dan setiap task tersebut memiliki batas waktu sekitar 14 hari pengerjaan. Setiap beberapa hari akan dilakukan pertemuan dengan Bapak Tovan Octa Ferdinan melaporkan progres dari tugas yang sedang dikerjakan, Bapak Tovan Octa Ferdinan akan mengulas tugas yang dikerjakan dan memberikan pendapatnya, namun terkadang ada revisi yang diberikan seperti perubahan, penambahan dan penghapusan fitur.

3.2 Tugas yang Dilakukan

Pada bulan pertama program kerja magang diawali dengan melakukan pembuatan *database* menggunakan MySQL serta membuat folder proyek ReactJS, menambahkan komponen *form input*, membuat fitur *register* dan *login*, menghubungkan *database* MySQL dengan sharepoint, serta membuat alur dari situs di aplikasi powerapps, membuat fitur *logout*. Dilanjutkan dengan membuat fitur *edit* di halaman *monitoring* yang berfungsi untuk melakukan *edit* data pengguna dan membuat halaman profil.

Pada bulan kedua diawali dengan memperbaharui tampilan *home* dengan mengganti beberapa *form*, menambahkan fitur *edit* data di halaman *monitoring*, menambahkan fitur *show password* di halaman *register* dan *login*, menambahkan Node Mailer untuk mengirim notifikasi *email*, menambahkan *role privilege* dan membuat halaman *error*. Dilanjutkan dengan membuat *database* baru MongoDB untuk proyek yang sama, mengubah semua *query database* MySQL menjadi MongoDB, membuat *data cloud* MongoDB untuk *hosting* dan *deploy*, serta *mendeploy* proyek.

Pada bulan Ketiga diawali dengan menghubungkan API ke *database* sharepoint menggunakan *custom connector* dan membuat *flow* untuk menambahkan data ke sharepoint, dilakukann pertemuan dengan Bapak Tovan

Octa Ferdinan untuk membahas proyek baru yaitu BSI Supplier Portal, membahas desain dari situs dan *flow*-nya. Kemudian mulai membuat folder proyek dan database menggunakan ReactJS dan MongoDB, membuat halaman *register*, *login* dan halaman utama, serta membuat desain untuk setiap halaman. Dilanjutkan dengan membuat struktur database dan mengimplementasikan *authorization* dan *authentication* untuk fitur *register* dan *login*, membuat navbar dan membuat salah satu dari fitur yaitu *Invoice Gateway*, membuat halaman utama untuk *Invoice Gateway*, halaman status, dan halaman status *detail*, membuat halaman *landing*, halaman profil. Kemudian membuat halaman *invoice gateway* status, membuat halaman status *detail*, membuat *edit form* di halaman status *detail*, membuat tombol setuju dan tombol tolak, membuat halaman *admin* untuk *invoice gateway* dan membuat fitur baru yaitu *outsourcing portal*. Setelah membuat fitur baru dilanjutkan dengan membuat *schema model outsourcing portal* dan *route* untuk *backend*.

Pada bulan keempat diawali dengan membuat *route* untuk melakukan *update* progres, membuat fitur baru *vendor registration*, menambahkan halaman histori and histori *detail* dan membuat *route* untuk *update* dan *delete*. Dilanjutkan dengan membuat tiga *route* yaitu *update* status, *submitted* dan *update* untuk fitur *vendor registration* di *backend*, menambahkan *carousel*, mengubah desain untuk tabel *outsourcing*, *invoice gateway* dan *vendor registration* serta menambahkan *route post* untuk sosial media.

Pada bulan kelima diawali dengan membuat fitur *live chat* menggunakan *library* Socket.io, mengubah *id* pengguna berdasarkan *email*, menambahkan ruangan obrolan *private* pada *live chat* dan mengimplementasikan fitur *populate* pada halaman sosial media. Kemudian membuat *comment container* untuk sosial media, membuat fitur *create profile*, membuat halaman profil *detail*, menambahkan *WYSIWYG Editor* pada sosial media dan menambahkan *route update* untuk sosial media. Dilanjutkan dengan menghapus *comment route* dan menambahkan *comment field* pada sosial media, mengimplementasi *like* and *dislike* fitur di sosial media, mengimplementasi fitur *comment* di sosial media, membuat halaman *news* untuk pengguna dan *admin*, membuat *news route*, membuat *route update* dan *delete* untuk *news*, menambahkan fungsi *search* di halaman *news*, membuat *update* dan *delete route* untuk *news*, membuat komponen *contact* di halaman *news* detail dan mengganti konten *carousel* di halaman home dengan konten *news*.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Pembuatan database menggunakan mysql serta membuat folder proyek react. Menambahkan komponen input form serta membuat fitur register dan login untuk website Vendor Registration. Menghubungkan database mysql dengan sharepoint.
2	Menambahkan input tipe file di form. Menambahkan beberapa halaman baru di website. Memperbaiki beberapa error di backend server. Mengupdate halaman register dan login. Menambahkan fitur logout.
3	Membuat projek BSI Registration. Membuat halaman monitoring, Menambahkan fitur edit data di halaman monitoring
4	Mendesign ulang halaman profile, dan memperbaiki bug. Menambahkan library moment untuk merubah format tanggal di halaman monitoring. Memperbaharui navbar. Mengaplikasikan dotenv pada backend dibagian koneksi database. Menambahkan fitur show password di halaman register dan login.
5	Menambahkan node mailer Menambahkan role privilege untuk setiap user
6	Mengganti database mysql dengan mongodb. Mengubah semua query database mysql menjadi mongodb.
7	Mendeploy project menggunakan netlify sebagai front-end dan cloud mongodb sebagai backend dan mengecek setiap route apakah masih terdapat error atau tidak Aplikasi berhasil dideploy, tapi masih ada beberapa error, seperti cookie yang tidak bisa dikases setelah dideploy.
	Lanjut pada halaman berikutnya

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
8	<p>Memperbaiki cookie yang bermasalah, dan aplikasi sudah selesai. Menghubungkan api ke database sharepoint menggunakan custom connector dan membuat flow untuk menambahkan data ke sharepoint per hari. Briefing / Meeting membahas mengenai project baru yaitu BSI Supplier Portal. Membahas web design, web flow. Memulai dengan membuat project folder dan database menggunakan react dan mongodb. Membuat halaman register, login dan homepage. Serta membuat design untuk masing-masing page. Membuat struktur database dan mengimplementasikan authorization dan authentication untuk register dan login. Membuat navbar dan membuat salah satu dari fitur yaitu Invoice Gateway(PayG). Membuat halaman utama untuk PayG, halaman status, dan halaman status detail. Membuat role previlage untuk masing-masing user. Membuat halaman input data Payg dan route post di backend .</p>
9	<p>Mendeploy backend server ke cloud, karena mongodb server tidak bisa diakses menggunakan wifi kantor. Membuat halaman landing page, halaman create profile serta route post and get untuk profile di backend. Membuat halaman Payg status Membuat halaman Payg status detail Membuat edit form di status detail</p>
10	<p>Membuat tombol approve dan reject untuk role <i>admin</i> di halaman Payg status detail. Membuat halaman <i>admin</i> untuk Payg invoice. Membuat Footer untuk setiap halaman, error page, dan toastify untuk setiap event</p>
	Lanjut pada halaman berikutnya

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.2.

Tabel 3.2. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
11	<p>Mengubah landing page dan menambahkan scroll to component fucntion on click.</p> <p>Membuat tabel baru disharepoint yang akan diisi oleh data dari route api Payg serta membuat flow untuk melakukan hal tersebut secara otomatis.</p> <p>Membuat fitur lainnya yang bernama Outsourcing Portal.</p> <p>Membuat dan mendesign halaman history dan history detail.</p> <p>Membuat fungsi input data outsourcing.</p> <p>Membuat fungsi edit data dan update message di halaman history detail</p>
12	<p>Mencoba connect ke sharepoint api dengna node sp-auth</p> <p>Fix value id undefined di fungsi edit outsourcing data.</p> <p>Update halaman home outsourcing portal.</p>
13	<p>Menambahkan route get by id dan delete by id untuk outsourcing portal.</p> <p>Mengganti homepage box design berdasarkan nama fitur.</p> <p>Membuat fitur baru Vendor Registration.</p> <p>Menambahkan halaman history and history detail, route untuk update dan delete.</p> <p>Menambahkan 3 route yaitu update status, submitted, dan update untuk fitur vendor registration di backend.</p>
14	<p>Menambahkan <i>carousel</i>, merubah design table outsourcing, payg dan vendor registration.</p> <p>Megganti fitur logout lama dengan yang baru dan menambahkan fitur search.</p> <p>Menambahkan row status di outsourcing model dan outsourcing home untuk <i>admin</i>.</p> <p>Menambahkan status field dan status route update untuk outsourcing detail.</p>
	Lanjut pada halaman berikutnya

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.3.

Tabel 3.3. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
15	<p>Fix logic button detail dan submit di vendor registration dan add toast.</p> <p>Menambahkan komponen iframe power bi di halaman outsourcing portal.</p> <p>Fix socket IO connection error.</p> <p>Fix tabel di vendor registration karena crash setelah masuk ke page detail.</p> <p>Disable socket.io connection.</p> <p>Fix email message value dan membuat fitur sosial media.</p>
16	<p>Menambahkan library mutler di sosmed profile routes.</p> <p>Menambahkan route post untuk sosial media post dan membuat modal untuk posting.</p> <p>Membuat live chat / chat room menggunakan library socket.io.</p> <p>Mengubah id user berdasarkan email user.</p> <p>Memperbaiki online user tidak tampil pada halaman live chat.</p> <p>Menambahkan ruangan private dan private chat pada live chat.</p> <p>Implementasi ref dan populate di social media profile dan post route.</p>
17	<p>Membuat comment container.</p> <p>Membuat route untuk delete user profile social media.</p> <p>Membuat fitur create profile di social media.</p> <p>Membuat halaman profile detail.</p> <p>Menambahkan WYSIWYG editor untuk create post social media.</p> <p>Menambahkan route update untuk social media post.</p> <p>Membuat delete dan update route untuk social media post.</p> <p>Update beberapa field yang akan dipopulate di social media post, profile, and user route.</p> <p>Memperbaiki fitur create post di social media.</p>
18	<p>Mempopulate sosmed profile model.</p> <p>Mengupdate comments route.</p> <p>Menghapus comment route dan menambahkan comment field di social media post.</p>
	Lanjut pada halaman berikutnya

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.4.

Tabel 3.4. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
18	Mengimplementasi like and dislike fitur di social media. Memperbaiki route unlike yang tidak berfungsi sebelumnya. Menambahkan kondisi di halaman social media untuk button like dan unlike.
19	Mengimplementasi fitur comment di social media. Membuat halaman news dan news untuk <i>admin</i> . Membuat news route. Membuat card komponen untuk setiap news. Membuat route post untuk mengupload news. Membuat route update dan delete untuk news route. Menambahkan fungsi getAllNews di backend. Membuat halaman news details.
20	Menambahkan fungsi search di halaman news. Membatasi input dari judul di news. Membuat update dan delete route untuk news. Membuat komponen contact me di halaman news details. Membuat FAB di halaman news details. Mengganti konten <i>carousel</i> di halaman home dengan konten news / announcement. Update commet container di halaman social media.
21	Membuat error handling untuk halaman register. Memperbaiki bug di halaman home bagian css. Mendeploy website untuk test mengguna vercel.

3.4 Proyek BSI Supplier Portal

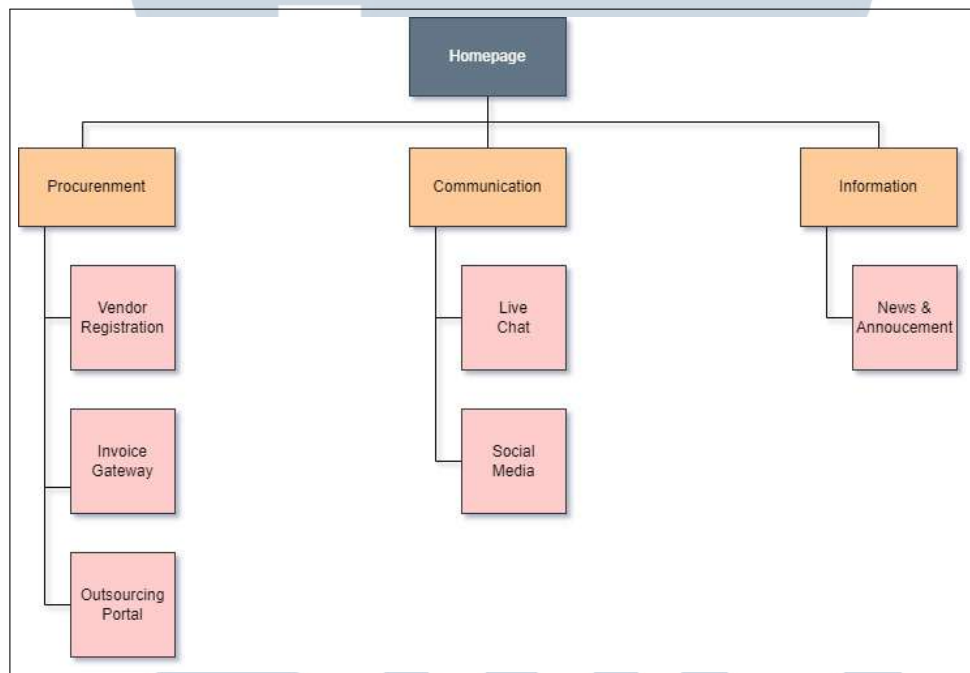
Proyek situs BSI Supplier Portal merupakan proyek yang dibuat untuk memenuhi kebutuhan penggunaanya dalam melakukan proses pendataan dan pendaftaran *vendor*. Dalam merancang dan membangun proyek BSI Supplier Portal ada beberapa tahapan-tahapan yang harus dilakukan sebelumnya, yaitu sebagai berikut :

3.4.1 Pengumpulan Data

Pengumpulan data dilakukan dengan mengadakan pertemuan dengan Bapak Tovan Octa Ferdinan dan juga beberapa tim dari divisi *General Affairs* dan *Procurement* untuk membahas desain, alur, dan fitur apa saja yang akan dibangun. Berdasarkan hasil *meeting* didapat sitemap, flowchart, dan *database*.

A. Sitemap

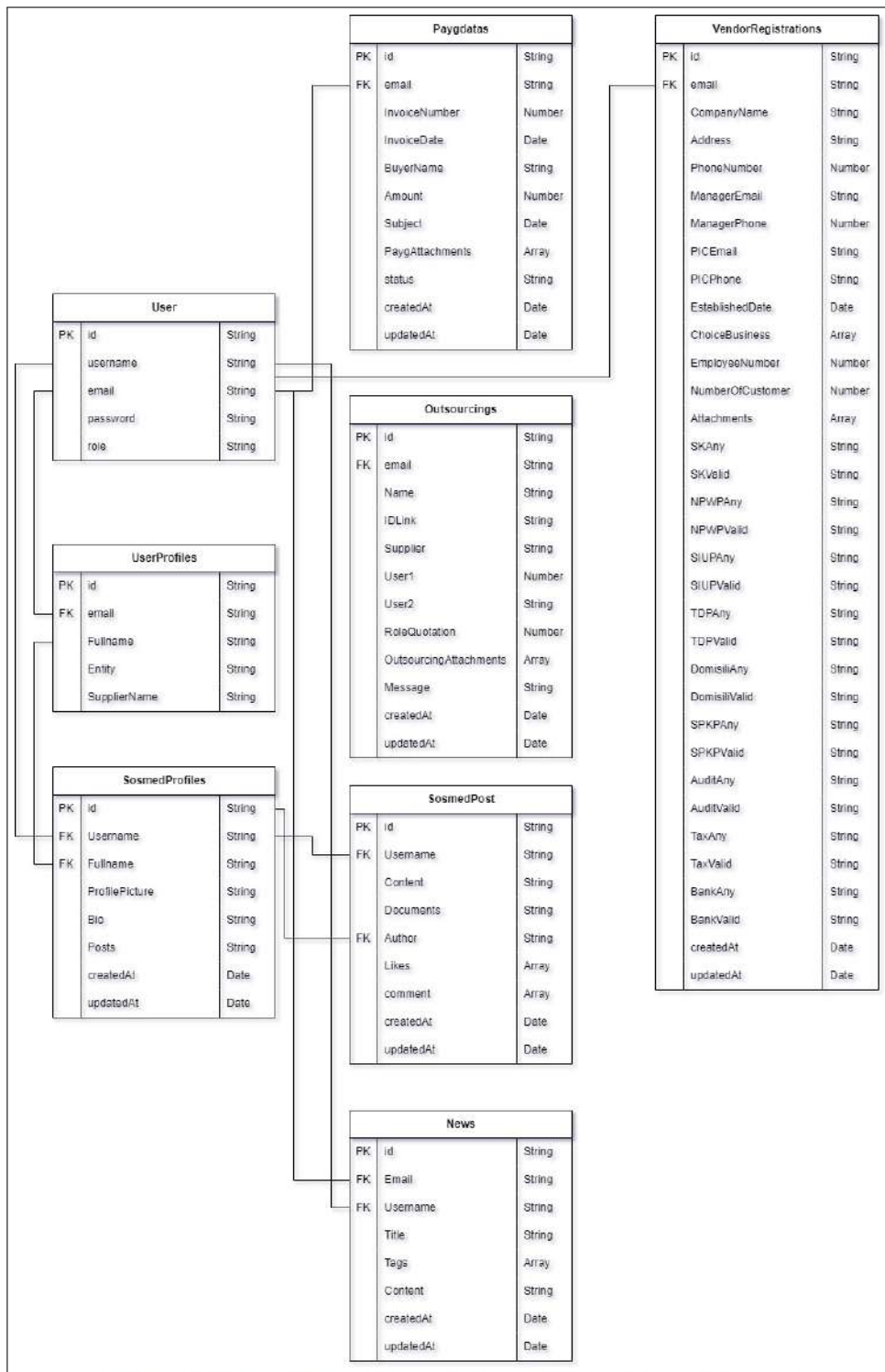
Terdapat tiga halaman utama, yaitu halaman *procurement*, *communication*, dan halaman *information*. Masing-masing dari halaman tersebut mempunyai fitur-fitur yang bisa diakses dan dijabarkan pada gambar 3.1.



Gambar 3.1. Sitemap Website

B. Database Skema

Terlihat tabel yang dibutuhkan dalam pembuatan situs, dimana dari masing-masing tabel memiliki *primary key* yang berupa *id* dari tabel itu sendiri dan juga mempunyai *foreign key (FK)* yang mempresentasikan relasi terhadap tabel lain dan dijabarkan pada gambar 3.2.



Gambar 3.2. Database Skema

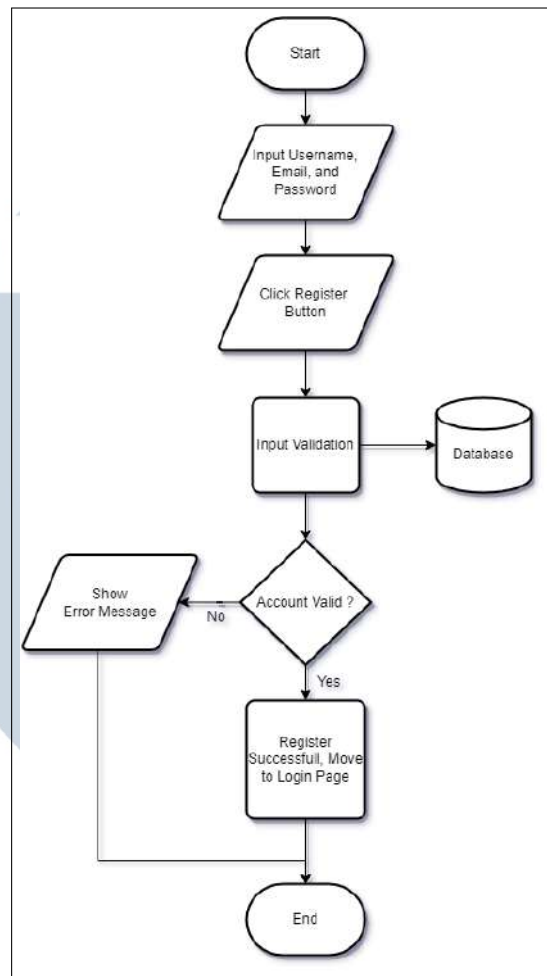
C. Flowchart

Pada proyek situs BSI Supplier Portal terdapat sebuah alur dalam aksi yang dapat digambarkan dalam sebuah flowchart. Terdapat beberapa flowchart yang menjelaskan aksi-aksi pada situs BSI Supplier portal seperti *register*, *login*, *input data vendor registration*, *input data invoice gateway*, *input data outsourcing portal*, *create profile social media*, *create post social media* dan *create news*.

1. Flowchart Register

Terdapat beberapa tahap ketika pengguna melakukan proses registrasi dimulai dari pengguna melakukan *username* dan *input password* pada situs BSI Supplier Portal. Ketika pengguna melakukan *input email*, tapi *email* tersebut telah terdaftar akan muncul *popup* notifikasi dan pengguna harus memasukan *email* lainnya. Ketika proses *registrasi* berhasil pengguna akan diarahkan menuju ke halaman *login* yang dijabarkan pada gambar 3.3.

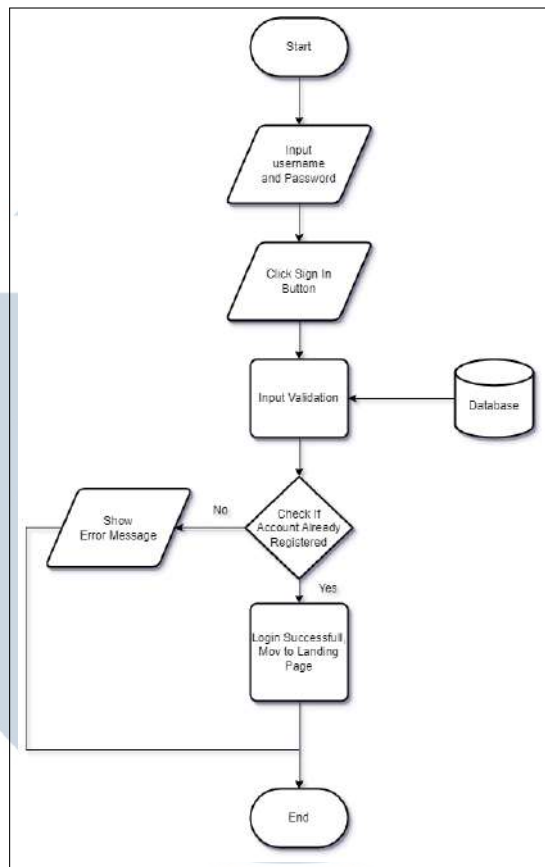




Gambar 3.3. Flowchart Register

2. Flowchart Login

Terdapat beberapa tahap ketika pengguna akan melakukan proses *login* ke situs, dimulai dari pengguna memasukkan *email* dan *password* pada situs BSI Supplier Portal. Ketika pengguna melakukan *input email* atau *password*, tapi *email* dan *password* tidak terdaftar di dalam *database* maka pengguna akan diminta untuk memasukkan ulang *email* dan *password* mereka, setelah berhasil pengguna akan diarahkan menuju halaman *landing* dari situs BSI Supplier Portal yang dijabarkan pada gambar 3.4

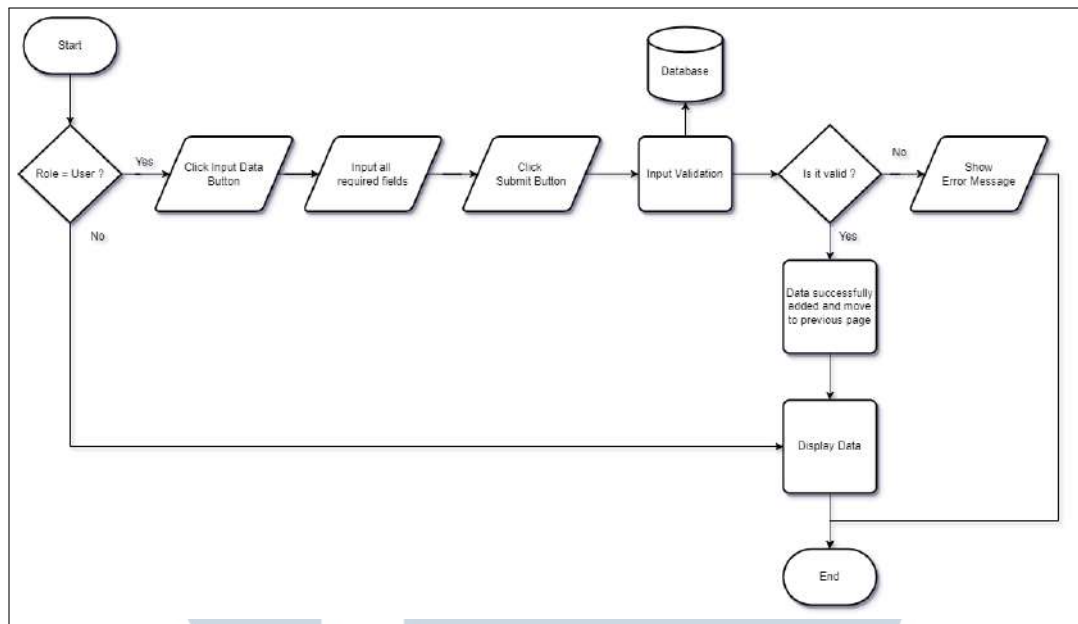


Gambar 3.4. Flowchart Login

3. Flowchart Input Data Vendor Registration

Pengguna dapat melakukan *input data* jika pengguna masuk ke situs sebagai *user* kemudian akan muncul *popup* yang pengguna harus isi, setelah itu pengguna dapat mengklik tombol *submit* dan data berhasil disimpan ke *server*. Sedangkan jika pengguna masuk sebagai *admin* pengguna tidak bisa melakukan aksi *input data*, tapi pengguna bisa melihat semua data-data tersebut yang dijabarkan pada gambar 3.5 ‘

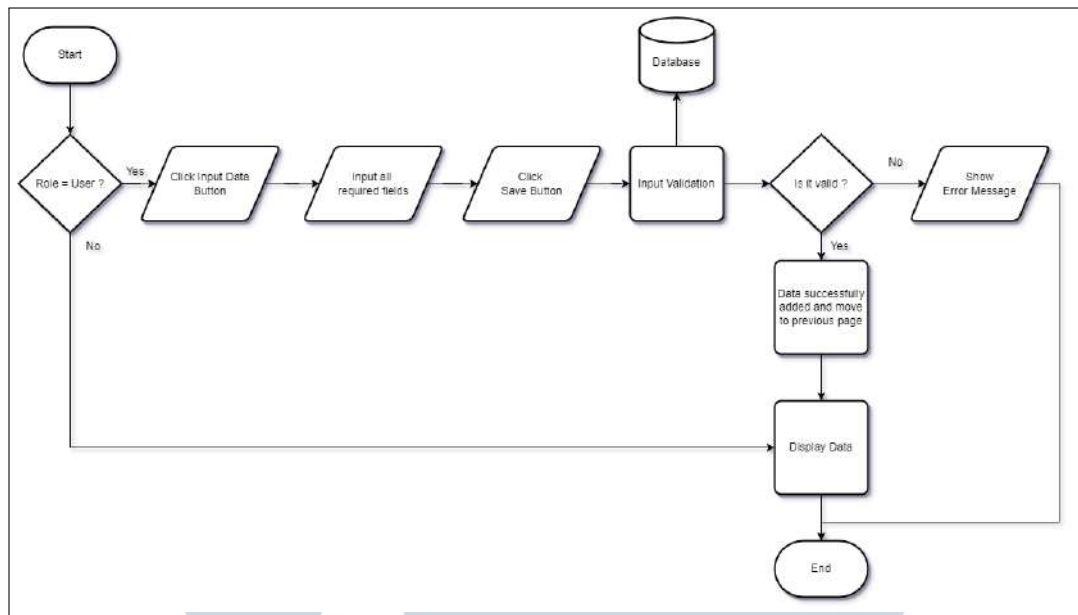
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. Flowchart Input Data Vendor Registration

4. Flowchart Input Data Invoice Gateway

Pengguna dapat melakukan *input data* jika pengguna masuk ke situs sebagai *user* kemudian akan muncul *popup* yang pengguna harus isi, setelah itu pengguna dapat mengklik tombol *submit* dan data berhasil disimpan ke *server*. Sedangkan jika pengguna masuk sebagai *admin* pengguna tidak bisa melakukan aksi *input data*, tapi pengguna bisa melihat semua data-data tersebut yang dijabarkan pada gambar 3.6. ‘

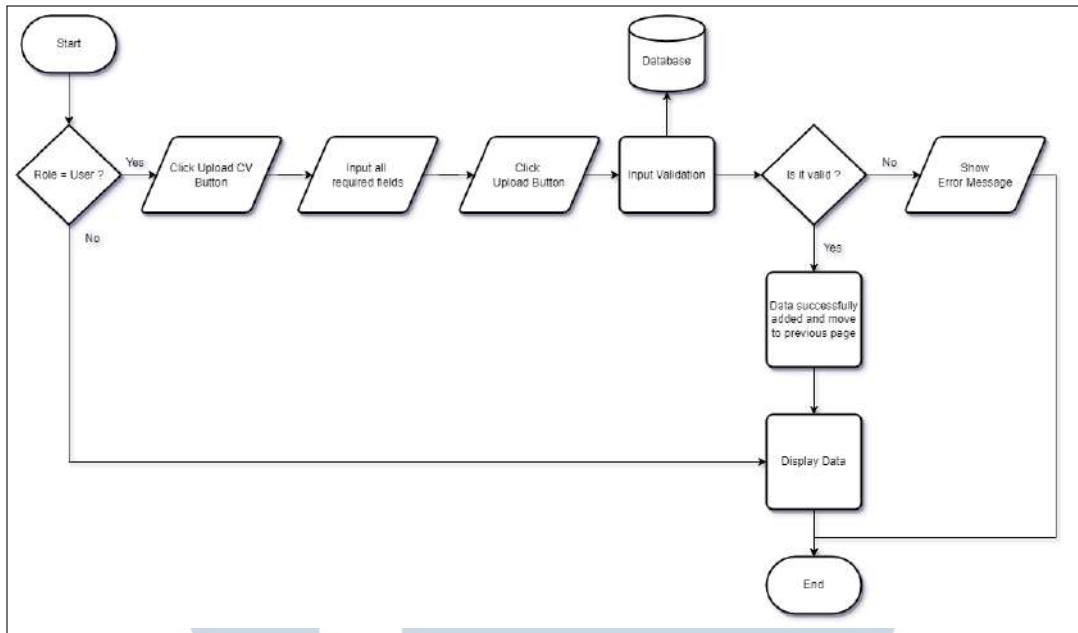


Gambar 3.6. Flowchart Input Data Invoice Gateway

5. Flowchart Input Data Outsourcing Portal

pengguna dapat melakukan *upload cv* jika pengguna masuk ke situs sebagai *user*, kemudian akan muncul *popup* yang pengguna harus isi, setelah itu pengguna dapat meng klik tombol *submit* dan data berhasil disimpan ke *server*. Sedangkan jika pengguna masuk sebagai *admin* pengguna tidak bisa melihat tombol *upload cv*, tapi pengguna bisa melihat semua data-data tersebut yang dijabarkan pada gambar 3.7. ‘

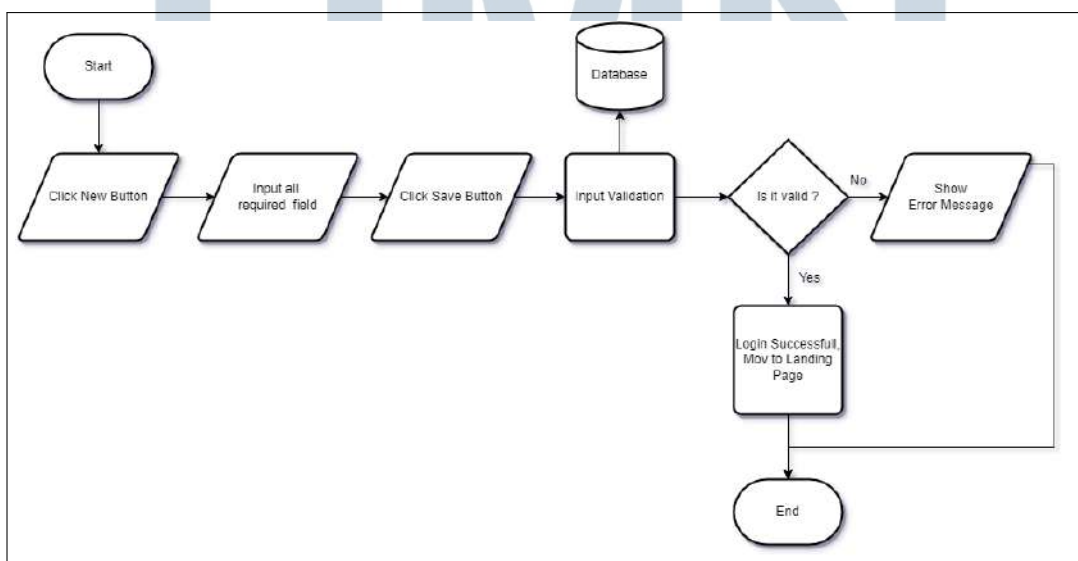
U M M N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



Gambar 3.7. Flowchart Input Data Outsourcing Portal

6. Flowchart Create Social Media Profile

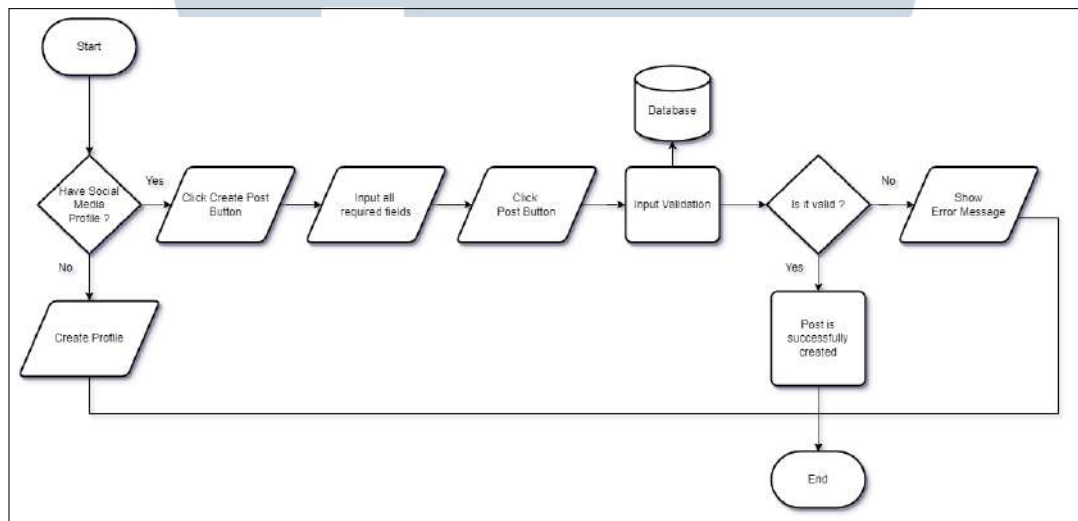
Pengguna dapat mengklik tombol *click new* untuk membuat *profile*, aksi yang muncul setelah pengguna mengklik tombol *click new* adalah munculnya *popup form*. Pengguna diminta mengisi *form* tersebut untuk dapat membuat *social media profile* yang dijabarkan pada gambar 3.8.



Gambar 3.8. Flowchart membuat sosial media profile

7. Flowchart Create Social Media Post

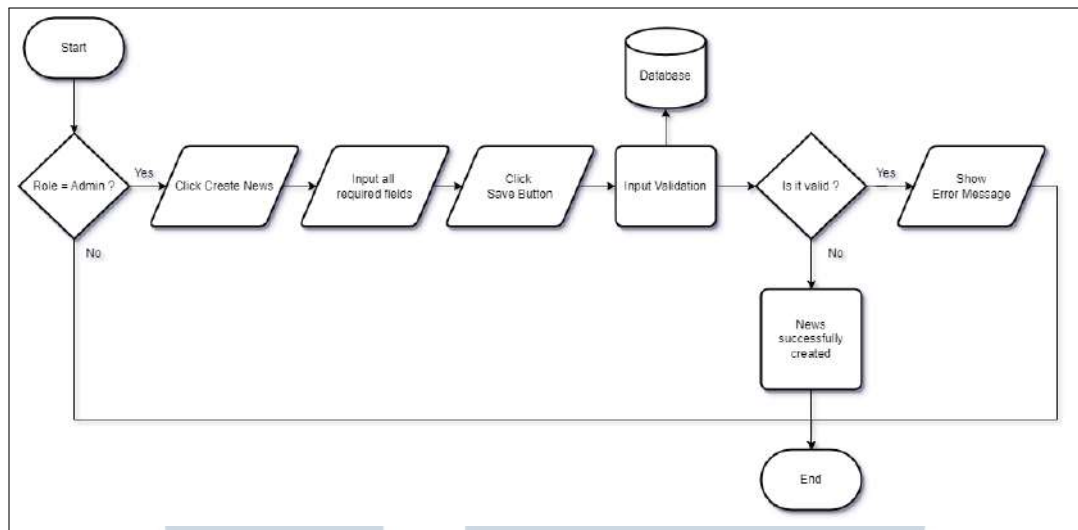
Proses yang terjadi ketika pengguna akan membuat *post social media*, sebelum pengguna bisa meng-*post* postingan pengguna harus mempunyai sebuah *social media profile* terlebih dahulu, jika pengguna belum membuat *social media profile* maka tombol *create post* tidak akan terlihat oleh pengguna. Jika pengguna sudah membuat *social media profile* maka akan muncul tombol *create post* ketika tombol tersebut di klik akan muncul *popup modal* yang pengguna harus isi. Setelah berhasil meng-*input* data, data tersebut akan tersimpan di *database* yang dijabarkan pada gambar 3.9.



Gambar 3.9. Flowchart membuat sosial media post

8. Flowchart Create News

Pengguna yang masuk sebagai *admin* saja yang bisa mengakses halaman *dashboard*, dihalaman *dashboard* terdapat fitur untuk membuat *news*, jika tombol *create news* di klik maka akan muncul *popup form*. Kemudian pengguna bisa meng-*input* data mereka dan data tersebut akan tersimpan di *database*. Pengguna yang *login* sebagai *user* hanya bisa melihat *news* yang sudah dibuat oleh *admin* yang dijabarkan pada gambar 3.10. ‘



Gambar 3.10. Flowchart membuat news

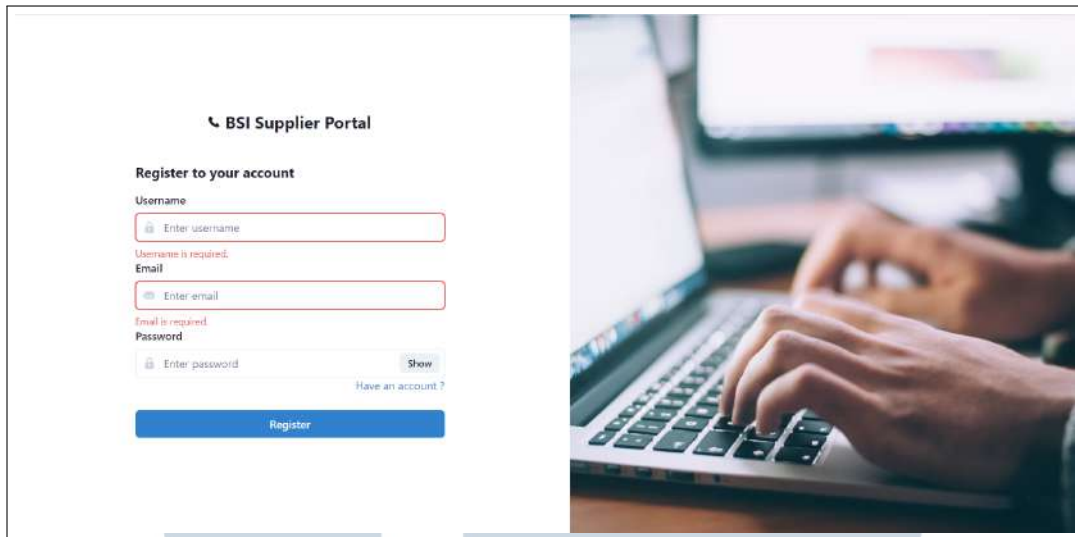
3.4.2 Implementasi

Implementasi dilakukan dengan mengubah desain *mockup* yang sudah dibuat kedalam bentuk kode dengan bahasa pemrograman javascript dan *framework* reactJs.

A. Halaman Register dan Login

Register dan *Login* merupakan salah satu fitur yang sangat penting untuk situs BSI Portal Supplier. *Login* merupakan proses pintu masuk bagi pengguna untuk bisa mengakses situs, sebelum pengguna bisa melakukan *login*, mereka terlebih dahulu harus menjalankan proses registrasi.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.11. Halaman Register

Pada Gambar 3.11, merupakan tampilan dari halaman *register* ketika pengguna mengakses *link* dari situs. Terdapat tiga *input* yang wajib diisi oleh pengguna yaitu *username*, *email*, dan *password*. Jika pengguna memasukkan *email* yang sama maka proses registrasi akan gagal dan pengguna akan diminta untuk memasukkan melakukan *input* ulang. Terdapat juga tombol *register*, ketika tombol tersebut di klik pengguna akan di arahkan ke halaman *login* ketika proses registrasi berhasil.

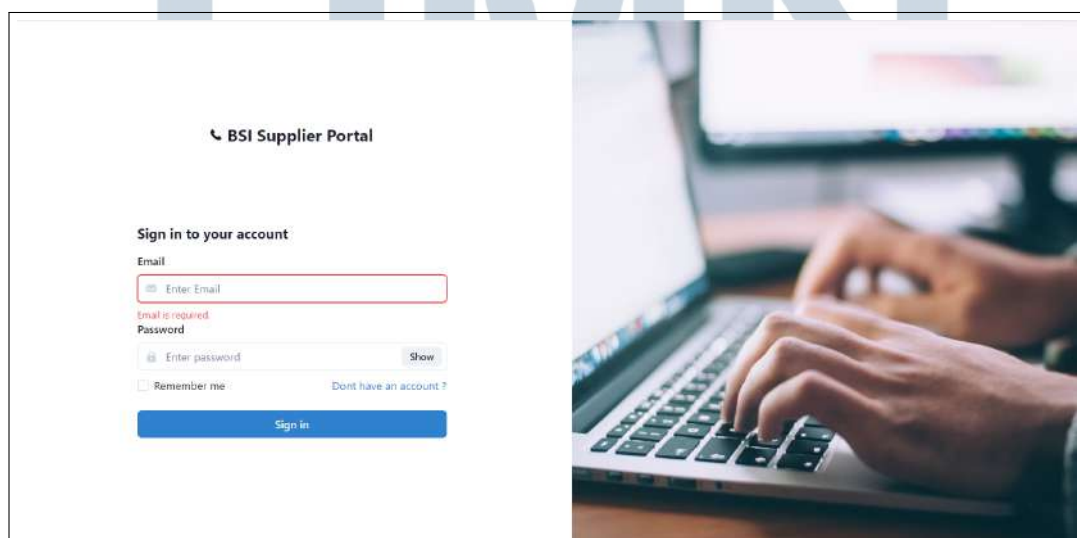
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Register

```
{  
  _id : <ObjectID>,  
  username : "username",  
  email : "email",  
  password : "password",  
  role : "role"  
}
```

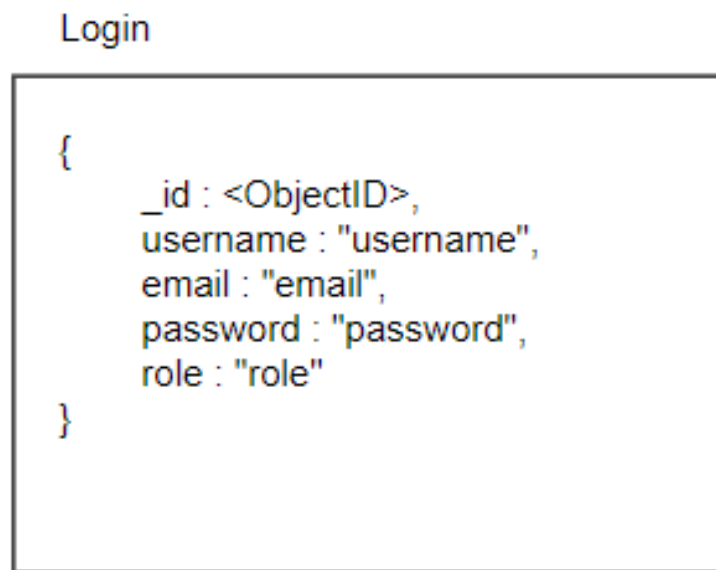
Gambar 3.12. Schema Register

Pada Gambar 3.12, merupakan *schema* dari *register model*. Di dalam *schema* tersebut terdapat beberapa *field* yang terdiri dari *id*, *username*, *email*, *password* dan *role*. Terdapat proses pengiriman data menuju ke *server* yang dilakukan pada *endpoint register*. Proses pengiriman data pada *endpoint register* menggunakan metode *post*, ketika pengguna berhasil mengirimkan data ke *server*, pengguna akan di beri *role*.



Gambar 3.13. Halaman Login

Pada Gambar 3.13, merupakan tampilan dari halaman *login* ketika pengguna berhasil melakukan proses registrasi sebelumnya. Di halaman *login* pengguna diminta untuk memasukan *credentials* yang sebelumnya sudah di daftarkan. Terdapat juga tombol *sign in*, ketika tombol *sign in* di klik, pengguna akan diarahkan ke halaman *landing* setelah proses *login* berhasil dan tidak ditemukan kendala.



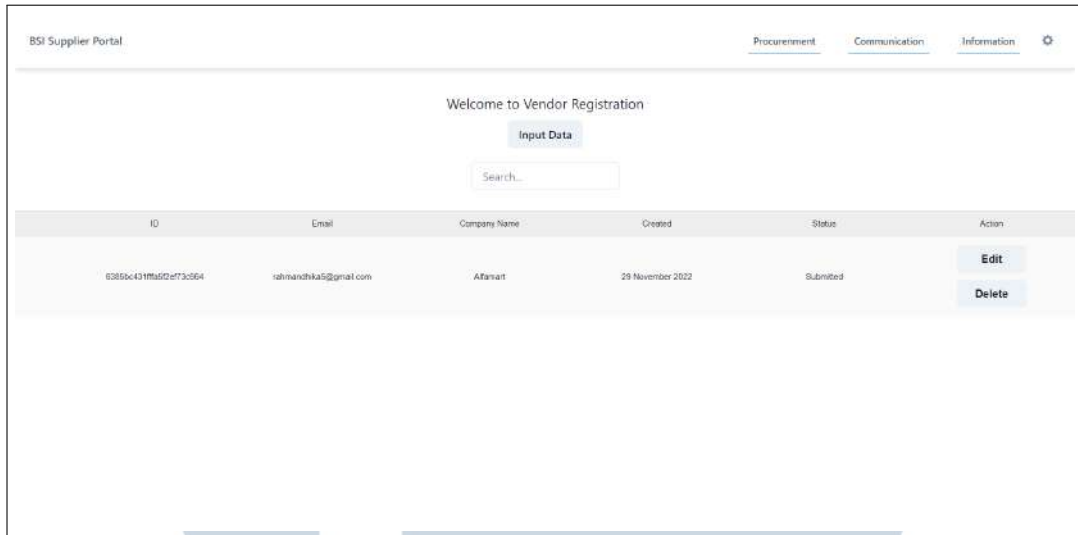
Gambar 3.14. Schema Login

Berdasarkan Gambar 3.14, gambar tersebut menampilkan *schmea* dari *login model*. Data dari *schema login* sama dengan *schema register*. Terdapat proses pengambilan data dari *server* dan proses pengiriman data menuju ke *server* dengan *endpoint login*, metode yang digunakan berupa *post* dan *get*. Metode *post* akan mengirimkan data menuju ke *server* sedangkan metode *get* akan mengambil data dari *server* dan menyimpannya di pihak *client*.

B. Halaman Vendor Registration

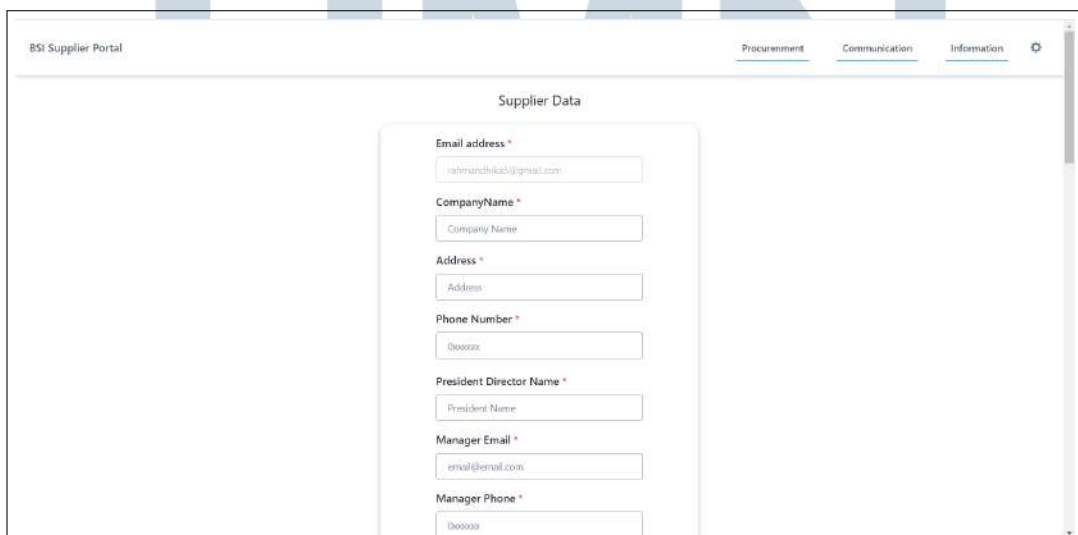
Vendor Registration merupakan fitur atau halaman yang mempunyai tujuan bagi pada pengguna untuk mendaftarkan diri mereka ke perusahaan melalui situs dengan melakukan *input* data diri. Halaman ini dibedakan menjadi dua yaitu halaman untuk pengguna dan *admin*. Di halaman pengguna, mereka bisa melakukan *input* data, *edit* data, dan *delete* data mereka sendiri. Sedangkan untuk

halaman *admin*, *admin* bisa menerima ataupun menolak data yang sudah di *submit*.



Gambar 3.15. Halaman Home Vendor Registration

Berdasarkan Gambar 3.15, terdapat sebuah tabel dengan kolom *id*, *email*, *company name*, *created date*, *status*, dan tombol *action*. Masing-masing kolom ditabel menampilkan sedikit bagian penting dari data yang pengguna. Terdapat tombol *input* data yang akan mengarahkan pengguna ke halaman *input* data ketika tombol tersebut di klik. Serta terdapat dua tombol aksi di tabel, tombol tersebut adalah *edit* dan *delete*.



Gambar 3.16. Halaman Input Data Vendor

Pada Gambar 3.16, terdapat sebuah container yang berisi beberapa kolom

input data yang wajib pengguna isi ketika ingin mendaftarkan diri mereka situs BSI Supplier Portal.

BSI Supplier Portal

Procurement Communication Information

Download Edit Submit

Supplier Data

Company Name : Alfamart
 Address : Banjar wijaya cluster asia b26 no 56
 Phone Number : 081280410775
 President Director (Name) : Adam
 Account Manager (Email / Phone) : adam@adam.com / 11111111111
 Escalation PIC (Email / Phone) : adam@adam.com / 22222222222
 Established Date : 29 November 2022
 Main Business : Others
 Employee Number : 500+
 Number of Customer : 500
 Attachments : https://empty-test-project.herokuapp.com/images/1669708864092.jpg

Document

Document	Any / Not	Valid / Invalid
SK Mentum & Akta Notaris	Not	Invalid
NPVWP	Any	Valid
SIUP	Not	Valid
TDP	Not	Valid
Surat Domisil	Not	Valid
SPKP	Any	Invalid
Latest Financial Report (Audit)	Any	Valid
Tax Report	Any	Valid
Bank Account	Any	Valid

Gambar 3.17. Halaman Details Vendor Registration

Berdasarkan pada Gambar 3.17 yang berada pada halaman *home vendor registration*, ketika pengguna mengklik tombol *edit* yang terletak di tabel aksi, pengguna akan diarahkan ke halaman *history detail*. Di halaman ini pengguna bisa melihat semua data-data yang mereka *input* disajikan dalam bentuk tabel dan tulisan. Dibagian atas terdapat tiga tombol yang berbeda, tombol *download* digunakan untuk mengunduh data dan merubahnya menjadi format pdf, tombol *edit* berguna untuk melakukan *edit* data yang ingin diubah, dan tombol *submit* berguna untuk mengirimkan notifikasi *email*, pihak perusahaan dapat menerima ataupun menolak data yang sudah di kirim.

Vendor Registration

```
{
  "_id": "63b99c015aab54527691dca7",
  "email": "rahmandhika5@gmail.com",
  "CompanyName": "BINUS",
  "Address": "Yaman Anggrek",
  "PhoneNumber": "081280410775",
  "PresidentName": "Fahry",
  "ManagerEmail": "fahry@fahry.com",
  "ManagerPhone": "022232144523",
  "PICEmail": "joker@joker.com",
  "PICPhone": "022213321456",
  "EstablishedDate": "2023-01-07T00:00:00.000Z",
  "ChoiceBusiness": [
    [
      "IT Outsourcing,IT Outsourcing,IT Outsourcing"
    ]
  ],
  "EmployeeNumber": "1-50",
  "NumberOfCustomer": 50,
  "Attachments": [
    [
      "https://bsi-portal-service-
      production.up.railway.app/images/1673108481487.jpg"
    ]
  ],
  "SKAny": "Not",
  "SKValid": "Valid",
  "NPWPAny": "Any",
  "NPWPValid": "Invalid",
  "SIUPAny": "Any",
  "SIUPValid": "Valid",
  "TDPAny": "Any",
  "TDPValid": "Valid",
  "DomisiliAny": "Any",
  "DomisiliValid": "Valid",
  "SPKPAny": "Not",
  "SPKPValid": "Valid",
  "AuditAny": "Any",
  "AuditValid": "Valid",
  "TaxAny": "Any",
  "TaxValid": "Valid",
  "BankAny": "Any",
  "BankValid": "Invalid",
  "createdAt": "2023-01-07T16:21:21.514Z",
  "updatedAt": "2023-01-07T16:21:21.514Z",
  "__v": 0
}
```

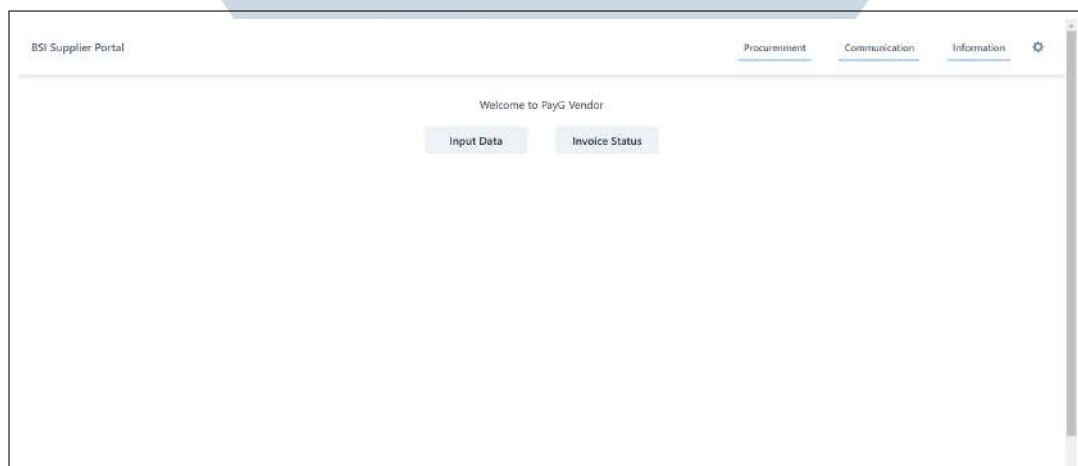
Gambar 3.18. Schema Vendor Registration

Berdasarkan Gambar 3.18, terdapat sebuah *schema vendor registration*, *schema* tersebut memiliki beberapa *field* yang terdiri dari *id*, *email*, *company name*, *address*, *phone number*, *president name*, *manager email*, *manager phone*, *pic email*, *pic phone*, *established date*, *choice business*, *employee number*, *number of customer*, *attachments*, *etc*. *Vendor registration model* memiliki delapan *endpoint* yang terdiri dari *endpoint registration* menggunakan metode *post*, *endpoint* ini digunakan untuk mengirimkan data dari *client side* ke *server*. *Endpoint* selanjutnya yaitu *registration*, berbeda dengan *endpoint* sebelumnya *endpoint registration* ini menggunakan

metode berbeda yaitu *get*, yang digunakan untuk mengirim data di *server* ke *client side*. *Endpoint* selanjutnya adalah *all*, *endpoint* ini menggunakan metode *get* yang berfungsi untuk mengambil semua data dari *server* dan mengirimnya ke *client*, *endpoint id* menggunakan metode *get*, *endpoint* ini berfungsi untuk mencari data di *server* yang memiliki *id* yang sama setelah menemukan *id* yang sama, data tersebut akan dikirimkan ke *client side*. Tiga *endpoint* selanjutnya menggunakan metode *put* yang mempunyai fungsi untuk meng-*update* data, dan *endpoint* terakhir adalah *delete*, *endpoint* ini menggunakan metode *delete* yang berfungsi untuk menghapus data yang tidak diinginkan.

C. Halaman Invoice Gateway

Invoice Gateway merupakan fitur atau halaman yang mempunyai tujuan untuk mendaftarkan *invoice* ke *sistem* perusahaan guna untuk menyimpan bukti.



Gambar 3.19. Invoice Gateway Home

Pada Gambar 3.19, merupakan tampilan awal dari halaman *Invoice Gateway*. Terdapat juga 2 tombol yaitu tombol *input data* dan tombol *invoice status*, masing-masing dari tombol tersebut mempunyai aksi yang berbeda ketika di klik.

Gambar 3.20. Invoice Gateway Input Data

Berdasarkan Gambar 3.20, jika tombol input data di klik, akan ada aksi yang mengarahkan ke halaman *input data*. Dimana di halaman *input data* pengguna akan melakukan *input data*. Setelah pengguna mengklik tombol *submit*, maka akan langsung diarahkan ke halaman *home invoice gateway*.

Invoice Number	Email	Buyer Name	Status	Action
122	nfmandhika@gmail.com	Town Ocle Festival	Approved	Edit Delete

Gambar 3.21. Invoice Gateway Status

Berdasarkan Gambar 3.21, jika tombol *invoice status* di klik, akan ada aksi yang mengarahkan ke halaman *invoce status*. Dimana di halaman ini pengguna dapat melihat data dan status data tersebut, status dapat diulas oleh pihak perusahaan, ulasan tersebut berupa diterima atau ditolak. Terdapat juga tombol *edit*

dan *delete*, dimana jika tombol *edit* di klik pengguna dapat melihat semua data mereka secara keseluruhan.

```
Invoice Gateway

{
  "_id": 1,
  "Email": "rahmandhika5@gmail.com",
  "InvoiceNumber": "123",
  "InvoiceDate": "2023-01-07T00:00:00.000Z",
  "BuyerName": "Tovan Octa Ferdinan",
  "Amount": 100000,
  "Subject": "Belanja",
  "PaygAttachments": [
    [
      "https://bsi-portal-service-
      production.up.railway.app/images/1673108856429.jpg"
    ]
  ],
  "status": "None",
  "createdAt": "2023-01-07T16:27:36.431Z",
  "updatedAt": "2023-01-07T16:27:36.431Z",
  "__v": 0
}
```

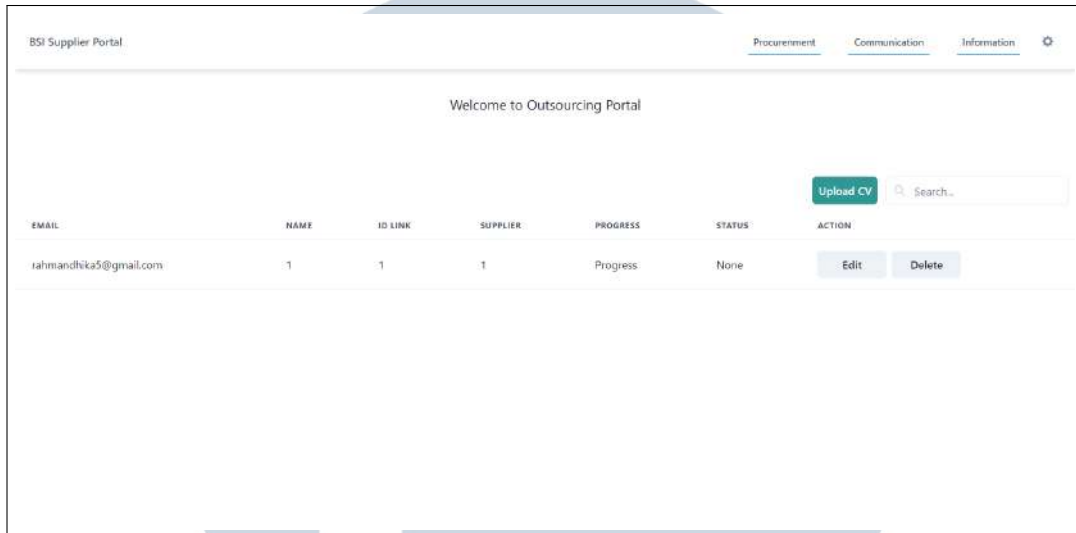
Gambar 3.22. Schema Invoice Gateway

Pada Gambar 3.22 merupakan tampilan data dari *schema invoice gateway*, *schema* ini memiliki *field* yang terdiri dari *id*, *email*, *invoice number*, *buyer name*, *amount*, *subject*, *attachments*, *status*, *created at*, dan *updated at*. *Schema* atau *model invoice gateway* mempunyai *endpoint* yang hampir sama dengan *schema vendor registration*. *Endpoint* tersebut terdiri dari *paygdata*, *all*, *email*, *id*, *submitted*, dan *status*. Masing-masing *endpoint* memiliki metode yang berbeda-beda, *endpoint paygdata* menggunakan metode *post*, *endpoint all*, *email*, dan *id* menggunakan metode *get*, sedangkan untuk *endpoint submitted*, dan *status* menggunakan metode *put*. Metode *post* digunakan untuk mengirimkan data dari pihak *client* yang akan disimpan di *server*, metode *get* digunakan untuk mengirimkan data dari server ke pihak *client*, data yang di kirim menuju *client* dapat berubah-ubah berdasarkan *logic* yang dibuat. Untuk metode *delete* sendiri digunakan untuk menghapus data yang tidak diinginkan, data dihapus berdasarkan data yang mempunyai *id* yang sama.

D. Halaman Outsourcing Portal

Outsourcing Portal merupakan fitur atau halaman yang berfungsi untuk melakukan perekrutan karyawan, ada dua halaman yang tersedia yaitu untuk pengguna dan *admin*. Di halaman pengguna, pengguna hanya akan bisa

mendaftarkan data mereka dan melihat data beserta statusnya. Di halaman *admin*, *admin* bisa melakukan *review* terhadap data, *edit*, dan *delete* data pengguna tersebut.



EMAIL	NAME	ID LINK	SUPPLIER	PROGRESS	STATUS	ACTION
rahmandika5@gmail.com	1	1	1	Progress	None	Edit Delete

Gambar 3.23. Outsourcing Portal Home

Pada Gambar 3.23, terdapat sebuah tabel yang menampilkan data dari pengguna yang sudah melakukan *upload CV* mereka ke sistem, tabel hanya akan menampilkan data berdasarkan *email* mereka. Kemudian ada tombol *upload cv* dan *search bar* yang digunakan untuk melakukan proses *filter* data berdasarkan *input* yang pengguna masukan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.24. Outsourcing Portal Upload CV

Berdasarkan Gambar 3.24, terdapat sebuah modal yang terbuka jika pengguna mengklik tombol *upload cv* di halaman *home outsourcing portal*, akan ada beberapa kolom yang pengguna wajib isi sebelum mereka bisa melakukan *upload cv*.

ID	Email	Name	ID Link	Supplier	User-1	User-2	Role Quotation	Attachments	Created	Updated	Progress	Status
639c14404cf6c682b07a2763	rahmandhika5@gmail.com	1	1	1	1	1	0	https://bsi-portal-service-production.up.railway.app/images/1671173184025.png	December 16th 2022, 1:46:24 pm	December 16th 2022, 1:46:24 pm	Current Progress	

Gambar 3.25. Outsourcing Portal Details

Pada Gambar 3.25, ketika pengguna meng klik tombol *edit* di tabel sebelumnya, pengguna akan diarahkan langsung ke halaman detail dimana halaman

ini akan menampilkan seluruh data dari pengguna. Terdapat juga tombol *edit* data progres, tombol *edit* data berfungsi untuk mengubah data yang ingin diubah. Juga terdapat tombol untuk mengulas data, ulasan tersebut berupa diterima dan ditolak serta hanya bisa yang hanya bisa dilihat dan digunakan jika pengguna masuk sebagai *admin*.

Outsourcing Portal

```

{
  "_id": "63b99e0b5aab54527691dcb4",
  "Email": "rahmandhika5@gmail.com",
  "Name": "Rahmandhika",
  "IDLink": "1",
  "Supplier": "BCA",
  "User1": "1",
  "User2": "1",
  "RoleQuotation": 0,
  "OutsourcingAttachments": [
    [
      "https://bsi-portal-service-
      production.up.railway.app/images/1673109002456.jpg"
    ]
  ],
  "Message": "Current Progress",
  "createdAt": "2023-01-07T16:30:03.024Z",
  "updatedAt": "2023-01-07T16:30:03.024Z",
  "_v": 0
}

```

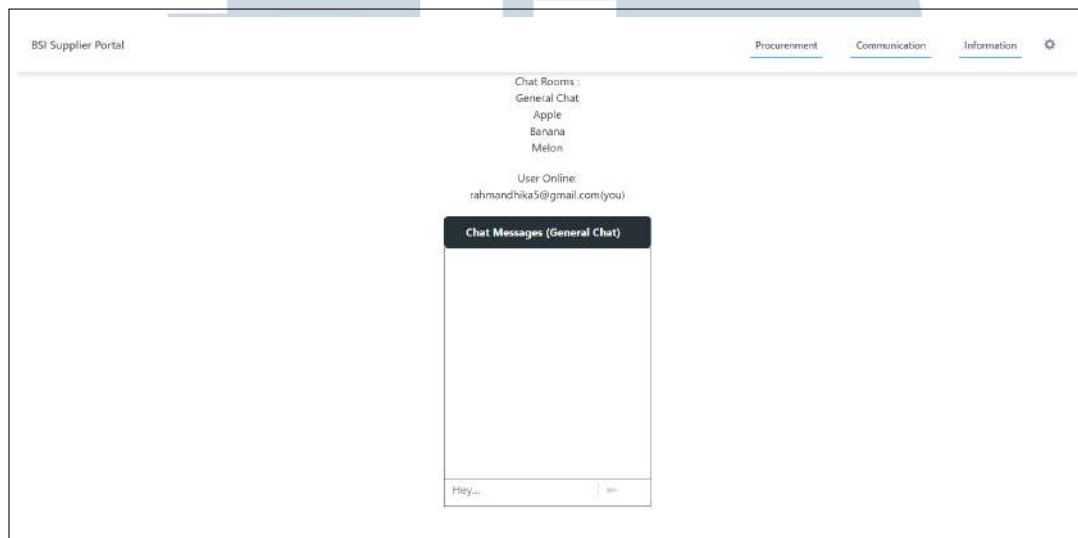
Gambar 3.26. Schema Outsourcing Portal

Pada Gambar 3.26 menampilkan gambar dari *schema outsourcing portal*, *schema* tersebut memiliki *field* yang terdiri dari *id*, *email*, *name*, *id link*, *supplier*, *user 1*, *user 2*, *role quotation*, *attachments*, *message*, *created at*, dan *updated at*. *Outsourcing portal schema / model* memiliki beberapa *endpoint* diantaranya adalah *outsourcing*, *all*, *id*, *update message*, *update status*, dan *delete*. *Endpoint outsourcing* mempunyai dua metode yang berbeda yaitu menggunakan metode *post* dan metode *get*, sementara *endpoint all*, *id* menggunakan metode *get* dan *endpoint update message* dan *update status* menggunakan metode *put* serta terdapat juga *endpoint delete*. Masing-masing dari *endpoint* memiliki fungsi yang berbeda-beda, *outsourcing* menggunakan 2 metode yang berbeda, metode *post* digunakan untuk mengirimkan data dari pihak *client* menuju ke *server*, sedangkan untuk metode *get* di *endpoint outsourcing* digunakan untuk mengirimkan data dari *server* ke pihak *client*. *Endpoint all* dan *id* menggunakan metode *get* yang mempunyai fungsi untuk mengirimkan data dari *server* ke pihak *client*, sementara *endpoint update message* dan *update status* menggunakan metode *put* yang mempunyai fungsi untuk melakukan *update* terhadap data yang di inginkan berdasarkan *id* dari data. *Endpoint* selanjutnya adalah *delete*, *endpoint* ini mempunyai fungsi untuk

menghapus data berdasarkan *id* yang dipilih.

E. Halaman Live Chat

Live chat adalah layanan pesan secara *online* dimana pengguna bisa mengajukan pertanyaan atau komplain terkait layanan yang disediakan oleh perusahaan. Dengan adanya *live chat* pengguna tidak harus menunggu lama untuk mendapatkan balasan karena pertanyaan diajukan secara *real time*.

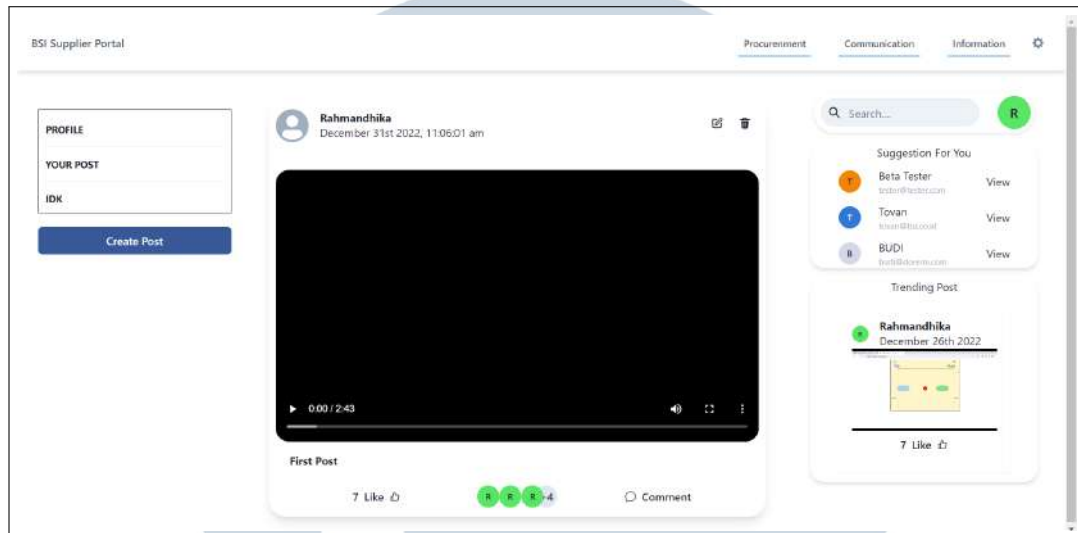


Gambar 3.27. Halaman Live Chat

Berdasarkan Gambar 3.27, terdapat sebuah *chat box* yang berfungsi untuk mengirim pesan sesama pengguna, pengguna bisa melihat pengguna lainnya yang sedang *online*. Terdapat juga *chat room* yang disediakan seperti *room apple*, *banana*, dan *melon*. Fitur *live chat* dibuat menggunakan bantuan dari *library Socket.io*, *Socket.io* sendiri adalah *real time engine* yang dibuat di atas *NodeJs*, dengan *Socket.io* pengguna bisa berkomunikasi secara *real time* yang berbasis *event*

Socket.io bekerja dengan cara yang cukup unik yaitu, *client* akan mengirimkan *request* ke *server* kemudian pihak *server* akan menerima *request* tersebut dan *request* yang diterima akan dikirimkan kembali ke *client*, biasanya *request* dan *response* tersebut akan disimpan didalam *database* sehingga ketika pengguna masuk ke halaman *live chat*, pengguna bisa melihat *history chat* yang telah dilakukan.

F. Halaman Social Media



Gambar 3.28. Social Media

Pada Gambar 3.28, terdapat sebuah halaman dari fitur *social media*, halaman ini merupakan halaman utama ketika tombol dari *social media* di klik pada halaman *home*. Di halaman ini pengguna dapat membuat *post* dan melakukan *upload post* mereka, terdapat juga tombol *like*, *comment*, serta tombol *delete*. Sebelum pengguna bisa membuat *post*, pengguna harus membuat profil terlebih dahulu, tombol untuk membuat profil terletak di sebelah kanan atas dibagian *search bar*.

Social Media Profile



Gambar 3.29. Schema Social Media Profile

Berdasarkan Gambar 3.29 terdapat data bertipe json yang dimiliki oleh *schema social media profile*, sosial media mempunyai dua *schema / model* yaitu *social media profile* dan *social media post*. *Social media profile* memiliki field

yang terdiri dari *id*, *username*, *fullname*, *picture*, *bio*, *posts*, *updated at* dan *created at*, *social media profile* memiliki *endpoint* yang terdiri dari *profile*, *all profile*, dan *profile entity*. *Endpoint profile* menggunakan dua metode yang berbeda metode pertama adalah *post*, metode ini berfungsi untuk mengirimkan data dari *client* ke *server*, sedangkan metode *get* digunakan untuk mengirim data dari *server* ke *client*.

Social Media Post

```

{
  "_id": "63b99ef05aab54527691dcc1",
  "Username": "rahmandhika5@gmail.com",
  "Content": "<p><strong>I Live In tangerang 2023.</strong></p>",
  "Documents": "https://bsi-portal-service-production.up.railway.app/images/1673109232458.png",
  "Author": {
    "_id": "63b99ed25aab54527691dcbc",
    "Username": "rahmandhika5@gmail.com",
    "FullName": "Rahmandhika",
    "ProfilePicture": "https://bsi-portal-service-production.up.railway.app/images/1673109202509.jpg",
    "Bio": "Tangerang Number One",
    "Posts": [],
    "createdAt": "2023-01-07T16:33:22.512Z",
    "updatedAt": "2023-01-07T16:33:22.512Z",
    "__v": 0
  },
  "Likes": [
    {
      "_id": "63b99ed25aab54527691dcbc",
      "Username": "rahmandhika5@gmail.com",
      "FullName": "Rahmandhika",
      "ProfilePicture": "https://bsi-portal-service-production.up.railway.app/images/1673109202509.jpg",
      "Bio": "Tangerang Number One",
      "Posts": [],
      "createdAt": "2023-01-07T16:33:22.512Z",
      "updatedAt": "2023-01-07T16:33:22.512Z",
      "__v": 0
    }
  ],
  "Comments": [
    {
      "PostedBy": "63b99ed25aab54527691dcbc",
      "Text": "Wow Cool Post",
      "_id": "63b99f065aab54527691dccc"
    }
  ],
  "createdAt": "2023-01-07T16:33:52.462Z",
  "updatedAt": "2023-01-07T16:34:14.190Z",
  "__v": 0
}

```

Gambar 3.30. Schema Social Media Post

Pada Gambar 3.30 merupakan gambaran data dari *schema social media post* yang memiliki *field id*, *username*, *content*, *dociments*, *author*, *likes*, *comments*, *created at* dan *updated at*. *Field author* menggunakan salah satu fitur yang disediakan oleh MongoDB yaitu *ref* dan *populate*, *populate* memiliki arti proses yang menggantikan *field* khusus di dokumen dengan *field* lainnya dari *collection*

yang berbeda. *Populate* memiliki fungsi yang sama dengan salah satu *operation* dari MySQL yaitu JOIN, perbedaannya adalah MongoDB merupakan *database* yang berbasis dokumen, sedangkan MySQL adalah *database* yang memiliki *relasi* antara setiap tabelnya.

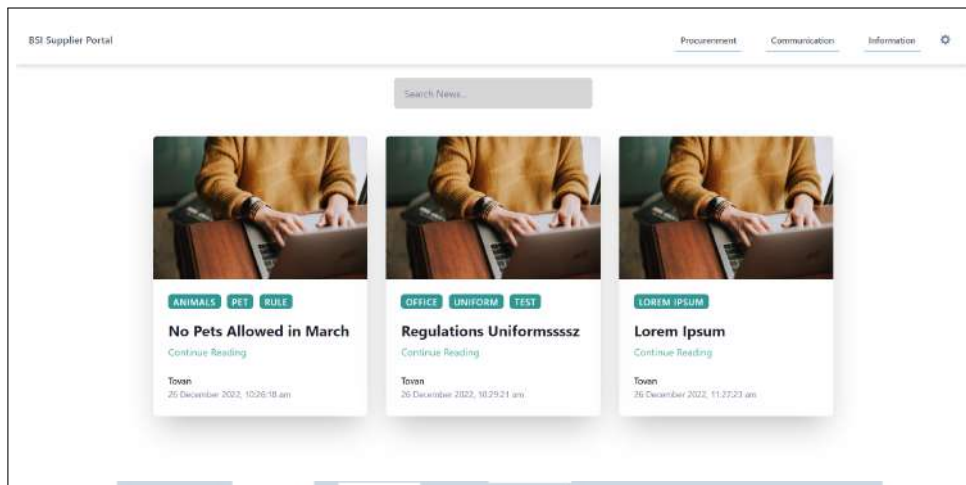
Di *schema / model social media post* penulis menggunakan *populate* untuk mem-*populate field author, likes, dan comment*, masing-masing *field* tersebut memiliki *field* yang mempunyai *value* yang sama terhadap *collection* targetnya. Sehingga penulis bisa melakukan *populate* dan mengambil data dari *collection* tujuan dan menggantikannya di *field* yang dipilih.

Schmea social media post memiliki beberapa *endpoint*, seperti *post, all, get by id, delete, like, unlike* dan *comment*. *Endpoint post* menggunakan metode *post* yang mempunyai fungsi untuk mengirimkan data dari *client* ke *server* kemudian *endpoint all* digunakan untuk mengambil dan mengirimkan data dari *server* ke *client* menggunakan metode *get*. *Endpoint get by id* mempunyai fungsi untuk mencari *post* berdasarkan *id* dari *post* tersebut, proses ini menggunakan metode *get*. *Endpoint delete* mempunyai fungsi untuk melakukan *delete* terhadap data yang ingin di *delete* berdasarkan *id* dari *post* yang dibuat, juga *endpoint delete* menggunakan metode *delete*. *Endpoint like* dan *unlike*, *endpoint* ini bertujuan untuk melakukan *like* dan *dislike* terhadap *post* yang diinginkan, proses *like* dan *dislike* akan mencari *social media post* berdasarkan *id*, ketika *post* ditemukan dan tombol *like* di klik *id* dari pengguna akan di tambahkan dan di *update* pada *field Likes*, sedangkan jika tombol *unlike* di klik maka *id* pengguna akan di hapus dari *field likes*.

Populate dilakukan pada *endpoint all* dan *id*, *endpoint* tersebut menggunakan metode *get*. Proses yang terjadi pada *endpoint* tersebut adalah *model social post* akan mencari berdasarkan *id* untuk *endpoint id*, ketika *id* berhasil ditemukan proses selanjutnya adalah melakukan *populate* terhadap *field author, comment* dan *likes*, sedangkan untuk *endpoint all model* akan mencari semua *social media post* setelah itu akan dilakukan *populate*.

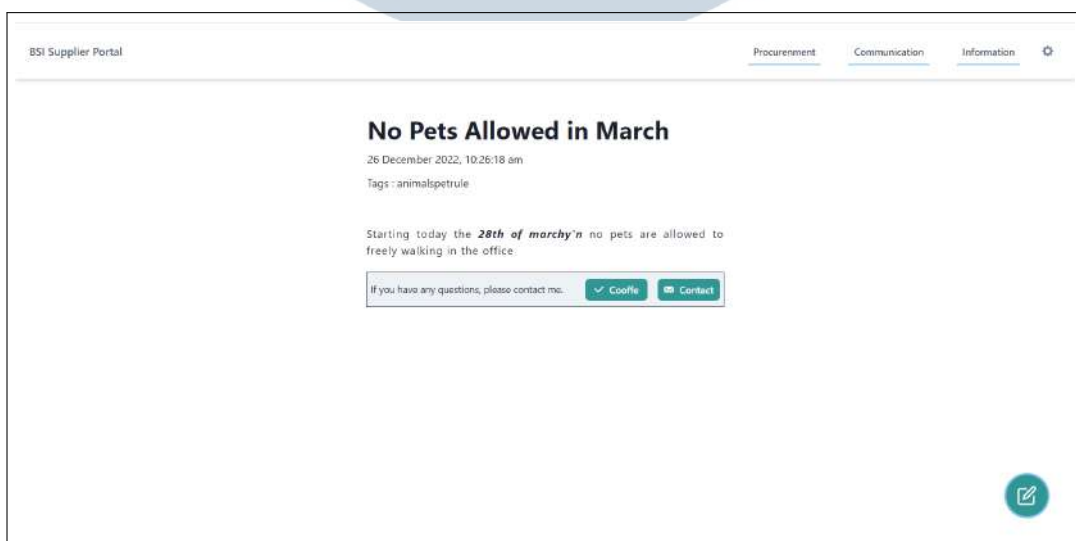
G. Halaman News and Announcement

News and announcement merupakan fitur atau halaman yang mempunyai fungsi seperti blog. Hanya *admin* yang dapat membuat, menghapus, dan mengubah *post*. Sedangkan pengguna bisa melihat *news and announcement* di halaman *home news and announcement*..



Gambar 3.31. Halaman News and Announcement

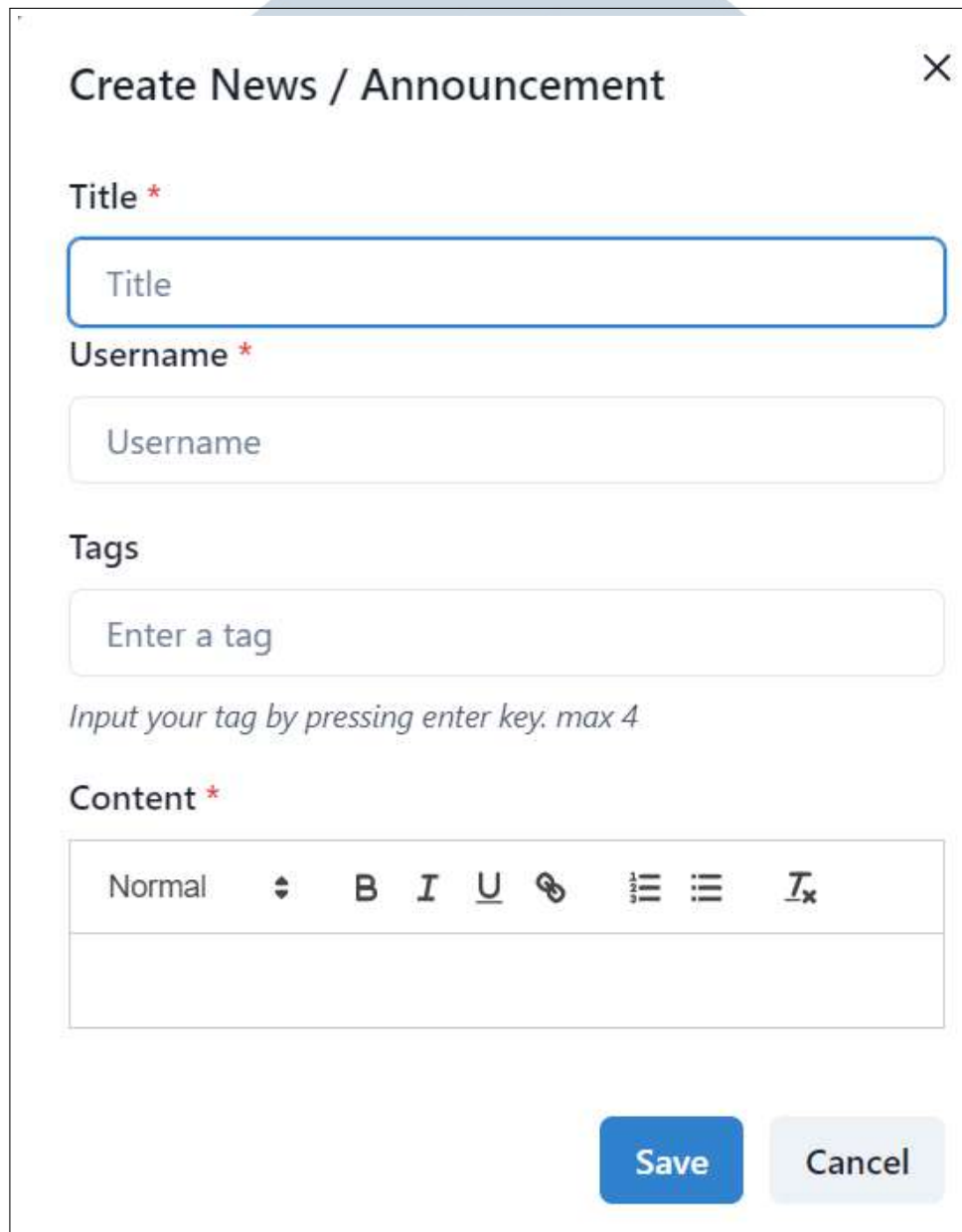
Berdasarkan Gambar 3.31, merupakan tampilan awal dari halaman *news and announcement*, ketika pengguna meng klik tombol *card* di halaman *home*, pengguna akan di arahkan langsung ke halaman *news and announcement*. Di halaman ini pengguna bisa melihat semua *news* yang dibuat seperti *gallery*.



Gambar 3.32. Halaman News and Announcement Details

Pada Gambar 3.31, merupakan tampilan yang disajikan ketika pengguna meng klik tombol *continue reading* di halaman sebelumnya. Pengguna akan diarahkan ke halaman *news and announcement* detail, dimana pengguna bisa melihat detail dari *news* tersebut secara keseluruhan. Terdapat juga tombol *edit news* di sebelah kanan bawah, jika tombol tersebut di klik akan ada *popup* yang muncul,

popup tersebut digunakan untuk mengubah konten *news*. Juga ketika pengguna masuk sebagai *admin* dan mengklik tombol *edit* akan muncul tombol *delete* yang terletak di bagian atas judul *news*.



Create News / Announcement ×




Title *

Username *

Tags

Input your tag by pressing enter key. max 4

Content *

Normal ↕ **B** *I* U    T_x

Save **Cancel**

Gambar 3.33. Halaman Input News

Pada Gambar 3.33 menampilkan tampilan modal dari *input news* pada halaman *admin*, fitur ini hanya bisa digunakan jika pengguna masuk sebagai *admin*. Ada beberapa *field* yang wajib diisi oleh *admin* sebelum bisa melakukan *save*, diantaranya terdapat *field title* dari *news*, *username admin*, *tags*, dan *content*

News and Announcement

```
{
  "_id": "63b9a03e5aab54527691dce0",
  "Email": "rahmandhika5@gmail.com",
  "Username": "Rahmandhika",
  "Title": "Regulasi Uniform",
  "Tags": [
    "Rules",
    "Uniform"
  ],
  "Content": "<p>Starting<strong> Monday </strong>employee not allowed to use white uniform.</p>",
  "createdAt": "2023-01-07T16:39:26.110Z",
  "updatedAt": "2023-01-07T16:39:26.110Z",
  "__v": 0
}
```

Gambar 3.34. Schema News

Berdasarkan Gambar 3.34 menampilkan data bertipe *json* pada *schema news*, *schema news* memiliki beberapa *field* yang terdiri dari *field id*, *email*, *username*, *title*, *tags*, *content*, *created at* dan *updated at*. Masing-masing dari *field* tersebut memiliki tipe yang berbeda-beda, terdapat tipe *string*, *object id*, *array* dan *time*. *Schema / model news* memiliki beberapa *endpoint* diantaranya adalah *create news*, *all news*, *id*, *update* dan *delete*, *endpoint create news* digunakan untuk mengirimkan data ke *server* menggunakan metode *post*, data yang berhasil dikirim ke *server* bisa dikirimkan kembali ke pihak *client* menggunakan *endpoint all news*. *Endpoint all news* berfungsi untuk mengambil semua data yang berada di *server* dan data tersebut akan dikirim ke pihak *client* untuk ditampilkan, *endpoint all news* menggunakan metode *get*. Untuk *endpoint id* digunakan untuk mencari *news* dengan *id* yang sama, *id* tersebut dikirimkan dari *client* ke *server* dan akan dicocokkan, jika cocok maka data yang sama akan dikirimkan ke *client*. Kemudian *endpoint update* digunakan untuk melakukan *update* pada data yang ingin diubah berdasarkan *id* data tersebut, *endpoint update* menggunakan metode *get* yang berfungsi untuk mengambil data dari *server*. Terakhir adalah *endpoint delete*, *endpoint* ini berfungsi untuk menghapus data yang tidak diinginkan, data dihapus berdasarkan *id* dari data tersebut, jika *id* yang dikirimkan ke *server* sama maka fungsi *delete* akan dijalankan.

3.4.3 Pengujian

Pengujian dilakukan oleh *supervisor* dan beberapa anggota divisi lainnya dengan melakukan pengecekan terhadap setiap fitur untuk memastikan apakah *input*

yang dimasukan sudah sesuai dan apakah *output* yang dihasilkan sudah sesuai. Jika tidak ada kendala dalam pengecekan maka aplikasi bekerja dengan seharusnya dan sudah dapat digunakan, namun jika terdapat masalah pada salah satu fitur maka akan dilakukan perbaikan sampai masalah tersebut selesai.

3.5 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan pada saat melakukan kerja magang di PT. Berlian Sistem Informasi yaitu mengenai *database*, proyek yang dibuat menggunakan *database* MongoDB sedangkan pihak perusahaan masih menggunakan *database* dari produk microsoft. Pihak perusahaan enggan jika data proyek situs ini tidak ada di *database* utama perusahaan, pihak perusahaan ingin setiap data yang dibuat masuk ke dalam *database* utama perusahaan.

Solusi yang ditemukan yaitu dengan menggunakan produk dari microsoft yaitu power automate, *database* yang digunakan yaitu MongoDB yang mempunyai fitur *MongoDB cloud* dan penulis menghubungkannya ke API sehingga data situs yang tersimpan di *database* terhubung dengan API yang bisa diakses secara *online*. Dengan menggunakan power automate penulis membuat *flow* yang akan mengambil data dari API sehingga data tersebut bisa disimpan di *database* utama perusahaan.

