

Real-Time BPMN Website Menggunakan Teknologi MERN Stack

Ricky Handoyo¹, Leo Willyanto Santoso², Alexander Setiawan³
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658

Email: hauwricky@gmail.com¹, leow@petra.ac.id², alexander@petra.ac.id³

ABSTRAK

Pada masa sekarang, teknologi merupakan suatu alat yang dikembangkan untuk membantu memudahkan aktivitas manusia. Pembuatan *diagram* bpmn yang kompleks dan banyak tidak mungkin dikerjakan oleh satu atau dua orang saja dan juga adanya orang yang melakukan pemeriksaan terhadap *diagram* yang telah dibuat. Sistem yang sekarang dilakukan dengan memanfaatkan *sharing file* dimana *file* BPMN yang telah dikerjakan dikirim melalui *email*. Oleh karena itu diperlukannya suatu sistem dimana suatu *project* dapat diakses oleh beberapa pengguna secara langsung.

Pembuatan website real-time bpmn dirancang dengan memanfaatkan MERN *stack technology* sebagai *single page application* dan *library javascript* yang dimanfaatkan dalam pembuatan website terutama *library bpmn* untuk pembuatan *diagram* dan *socket* untuk komunikasi dua arah secara *real-time* antara *user* dengan sistem.

Hasil akhir dari skripsi adalah *website real-time bpmn* digunakan untuk membantu user melakukan pembuatan *diagram bpmn* yang dapat dilakukan oleh beberapa *user*. Adanya fitur-fitur seperti *sharing system*, *real-time diagram*, dan *real-time chat* yang dibuat untuk *user* dapat berbagi *project* dan membuat *diagram* bersama-sama.

Kata Kunci: BPMN, MERN, *Realtime*, *Website*

ABSTRACT

At present, technology is a tool developed to help facilitate human activities. The making of complex and many bpmn diagrams is impossible to do by one or two people and also there are people who do an examination of the diagram that has been made. The current system is done by utilizing file sharing where BPMN files that have been done are sent via email. Therefore we need a system where a project can be accessed by several users directly.

The making of real-time bpmn website is designed by utilizing the MERN stack technology as a single page application and javascript library which is used in making websites, especially library bpmn js for making diagrams and sockets IO for real-time two-way communication between users and system.

The final result of this paper is a real-time website bpmn is used to help users make bpmn diagrams that can be done by several users. The presence of features such as sharing systems, real-time diagrams, and real-time chat are made for users to be able to share projects and make diagrams together.

Keywords: BPMN, MERN, *Realtime*, *Website*

1. PENDAHULUAN

Pada zaman ini, teknologi informasi terus berkembang dan teknologi informasi dimanfaatkan serta dikembangkan manusia

untuk memudahkan aktivitas mereka. Beberapa contoh dari teknologi informasi dapat berupa aplikasi, *website*, *software desktop*, dan internet. Internet merupakan salah satu yang memberikan dampak bagi kemajuan teknologi informasi dimana informasi-informasi dapat diakses maupun dibagikan secara langsung melalui internet.

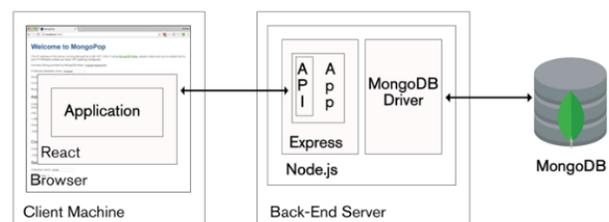
Business Process Model and Notation atau yang biasa disebut sebagai BPMN merupakan sebuah standar yang menggambarkan proses bisnis dalam suatu perusahaan yang ditampilkan dalam bentuk diagram. BPMN ini penting bagi perusahaan untuk membuat proses bisnis, karena tanpa adanya proses bisnis yang jelas akan membuat bisnis tidak memiliki konsistensi dan arah tujuan dari bisnis tersebut. Di lain sisi BPMN juga merupakan hal yang penting dalam dunia pendidikan karena pembuatan BPMN merupakan salah satu subjek yang diajarkan di universitas.

Kendala yang dihadapi sekarang adalah pembuatan BPMN dalam perusahaan dan universitas dimana BPMN yang kompleks dan banyak tidak mungkin hanya dikerjakan oleh satu atau dua orang saja. Aplikasi-aplikasi sekarang yang ada hanya memanfaatkan *sharing file* dimana BPMN yang telah dikerjakan akan disimpan di Google Drive atau komputer pribadi lalu di *share* terhadap orang lain baik melalui *email* atau *link*.

2. TINJAUAN PUSTAKA

2.1 MERN Stack (MongoDB, Express, React, Node)

Pembuatan *website* biasanya dibuat dengan beberapa teknologi menjadi satu. Kombinasi dari beberapa teknologi ini yang biasa disebut dengan “stack”. Dipopulerkan oleh LAMP stack yang merupakan akronim untuk Linux, Apache, MySQL, dan PHP yang semuanya merupakan komponen *open-source*. Sebagai *web* yang pengembangan menjadi matang dan interaktivitas muncul kedepan yaitu *Single Page Applications (SPA)* menjadi lebih populer. SPA merupakan suatu paradigma aplikasi *web* yang menghindari *refresh* pada halaman *web* untuk menampilkan konten baru. Cara yang digunakan adalah melakukan panggilan ke server untuk mendapatkan beberapa data untuk memperbarui halaman *web*. Hasilnya terlihat cukup bagus jika dibandingkan dengan cara lama yang melakukan *refresh* halaman secara keseluruhan. Hal ini membawa peningkatan dalam kerangka *front-end*. [1]



Gambar 1. Konsep MERN Stack

Sumber : (<https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack>)

Pada Gambar 1 merupakan gambar konsep dari cara kerja MERN *stack* dimana React Js akan berkomunikasi dengan melakukan request terhadap Node Js yang menggunakan framework Express. Node Js akan menghubungkan *back-end* server dengan MongoDB yang digunakan sebagai *database* lalu memberikan respon dalam bentuk JSON terhadap *front-end*.

2.2 BPMN (*Business Process Model and Notation*)

Merupakan standar dari BPMN yang memberikan bisnis kemampuan memahami prosedur bisnis *internal* dalam bentuk notasi grafis dan akan memberi kemampuan kepada perusahaan untuk mengkomunikasikan prosedur ini secara standar. Lebih jauh lagi, notasi grafis akan memfasilitasi pemahaman tentang kolaborasi kinerja dan transaksi bisnis antara perusahaan. Ini akan memastikan bahwa bisnis akan memahami dirinya sendiri dan juga peserta dalam bisnis serta untuk menyesuaikan diri dengan keadaan bisnis *internal* dan B2B yang baru dengan cepat. [2]

2.3 Node Js

Node Js merupakan *open-source, cross-platform runtime environment* yang digunakan pada sisi server dari aplikasi *website*. Node Js ditulis dalam JavaScript dan dapat dijalankan berbagai macam sistem operasi. Node Js didasarkan pada *event-driven architecture* dan *non-blocking input/output* API yang dirancang untuk mengoptimalkan keseluruhan aplikasi dan skalabilitas untuk *website* yang bersifat *realtime*. [3]

Beberapa dari dependencies pada Node Js :

- Bcrypt : *Library* yang digunakan untuk membantu dalam melakukan hash *password* ketika melakukan *insert* data ke dalam suatu *database* dan membantu melakukan proses *compare* untuk mendapatkan data kembali
- Body-parser : *Library* yang digunakan untuk melakukan *extract* terhadap *body* dari *request* yang ada dan dimasukkan kedalam *middleware* sebelum ke *handler* dan untuk mengakses melalui *req.body*
- Concurrently : *Library* yang digunakan untuk menjalankan beberapa *command* secara bersamaan
- Cors : *Library* untuk memungkinkan sumber daya terbatas pada halaman *web* untuk diminta dari domain lain di luar domain tempat resource diberikan
- Express : *Framework* yang digunakan pada Node Js
- Jsonwebtoken : Sarana untuk mewakili klaim yang akan ditransfer antara dua pihak. Jsonwebtoken dikodekan sebagai objek JSON yang ditandatangani secara digital menggunakan JSON Web Signature atau dienkripsi menggunakan JSON Web Encryption
- Mongoose : Alat yang digunakan untuk melakukan modeling pada MongoDB dan bekerja dalam lingkungan *asynchronous*
- Nodemon : Alat yang digunakan untuk membantu dalam develop Node Js dengan melakukan *restart* secara otomatis kepada Node ketika terjadi perubahan *files* yang terdeteksi
- Socket IO : *Library* yang digunakan untuk komunikasi dua arah secara *real-time*. [4]

2.4 Express

Express Js merupakan framework yang berada di atas fungsi server web Node.js untuk menyederhanakan API dan menambahkan fitur baru yang bermanfaat. Membuat lebih mudah untuk mengatur fungsionalitas aplikasi dengan *middleware* dan *routing*. Hal

tersebut menambah manfaat utilitas untuk objek HTTP Node.js, memfasilitasi rendering tampilan HTML dinamis dan mendefinisikan standar ekstensibilitas yang mudah diimplementasikan. [5]

2.5 React Js

React merupakan UI *library* yang dikembangkan oleh Facebook untuk membantu dalam pembuatan yang interaktif, stateful, dan komponen dari UI yang dapat digunakan kembali. React Js memiliki 3 fitur utama yaitu :

- *Declarative* : React membantu dalam pembuatan UI yang interaktif. Desain tampilan yang sederhana untuk setiap state dalam aplikasi, dan React akan secara efisien memperbarui dan memberikan komponen yang tepat ketika terjadi *datachanges*. Tampilan deklaratif ini membuat kode lebih mudah di debug
- *Component-based* : Membangun enkapsulasi komponen yang mengelola state mereka sendiri, lalu digunakan untuk membuat UI yang kompleks. Logika komponen ditulis dalam JavaScript sehingga memudahkan dalam pass data melalui aplikasi
- *Learn Once, Write Anywhere* : React Js tidak membuat asumsi akan stack dari teknologi lain yang digunakan sehingga dapat mengembangkan fitur dari React tanpa menulis ulang kode yang ada. React juga dapat render pada server dengan menggunakan Node dan aplikasi mobile dengan React Native [6]

Beberapa dari dependencies yang digunakan pada React Js untuk membantu dalam proses pembuatan yaitu :

- Axios : Promise berdasarkan HTTP *client* untuk *browser* dan Node Js. Beberapa fitur yang dimiliki :
 - o Membuat XMLHttpRequest melalui browser
 - o Membuat *http request* dari Node Js
 - o Support Promise API
 - o *Intercept request* dan response
 - o Mengubah data dari *request* dan *response*
 - o Membatalkan *request*
 - o Otomatis mengubah data ke JSON
- BPMN Js : *Library* yang digunakan untuk melihat atau melakukan perubahan *diagram* BPMN 2.0
- React : JavaScript *library* yang digunakan untuk membuat *user interface*
- React-dom : Bekerja sebagai *entry point* untuk DOM dan server *render* untuk React
- React-router-dom : DOM binding untuk React router
- React-scripts : *Package* yang berisikan script dan pengaturan yang digunakan dalam command *create-react-app*
- Socket IO-client : *Library* yang digunakan untuk komunikasi dua arah secara *real-time*. Socket IO client dibuat untuk *client* atau *frontend*. [4]

2.6 MongoDB

MongoDB adalah penyimpanan data yang *powerful, flexible*, dan *scalable* dalam basis data. MongoDB menggabungkan kemampuan untuk memperkecil skala dengan fitur seperti indeks sekunder, *range queries*, pengurutan, agregasi, dan indeks geospasial. MongoDB merupakan *document-oriented database*, bukan merupakan database yang relasional. Alasan utama untuk bergerak jauh dari relasional adalah untuk membuat skala lebih mudah dan meningkatkan performa. [7]

3. ANALISA DAN DESAIN

3.1 Analisis Sistem

Sistem yang dibuat untuk membantu dalam membuat diagram BPMN yang dapat dilakukan oleh beberapa user dimana user memiliki akses terhadap *diagram* baik sebagai pemilik atau sebagai user yang telah mendapatkan hak akses. Hal ini untuk mencegah terjadinya *miscommunication* antara user saat mengerjakan suatu *project* yang membutuhkan *diagram* BPMN kompleks. Beberapa fitur yaitu menyimpan data *diagram* BPMN dalam server, memberi akses terhadap user lain kedalam *project.chat room* untuk tempat berkomunikasi, dan *friend management*. Sistem akan dibuat dengan menggunakan teknologi MERN stack (MongoDB, Express, React, Node).

3.2 Analisis Kebutuhan

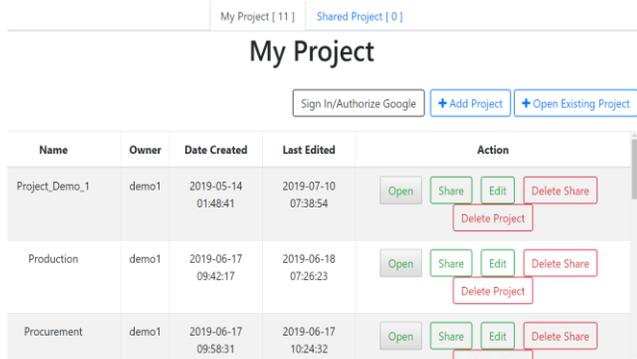
Dari analisis sistem, ada beberapa hal yang akan dibuat dalam sistem untuk membantu user dalam melakukan pembuatan diagram BPMN sebagai berikut:

1. Sistem yang dapat bekerja secara *real-time* dalam pemuatan *diagram* BPMN dan fitur *chat* yang memudahkan untuk melakukan komunikasi
2. Sistem yang dapat melakukan penyimpanan data, gambar dalam server dan melakukan *print*.
3. Sistem yang dapat membuat *user* melakukan *friend management* dan proses *share* terhadap *user* lain.
4. Sistem yang dapat menunjukkan kepada *user project* yang dimiliki dan juga *project* yang telah di *share*

4. HASIL DAN PEMBAHASAN

4.1 Membuat Project

Dalam studi kasus *user* dengan nama *demo1* akan membuat *project* baru. Pengujian dimulai setelah *demo1* berhasil melakukan *login* dan telah membaca *guide* yang ada, *demo1* akan memulai membuat *project* dengan pergi ke halaman *my project*. Halaman *my project* dapat dilihat pada Gambar 2.



Gambar 2. Halaman *my project*

Setelah *demo1* berada pada halaman *my project*, *demo1* membuat *project* dengan memilih membuat *project* baru dan akan keluar *form* untuk mengisi nama dari *project* yang akan dibuat. *Demo1* dapat melihat *project* yang telah dibuat. *Project* baru yang telah dibuat akan ditampilkan pada halaman *my project*.

Setelah *demo1* telah berhasil membuat *project* baru dan *demo1* masuk ke dalam *project* untuk membuat *diagram* yang akan digunakan. *Demo1* memilih *open* dan *demo1* diarahkan ke halaman *workspace* sebagai tempat untuk pembuatan *diagram*.

Pada halaman *workspace* *demo1* dapat menggunakan beberapa fitur yang telah disediakan untuk membantu dalam pekerjaan seperti simbol bpmn, menyimpan *file* ke komputer pribadi,

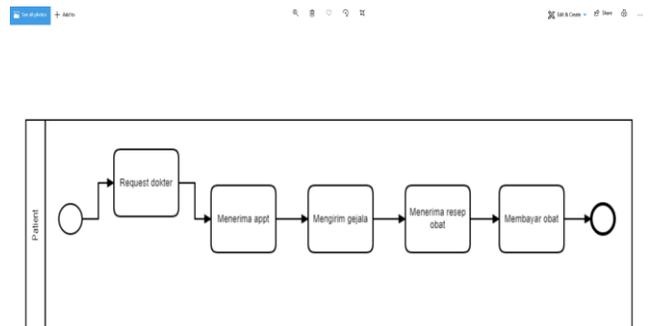
menyimpan gambar, melakukan *print*, menyimpan *file* ke Google Drive. Dalam studi kasus user akan memulai membuat *diagram* bpmn kemudian menyimpan *file* dan gambar ke komputer pribadi serta melakukan *print*.

Demo1 melakukan pembuatan *diagram* bpmn dengan memasukkan simbol-simbol yang ada. Setelah *demo1* melakukan pembuatan *diagram*, *demo1* ingin melakukan penyimpanan *file diagram* untuk digunakan kembali sebagai cadangan jika *file* tidak sengaja terhapus. Tampilan hasil penyimpanan dalam bentuk xml dapat dilihat pada Gambar 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:bpmn2:
<bpmn2:collaboration id="Collaboration_0c9hvx8" processRef="Process_1" />
</bpmn2:collaboration>
<bpmn2:process id="Process_1" isExecutable="false">
<bpmn2:startEvent id="StartEvent_1" />
<bpmn2:task id="Task_11cqxm" />
</bpmn2:process>
</bpmn2:definitions>
<bpmndi:BPMNDiagram id="BPMNDiagram_1">
<bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="Collaboration_0c9hvx8">
<bpmndi:BPMNShape id="Participant_0cyhvx8_di" bpmnElement="Participant_0cyh
<dc:Bounds x="130" y="220" width="600" height="250" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="BPMNShape_StartEvent_2" bpmnElement="StartEvent_1">
<dc:Bounds x="214" y="308" width="36" height="36" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Task_11cqxm_di" bpmnElement="Task_11cqxm">
<dc:Bounds x="413" y="286" width="100" height="80" />
</bpmndi:BPMNShape>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
```

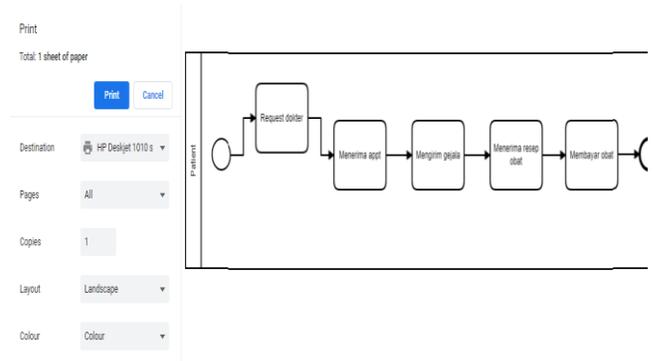
Gambar 3. Hasil penyimpanan *file*

Demo1 ingin melakukan penyimpanan dalam bentuk gambar dengan *format* .PNG. Tampilan hasil penyimpanan gambar dengan *format* .PNG dapat dilihat pada Gambar 4.



Gambar 4. Hasil penyimpanan gambar .PNG

Demo1 ingin melakukan *print diagram* dalam bentuk gambar secara langsung ke *printer*. *Demo1* akan melakukan pengaturan terlebih dahulu terhadap gambar yang ingin di *print*. Tampilan hasil *print* dapat dilihat pada Gambar 5.



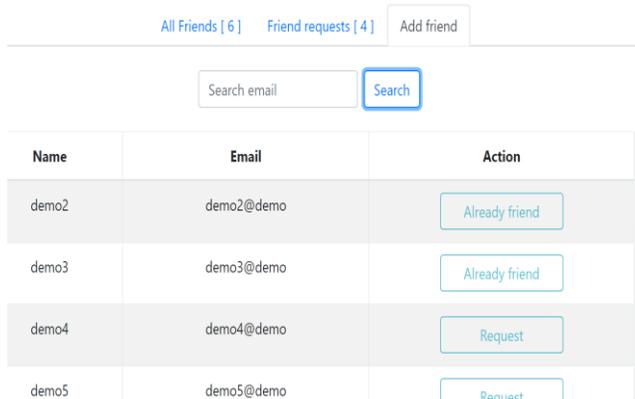
Gambar 5. Pengaturan *print*

Demo1 akan melakukan penyimpanan *file* ke Google Drive sebagai *backup* kedua jika terjadi sesuatu terhadap komputer dari demo1. Demo1 melakukan *login* terlebih dahulu ke Google Drive dan *user* mendapatkan permintaan oleh *website* agar dapat mengakses untuk melakukan *upload* dari *website*. Setelah berhasil memberikan akses, demo1 melakukan *upload file* dan *file* tersimpan dalam Google Drive.

4.2 Membagikan Project

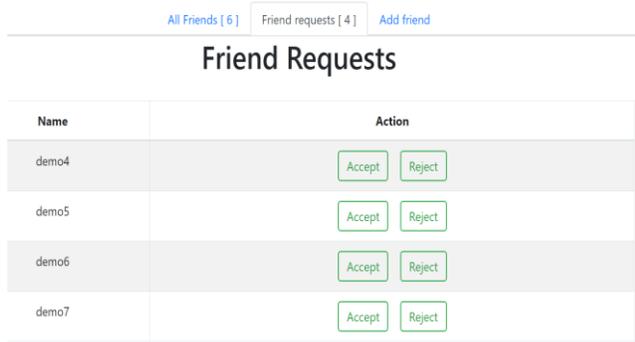
Dalam studi kasus ini setelah demo1 selesai membuat *project* dan melakukan penyimpanan, demo1 ingin membagikan *project* yang telah dibuat untuk diperiksa oleh demo2 yang telah memiliki akun dalam *website*.

Demo1 telah membaca *guide* yang telah disediakan dan untuk membagikan *project* ke demo2 maka perlu dilakukan penambahan teman terlebih dahulu. Demo1 pergi ke halaman *friends* dan pergi ke bagian *tab add friend* untuk menambah teman. Kemudian, demo1 melakukan pencarian terhadap demo2 yang akan di *share project* dan telah mendaftar pada *website* ini. Tampilan untuk penambahan teman yang ditunjukkan pada demo1 dapat dilihat pada Gambar 6.



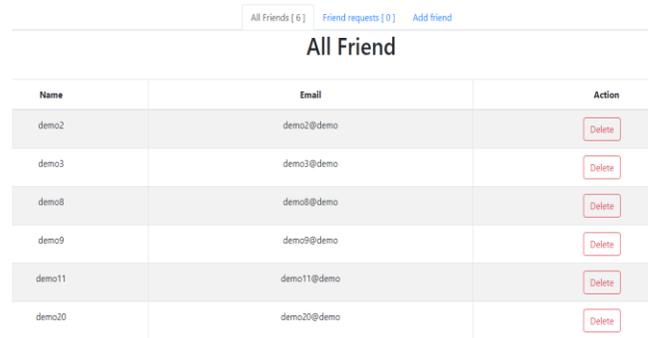
Gambar 6. Halaman *friends* tab *add friend* demo1

Setelah melakukan penambahan teman, demo1 menunggu demo2 untuk melakukan penerimaan terhadap *friend request* yang ada. Tampilan untuk melakukan proses penerimaan atau penolakan teman yang ditunjukkan pada demo2 dapat dilihat pada Gambar 7.



Gambar 7. Tampilan *friend request* demo2

Demo2 melakukan penerimaan *friend request* yang ada pada tab *friend request*. Kemudian, setelah dilakukan penerimaan *user* melakukan pengecekan pada *tab all friend* apakah demo2 sudah menerima *friend request* atau belum. Tampilan list teman yang dimiliki demo1 dapat dilihat pada Gambar 8.



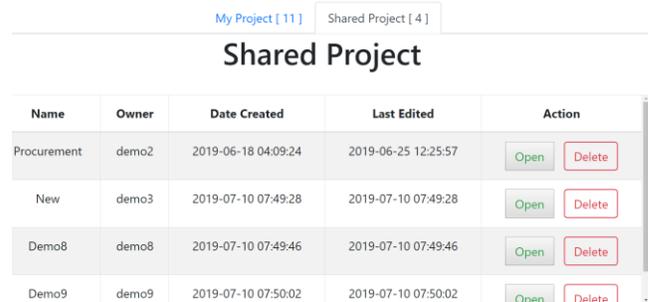
Gambar 8. Halaman *friends* tab *all friend* demo1

Untuk membagikan *project*, demo1 kembali ke halaman *my project* dan memilih *project* yang akan dibagikan terhadap demo2. *User* memilih *share* dan memilih demo2 untuk di *share*. Tampilan untuk melakukan *share* dari demo1 untuk demo2 dapat dilihat pada Gambar 9



Gambar 9. Tampilan *share project* demo1

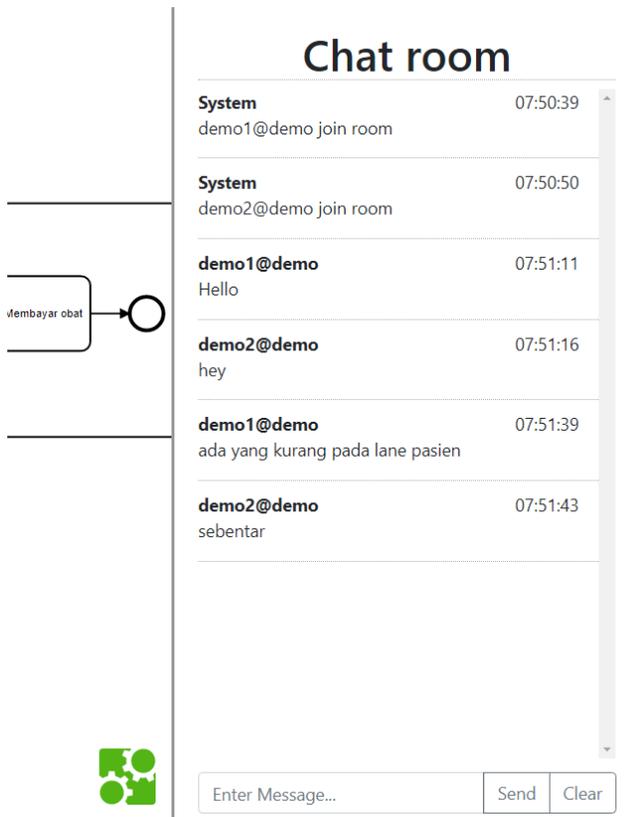
Demo2 memperoleh *project* yang di *share* dan dapat membuka *project* tersebut untuk melakukan pemeriksaan. Tampilan dapat dilihat pada Gambar 11.



Gambar 10. Tampilan *shared project* demo2

4.3 Real-Time Chat

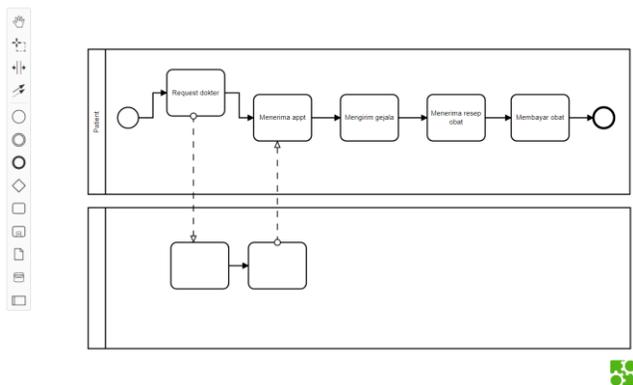
Sistem chat bersifat *live* dimana komunikasi yang terjadi tidak akan tersimpan di dalam *database* dan *user* yang baru masuk ke dalam *project* tidak akan dapat melihat *chat* yang telah terjadi sebelumnya. Studi kasus dimulai pada saat demo1 sebagai *user* yang pertama kali berada dalam *project* dan akan mendapatkan notifikasi pada tempat *chat* jika ada *user* lain yang masuk. Demo2 masuk ke dalam *project*. Tampilan yang ditunjukkan pada demo1 seperti pada Gambar 11.



Gambar 11. Pengujian chat demo1

4.4 Real-Time Pembuatan Diagram

Pengujian sistem real-time dalam pembuatan *diagram* dimana jika ada salah satu *user* yang melakukan perubahan, penghapusan, atau penambahan dalam *diagram* maka data tersebut akan dikirim ke semua *user* lain yang berada dalam project. *User* yang baru masuk ke dalam *project* akan mendapatkan data yang terbaru. Studi kasus dimulai pada saat demo1 mulai memasukkan simbol-simbol *diagram*. Tampilan dari demo1 dapat dilihat pada Gambar 12. Demo2 akan mendapatkan penambahan simbol yang dilakukan.



Gambar 12. Pengujian diagram demo1

4.5 Pengujian Teknologi MERN Stack

Pada bagian ini akan menjelaskan kenapa teknologi MERN Stack ini cocok digunakan untuk pembuatan tools yang berhubungan dengan sistem real-time. Teknologi yang digunakan dalam pembuatan website adalah MERN yang terdiri dari MongoDB, Express, React Js, dan Node Js. MongoDB sebagai database yang

berbentuk dokumen dan cocok untuk perubahan data yang terus-menerus. MongoDB compatible dengan Node Js dimana Node Js memanfaatkan WebSocket sehingga dapat melakukan perubahan data secara terus-menerus dan real-time terhadap database. React Js sebagai front-end memiliki fitur yang digunakan dalam pembuatan website yaitu state dimana saat terjadi perubahan data terus-menerus yang didapat dari back-end, halaman website tidak perlu melakukan load ulang tampilan keseluruhan dan hanya melakukan perubahan pada state. Fitur ini cocok digunakan dalam pembuatan sistem real-time.

Teknologi MERN Stack direkomendasikan untuk pembuatan sistem yang bersifat realtime seperti chat, game, skor pertandingan, penggambaran diagram. Kekurangan dari MERN Stack tidak cocok digunakan untuk pembuatan database yang memerlukan relasi kompleks dan tidak cocok digunakan jika di host di beberapa server.

5. KESIMPULAN

Dari hasil perancangan, pembuatan, dan pengujian *Real-Time BPMN Website* Menggunakan Teknologi *MERN Stack*, dapat diambil kesimpulan antara lain:

- Aplikasi *website* dibuat dengan memanfaatkan teknologi MERN Stack. Teknologi MERN stack membantu dalam pembuatan *website* yang realtime karena *website* dapat melakukan perubahan data tanpa memuat ulang halaman *website* dan adanya *library* WebSocket yang *compatible* sehingga memudahkan untuk membuat komunikasi secara *realtime* antara *client* dan *server*.
- Aplikasi *website* menerapkan sistem *realtime* yang membantu user dalam pembuatan *diagram* dan komunikasi yang dilakukan secara langsung. Pembuatan *diagram* dapat dilakukan oleh beberapa *user* secara bersamaan dan juga *realtime chat* untuk mengurangi terjadinya miskomunikasi antar *user*. Sistem *realtime* dibuat dengan memanfaatkan *library* Socket IO
- Aplikasi *website* dapat melakukan penyimpanan *diagram* sesuai dengan *format* ke *local storage* dan Google Drive sehingga dapat digunakan kembali dan penyimpanan gambar dengan *format* .PNG, .JPG, .JPEG ke *local storage*
- Aplikasi *website* memiliki fitur *friend management* yang digunakan untuk mengatur *list* dari teman yang dimiliki oleh *user* dan merupakan fitur yang diperlukan dalam proses berbagi *project* dengan *user* lain
- Aplikasi *website* memiliki fitur *project management* yang dapat digunakan untuk mengatur *project* yang dimiliki atau telah di *share* terhadap *user*. Merupakan fitur yang diperlukan untuk melakukan pembuatan *diagram*
- Aplikasi *website* dapat melakukan *print* ke *printer* dalam bentuk gambar
- Berdasarkan hasil kuisioner untuk penilaian terhadap aplikasi secara keseluruhan dengan nilai standar 3 (cukup) sebagai patokan dimana aplikasi dapat berjalan dan memenuhi kebutuhan dalam pembuatan *diagram*. Nilai terhadap aplikasi secara keseluruhan adalah 30% cukup, 50% baik, dan 20% sangat baik. Melalui hasil survei dapat dikatakan bahwa aplikasi dapat berjalan dengan baik dan sudah memenuhi kebutuhan dalam pembuatan *diagram*

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut untuk aplikasi ini adalah sebagai berikut:

- Dapat menggunakan *techonology stack* lainnya seperti MEAN (MongoDB, Express, Angular Js, dan Node Js)

- Membuat versi *mobile* dari aplikasi baik *android* atau *iOS*
- Penambahan beberapa fitur seperti *forum* dan juga fitur untuk membagikan *project* yang dapat digunakan oleh *user* lain

6. DAFTAR REFERENSI

- [7] Bradshaw, S, & Chodorow, K. 2019. “*MongoDB: The Definitive Guide : Powerful and Scalable Data Storage*”. California: O’Reilly Media, Inc.
- [6] Facebook, Inc. 2018. React Js. Retrieved March 1,2019, from <https://reactjs.org/>.
- [5] Hahn, E.M. 2016. “*Express in Action : Writing, buildings, and testing Node.js applications*”. United States : Manning Publications.
- [4] Npm, Inc. 2019. Npm. Retrieved March 1, 2019, from <https://www.npmjs.com/>.
- [2] Object Management Group, Inc. 2018. Charter. Retrived December 3, 2018, from <http://www.bpmn.org>.
- [3] Rungta, K. 2016. “*Learn NodeJs in 1 Day : Complete NodeJs Guide with Examples*”. Independently Published
- [1] Subramanian, V. 2017. “*Pro Mern Stack : Full Stack Web App Development with Mongo, Express, React, and Node*”, India