

Penggunaan *Event-Source* Pada Pengiriman Notifikasi Donasi Siaran Langsung

Albert Abimanyu, Hapnes Toba

SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Prof. drg. Surya Sumantri, M.P.H. No. 65, Bandung 40164

¹albert.abimanyu@gmail.com

²hapnestoba@it.maranatha.edu

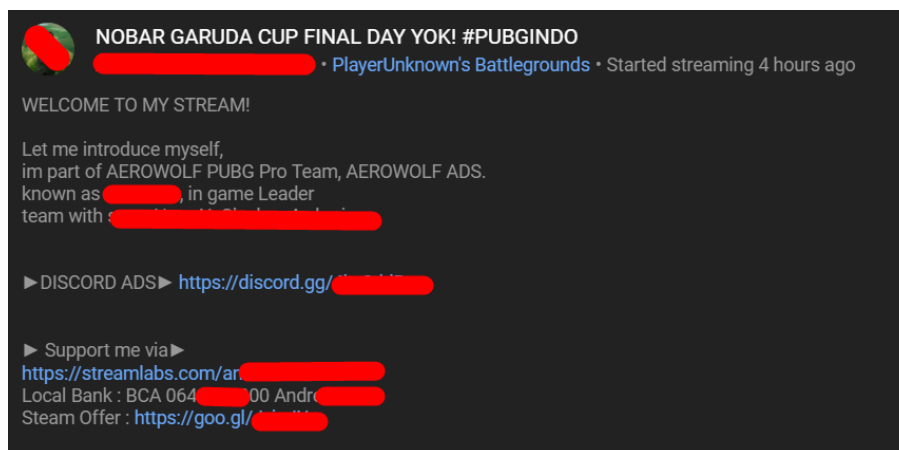
Abstract— *An internet live stream usually considered as a source of popularity and income generator. Some internet personality, sometimes also known as streamer, uses live streaming platforms to meet their daily needs. They depend on donations from viewers and ads revenue or sponsorships. Streamer in Indonesia usually uses their own bank account number as a means of receiving donations. As a means of thanking the viewers who donate, the streamer usually shows the sender name and amount of donation received in their stream description. Here, the author proposes a system that use browser source feature to be used in Open Broadcaster Software to show donation alerts on streamer's live stream. A system that only uses Indonesian Rupiah to make streamers easier to withdraw the donations received compared to another means such as via Paypal where streamer need to exchange Unites States Dollar (USD) to Rupiah. Sending out donation data in a timely manner is an important point for the information to be used in broadcaster's live stream. So a feature in a donation management system need to be able to send the donation notification data with a fast method such as event-source.*

Keywords— *Donation, Event Source, Live Stream, Notification*

I. PENDAHULUAN

A. Latar Belakang

Penyiar daring atau yang biasa disebut *Online Streamer* atau *Live Streamer* atau *YouTuber* belakangan ini sedang berkembang dengan cukup pesat di Indonesia. Hal ini dapat dilihat dari tren orang-orang yang menjadi terkenal dari konten yang dapat ditemukan di internet. *Streamer* biasanya menerima donasi saat melakukan siaran langsungnya. Yang sering ditemukan adalah penyiar mencantumkan nomor rekening bank pribadi untuk menerima donasi. Hal tersebut merupakan hal yang cukup berbahaya karena nomor rekening dapat digunakan untuk hal yang tidak diinginkan seperti penipuan dan pencemaran nama baik bagi pemilik rekening. Hal ini dapat dilihat misalnya pada Gambar 1 berikut.



Gambar 1. Tangkapan Layar Contoh Deskripsi Siaran Langsung

Pada tangkapan layar tersebut dapat dilihat bahwa penyiar tersebut menggunakan *platform* StreamLabs yang hanya menerima mata uang yang diterima oleh Paypal di mana mata uang Rupiah tidak termasuk di dalamnya. Dana yang diterima oleh penyiar biasanya digunakan untuk meningkatkan kualitas siaran, membeli perangkat komputer yang lebih baik,

ataupun untuk kebutuhan harian dari peniar tersebut. *Platform* yang biasa digunakan oleh peniar untuk melakukan siaran langsung diantaranya adalah YouTube, Twitch.tv dan lain sebagainya. Namun, platform-platform tersebut belum dapat menerima mata uang rupiah sebagai mata uang donasinya. Penulis berharap dengan dikembangkannya sistem ini, peniar dapat mulai menerima donasi dengan mata uang rupiah dan dengan cara yang lebih aman. Sistem ini dikembangkan dengan teknologi *Server Sent Event* atau *EventSource*.

B. Ruang Lingkup

Berikut ini adalah ruang lingkup kajian dalam pembuatan sistem ini:

1. Sistem hanya menerima deposit dan penarikan dana dalam bentuk mata uang Indonesia, yaitu Rupiah.
2. Sistem menggunakan HTTP API *Server Sent Event* (SSE) untuk mengirimkan data notifikasi secara cepat.

II. LANDASAN TEORI

A. JavaScript / JQuery

JavaScript merupakan sebuah bahasa pemrograman yang memungkinkan pengembang untuk mengembangkan fitur HTML dengan animasi, interaksi, dan efek visual dinamis. JavaScript dapat membuat halaman web lebih bermakna dengan menyediakan timbal balik instan. Sebagai contoh, halaman belanja yang dikembangkan menggunakan JavaScript dapat menampilkan jumlah harga secara langsung ketika pengguna memilih barang yang ingin dibeli. JavaScript juga dapat menampilkan pesan kesalahan secara cepat setelah seseorang mencoba untuk mengirimkan formulir web yang memiliki kekurangan informasi yang diwajibkan [1].

JQuery merupakan sebuah kamus JavaScript yang ditujukan untuk membuat pemrograman JavaScript lebih mudah dan menyenangkan. Sebuah kamus JavaScript adalah kumpulan yang kompleks dari kode JavaScript yang memudahkan tugas yang sulit dan menyelesaikan masalah peramban web yang berbeda-beda [1].

B. Server-Sent Event / EventSource

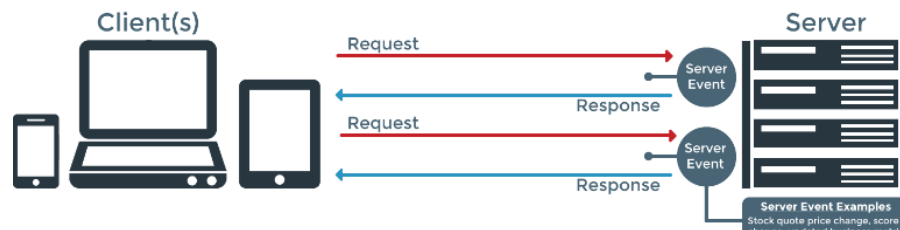
Dalam sebuah situs web, seperti www.google.com/finance, pengunjung dapat melihat bahwa isi dari halaman web tersebut terjadi pembaharuan data secara reguler. Pada situs www.bbc.co.uk/news juga memiliki fitur yang kurang lebih sama. Dalam contoh-contoh tersebut, situs di atas menggunakan teknik yang disebut *polling*. Secara periodik, halaman melakukan pengecekan ke *server* web guna memperoleh data yang baru.

Untuk mengimplementasikan desain seperti ini, pengembang dapat menggunakan *setInterval()* atau *setTimeout()* dari JavaScript yang menjalankan kode program setelah durasi yang ditentukan. *Polling* merupakan solusi yang masuk akal, namun terkadang tidak efisien. Dalam banyak kasus, menggunakan kode di atas berarti melakukan pemanggilan *server* web dan membuat koneksi baru yang terkadang hanya mendapati tidak adanya data baru. Kalikan prosedur tersebut dengan ribuan orang yang menggunakan aplikasi tersebut dalam satu waktu dan hal tersebut dapat membuat beban pada *server* web yang tidak berguna. Salah satu solusi yang dapat digunakan adalah *Server-Sent Events*, dengan tujuan sebuah halaman web mempertahankan koneksi ke *server* web. *Server* web dapat mengirimkan data baru kapan saja, dan tidak lagi membutuhkan secara terus menerus untuk memutuskan koneksi, lalu membuka koneksi baru, menjalankan kode program dari awal kecuali dibutuhkan [2, p. 386].

EventSource merupakan teknologi HTML5 yang menggunakan bahasa JavaScript untuk menjalankannya. *EventSource* bertugas 'mendengarkan' *event trigger* dari kode program yang berjalan pada sisi *server* dan menjalankan kode JavaScript yang ditentukan. *Event* dapat dijalankan dapat menggunakan *AddEventListener()* pada kode JavaScript yang dibuat untuk menambahkan *event* yang dapat dijalankan oleh aplikasi [3].

C. Polling

Pada awalnya, aplikasi web dikembangkan menggunakan model *client-server* di mana web *client* sebagai inisiator dari transaksi data, meminta data dari *server*. Maka dari itu, tidak adanya mekanisme dari *server* untuk mengirimkan data ke sisi *client* tanpa *client* melakukan permohonan terlebih dahulu. Untuk mengatasi kekurangan ini, pengembang sistem berbasis web dapat mengimplementasi sebuah teknik yang disebut HTTP *Long Polling*, yang memungkinkan *client* melakukan *poll* terhadap *server* untuk meminta informasi baru (lihat Gambar 2). *Server* menahan permohonan yang masih terbuka sampai informasi baru tersedia. Ketika informasi baru tersedia, *server* membalas permohonan dan mengirimkan informasi tersebut. Ketika *client* menerima informasi terbaru, *client* langsung mengirimkan permohonan yang sama, dan operasi tersebut dapat diulang terus-menerus [4].



Gambar 2 HTTP Long Polling Diagram [5]

Pada JavaScript, *polling* dapat dilakukan menggunakan kode berikut:

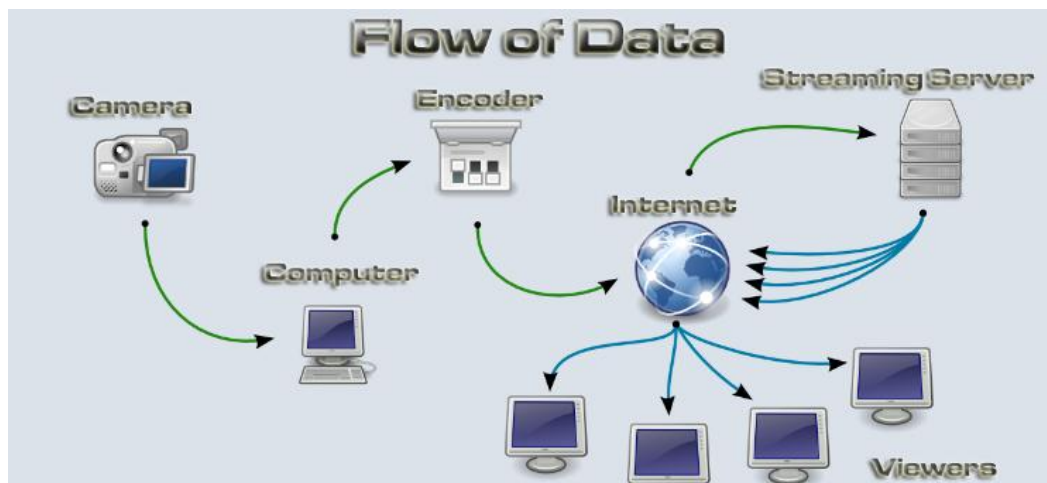
```
setInterval (fungsiMintaData () , interval );
```

Fungsi *fungsiMintaData()* adalah sebuah *Ajax Request* yang meminta data terbaru dari server dan interval merupakan jangka waktu dalam milidetik yang diatur untuk memanggil kembali fungsi tersebut. Cara ini merupakan solusi yang masuk akal, namun terkadang tidak efisien karena cara ini memiliki kelemahan yaitu tertundanya informasi yang akan diterima selama interval yang telah di atur pada fungsi. *Polling* dengan metode ini juga memberatkan sisi *server* dengan koneksi *client-server* yang terus terbuka walaupun tidak ada data baru yang diterima.

Salah satu solusi dari masalah tersebut di atas adalah menerapkan konsep *Server Sent Events(SSE)*, yang memungkinkan halaman web mempertahankan koneksi ke *server* web. *Server* web dapat mengirimkan data informasi baru kapan saja. *EventSource* merupakan teknologi HTML5 yang menggunakan bahasa JavaScript untuk menjalankannya. *EventSource* menunggu *event trigger* dari kode program yang berjalan pada sisi server dan kemudian menjalankan kode fungsi JavaScript yang ditentukan [3]. Penulis mengirimkan data dari server menggunakan format JSON untuk mempermudah transmisi data dan pemrosesan data.

D. OBS Studio

Open Broadcaster Software(OBS) Studio merupakan sebuah perangkat lunak gratis dan *open source* untuk merekam video dan menyiarkan secara langsung (*Live Streaming*). Fitur utamanya melingkupi *encoding* menggunakan H264 (x264), AAC, RTMP *Streaming* ke Twitch, YouTube, Facebook, dan lainnya. Luaran berkas rekaman dihasilkan dalam bentuk MP4 atau FLV melalui rekaman layar cepat [6]. *Open Broadcaster Software Studio* merupakan sebuah *client* RTMP (*Real Time Messaging Protocol*). RTMP adalah teknologi yang dikembangkan oleh Macromedia(Adobe) yang pada mulanya di rancang untuk transmisi suara, video, dan data dengan performa tinggi antar *platform* teknologi Adobe Flash[7][8].



Gambar 3 Alur Kerja Streaming

(Sumber: <http://www.simplethoughtproductions.com/2008/12/03/streaming-live-video/>)

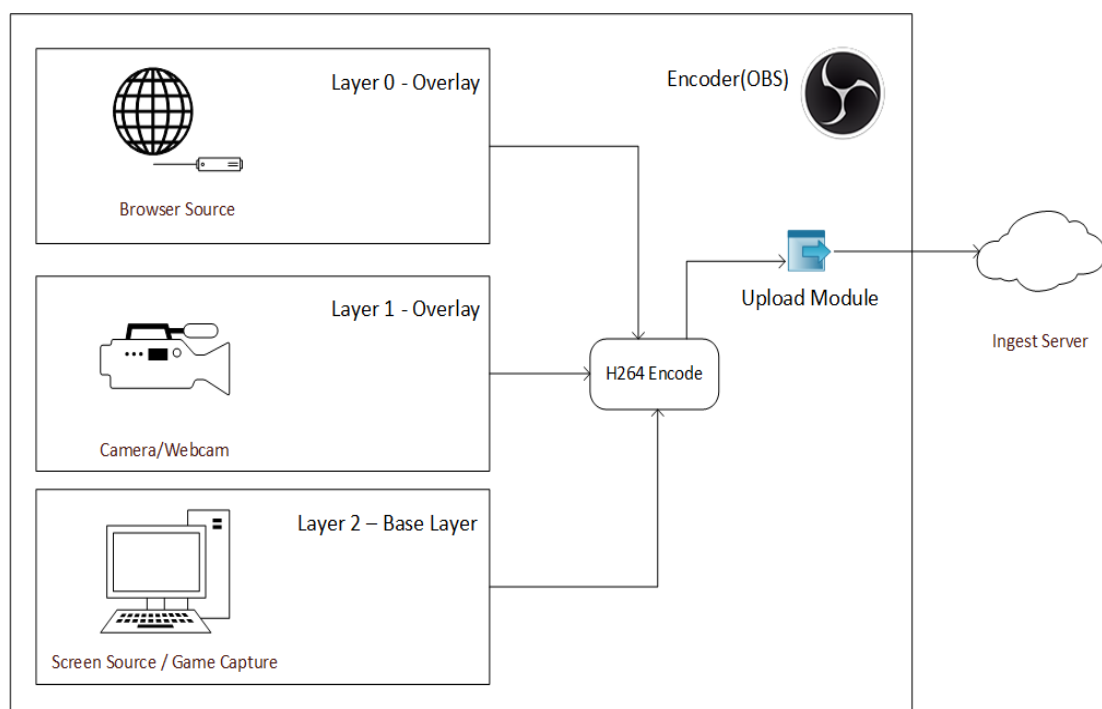
Pada saat ini, OBS Studio lebih banyak digunakan oleh generasi sosial media yang suka membagikan kegiatan yang dilakukannya supaya dapat dilihat oleh masyarakat luas. OBS Studio berfungsi sebagai *encoder* yang berada di dalam komputer pengguna. OBS Studio menerima gambar atau data masukan dari kamera, biasanya berupa *webcam* atau *action camera* yang disambungkan ke komputer, *browser source* yang merupakan tampilan dari peramban internet, maupun *screen capture* atau *game capture* yang menangkap tampilan dari layar maupun dari permainan komputer yang sedang dimainkan.

Para penyiar daring biasanya menerima donasi yang kemudian ditampilkan datanya, seperti: namapengirim, jumlah, dan pesan dari pengirim pada siarannya. OBS Studio melakukan kompresi gambar yang telah di-capture yang biasa disebut sebagai *encoding process* sebelum dikirimkan melalui internet ke *ingest server*. *Ingest Server* merupakan *server* yang menerima data gambar siaran. Sebagai contoh, pada layanan YouTube Live, *ingest server* berada di tautan <rtmp://a.rtmp.youtube.com/live2> dan pada Twitch.tv berada pada tautan rtmp://live-sin.twitch.tv/app/{stream_key} dimana *stream_key* merupakan karakter alfanumerik yang digunakan sebagai identifikasi pengirim data siaran langsung. Pada Gambar 3 memberikan sketsa tampilan konsep kerja OBS Studio.

III. ANALISIS DAN RANCANGAN

A. Proses Kerja Open Broadcaster Studio

OBS Studio berfungsi sebagai *encoder* yang menerima masukan data gambar dari tangkapan layar maupun tangkapan gambar permainan komputer yang sedang dimainkan sebagai latar lapisan pertama (*Base Layer*) yang berada di paling belakang, tangkapan gambar dari kamera yang biasanya berada di depan dari lapisan latar (*Intermediate Layer*), dan notifikasi donasi yang menggunakan fitur Browser Source pada OBS Studio yang diletakkan pada lapisan paling depan (*Top Layer*). Setelah lapisan-lapisan gambar tersebut tersusun, modul *encoder* pada OBS Studio melakukan *encode* terhadap gambar yang dihasilkan. Gambar yang telah di *encode* akan dikirimkan pada *ingest server* melalui internet. Pada Gambar 4 ditampilkan alur kerja OBS Studio.

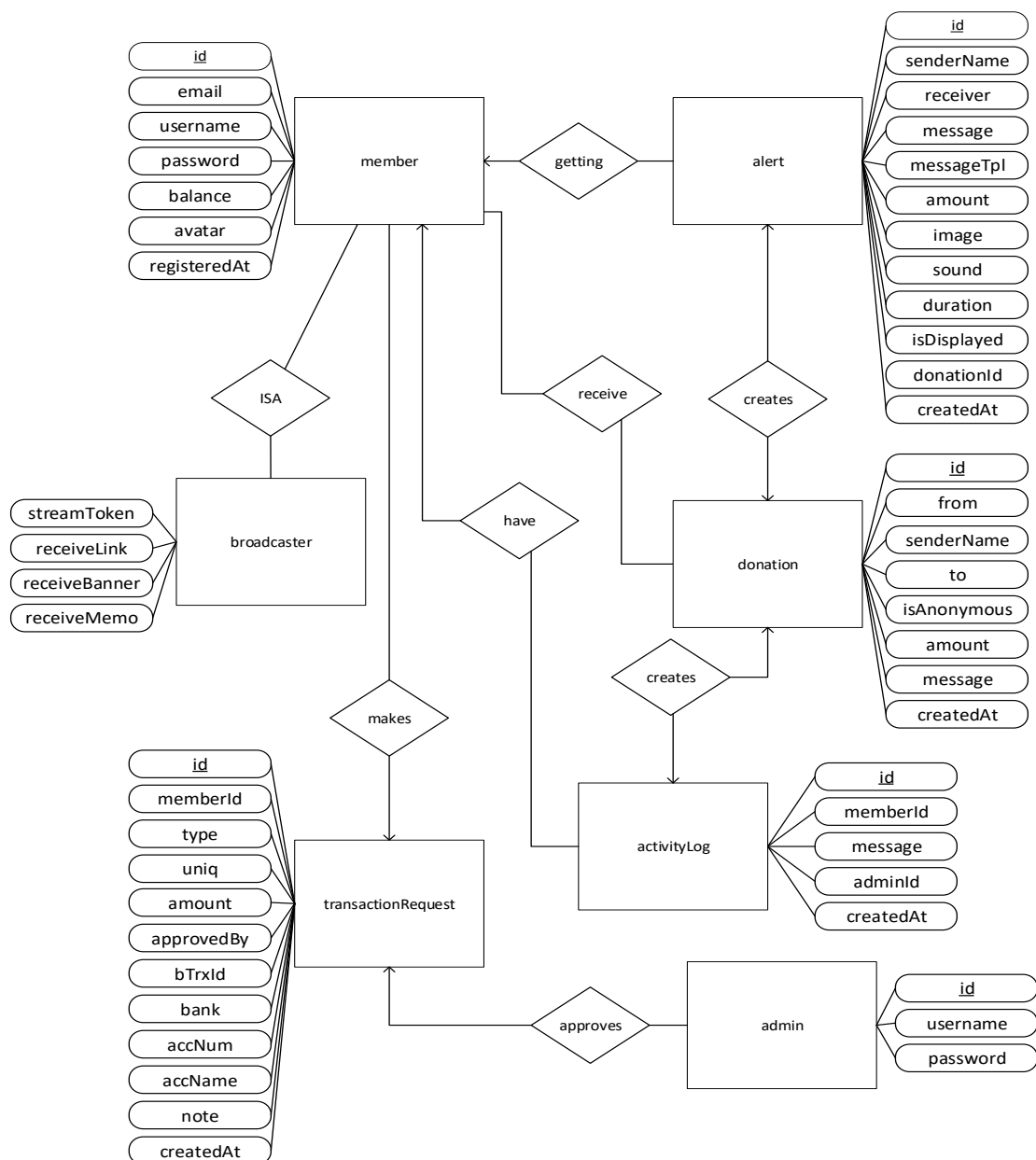


Gambar 4 Proses Kerja Open Broadcaster Software

Penggunaan *EventSource* terletak pada pengiriman data yang akan ditampilkan pada modul *Browser Source* dalam *Open Broadcaster Software* untuk menampilkan notifikasi donasi yang diterima oleh penyiar.

B. Entity Relationship Diagram

Pada Gambar 5 berikut ini, digambarkan relasi entitas pada sistem yang dikembangkan.



Gambar 5 Entity Relationship Diagram

Pada Gambar 5, relasi entitas di atas, *member* sebagai donatur, bisa juga menjadi sebagai penyiar. Penyiar memiliki *token streaming* yang digunakan untuk menampilkan halaman notifikasi. *Member* dapat menerima banyak donasi dari *member* lainnya. *Member* mendapatkan notifikasi dari donasi yang terjadi melalui sistem yang juga tercatat dalam *logaktivitas*. Setiap donasi yang diterima maupun dikirimkan oleh pengguna dicatat dalam *logaktivitas*. Pengguna dapat melakukan permohonan transaksi yang dapat berupa penambahan maupun penarikan dana. Setiap permohonan transaksi harus disetujui oleh admin dari sistem yang akan dicatat dalam rekaman transaksi.

Pada tabel-tabel berikut ini dijelaskan transformasi dari diagram relasi entitas.

TABEL I
TABEL MEMBER

Nama Field	Key	Null	Tipe Data
id	Index Primary	Not Null	Int(11)
email		Not Null	Text
username		Not Null	Text
password		Not Null	Text

Nama Field	Key	Null	Type Data
balance		Not Null	Double(11,2)
avatar		Null	Text
registeredAt		Not Null	DateTime

TABEL II
TABEL BROADCASTER

Nama Field	Key	Null	Type Data
memberId	Foreign Key (member), Index Primary	Not Null	Int(11)
streamToken		Not Null	Text
receiveLink		Not Null	Text
receiveBanner		Null	Text
receiveMemo		Null	Text

TABEL III
TABEL DONASI

Nama Field	Key	Null	Type Data
id	Index Primary	Not Null	Int(11)
from	Foreign Key (member)	Not Null	Int(11)
senderName		Null	Text
to	Foreign Key (member)	Not Null	Int(11)
isAnonymous		Not Null	Boolean
amount		Not Null	Double(11,2)
message		Null	Text
createdAt		Not Null	DateTime

TABEL IV
TABEL ALERT

Nama Field	Key	Null	Type Data
id	Index Primary	Not Null	Int(11)
senderName		Null	Text
receiver	Foreign Key (member)	Not Null	Int(11)
message		Null	Text
messageTpl		Not Null	Text
amount		Not Null	Double(11,2)
image		Null	Text
sound		Null	Text
duration		Not Null	Int(11)
isDisplayed		Not Null	Boolean
donationId	Foreign Key (donation)	Not Null	Int(11)
createdAt		Not Null	DateTime

TABEL V
TABEL PERMOHONAN TRANSAKSI

Nama Field	Key	Null	Type Data
id	Index Primary	Not Null	Int(11)
memberId	Foreign Key (member)	Not Null	Int(11)
type		Not Null	Enum(Dp, Wd)
uniq		Null	Int(4)
amount		Not Null	Double(11,2)
approvedBy	Foreign Key (admin)	Null	Int(11)
bTrxId		Null	Text

Nama Field	Key	Null	Tipe Data
bank		Not Null	Text
accNum		Not Null	Text
accName		Not Null	Text
note		Null	Text
createdAt		Not Null	DateTime

TABEL VI
TABEL LOG AKTIVITAS

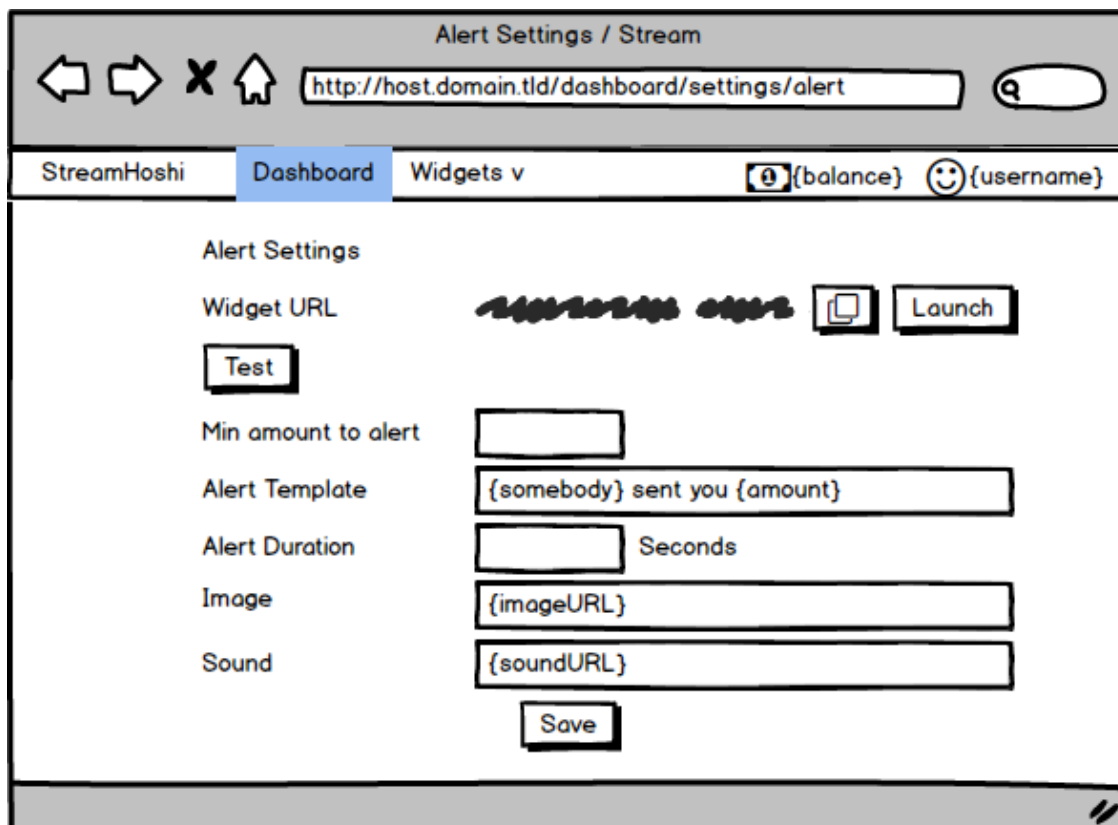
Nama Field	Key	Null	Tipe Data
id	Index Primary	Not Null	Int(11)
memberId	Foreign Key (member)	Not Null	Int(11)
message		Not Null	Text
adminId	Foreign Key (admin)	Null	Int(11)
createdAt		Not Null	DateTime

TABEL VII
TABEL ADMIN

Nama Field	Key	Null	Tipe Data
id	Index Primary	Not Null	Int(11)
username		Not Null	Text
password		Not Null	Text

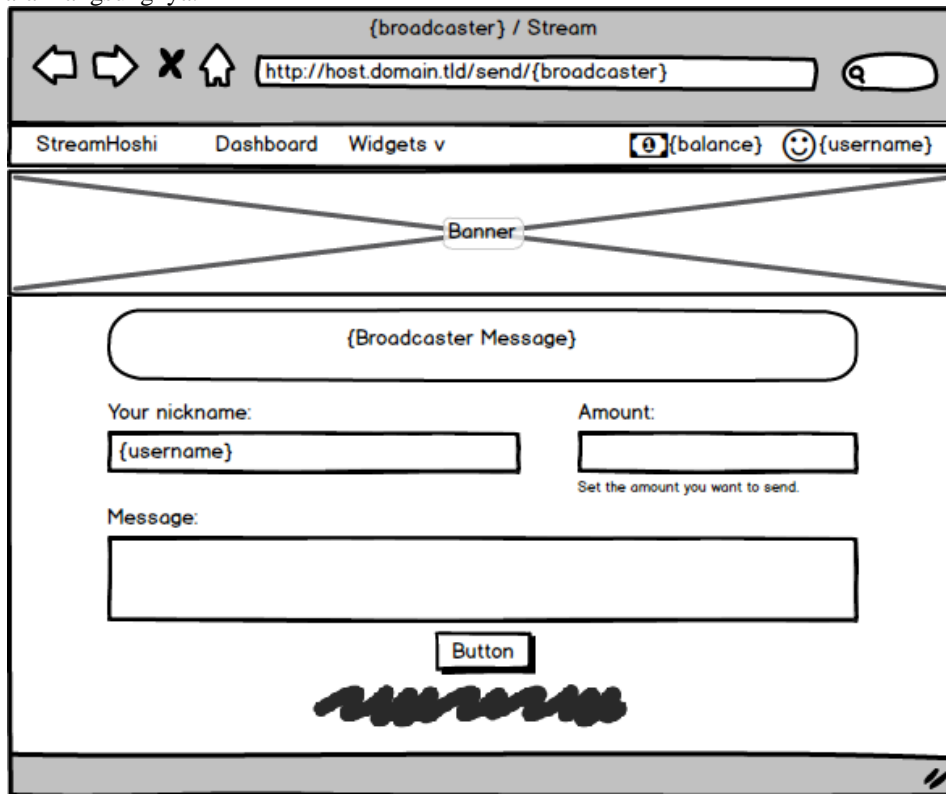
C. Sketsa Tampilan

Gambar 6 memberikantampilan utama untuk fitur notifikasi yang menggunakan fitur *EventSource*.



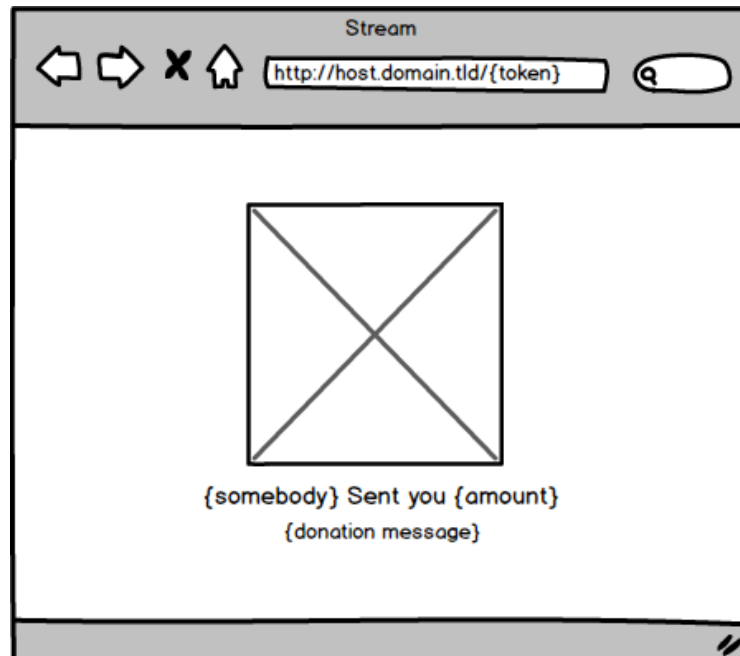
Gambar 6 Sketsa Tampilan Halaman Pengaturan Notifikasi Donasi

Gambar 6 menggambarkan tampilan pengaturan notifikasi yang dapat di atur oleh penyiar untuk kemudian di tampilkan pada siaran langsungnya.



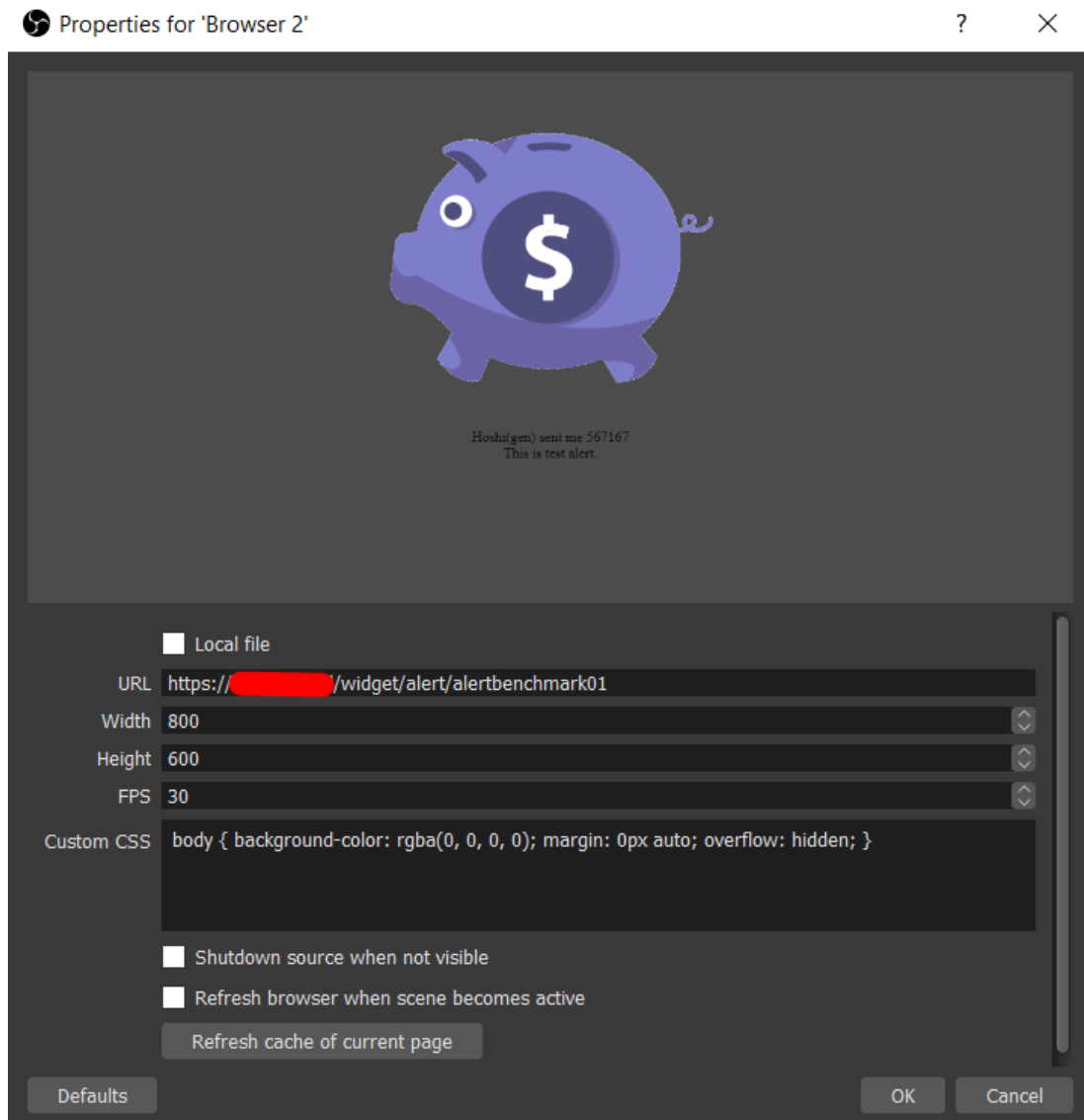
Gambar 7 Sketsa Tampilan Halaman Pengiriman Donasi

Gambar 7 menggambarkan tampilan halaman untuk mengirimkan donasi kepada penyiar.



Gambar 8 Sketsa Tampilan Halaman Notifikasi Donasi

Gambar 8 menggambarkan tampilan halaman notifikasi donasi yang di terima oleh penyiar.



Gambar 9 Contoh Konfigurasi Pada Aplikasi *Open Broadcaster Software*

Gambar 9 menggambarkan tampilan konfigurasi pada aplikasi *Open Broadcaster Software*. Melalui tampilan ini, penyiar dapat mengatur untuk menggunakan tampilan notifikasi donasi yang telah diatur pada Gambar 7.

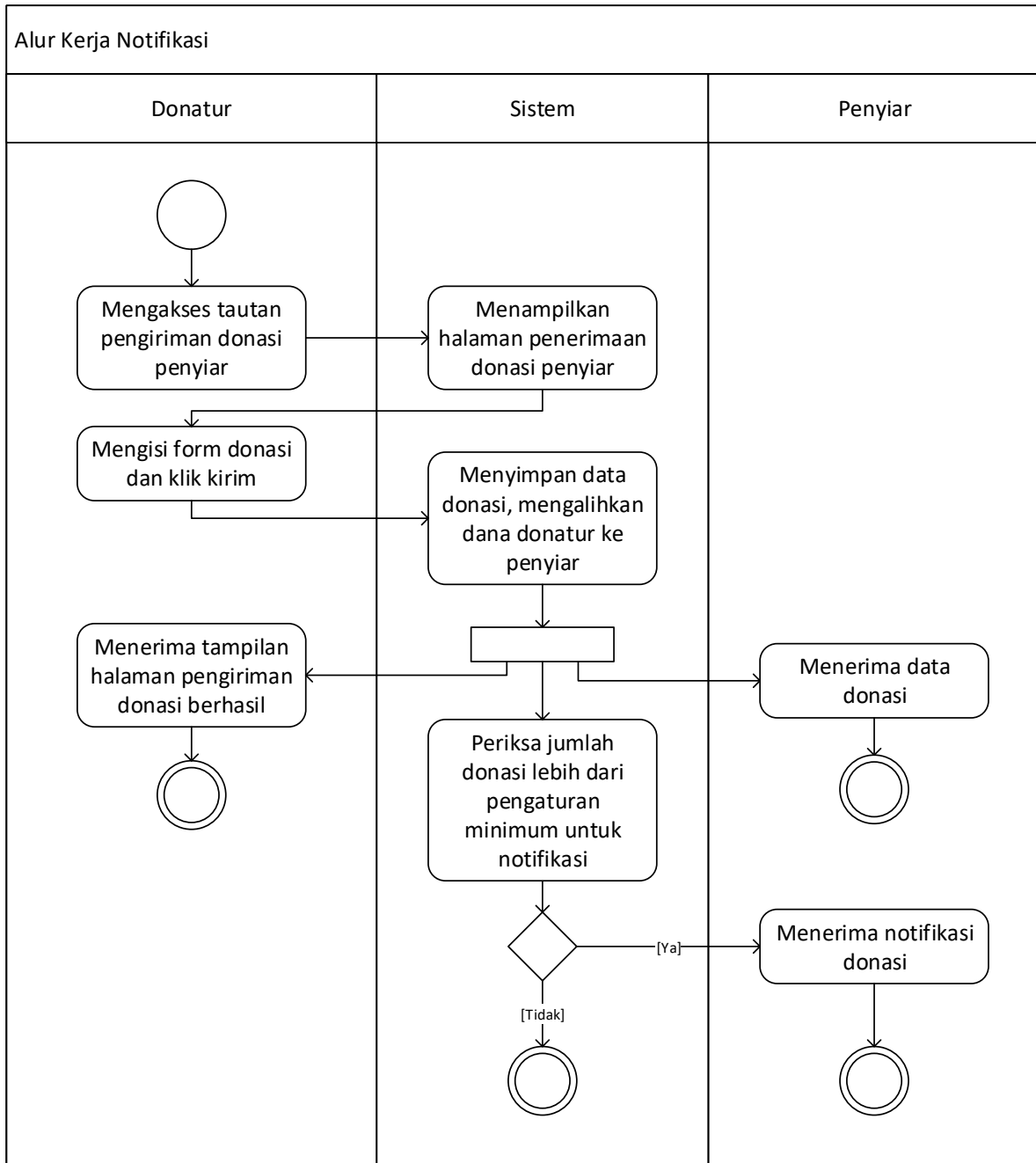
IV. HASIL DAN KESIMPULAN

A. Alur Kerja Notifikasi

Gambar 10 menggambarkan alur kerja notifikasi donasi, dengan kemampuan donatur untuk mengakses tautan khusus pengiriman donasi dari penyiar. Kemudian sistem akan menampilkan halaman penerimaan donasi, dan donatur mengisi formulir pengiriman donasi yang telah disediakan. Setelah donatur mengisi formulir donasi dan mengirimkan formulir tersebut, sistem mengalihkan jumlah donasi yang telah diisi oleh donatur pada formulir donasi. Lalu, sistem menampilkan halaman pengiriman donasi berhasil kepada donatur dan melakukan pengecekan atas jumlah donasi yang di kirimkan.

Apabila jumlah donasi melebihi pengaturan jumlah donasi minimum untuk notifikasi, maka sistem mengirimkan data notifikasi donasi dengan tampilan seperti pada Gambar 8 menggunakan *EventSource* kepada penyiar untuk dapat di tampilkan pada siaran yang sedang berlangsung. Namun, apabila jumlah di bawah pengaturan jumlah donasi minimum

untuk notifikasi, sistem hanya mengirimkan notifikasi donasi baru diterima kepada penyiar tanpa melalui fitur notifikasi donasi (*widget alert*).



Gambar 10 Diagram Alur Kerja Notifikasi Donasi

B. Pengujian

Berdasarkan hasil pengujian menggunakan metode *benchmarking* pada fitur notifikasi yang ada di dalam sistem yang telah dikembangkan, didapatkanlah hasil sebagai berikut:

TABEL VIII
HASIL PENGUJIAN BENCHMARK

Jumlah Notifikasi	Keterlambatan terkecil		Keterlambatan terbesar	
	Lokal	Server	Lokal	Server

Jumlah Notifikasi	Keterlambatan terkecil		Keterlambatan terbesar	
	Lokal	Server	Lokal	Server
10	0.49473991394042	0.52678003311157	2.80783009529113	2.85652995109558
50	0.82157198429107	0.70962801933288	2.25145204544067	2.13667604446411
100	1.46255090951919	1.34122000694274	2.61883992433547	2.42959598302841
500	0.82485253715515	0.55878601264953	3.44455285263061	2.73752840948104
1000	0.86488005156452	0.31173243877050	3.41843916964530	2.17271880269050

(Hasil dalam satuan milidetik.)

Dari hasil di atas, dapat disimpulkan bahwa keterlambatan pengiriman data notifikasi tidak signifikan walaupun memiliki data yang banyak. Dan hasil tersebut dapat bersifat fluktuatif berdasarkan penggunaan sumber daya *server* di mana sistem berada.

DAFTAR PUSTAKA

- [1] D. S. McFarland, *JavaScript & jQuery: The Missing Manual*, 3 penyunt., Sebastopol, CA: O'Reilly Media, Inc., 2014.
- [2] M. MacDonald, *HTML5: The Missing Manual*, 2nd Edition, Sebastopol, CA: O'Reilly Media, Inc., 2014.
- [3] Web Hypertext Application Technology Working Group, "HTML Standard," [Online]. Tersedia: <https://html.spec.whatwg.org/multipage/server-sent-events.html#server-sent-events>.
- [4] J. Hanson, "What is HTTP Long Polling? | PubNub," PubNub Inc., [Online]. Tersedia: <https://www.pubnub.com/blog/2014-12-01-http-long-polling/>
- [5] J. Jaitla, "WebSockets and Long Polling in JavaScript, Ruby and Python | PubNub," PubNub Inc., [Online]. Tersedia: <https://www.pubnub.com/blog/2014-10-01-websockets-and-long-polling-in-javascript-ruby-and-python/>.
- [6] R. Hill-Whittall, *The Indie Game Developer Handbook*, Burlington, Massachusetts: Focal Press, 2015.
- [7] Adobe Systems Incorporated, "Real-Time Messaging Protocol (RTMP) specification | Adobe Developer Connection," Adobe, [Online]. Tersedia: <http://www.adobe.com/devnet/rtmp.html>.
- [8] T. Radius dan I. Sushendra, "Perancangan dan Implementasi Aplikasi Android Streaming (Studi Kasus FTI Universitas Kristen Satya Wacana Salatiga)," *JuTISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 1, no. 3, p. 204, Desember 2015.