Rancang Bangun Aplikasi *Event Management* Untuk Manajemen Data Peserta KLiKS Dengan *Secure Web* API Berdasarkan OWASP API Top Ten 2019

Ismail Sofyan Tsany[1)], Nurul Qomariasih[2)]
(1) Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, mai.tsany@gmail.com
(2) Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, nurul.qomariasih@bssn.go.id

Abstrak

Dalam rangka menjalankan tugasnya untuk mengedukasi masyarakat mengenai keamanan siber, BSSN melalui Subdirektorat Proteksi Keamanan Informasi Publik (PKIP), Direktorat Ekonomi Digital, menyelenggarakan seminar Kampanye Literasi Keamanan Siber (KLiKS). Akan tetapi pada pelaksanaannya masih dijumpai permasalahan mengenai manajemen data peserta yang mendaftar dimana masih dilakukan secara manual, menggunakan aplikasi pihak ketiga, dan ketimpangan data di lapangan. Selain itu dibutuhkan juga pengintegrasian antara aplikasi registrasi, manajemen data, dan registrasi ulang gawai KLiKS milik PKIP. Penelitian ini akan membangun aplikasi event management untuk manajemen data peserta KLiKS berbasis web menggunakan metode SDLC Prototyping guna mengatasi permasalahan manajemen data peserta KLiKS. Selain itu, penelitian ini juga akan membangun layanan API untuk komunikasi data dengan aplikasi API client dengan sistem keamanan JWT guna melindungi dari kerawanan broken user authentication yang menjadi kerawan peringkat kedua pada OWAPS API Top Ten 2019. Hasilnya ditemui bahwa penerapan aplikasi event management KLiKS ini dapat mempermudah proses manajemen data peserta KLiKS, bebas dari aplikasi pihak ketiga, dan data yang disajikan selaras. Penerapan JWT juga dapat mengatasi terjadinya kerawanan broken user authentication karena setiap request yang dikirim perlu memiliki token yang valid.

Kata kunci: API JWT, Aplikasi *event management*, KLiKS BSSN, SDLC *Prototyping*

1. PENDAHULUAN

Ditengah kemajuan teknologi saat ini, keamanan siber menjadi bekal yang wajib dimiliki oleh setiap pengguna internet. Menanggapi hal ini pemerintah Indonesia melalui Badan Siber dan Sandi Negara (BSSN) memiliki tugas untuk mengedukasi masyarakat Indonesia mengenai keamanan siber [1]. Untuk mewujudkannya BSSN menyelenggarakan kegiatan rutin yaitu seminar Kampanye Literasi Keamanan Siber (KLiKS) yang diselenggarkan oleh Subdirektorat Proteksi Keamanan Informasi Publik (PKIP), Direktorat Proteksi Ekonomi Digital. Akan tetapi pada pelaksanaannya, proses manajemen data peserta kegiatan KLiKS masih mengandalkan aplikasi pihak ketiga serta masih bersifat manual [2]. Dengan kondisi seperti ini banyak ditemui ketimpangan data di lapangan antara jumlah peserta terdaftar, peserta yang hadir saat kegiatan, peserta yang mengisi *pre questionnaire* dan *post questionnaire*, dan peserta yang menerima sertifikat. Hal ini juga diperburuk dengan kondisi pelaksanaan kegiatan yang berkelanjutan dengan jumlah peserta berkisar antara 1.000 - 1.900 peserta [3]. Faktanya untuk menghadapi revolusi industri 4.0, penting bagi pemerintahan saat ini untuk mulai menerapkan *e-Government* [4]. Dengan menggunakan infrastruktur mandiri pemerintah dapat meningkatkan efektivitas, efisiensi, kesinambungan, aksesibilitas, dan keamanan dari setiap kegiatan yang dilaksanakannya.

Upaya yang dapat dilakukan dari permasalah di atas adalah dengan membangun aplikasi *event management* yang dapat membantu kegiatan KLiKS utamanya pada proses manajemen data peserta seperti proses tambah, hapus, sunting, dan pantau data peserta [2]. Selain itu dikarenakan pihak PKIP saat ini juga tengah membangun aplikasi registrasi ulang KLiKS berbasis gawai dan web pendaftaran peserta KLiKS, diperlukan layanan web API untuk integrasi aplikasi. Meningkatnya serangan terhadap API pun perlu menjadi pertimbangan pada saat menambahkan layanan web API pada aplikasi web. Pada tahun 2019 lalu terhitung lebih dari 75% serangan *credential stuffing* mengincar layanan API [5]. Pada tahun yang sama OWASP mengeluarkan projek yang diberi nama *OWASP API Top Ten 2019* yang berisi sepuluh kerawanan yang paling banyak ditemui pada API dengan kerawanan kedua paling banyak ditemui adalah *Broken User Authentication* yaitu kerawanan yang terjadi akibat kesalahan mekanisme autentikasi yang digunakan untuk identifikasi pengguna [6]. Oleh karena itu diperlukan langkah mitigasi dari kerawanan tersebut dengan menerapkan tahap identifikasi pengguna untuk memperbaiki mekanisme autentikasi seperti penggunaan *username* dan *password* untuk mendapatkan JSON Web Tokens (JWT) [5].

Penelitian ini akan membangun aplikasi *event management* kegiatan KLiKS berbasis web yang ditujukan untuk manajemen data peserta kegiatan KLiKS. Manajemen data yang dimaksud adalah proses baca, tambah, sunting, dan hapus data. Aplikasi juga akan dilengkapi dengan layanan web API yang dilengkapi JWT sebagai penghubung dengan aplikasi lain yang juga menggunakan data yang sama. Dari penelitian ini diharapkan dihasilkan aplikasi *event management* yang cocok untuk memanajemen data peserta kegiatan KLiKS sehingga bisa menjadikan proses bisnis yang berjalan secara digital dan mandiri.

Aplikasi juga diharapkan dapat memberikan layanan API yang aman dari kerawanan *Broken User Authentication*.

2. LANDASAN TEORI

2.1. Aplikasi *Event Management*

Pada penelitian yang dilakukan Pinjari, aplikasi *event management* pada dasarnya merupakan implementasi dan manajemen suatu kegiatan melalui berbagai macam teknologi perangkat lunak yang tersedia [7]. Pada penelitiannya tersebut aplikasi yang dibangun berisikan fitur dan fungsi yang berhubungan dengan penyelenggaraan kegiatan seperti menambahkan data, menampilkan data, dan memberikan laporan. Dari penelitian tersebut menunjukan hasil bahwa aplikasi event management yang dibangun dapat mengurangi usaha yang diperlukan serta mempermudah pekerjaan.

2.2. Web API

Penggunaan API ditujukan sebagai jalur komunikasi antara satu aplikasi dengan aplikasi lainnya [8]. Dengan menggunakan web API ini data center yang digunakan dua aplikasi tersebut menjadi sama. *Web service* sendiri ditujukan untuk memenuhi kebutuhan dari situs atau aplikasi. Klien dapat menggunakan API untuk berkomunikasi dengan *web service.*

Ditengah meningkatnya serangan terhadap web API, perlu dilakukan mitigasi untuk mencegah terjadinya serangan seperti menggunakan JSON Web Token (JWT) [5]. JWT adalah standar *open industry* yang dapat mengamankan komunikasi dua pihak [9]. JWT terdiri dari tiga bagian yaitu *header, payload*, dan *signature*. Masing-masing bagian menggunakan basis *Base64 encoded* dan dipisahkan menggunakan titik. Pada bagian *header* berisikan tipe token dan juga algoritma yang digunakan, *payload* berisikan data yang dikomunikasikan dan bagian *signature* berfungsi untuk menjaga integritas dari JWT.

2.3. OWASP API Security Top Ten

OWASP API Security Top Ten keluar pada tahun 2019 sebagai respon terhadap meningkatnya penggunaan dan serangan terhadap *Web API* [6]. Pada API Security, bahasan utama yang dibahas adalah mengenai strategi dan solusi untuk memahami dan memitigasi kerawanan dan penilaian keamanan yang khusus berhubungan dengan API. Contoh kerawanan yang dibahas adalah API1:2019 – *Broken Object Level Authorization*, API2:2019 - *Broken User Authentication*, dan API3:2019 - *Excessive Data Exposure*. Dari proyek ini pengembang aplikasi dapat meningkatkan fitur keamanan web APInya dengan berpedoman panduan di dalamnya [5] .

Fokus dari penelitian ini adalah untuk mengatasi kerawanan API2:2019 – *Broken User Authentication*. Kerawanan ini terjadi saat adanya kebocoran data/*resource* karena adanya kesalahan saat proses autentikasi pengguna [6]. Kerawanan ini berasal dari tidak adanya mekanisme autentikasi pengguna ataupun kesalahan penerapan mekanisme autentikasi seperti tidak menerapkan batasan waktu penggunaan token untuk komunikasi API. Dampak dari kerawanan ini adalah penyerang dapat mendapatkan kontrol terhadap akun pengguna, mendapatkan akses terhadap data/*resource*, dan melakukan aksi sensitif seperti pengiriman pesan pribadi atau transaksi keuangan. Proses autentikasi pengguna pada API sendiri bisa dilakukan dengan melakukan identifikasi *username* dan *password* [5]. Adapun pada penelitian ini diterapkan alur autentikasi *username* dan *password* untuk menghasilkan token JWT yang kemudian digunakan untuk proses komunikasi API. Token JWT yang dibuat pun diatur lama kedaluarsanya sehingga hanya bisa digunakan untuk rentan waktu tertentu saja.

2.4. *Unified Modelling Language* (UML)

Unified Modelling Language (UML) bertujuan untuk menyediakan bahasa yang umum untuk istilah *object-based* dan diagram yang fleksibel untuk memodelkan setiap sistem proyek yang dikembangkan dari tahap analisis sampai desain. Ada banyak jenis dari diagram UML namun terdapat empat jenis diagram fundamental yang menjadi inti penggunaan UML saat ini, antara lain *use case diagram, class diagram, sequence diagram*, dan *behavioral state machine diagram* [10].

2.5. SDLC *Prototyping*

Prototyping merupakan salah satu metode pendekatan *System Development Life Cycle* (SDLC) yang melakukan tahap analisis, desain, dan implementasi secara berkelanjutan untuk mengembangkan dengan cepat versi sederhana dari sistem yang dituju dan memberikannya kepada *user* untuk dievaluasi dan ditanggapi [10]. Setelah diberikan evaluasi dan tanggapan, pengembang melakukan ulang tahap analisis, desain, dan implementasi untuk memperbaiki *prototype* pertama sehingga dihasilkan *prototype* kedua. Tahap ini terus berlanjut sampai dengan analis, *user*, dan sponsor setuju bahwa *prototype* sudah sesuai dan siap digunakan. Sama seperti tahap SDLC lainnya, metode *prototype* terdiri dari empat tahap utama *planning, analysis, design,* dan *implementation* [10].

3. METODE PENELITIAN

Pada penelitian ini yang menjadi obyek penelitian adalah Subdirektorat Proteksi Keamanan Informasi Publik (PKIP), Direktorat Proteksi Ekonomi Digital, BSSN. Subdirektorat Proteksi Keamanan Informasi Publik mempunyai tugas melaksanakan penyiapan penyusunan, koordinasi, pelaksanaan, pengendalian, evaluasi, dan pelaporan kebijakan teknis di bidang strategi dan tata kelola keamanan informasi publik,

budaya keamanan informasi publik, edukasi keamanan siber, dan strategi dan koordinasi dalam pembinaan komunitas keamanan siber. Salah satu upaya yang dilakukan Subdirektorat PKIP untuk melaksanakan tugasnya di bidang budaya keamanan informasi publik dan edukasi keamanan siber adalah melalui kegiatan KLiKS yang dibahas pada penelitian ini.

Jenis penelitian yang dilakukan pada penelitian ini adalah penelitian kualitatif. Pengumpulan data dilakukan dengan wawancara mendalam (*in-depth interview*) sehingga dihasilkan *user requirement* yang lebih valid [11]. Selama dilakukan penelitian tetap dilakukan studi literatur yaitu berpedoman pada standar dan teori yang sudah ada sebelumnya seperti standar OWASP API. Tahap analisis dilakukan dengan berfokus pada penyelesaian rumusan masalah yaitu pembangunan aplikasi *event management* dan *secure API* juga analisis pengujian fungsional dan non-fungsional, termasuk pengujian keamanan.

Metode pengembangan yang dilakukan pada penelitian ini yaitu metode SDLC dengan pendekatan *Prototyping*. Terdapat empat tahap pada metode ini yaitu *planning*, *analysis*, *design*, dan *implementation* sebelum akhirnya dihasilkan *system prototype*. *Prototype* kemudian akan disampaikan kepada pihak lokus untuk dipastikan apakah *system prototype* sudah memenuhi ekspektasi sesuai dengan pembatasan yang sudah ditetapkan dengan lokus. Pada tahap paling awal ini peneliti melakukan identifikasi proses bisnis dari lokus mengenai kegiatan KLiKS melalui *system request* singkat sehingga bisa diketahui permasalahan yang ditemui lokus dan permintaannya. Pada tahap *analysis* dilakukan *requirement gathering* dengan menggunakan metode wawancara kepada narasumber untuk mengulik lebih jauh mengenai sistem yang dibutuhkan. Pada tahap ini kebutuhan fungsional dan non fungsional akan diidentifikasi. Tahap *design* merupakan tahap yang merancang bagaimana sistem akan bekerja dilihat dari berbagai aspek komponen. Pada tahap ini rancangan akan dimodelkan menggunakan *Unified Modeling Language* (UML) dengan tiga jenis model diagram yaitu, sequence diagram, *class* diagram, dan behavior state machine diagram. Pemilihan diagram ini dirasa cukup bisa mewakilkan setiap alur dan komponen yang ada pada sistem KLiKS yang akan dibuat. Selain itu untuk menggambarkan hubungan tabel pada *database* dibuat juga ERD. Pada tahap implementation alur kerja sistem yang sudah dimodelkan pada tahap desain diimplementasikan untuk dibangun. Pada penelitian ini sistem dibangun menggunakan *platform* web menggunakan bahasa pemograman PHP. Selain itu pembangunan sistem menggunakan *framework* CodeIgniter versi 3.0. *Database* yang digunakan adalah MySQL. Selain itu pada tahap implementation dilakukan proses pengujian perangkat lunak. Pengujian terdiri dari pengujian fungsional dan non fungsional berupa *unit testing*, *integration testing*, *performance testing*, *user acceptance testing*, dan *security testing*. Semua pengujian tersebut dilakukan dengan metode pengujian *black box*. Dikarenakan penelitian ini tidak menggunakan DFD maka *Integration testing* yang dilakukan adalah *user interface testing* dan *use case testing*. *Performance testing* dilakukan menggunakan *online tools* GTmetrix. Pada UAT, user akan menguji sendiri aplikasi yang dibangun dan diintegrasikan dan diberikan dokumen *test case* sebagai panduan.

4. HASIL DAN PEMBAHASAN

4.1. *Planning*

Aplikasi ini ditujukan untuk membantu proses manajemen data peserta yang saat ini masih berjalan secara manual dan menggunakan aplikasi pihak ketiga sehingga diharapkan aplikasi dapat melakukan proses tambah, hapus, sunting, dan pantau data peserta [2]. Aplikasi yang dibangun juga harus mendukung proses integrasi, yaitu dengan menyediakan layanan API [2]. Akan tetapi seiring dengan meningkatnya serangan terhadap layanan API pada penelitian ini akan ditambahkan sistem keamanan khusus untuk API yaitu dengan menggunakan JSON Web Token (JWT) untuk menghindari kerawanan *Broken User Authentication* [6]. Dari latar belakang dan perencaan tersebut tahap selanjutnya adalah dengan melakukan analisis kebutuhan proyek melalui *system request*. *System request* pada penelitian ini dijelaskan pada Tabel 1.

Tabel 1. *System Request*

System Request Aplikasi *Event Management* KLiKS dengan *Secure Web API*
Business Need: Penelitian ini bertujuan untuk membangun aplikasi mandiri yang dapat mengatur data peserta kegiatan KLiKS dan dapat terhubung dengan aplikasi lainnya.
Business Request: Aplikasi dibangun menggunakan *platform* web dengan fitur sebagai berikut: • Mampu memanajemen data peserta. • Mampu menyediakan layanan API yang aman dari kerawanan *Broken User Authentication*.
Business Value: Dengan adanya aplikasi ini diharapkan dapat membantu proses manajemen data peserta seminar KLiKS menjadi berbasis aplikasi digital mandiri serta dapat membantu proses integrasi aplikasi gawai daftar ulang dan web pendaftaran KLiKS.

4.2. Penelitian Tahap I

a. ***Analysis***

Analisis yang dilakukan dengan cara wawancara dengan pihak narasumber. Hasil dari terjabarkannya kebutuhan dari aplikasi ini kemudian yang menjadi landasan dalam tahap selanjutnya, *design* dan *implementation*.

1) Kebutuhan Fungsional

Kebutuhan fungsional atau fitur utama pada tahap pertama ini antara lain:

a) Aplikasi memiliki fitur akses kontrol melalui proses *login* dan *logout*.
b) Aplikasi memiliki fitur mengolah data peserta yang terdiri dari proses tambah, sunting, dan hapus data peserta.
c) Aplikasi dapat menyediakan berkas inputan untuk kepentingan audit dan daftar pengiriman sertifikat berupa berkas ekstraksi data peserta kedalam format pdf dan xls.
d) Aplikasi memiliki fitur untuk mengolah petugas yang terdiri dari proses tambah, sunting, dan hapus data petugas.

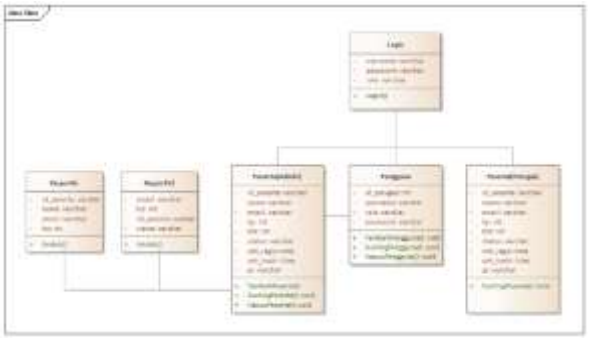
2) Kebutuhan Non Fungsional
Kebutuhan non fungsional atau requirement yang dimiliki sistem pada aplikasi ini antara lain:
a) Aplikasi menggunakan *platform* web.
b) Aplikasi memiliki dua jenis role yaitu admin dan petugas biasa.

b. ***Design***

1) *Use case Diagram*
Use case diagram merupakan diagram utama pada UML yang merepresentasikan beberapa jalur yang dapat dilakukan *user* terhadap sistem. Adapun penelitian tahap I ini digambarkan dengan diagram *use case* di bawah ini.

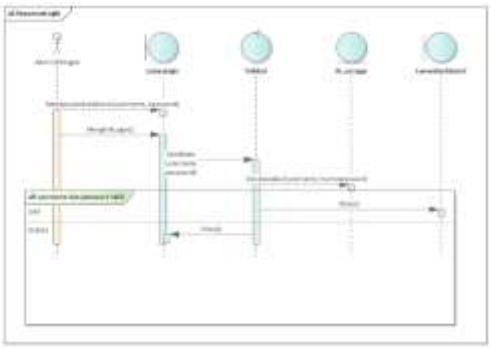
Gambar 1 *Use case Diagram System prototype I*

2) *Class Diagram*
Class diagram merupakan model statis yang mendukung tampilan statis dari perkembangan sistem. Pada penelitian ini *class diagram* digambarkan pada Gambar 2 di bawah ini.



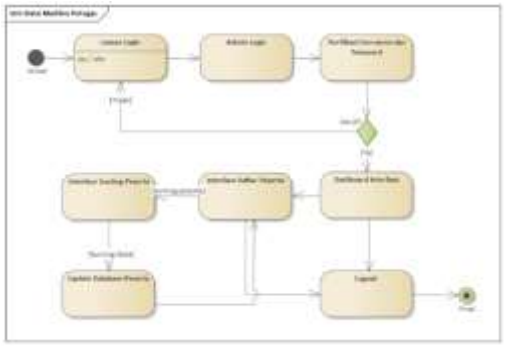
Gambar 2 *Class Diagram System prototype I*

3) *Sequence Diagram*
Sequence diagram menggambarkan objek yang digunakan pada *use case* dan pesan yang dikirimkan di antara objek-objek tersebut pada satu buah *use case*. Pada diagram ini digambarkan alur kerja dan keterangan lainnya dari sebuah *use case*. Sesuai dengan jumlah *use case*, pada penelitian tahap I ini terdapat sembilan *sequence diagram* salah satunya adalah diagram di bawah ini.



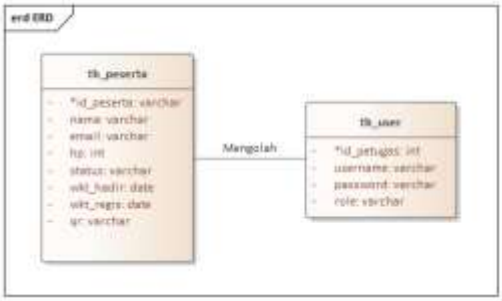
Gambar 3 *Sequence Diagram System prototype I*

4) *Behavioral State Machine Diagram*
Behavioral state machine diagram atau biasa disebut juga dengan *state machine diagram* merupakan salah satu *behavioral diagram* yang memodelkan *state* berbeda yang suatu *class* lewati pada saat masa aktifnya sebagai respon terhadap suatu kejadian beserta respon balasan dan aksinya. Pada *system prototype I* ini dibagi menjadi dua buah *state machine diagram* sesuai dengan *role* yang ada pada aplikasi sehingga menggambarkan alur *class* yang bekerja pada *role* tersebut. Berikut merupakan gambar dari *state machine diagram role* petugas.



Gambar 4. *Behavioral State Machine Diagram System prototype I*

5) *Entity Relationship Diagram*
Diagram di atas menggambarkan hubungan antara tabel yang digunakan pada *system prototype I* ini. Hanya terdapat dua tabel yaitu tabel tb_peserta yang berisikan *field* data peserta dan tabel tb_user yang berisikan *field* data petugas. Hubungan antara dua tabel ini adalah isi dari tb_peserta diolah oleh petugas yang terdapat di tb_user.



Gambar 5 *ERD System prototype I*

c. ***Implementation***

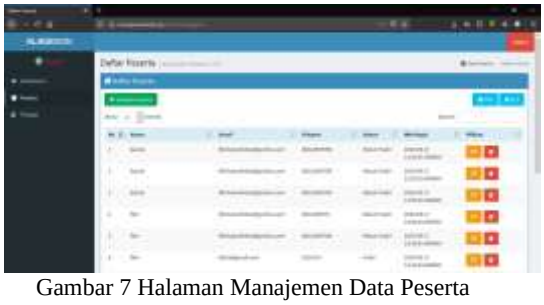
1) *Login*

Untuk bisa mengakses aplikasi, pengguna harus melakukan *login* terlebih dahulu menggunakan *username* dan *password* yang sudah terdaftar pada aplikasi. Pemberian fitur *login* disini ditujukan sebagai akses kontrol terhadap aplikasi. Terdapat dua *role* pada aplikasi ini yaitu admin dan petugas. Pengguna yang memasukan *username* dan *password* dengan benar akan bisa masuk ke halaman *dashboard* aplikasi. Pengguna yang salah memasukan *username* dan/atau *password* akan diarahkan kembali ke halaman *login* yang baru.

Gambar 6 Halaman *Login*

2) Manajemen Data Peserta

Proses implementasi dari manajemen data peserta terletak pada halaman khusus yang menampilkan daftar data peserta. Pada halaman ini terdapat lima fitur pada aplikasi antara lain tambah peserta, sunting peserta, hapus peserta, dan ekspor menjadi data pdf dan xls. Data peserta yang dapat diolah antara lain nama lengkap, email, nomor handphone, dan status kehadiran dari peserta. Selain data tersebut, secara otomatis sistem akan membangkitkan *id* peserta, waktu pendaftaran (proses tambah peserta), dan *string QR Code* ke *database*.



Gambar 7 Halaman Manajemen Data Peserta

3) Manajemen Data Petugas

Proses manajemen data petugas terletak pada halaman khusus yang hanya dapat diakses oleh *role* admin yaitu halaman daftar data petugas. Pada halaman ini ditampilkan data petugas berupa username dan *role*. Terdapat tiga fitur untuk melakukan manajemen data petugas antara lain tambah petugas, sunting petugas, dan hapus petugas. Data petugas yang diolah antara lain *username, password*, dan *role*.

Gambar 8 Halaman Manajemen Data Petugas

d. ***Testing***

1) *Unit Testing*

Pengujian ini akan dilakukan terhadap fungsi-fungsi yang terdapat pada enam kelas yang ada pada aplikasi. *Unit testing* dilakukan menggunakan metode *blackbox* dengan tujuan memastikan setiap fungsi pada kelas aplikasi berjalan dengan benar. Hasil dari pengujian menunjukan hasil yang sesuai semua.

2) *Integration Testing*

Pada penelitian ini terdapat dua jenis integration testing yang akan dilakukan, *user interface testing* dan *use scenario testing*. *User interface testing* ditujukan untuk menguji tampilan aplikasi saat diberikan inputan yang valid maupun tidak valid. Dari 10 halaman yang diuji kesemuanya menunjukan hasil yang sesuai.

Pengujian *use scenario testing* dilakukan dengan memberikan skenario terhadap aplikasi sesuai dengan skenario desainnya. Tujuannya adalah melihat apakah aplikasi sanggup melakukan skenario yang diberikan. Dari sembilan skenario yang diujikan semuanya menghasilkan hasil yang sesuai.

3) *Performance Testing*

Performance testing dilakukan dengan memanfaatkan *automated tools* GTmetrix. Aspek performa yang diuji pada pengujian ini diantaranya *first contentful paint, speed index, largest contentful paint, time to interactive, total blocking time*, dan *cumulative layout shift*. Dari pengujian ini didapatkan hasil pengujian bernilai grade B atau sudah baik namun dapat diperhitungkan untuk perbaikannya.

4) *Security Testing*

Pengujian dilakukan dengan menjalankan *test case* dan kesuaian hasilnya. Pada *system prototype I* ini pengujian dijalankan pada lingkungan aplikasi manajemen. Terdapat lima

buah *test case* yang diujikan dan kesemuanya sesuai.

5) User Acceptance Testing

Pada penelitian ini UAT dilakukan dengan menggunakan metode likert dengan memberikan pertanyaan seputar fungsionalitas, tampilan, dan *user experience* dari aplikasi. Dari pengujian ini kemudian didapatkan kebutuhan untuk *system prototype II*. Hasil pengujian UAT ini disetujui dengan persentase 91.43%.

4.3. Penelitian Tahap II

a. ***Analysis***

Pada analysis tahap II ini kebutuhan fungsional dan non-fungsional dari aplikasi akan diidentifikasi dan dianalisis melalui hasil UAT pada tahap I. Selain itu pada penelitian tahap II ini juga akan mulai dilakukan integrasi dengan aplikasi registrasi KLiKS dan aplikasi gawai KLiKS sehingga terdapat beberapa kebutuhan untuk menyelaraskan kinerja ketiga aplikasi.

1) Kebutuhan Fungsional

Kebutuhan fungsional atau fitur utama pada tahap kedua ini antara lain:

a. Aplikasi memiliki fitur layanan REST API yang dapat melakukan olah data peserta dan petugas yang terdiri dari ambil semua data, ambil data berdasarkan *id* dan nilai QR Code, tambah data, hapus data, dan sunting data.
b. Aplikasi memiliki fitur keamanan REST API berupa JSON Web Token (JWT) untuk menghindari serangan Broken User Authentication pada layanan API.
c. Aplikasi memiliki halaman dashboard.
d. Data pribadi peserta berupa nama, nomor handphone, email, dan NIK disimpan pada *database* dalam kondisi tereknripsi.
e. Pada fitur sunting peserta oleh admin, status peserta tidak bisa diubah.
f. Penambahan keterangan waktu dan tanggal pada format penamaan berkas unduhan pdf dan xls.
g. Perubahan nama tabel data peserta dan penambahan field NIK.

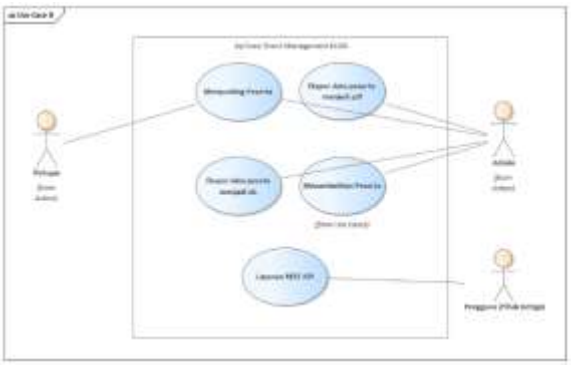
2) Kebutuhan Non Fungsional

Kebutuhan non fungsional pada aplikasi tahap kedua ini berupa perbaikan tampilan pada halaman tambah dan sunting peserta yaitu menghilangkan tombol atur angka pada field nomor hp.

b. ***Design***

1) *Use case Diagram*

Hubungan antara *user* dengan sistem pada penelitian tahap II digambarkan dengan diagram *use case* pada Gambar 8 dan Gambar 9.



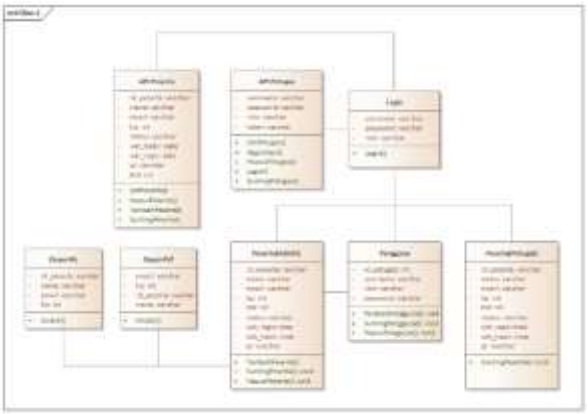
Gambar 8 *Use case Diagram System prototype II (a)*

Adapun *use case* Layanan REST API dapat dijabarkan kembali menjadi beberapa *use case* yang khusus menjelaskan mengenai lingkungan REST API seperti yang digambarkan pada *use case* dibawah ini.

Gambar 9 *Use case Diagram System prototype II (b)*

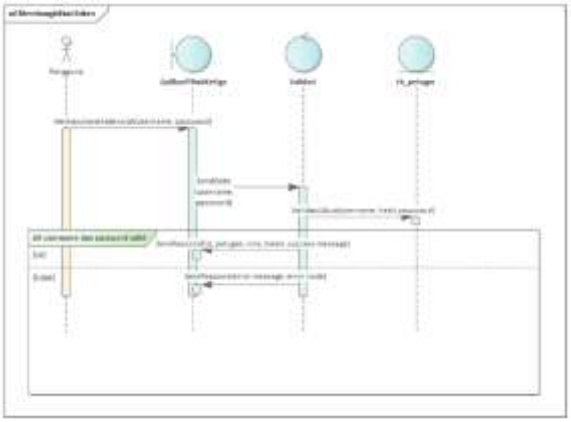
2) *Class Diagram*

Berikut merupakan *class diagram system prototype II*.



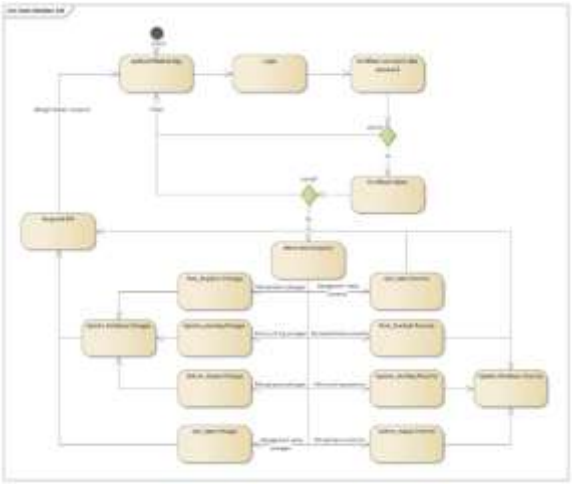
Gambar 10 *Class Diagram System prototype II*

3) *Sequence Diagram*
Sesuai dengan jumlah *use case*, pada penelitian tahap II ini terdapat lima belas *sequence diagram*. Berikut merupakan salah satu contoh dari *sequence diagram system prototype II*.



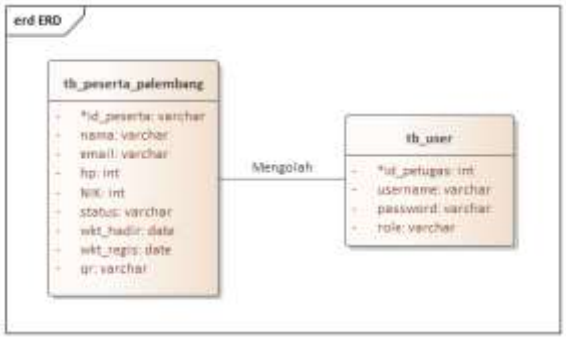
Gambar 11 *Sequence Diagram System prototype II*

4) *Behavioral State Machine Diagram*
Behavioral state machine diagram pada *system prototype II* dibagi menjadi tiga diagram sesuai dengan jumlah aktor pada *use case diagram* yaitu petugas, admin, dan pengguna. Berikut merupakan contoh dari *state machine diagram* pengguna.



Gambar 12 *Behavioral State Machine Diagram System prototype II*

5) *Entity Relationship Diagram*
Pada penelitian II ini tidak banyak mengubah *database* dan hubungannya. Masih tetap ada dua tabel dengan hubungan bahwa petugas memiliki akses untuk mengolah data peserta. Perbedaan yang ada hanyalah penamaan tabel dimana tabel untuk peserta yang awalnya bernama tb_peserta diubah menjadi tb_peserta_palembang untuk menyesuaikan dengan proses bisnis aplikasi registrasi KLiKS.



Gambar 13 *ERD System prototype II*

c. ***Implementation***

1) Enkripsi - Dekripsi *Database*
Fitur ini merupakan masukan pada UAT I sebagai upaya untuk melindungi data pibadi peserta KLiKS. Sebelum disimpan di *database* data pribadi peserta yaitu nama lengkap, nomor handphone, email, dan NIK melewati proses enkripsi data. Dengan memanfaatkan *library open*SSL data peserta dienkripsi menggunakan algoritma AES-256 dengan kunci dan iv yang di-*hash* menggunakan SHA 256.

2) Manajemen Data Peserta
Pada bagian manajemen peserta, *system prototype II* hanya membutuhkan sedikit perbaikan pada fiturnya dan juga proses kerjanya yaitu proses enkripsi – dekripsi *database*. Selain itu pada *system prototype* II ini juga terdapat penambahan halaman yaitu halaman dashboard yang menampilkan data simpulan kehadiran peserta dan petugas

3) REST API
REST API yang dibangun ditambahkan sistem keamanan JWT sehingga *request* yang diajukan harus berasal dari pengguna yang tervalidasi. Fitur yang dibangun antara lain tambah, sunting, hapus data peserta dan petugas. Akan tetapi semua fitur tersebut baru bisa dilakukan setelah melakukan *login* dan mendapatkan JWT.

Gambar 14 Tampilan Komunikasi API

d. ***Testing***

1) *Unit Testing*
Unit Testing dilaksanakan dengan menguji fungsi setiap kelas pada fokus pengembangan aplikasi di *system prototype II*. Dari lima kelas yang diujikan dengan beberapa fungsi didalamnya semua

menghasilkan hasil yang sesuai semua.

2) Integration Testing

Sama seperti pada *system prototype I, Integration Testing* dilakukan menggunakan dua pendekatan, *User Interface Testing* dan *Use Scenario Testing*. Untuk *system prototype* II ini terdapat enam halaman yang diuji pada *user interface testing* dan 15 skenario yang diujikan pada *use scenario testing* dan semuanya sesuai.

3) Performance Testing

Dalam menguji nilai performa aplikasi, aspek yang diuji pada pengujian ini diantaranya *first contentful paint, speed index, largest contentful paint, time to interactive, total blocking time,* dan *cumulative layout shift*. Pada *performance testing system prototype II* didapatkan hasil *grade* B dengan nilai *performance* 90%. *Performance* dari web dinyatakan sudah baik dan tidak memerlukan perbaikan tambahan.

4) Security Testing

Pengujian ini ditujukan untuk menguji keamanan dari API yang dibangun pada *system prototype II* ini terhadap serangan *Broken User Authentication*. Metode yang digunakan adalah dengan menggunakan *test case* berdasarkan *test case for web security testing* dari penelitian Kundu [12]. Dari beberapa pengujian terdapat pengujian untuk menguji ketahanan terhadap kerawanan *broken user authentication* yaitu dengan melakukan *direct request* dan memasukan kredensial yang salah. Hasilnya API masih tahan dan menghasilkan hasil yang sesuai dengan harapan.

5. KESIMPULAN

Berdasarkan hasil rangkaian penelitian dari mulai identifikasi masalah sampai pada dua tahap penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

a) Aplikasi *event management* KLiKS yang dibangun dapat mengatasi permasalahan manajemen data peserta KLiKS sehingga menjadi bebas dari aplikasi pihak ketiga dengan data peserta yang sinkron disetiap tahapnya.

b) Aplikasi *event management* KLiKS mendukung proses integrasi dengan aplikasi registrasi KLiKS dan aplikasi gawai KLiKS melalui penggunaan *database* yang sama dan layanan REST API.

c) Implementasi JWT pada REST API dapat mengamankan layanan API dari kerawanan *Broken User Authentication.* Hal ini dibuktikan melalui beberapa hasil pengujian yang menyatakan bahwa API dapat melakukan otorisasi *request* yang dikirimkan pengguna sehingga setiap *request* yang dieksekusi berasal dari pengguna yang terautentikasi.

REFERENSI

[1] Peraturan Presiden Republik Indonesia Nomor 53 Tahun 2017 Tentang Badan Siber dan Sandi Negara, 2017.

[2] B. E. Sukmono, “Transkrip Wawancara.” [Wawancara], 2020.

[3] BSSN, KLiKS BSSN Untuk Negeri. Jakarta, 2019.

[4] V. Wirawan, “Penerapan E-Government dalam Menyongsong Era Revolusi Industri 4.0 Kontemporer di Indonesia,” J. Penegakan Huk. dan Keadilan, 2020, doi: 10.18196/jphk.1101

[5] E. Cabezas, “Leveraging the OWASP API Security Top 10 to Build Secure Web Service,” SANS Institute, 2020.

[6] The OWASP Foundation, “OWASP API Security Top 10 2019,” 2019. [Online]. Available: https://raw.githubusercontent.com/OWASP/API-Security/master/2019/en/dist/owasp-api-security-top-10.pdf.

[7] M. Shah, V. Shewale, M. Sukal, and S. Yadav, “Event Management,” Int. Res. J. Eng. Technol., vol. 6, no. 1, pp. 1507–1511, 2020.

[8] M. Mark, REST API *Design* Rulebook. 2013.

[9] P. Iversen, “Specification-based security analysis of REST APIs,” no. June, 2018, [Online]. Available: https://brage.bibsys.no/xmlui/bitstream/handle/11250/2560780/18062_FULLTEXT.pdf?sequence=1&isAllowed=y

[10] A. Dennis, B. H. Wixom, and R. M. Roth, *System* Analysis and *Design* 5th Edition. 2012.

[11] P. D. Sugiyono, Metode Penelitian Kuantitatif, Kualitatif, dan R&D. 2016.

[12] S. Kundu, “Web Testing : Tool , Challenges and Methods,” vol. 9, no. 2, pp. 481–486, 2012.