

# Pengembangan Aplikasi web untuk Pre-Processing Business Intelligence dengan Integrasi ChatGPT, dan Google OAuth 2.0

M Yusril Fauzan Harahap  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
yusrilhrp@student.telkomuniversity.ac.id

Roswan Latuconsina, ST. MT  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
roswan@telkomuniversity.ac.id

Dr. Purba Daru Kusuma, ST., MT  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
purbodaru@telkomuniversity.ac.id

**Abstrak**— Dalam era digital yang berkembang dengan pesat, kemampuan untuk mengelola dan memanfaatkan data telah menjadi salah satu faktor kunci dalam menentukan keberhasilan suatu organisasi. Aplikasi web untuk *pre-processing business intelligence* dikembangkan untuk mengintegrasikan teknologi kecerdasan buatan dan Google OAuth 2.0 untuk kemudahan pengguna dalam menggunakan aplikasi. Pengembangan ini melibatkan pembuatan frontend, integrasi database, serta integrasi API untuk autentikasi dan analisis data.

**Kata kunci**— aplikasi web, API, *business intelligence*, frontend, database

## I. PENDAHULUAN

Dalam era digital yang berkembang dengan pesat, kemampuan untuk mengelola dan memanfaatkan data telah menjadi salah satu faktor kunci dalam menentukan keberhasilan suatu organisasi. Bagi perusahaan baru yang ingin bersaing di pasar global, pemanfaatan data secara efektif tidak hanya sekedar kebutuhan, tetapi merupakan suatu keharusan. Data yang ada dalam sebuah perusahaan memiliki potensi besar untuk memberikan wawasan berharga yang dapat membantu dalam pengambilan keputusan strategis, peningkatan efisiensi operasional, serta dalam mempertajam daya saing.

Perkembangan teknologi informasi yang pesat telah mempengaruhi berbagai aspek kehidupan manusia, termasuk dalam bidang komunikasi dan pengolahan data. Aplikasi web kini menjadi salah satu alat yang paling banyak digunakan untuk berbagai keperluan, mulai dari layanan informasi hingga transaksi bisnis. Dalam beberapa tahun terakhir, kebutuhan untuk mengintegrasikan berbagai layanan dalam satu platform semakin meningkat, baik itu integrasi dengan sistem autentikasi eksternal seperti Google OAuth 2.0 maupun penambahan fitur cerdas seperti *chatbot* berbasis AI.

Dengan adanya kebutuhan tersebut, pengembangan aplikasi web yang mampu menyediakan fitur-fitur ini menjadi penting. Penggunaan teknologi seperti MySQL untuk manajemen data pengguna, Google OAuth 2.0 untuk autentikasi yang lebih aman, dan integrasi ChatGPT untuk interaksi pengguna yang lebih cerdas, dapat meningkatkan fungsionalitas aplikasi secara signifikan.

Dalam pengembangan aplikasi web modern, terdapat beberapa tantangan yang perlu diselesaikan antara lain seperti Keseluruhan Aplikasi, Autentikasi Pengguna, serta Interaksi

pengguna merupakan bagian penting dalam pengembangan aplikasi ini sehingga memerlukan solusi yang efektif dan efisien.

## II. KAJIAN TEORI

Dalam Pengembangan aplikasi web ini PHP digunakan sebagai bahasa pemrograman utama untuk sisi frontend. PHP yang merupakan singkatan dari "*Hypertext Preprocessor*" adalah bahasa skrip server-side yang sangat populer dan sering digunakan untuk pengembangan web karena kemampuannya untuk menghasilkan konten dinamis dan efisien. Pada aplikasi ini, PHP berperan dalam mengelola dan memproses data dari server sebelum disajikan kepada pengguna, serta untuk membangun halaman web yang interaktif.

Di dalam struktur PHP, HTML digunakan untuk menyusun dan mengatur elemen-elemen dasar dari halaman web, seperti teks, gambar, dan form. HTML menyediakan kerangka dasar yang memungkinkan PHP untuk menyisipkan konten dinamis secara *real-time*. Selain itu, *Javascript* digunakan untuk menambahkan interaktivitas pada halaman, seperti navigasi, dan juga validasi form. *Javascript* memungkinkan pengalaman pengguna yang lebih responsif dan cepat. Selain itu, CSS (*Cascading Style Sheets*) digunakan untuk mengatur tampilan dan tata letak halaman web. CSS memisahkan gaya visual dari konten dan logika, memungkinkan untuk menyesuaikan tampilan web tanpa mempengaruhi fungsionalitas atau struktur dasarnya. Dalam aplikasi web ini.

Untuk menyimpan *database* pengguna digunakan MySQL sebagai sistem manajemen basis data (DBMS) serta mengelola semua informasi yang berkaitan dengan pengguna serta data lain seperti file yang diupload oleh pengguna. MySQL yang merupakan salah satu DBMS *open-source* yang populer mudah digunakan dalam interpretasi untuk data user.

Aplikasi ini juga mengintegrasikan Google OAuth 2.0 sebagai metode autentikasi pengguna. OAuth 2.0 adalah protokol terbuka yang memungkinkan pengguna untuk login ke aplikasi web dengan menggunakan akun Google mereka tanpa harus mendaftar akun.

Untuk meningkatkan interaksi pengguna serta proses analisis data yang lebih mudah, aplikasi ini menggunakan API dari OpenAI ChatGPT yang didukung oleh teknologi kecerdasan buatan dari OpenAI, melalui integrasi API ini,

pengguna dapat berkomunikasi dengan kolom chat yang dapat menjawab pertanyaan dari pengguna secara langsung.

### III. METODE

Dalam pembuatan *frontend*, *database*, serta integrasi Google OAuth 2.0 dan juga API ChatGPT digunakan beberapa metode yang dimulai dari :

#### A. Pembuatan *Frontend* Aplikasi

Hal pertama yang diperlukan dalam membuat dan mengembangkan aplikasi ini adalah membuat struktur aplikasi dengan PHP serta seluruh proses yang ada didalam aplikasi untuk tiap halaman. Dimulai dari proses awal pengguna melakukan Login, Daftar, Lupa Password, Input File, View Data, dan Visualisasi. Semua halaman yang ada di dalam aplikasi dibuat dengan struktur yang sesuai dengan proses yang ada dan juga dilakukan penyesuaian halaman agar membuat tampilan lebih menarik dan proses didalam aplikasi dapat dilakukan dengan mudah oleh pengguna.

#### B. Pembuatan dan Integrasi *Database*

*Database* yang digunakan untuk menyimpan data pengguna, kepemilikan *file* yang telah diunggah agar hanya pengguna tertentu yang bisa melihat *file* yang mereka unggah, serta menyimpan *file* yang telah diunggah kedalam database menggunakan database MySQL. Penggunaan MySQL cukup mudah karena merupakan salah satu database yang cukup populer sehingga tidak asing dalam proses integrasinya.

#### C. Integrasi API ke aplikasi

API yang digunakan adalah Google OAuth 2.0 dan juga API *OpenAI* GPT. OAuth 2.0 merupakan API yang digunakan untuk memudahkan pengguna dalam melakukan *Login* ke dalam aplikasi ini sehingga pengguna bisa untuk tidak mendaftar akun terlebih dahulu dan langsung masuk menggunakan akun Google yang mereka miliki.

API *OpenAI* GPT digunakan untuk pengguna dalam melakukan bantuan dalam menganalisis data hasil visualisasi, pengguna dapat melakukan interaksi pada kolom chat yang telah disediakan dan dapat bertanya langsung dengan *bot* yang terdapat kecerdasan buatan didalamnya.

### IV. HASIL DAN PEMBAHASAN

#### A. Pembuatan *Frontend* Aplikasi

Pembuatan *frontend* aplikasi dilakukan terlebih dahulu untuk proses kerangka dan struktur tiap halaman aplikasi. Proses penulisan kode pemrograman dilakukan pada aplikasi Visual Studio Code dengan urutan seperti berikut :

##### a. Pembuatan *Landing Page*

*Landing page* merupakan halaman awal aplikasi yang menampilkan garis besar mengenai fungsi dari aplikasi ini. Setelah aplikasi ini dibuat, maka pada file *.htaccess* didalam folder aplikasi ini dituliskan perintah agar ketika aplikasi dijalankan maka halaman awal yang akan ditampilkan adalah *Landing page* ini.

##### b. Pembuatan halaman Login

Halaman login berisikan logo aplikasi Visual Hub di bagian kiri halaman, serta di bagian kanan halaman terdapat form yang harus diisi oleh pengguna untuk bisa masuk kedalam aplikasi. Jika pengguna belum memiliki akun, maka

bisa menekan tombol "*Register*" untuk membuat akun baru dan jika pengguna sudah memiliki akun tetapi lupa kata sandi, maka bisa menekan tombol "*Lupa password*" dan juga jika pengguna ingin masuk ke aplikasi menggunakan akun google mereka, maka bisa langsung menekan tombol "*Login with Google*".

##### c. Pembuatan halaman Daftar

Pada halaman daftar, pengguna harus mengisi form yang ada pada bagian kanan halaman berupa identitas seperti "*username*" "*email*" dan "*Password*" yang kemudian akan disimpan kedalam database MySQL sebagai data pengguna.

##### d. Pembuatan halaman Lupa password

Pada halaman Lupa password, pengguna diminta untuk mengisi *Username* yang dimiliki dan diminta untuk mengisi *email* yang bertujuan untuk melakukan verifikasi terhadap akun yang telah terdaftar, proses ini berhubungan dengan data pengguna yang tersimpan di database.

##### e. Pembuatan halaman Input Data

Pada halaman input data, pengguna diminta untuk melakukan *upload file* dalam format *.csv* yang nanti akan disimpan kedalam database *file*, di halaman ini tidak hanya fitur *upload* saja, melainkan terdapat navigasi halaman untuk pengguna berpindah ke halaman lain seperti *View data*, visualisasi, dan profil.

##### f. Pembuatan halaman Visualisasi

Pada halaman visualisasi berisikan tampilan dari grafik yang telah dihasilkan dari proses algoritma aplikasi dengan berbagai macam grafik seperti *bar chart*, *line chart*, *pie chart*, *donut chart*, *scatter plot*, *table*, serta dibagian bawah halaman visualisasi terdapat kolom chat yang telah diintegrasikan dengan kecerdasan buatan ChatGPT untuk pengguna bertanya secara langsung.

##### g. Pembuatan halaman Profil

Pembuatan halaman profil bertujuan untuk menampilkan identitas pengguna berupa *username*, *email*, *password* yang dirahasiakan, di halaman ini juga terdapat perintah *logout* untuk pengguna mengakhiri sesi dan keluar dari aplikasi serta fitur tambahan seperti foto profil pengguna.

#### B. Pembuatan dan Integrasi *Database*

Database yang digunakan adalah MySQL, database berfungsi untuk menyimpan data dan informasi pengguna mulai dari akun, kepemilikan *file*, dan juga foto profil pengguna. Langkah awal adalah membuat database dan menghubungkannya melalui perintah kode yang dijalankan di Visual Studio Code.

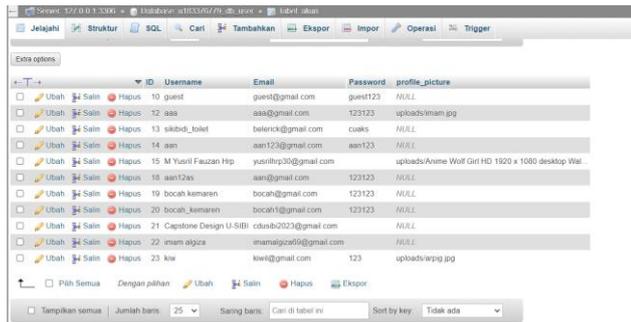
```
$db_servername = "localhost";
$db_username = "root";
$db_password = "";
$db_name = "db_user";

// Buat koneksi
$conn = new mysqli($db_servername, $db_username, $db_password, $db_name);

// Periksa koneksi
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
```

Gambar 1 Koneksi ke database db\_user

Gambar diatas merupakan perintah kode yang akan menghubungkan koneksi antar aplikasi kedalam *database db\_user* yang berisikan informasi pengguna.



Gambar 2 Database User

Gambar diatas menunjukkan pengguna yang telah berhasil melakukan *login* kedalam aplikasi. Tabel menunjukkan informasi pengguna berupa *username*, *email*, *password*, *profile picture* serta ID yang menjadi *primary key* untuk memisahkan sesi tiap akun.

C. Integrasi API

Penggunaan API Google OAuth 2.0 membutuhkan langkah langkah yang harus diatur pada Google Console. Setelah melakukan *login* ke Google console, maka berpindah ke bagian APIs & Service, lalu pilih Credentials, kemudian membuat credentials baru di bagian OAuth 2.0 untuk mendapatkan Client ID yang nantinya digunakan di kodingan



Gambar 3 API Google OAuth 2.0

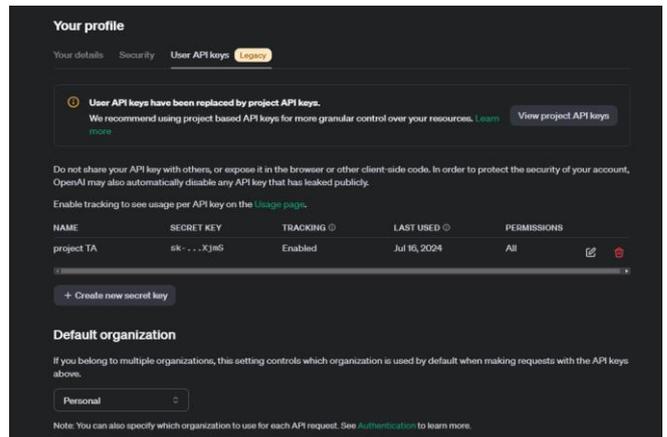
Client ID tadi dimasukkan kedalam kodingan halaman *login* agar bisa menampilkan pilihan *login* dengan akun *google*

```
<div id="g_id_onload"
  data-client_id="757773955445-pkhne824djmh68bd9s7bh75kj6ppnp4.apps.googleusercontent.com"
  data-callback="handleCredentialResponse">
</div>
```

Gambar 4 Client ID API

Setelah itu *login* sudah bisa menggunakan akun google pengguna tetapi tidak lupa untuk membuat file kodingan yang akan menyimpan informasi *login* ini yang nantinya akan menyimpan *username* dan *email* pengguna ke dalam database.

Setelah API Google OAuth 2.0 berhasil diintegrasikan kedalam aplikasi, API yang digunakan selanjutnya adalah API *OpenAI* ChatGPT. Sama seperti google console harus mendaftarkan akun terlebih dahulu untuk melakukan pembelian API dan akan mendapatkan *secret key* dari *OpenAI* untuk nantinya dipanggil di kodingan.



Gambar 5 OpenAI API Key

Setelah mendapatkan secret key, langkah selanjutnya adalah mengikuti *tutorial* yang disediakan untuk melakukan pemanggilan terhadap API kedalam aplikasi dengan membuat *file* baru untuk integrasi API tersebut.

```
// Proses permintaan ke OpenAI API
$apiKey = "sk-proj-gL45...";

$data = [
  "model" => "gpt-3.5-turbo",
];

$ch = curl_init("https://api.openai.com/v1/chat/completions");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, [
  'Content-Type: application/json',
  'Authorization: Bearer ' . $apiKey,
]);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));

$response = curl_exec($ch);
```

Gambar 6 Integrasi API Key

Setelah proses integrasi API *key* dilakukan, maka langkah selanjutnya adalah membuat kolom chat tersebut di halaman visualisasi.

```
</div>
<div class="chat-section">
  <h2>Chat dengan ChatGPT</h2>
  <div class="chat-box" id="chat-box">
    <!-- Pesan Chat akan ditampilkan di sini -->
  </div>
  <form id="chat-form">
    <input type="text" id="user-message" placeholder="Ketik pesan Anda di sini..." required autocomplete="off">
    <button type="submit">Kirim</button>
  </form>
</div>
```

Gambar 7 Membuat chatbox

Setelah pembuatan kolom chat untuk pengguna berinteraksi, maka semua fitur yang pengguna akan lakukan untuk bertanya akan otomatis bisa digunakan selagi saldo di *OpenAI* masih ada dan belum habis.

D. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk melakukan pengecekan terhadap aplikasi dari berbagai aspek, pada bagian ini dilakukan *Penetration test* untuk mengetahui celah yang bisa diakses oleh pihak terlarang. *Penetration test* dilakukan dalam beberapa tahapan seperti :

- a. Pemetaan jaringan dengan Nmap
  - Dilakukan pemetaan jaringan untuk menguji IP terhadap port yang terbuka

```

PS C:\Users\user> nmap -O 192.168.18.39
Starting Nmap 7.95 ( https://nmap.org ) at 2024-08-08 10:14 SE Asia Standard Time
Nmap scan report for 192.168.18.39
Host is up (0.88834s latency)
Not shown: 993 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
2222/tcp  open  EtherNet/IP-1
3306/tcp  open  mysql
Device type: general purpose
Running: Microsoft Windows 10[11]
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_11
OS details: Microsoft Windows 10 1607 - 11 23H2
Network Distance: 0 hops

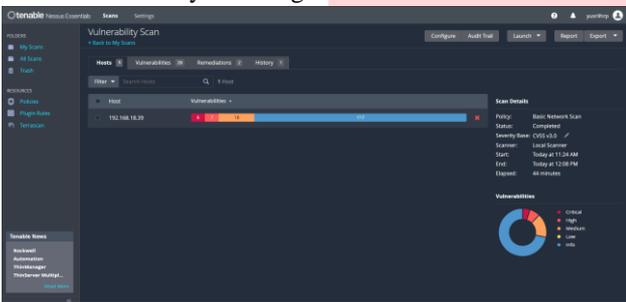
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.51 seconds

```

Gambar 8 Pemetaan dengan Nmap

Port yang terbuka pada aplikasi antar lain HTTP, HTTPS, MySQL, dan yang lainnya, serta sistem operasi yang digunakan adalah Windows. Pengujian ini bertujuan untuk mengidentifikasi dan mengeksploitasi kerentanan dalam sistem.

b. Vulnerability scanning



Gambar 9 Vulnerability scanning

Vulnerability Scanning bertujuan untuk melakukan pengecekan aplikasi keseluruhan untuk mengidentifikasi sejumlah kerentanan. Dari hasil tersebut terlihat berbagai jenis kerentanan dari Critical, High, Low. Beberapa hasil tersebut melibatkan versi OpenSSL dan Apache yang rentan yang mengakibatkan celah ini menjadi sangat mudah dieksploitasi oleh penyerang.

c. Eksploitasi kerentanan

| Host                                   | Strength | Progress | Elapsed          | Reqs         | Alerts     | Status |
|--|----------|----------|------------------|--------------|------------|--------|
| XML External Entity (XXE)              | Medium   | 100%     | 00:00:11         | 0            | 0          | ✓      |
| Generic Padding Oracle                 | Medium   | 100%     | 00:00:082        | 0            | 0          | ✓      |
| Cloud Metadata Potentially Exposed     | Medium   | 100%     | 00:00:006        | 4            | 0          | ✓      |
| Server Side Template Injection         | Medium   | 100%     | 00:07:333        | 1064         | 0          | ✓      |
| Server Side Template Injection (Blind) | Medium   | 100%     | 00:06:248        | 912          | 0          | ✓      |
| Directory Browsing                     | Medium   | 100%     | 00:03:058        | 490          | 91         | ✓      |
| Buffer Overflow                        | Medium   | 100%     | 00:00:964        | 76           | 0          | ✓      |
| Format String Error                    | Medium   | 100%     | 00:01:992        | 208          | 0          | ✓      |
| CRLF Injection                         | Medium   | 100%     | 00:03:490        | 532          | 0          | ✓      |
| Parameter Tampering                    | Medium   | 100%     | 00:03:985        | 532          | 0          | ✓      |
| ELMMH Information Leak                 | Medium   | 100%     | 00:00:007        | 1            | 0          | ✓      |
| Trace and Information Leak             | Medium   | 100%     | 00:00:161        | 73           | 0          | ✓      |
| .htaccess Information Leak             | Medium   | 100%     | 00:00:114        | 68           | 0          | ✓      |
| .env Information Leak                  | Medium   | 100%     | 00:00:101        | 68           | 0          | ✓      |
| Hidden File Finder                     | Medium   | 100%     | 00:00:120        | 50           | 2          | ✓      |
| Spring Actuator Information Leak       | Medium   | 100%     | 00:00:004        | 2            | 0          | ✓      |
| XSLT Injection                         | Medium   | 100%     | 00:04:400        | 588          | 0          | ✓      |
| GET for POST                           | Medium   | 100%     | 00:00:093        | 0            | 0          | ✓      |
| User Agent Fuzzer                      | Medium   | 100%     | 00:18:278        | 5880         | 695        | ✓      |
| Script Active Scan Rules               | Medium   | 100%     | 00:00:001        | 0            | 0          | ✓      |
| SOAP Action Spoofing                   | Medium   | 100%     | 00:00:007        | 0            | 0          | ✓      |
| SOAP XML Injection                     | Medium   | 100%     | 00:00:111        | 0            | 0          | ✓      |
| <b>Totals</b>                          |          |          | <b>02:47:632</b> | <b>25803</b> | <b>790</b> |        |

Gambar 10 Eksploitasi kerentanan

Proses Eksploitasi kerentanan dilakukan dengan aplikasi Zed Proxy Attack (ZAP) untuk mencoba mengeksploitasi aplikasi secara keseluruhan melalui IP yang digunakan. Hasil yang ditampilkan oleh ZAP menekankan pentingnya tindakan keamanan yang lebih ketat terhadap aplikasi untuk mengonfigurasi server web dengan benar dan memastikan bahwa informasi sensitif tidak dapat diakses oleh pihak yang tidak berwenang

V. KESIMPULAN

Pengembangan aplikasi Web ini dirancang untuk mendukung proses Business Intelligence dengan memanfaatkan integrasi teknologi ChatGPT dan juga Google OAuth 2.0. Proses pengembangan mencakup pembuatan frontend aplikasi, integrasi database MySQL untuk manajemen data, serta dilakukan pengujian berupa Penetration testing terhadap aplikasi. Hasil dari tiap tahapan pengembangan menunjukkan bahwa aplikasi dapat berfungsi dengan baik sesuai tujuan yang diharapkan, meskipun masih memerlukan beberapa langkah perbaikan dalam hal keamanan. Aplikasi ini bertujuan untuk mempermudah pengelolaan dan analisis data, serta meningkatkan interaksi pengguna melalui fitur-fitur cerdas seperti autentikasi yang aman dengan google OAuth 2.0 dan komunikasi interaktif dengan OpenAI API chatbot berbasis GPT

REFERENSI

- [1] S. Scheeps, *Business Intelligence for Dummies*. Hoboken, NJ, USA: John Wiley & Sons, 2008.
- [2] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for ETL processes," in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 14-22
- [3] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2016.
- [4] A. Solihin, *Pemrograman Web dengan PHP dan MySQL*. Yogyakarta, Indonesia: ANDI, 2016.
- [5] G. Santoso, J. Setiawan, and A. Sulaiman, "Development of OpenAI API-based chatbot to improve user interaction on the JBMS website," *G-Tech: J. Teknol. Terapan*, vol. 7, no. 4, pp. 1606-1615, 2023. doi: 10.33379/gtech.v7i4.3301.
- [6] "Evaluation of an OAuth 2.0 protocol implementation for web server applications," in *Proc. 2015 Int. Conf. Workshop Comput. Commun. (IEMCON)*, Vancouver, BC, Canada, Oct. 2015, pp. 1-6. doi: 10.1109/IEMCON.2015.7344461.