



## ***Clash-of-Clans API untuk Pengelolaan Klan***

**Mohamad Syafri Tuloli\*, Mukhlisulfatih Latief, Manda Rohandi**

Program Studi Sistem Informasi, Fakultas Teknik, Universitas Negeri Gorontalo, Indonesia

***\*Penulis korespondensi, email: [syafri.tuloli@ung.ac.id](mailto:syafri.tuloli@ung.ac.id)***

### ***Abstract***

*Many factors influenced game success, including the player's participation in developing the game. Player involvement can be facilitated by providing Application Programming Interface (API) so that players can use it to reach their objectives, this strategy is used by Clash-of-Clans (COC) developer. There are two main problems in the COC game clan management: there are no indicators of player activity, and there is also no good clan war strategy. This research uses prototyping software design and development methods to explore the use of COC API services to overcome these problems. The results showed that the COC API service was able to support the development of clan management applications, succeeded in increasing player activity (in total online time for all players), and the proposed war strategy algorithm was proven to increase war results (by 12%).*

***Keywords:*** *Application Programming Interface; Clash-of-Clans; mobile games*

### **Abstrak**

Kesuksesan suatu permainan sangat dipengaruhi oleh banyak hal, termasuk keterlibatan pemain untuk ikut mengembangkan permainan. Keterlibatan pemain dapat difasilitasi dengan penyediaan *Application Programming Interface (API)* sehingga membuat pemain dapat memanfaatkannya untuk memenuhi tujuan-tujuannya. Sebagai permainan yang sukses, *Clash-of-Clans (COC)* juga telah memberikan layanan *API*. Terdapat dua permasalahan utama dalam permainan *COC* terkait pengelolaan klan, yaitu belum adanya indikator keaktifan pemain, dan juga belum ada pengaturan strategi perang yang baik. Penelitian ini menggunakan metode perancangan dan pengembangan perangkat lunak prototyping melakukan eksplorasi terhadap penggunaan layanan *COC API* untuk mengatasi permasalahan tersebut. Hasil menunjukkan bahwa layanan *COC API* mampu untuk mendukung pengembangan aplikasi pengelolaan klan, berhasil meningkatkan keaktifan pemain (dalam total waktu *online* seluruh pemain), dan penghitungan strategi perang terbukti meningkatkan hasil perang (sebesar 12%).

**Kata kunci:** *Application Programming Interface; Clash-of-Clans; permainan mobile*

## PENDAHULUAN

*Game* (permainan) pada zaman sekarang sudah banyak berkembang. *Game* sudah banyak dipergunakan bukan hanya untuk hiburan, tetapi sudah untuk peran-peran lain seperti untuk pembelajaran ataupun untuk peran-peran baru yang tidak terpikirkan sebelumnya, seperti media untuk menumbuhkan kesadaran (Mortensen dkk, 2019), atau metode untuk meramalkan masa depan (Vervoort, 2019). Salah satu *game* sukses adalah *game Clash-of-Clans (COC)*, *game* ini sempat meledak menjadi *game mobile* yang paling populer dan mendatangkan banyak keuntungan bagi perusahaan pembuatnya (*SuperCell*), bahkan sudah beberapa tahun dari tahun peluncurannya (2012), *COC* masih memiliki popularitas yang baik di dunia, termasuk di kalangan pemain (*gamer*) di Indonesia.

Kesuksesan *COC* sudah dipelajari untuk membentuk model gamifikasi yang baik untuk meningkatkan motivasi karyawan (Nicholas dkk., 2015), juga diadopsi untuk peningkatan motivasi pelajar (Preist & Jones, 2015). *Game COC* memberikan kemungkinan untuk memunculkan pola interaksi kooperatif yang dinamis (tergantung *event*) untuk mencapai tujuan bersama (Makkarumpa, 2016). Pendorong kesuksesan *COC* juga disebabkan oleh aspek positif yang tinggi, tingkat *immersion* (pemain tenggelam dalam pengalaman bermain) yang tinggi, dan tingkat kompetensi yang dapat dirasakan langsung oleh pemain (Fahmi dkk., 2018).

Manfaat nyata dari *game COC* juga sudah dibuktikan dengan kemampuannya untuk meningkatkan kosa kata bahasa Inggris (Veronika dkk., 2018), meningkatkan kreatifitas (Rino Saputra, 2016), dan juga mempermudah mendapatkan teman *online* (Damayanti, 2015). Tentu saja semua *game*, termasuk *COC*, tetap saja memiliki permasalahan, seperti (tidak adanya) transfer keterampilan (Syarif dkk., 2011), efek buruk yang dapat muncul (Hirvy dalam Alanur dkk., 2016), termasuk diantaranya adalah kecanduan bermain (Damayanti, 2015), menjauh dari teman-teman nyata di lingkungannya (Dinata, 2017), dan juga mempengaruhi prestasi akademiknya (Saputra, 2016), termasuk terlibat jual beli yang tidak sah meskipun pendapat ini masih dalam perdebatan (Meliani, 2017; Yuliasutik, 2016).

Untuk menghasilkan *game* sukses seperti *COC*, banyak cara yang dilakukan oleh pengembang *game*, seperti penekanan pada kemudahan penggunaan (*usability*) dari *games* (Rajanen & Nissinen, 2015), pengaturan tingkat kesulitan permainan/tantangan (Reis dkk., 2019), menambahkan intelegensi buatan (Aruan & Mulyani, 2018), dll. Satu hal yang dapat meningkatkan ketertarikan bermain adalah keterlibatan pengguna untuk ikut mengembangkan permainan, misalnya dengan membuka *source-code*, perangkat, *library*, *game API*, dll. Dengan adanya keterlibatan pemain pada pengembangan dapat menimbulkan motivasi yang kuat untuk memainkan.

*SuperCell* juga melakukan hal yang serupa, yaitu membuka *API* yang dapat digunakan untuk mengembangkan permainan yaitu *COC API (developer.clashofclans.com)*. Beberapa pengembang/pemain menggunakan fasilitas *COC API* untuk melakukan pencarian data pemain/klan (*Clashofstats.com*), layanan peramalan waktu bermain (*loot*) yang terbaik (*www.clashofclansforecaster.com*), dan layanan-layanan lain. Layanan-layanan ini sangat membantu pemain, misalnya untuk mencari dan menilai pemain dan klan atau juga untuk menentukan waktu bermain yang tepat, terutama karena permasalahan pada *COC* yang memiliki waktu tunggu yang lama apabila terlalu banyak pemain yang *online* pada saat yang bersamaan (terutama pada tingkat trofi tinggi seperti *titan* dan *legend*).

## PERMAINAN COC

Layanan *COC API* memberikan banyak manfaat, seperti hanya layanan untuk memberikan informasi keadaan umum dari klan, seperti tingkat *rank*, jabatan, tingkat donasi/terima, dan informasi lain. Hal ini diperlukan oleh calon anggota untuk menentukan apakah klan tersebut adalah klan yang diinginkannya. Hal ini karena terdapat berbagai tujuan pendirian klan (*Push Clan*, *War Clan*, *Req & Leave*, Klan Pribadi/kelompok, dll). Dengan beragamnya jenis klan dan juga beragamnya karakteristik dari pemain, maka baik klan dan pemain membutuhkan suatu cara untuk menilai klan/pemain. Pemain dapat menggunakan informasi yang diberikan oleh layanan *COC API* mengenai informasi klan untuk menentukan apakah pemain cocok untuk masuk ke dalam klan tersebut.

### 1. Kurang indikator keaktifan

Dalam suatu klan, keaktifan pemain sangatlah penting, keaktifan dalam memberikan bantuan, dalam perang, ataupun sekadar dalam berkomunikasi (*chat*) akan membuat lingkungan permainan yang menyenangkan. Akan meningkatkan rasa kebersamaan dan kekompakan anggota pemain sehingga mempercepat progres peningkatan akun pemain (*trofi*, *level*, *experience*, dll), minimal untuk membuat pemain merasa betah untuk tinggal di klan tersebut.

Dalam aplikasi *COC* belum tersedia indikator yang kuat dalam menilai keaktifan pemain, seringkali pengelola klan (*leader/coleader/senior*) memerlukan waktu untuk menilai tingkat keaktifan pemain. Pengelola klan harus mengamati keaktifan pemain berdasarkan lama *online*, keterlibatan dalam berkomunikasi, keadaan *base*-nya (terawat atau tidak), perkembangan tingkat donasi/terimanya, dan statistik lainnya, sehingga memerlukan usaha yang lebih. Hal ini menjadi sulit terutama dengan jumlah pemain dalam suatu klan yang banyak (maksimal 50 pemain), dan apalagi untuk menyeleksi calon pemain (dari *global chat*) yang tertarik untuk masuk ke dalam klan yang jumlahnya sangat banyak setiap hari.

### 2. Pengaturan Perang yang tidak tepat

Dalam melakukan perang setiap klan memiliki pengaturan perang yang beragam. Paling banyak klan yang menggunakan pengaturan perangnya berdasarkan urutan *base*, yaitu pemain yang urutan *base*-nya nomor 1 ditargetkan untuk menyerang *base* musuh urutan nomor 1, nomor 2 serang nomor musuh nomor 2, dst. Pengaturan ini memiliki kelemahan besar yaitu urutan nomor *base* ini disusun oleh aplikasi berdasarkan nilai kekuatan pertahanan dari *base* pemain, yaitu berdasarkan tingkat kekuatan menara pertahanan, perangkap, dan tingkat kekuatan dinding dari *base*, sehingga tidak menggambarkan kekuatan serangan dari pemain. Sehingga pemain nomor 1 adalah pemain dengan kekuatan pertahanan paling tinggi, tapi belum tentu adalah pemain dengan kekuatan serangan yang paling tinggi, ataupun kemampuan strategi serangan yang paling tinggi.

Hal ini juga didukung dengan adanya pemain yang tidak seimbang antara kekuatan pertahanan dengan kekuatan serangannya. Misalnya pemain *minimax* (Djawa, 2018), yang terburu-buru dalam meningkatkan *town-hall*-nya untuk segera meningkatkan kekuatan serangannya, dan dengan sengaja tidak meningkatkan kekuatan pertahanannya. Hal ini membuat strategi berdasarkan urutan *base* akan membuat pemain jenis ini harus menyerang target yang rendah, hanya karena kekuatan pertahanan pemain ini lebih rendah, padahal tingkat kekuatan serangannya mampu untuk menyerang *base* urutan yang lebih atas.

Demikian pula sebaliknya, bagi pemain yang mementingkan peningkatan pertahanan dan tidak terlalu meningkatkan kekuatan serang, menggunakan strategi nomor urut *base*, terpaksa harus menyerang *base* dengan nomor urut atas, hanya karena *base*-nya berada pada nomor urut atas. Pemain ini akan mengalami kesulitan atau bahkan *zonk* (Djawa, 2018), karena menyerang *base* yang tidak mampu. Hal ini juga-lah yang membuat pemain dapat merasa tegang dan apalagi

pada saat tingkat tekanan tinggi (sedang ketinggalan poin) sehingga setiap kesalahan pemain dapat berakibat kekalahan klan (Fahmi dkk., 2018).

Selain itu ketidak sesuaian antara tingkat serangan dan pertahanan, juga terdapat ketidaksesuaian antara tingkat serangan dengan keterampilan pemain. Hal ini dapat terjadi karena dimungkinkannya transaksi jual beli akun pemain (Meliani, 2017), sehingga dapat saja pemain yang baru belajar bermain *COC* tetapi memiliki tingkat serangan yang *mentok* atau tingkat tertinggi pada kekuatan serangan dan pertahanannya (Djawa, 2018), atau juga karena akun pinjaman dari orang lain.

Hal lain yang mempengaruhi adalah juga tingkat fokus pemain yang bisa berubah-ubah, karena karakteristik pemain *COC* yang beragam lokasi dan latar belakangnya (siswa, mahasiswa, karyawan, dll) sehingga dapat memiliki kesibukan yang berbeda-beda setiap minggunya (Malik & Hidayah, 2017). Hal ini mempengaruhi keefektifan melakukan serangan, walaupun dengan kemampuan bermain yang tetap. Hal ini ditambah lagi dengan kebijakan klan yang kadang memperbolehkan untuk ikut perang sambil meningkatkan (*upgrade*) pahlawannya sehingga mempengaruhi kekuatan serangannya, dan harus diatur untuk diberikan target serang yang lebih mudah. Permasalahan lain juga adalah waktu aktif yang berbeda-beda, sehingga seringkali dibutuhkan untuk menunggu para pengelola untuk aktif dulu baru bisa memulai serangan, hal ini dapat mengganggu, karena dapat saja waktu *online* seorang pemain yang terbatas, membuatnya tidak dapat menunggu pengumuman target perang sehingga menjadi menyerang sesuka hatinya atau malah terlupa untuk menyerang.

Semua hal itu mengindikasikan perlunya suatu pengaturan strategi perang yang menggunakan kekuatan serangan pemain sebagai acuan dasarnya. Tetapi juga memperhitungkan faktor lainnya seperti tingkat fokus, keterampilan dan keadaan pahlawan (*heroes*) dari pemain. Pengaturan strategi ini harus bisa beradaptasi dengan perkembangan tingkat (*level*) dari serangan dari pemain secara otomatis, dan informasi ini harus dapat diakses oleh semua pemain.

Pembahasan mengenai strategi perang klan yang baik belum pernah dibahas secara ilmiah. Yang ada hanya pembahasan di forum mengenai pengelolaan perang pada klan masing-masing, ataupun tips-tips umum dalam pengelolaan perang di klan. *SuperCell* juga sudah memberikan fasilitas tambahan pada permainan untuk mengatur perang klan, yaitu: fasilitas pemain untuk memberitahukan target serang, atau pun rekomendasi target serang yang baik untuk pemain, tetapi kurang efektif. Hal ini karena pemain masih kurang baik dalam menentukan target serang yang efektif, dan rekomendasi target yang diberikan juga kurang dapat memperhatikan kondisi pemain (sibuk, meningkatkan pahlawan, dll). Paling banyak artikel pada *blog* ataupun pada video adalah mengenai strategi serangan dan bukan manajemen perang klan.

Tujuan dari penelitian ini adalah melakukan eksplorasi penggunaan *COC API* untuk memecahkan dua permasalahan utama dalam permainan *COC*, yaitu: kurangnya indikator keaktifan pemain, dan pembuatan strategi perang secara otomatis.

## METODE

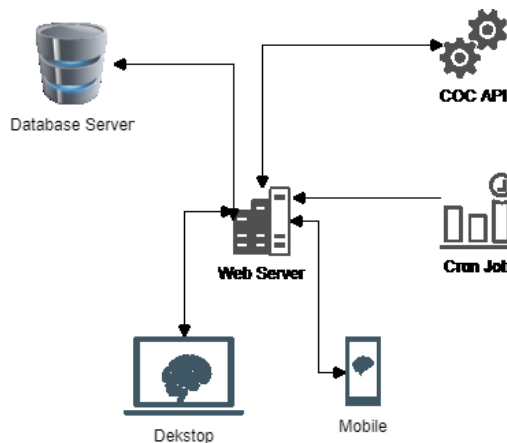
Metode pengembangan yang dilakukan adalah menggunakan metode perancangan dan pengembangan, yaitu: rancangan struktur aplikasi, rancangan algoritma, dan rancangan antarmuka dibuatkan suatu program berbasis *web*. Adopsi terhadap model pengembangan yang lebih lengkap (misalnya: *Waterfall*, *prototyping*, dan *xtreme programming*) tidak dilakukan mengingat sifat dari pengembangan yang dilakukan hanya oleh satu orang pengembang yang sekaligus merupakan pemilik dan pengguna dari perangkat lunak, sehingga proses pendefinisian

dan analisis kebutuhan, perencanaan sampai implementasi tidak perlu menggunakan proses yang lebih lengkap.

### Arsitektur Sistem

Adapun rancangan struktur sistem secara umum adalah seperti pada Gambar 1. Aplikasi berjalan dengan memanfaatkan *COC API*, yaitu dengan mengirimkan *request* data klan, data pemain, data war biasa, dan war *CWL (Clan War League)* kepada *server COC*. Data yang diterima kemudian diproses oleh aplikasi untuk menjalankan fungsionalitas yang telah ditentukan. Untuk data yang jarang mengalami perubahan akan disimpan ke dalam *database server*.

Untuk fungsionalitas monitoring terhadap keaktifan pemain dan juga fungsionalitas terhadap pembuatan strategi perang membutuhkan monitoring terhadap perubahan data, aplikasi perlu diintegrasikan dengan layanan penjadwalan. Layanan tersebut akan memicu aksi pembaharuan terhadap data, misalnya data indikasi keaktifan pemain dan juga terhadap perubahan data tingkat kekuatan serangan (pasukan/mantra/pahlawan) yang dipergunakan untuk membuat strategi perang. Aplikasi juga mendukung pengaksesan menggunakan perangkat *desktop* dan juga perangkat *mobile*.



Gambar 1. Arsitektur Sistem Pengelolaan Clan

### Rancangan Strategi Perang

Penentuan target serang bagi setiap pemain (strategi perang) adalah dengan menghitung kekuatan serangan pemain. Kekuatan serangan pemain dihitung berdasarkan tingkat (*level*) pasukan (*troop*) pemain, tingkat mantra (*spell*) pemain, dan tingkat pahlawan (*hero*) pemain. Perhitungan tingkat kekuatan adalah menggunakan rumus (1). Rerata level Pasukan dan mantra akan menghasilkan nilai antara 0 sampai maksimal-tingkat-pasukan (bernilai sekitar 5-7), tingkat Pahlawan akan berada di antara nilai 0-65, dan tingkat *Town-Hall* akan berada di antara nilai 1-12, sedangkan tingkat *Balancer* akan berada di antara nilai -12 sampai +12. Karena skala nilai yang berbeda-beda ini maka pada rumus ditambahkan bobot, yaitu untuk mengatur pengaruh dari masing-masing nilai, kecuali untuk tingkat pasukan yang akan dipergunakan sebagai sebagai nilai patokan ukuran.

Rancangan persamaan yang bersifat linear dipilih dikarenakan asumsi bahwa nilai aspek-aspek tersebut (rTPM, rTM, dll) memiliki pengaruh yang cukup merata terhadap kemampuan pemain dalam melakukan serangan dan juga semua aspek tersebut memberikan kontribusi positif (menambahkan) pada kekuatan serangan. Perubahan (peningkatan) terhadap nilai aspek-aspek tidak ada yang memiliki pengaruh yang terlalu signifikan terhadap peningkatan kemampuan serangan, oleh karenanya tidak digunakan faktor peningkatan yang lebih tinggi (eksponensial) terhadap perubahan aspek tersebut.

$$totalKekuatan = rTP + rTM + (rTPah \times bPah) + (B \times bB) + (TTH \times bTH) \quad (1)$$

rTP	: rerata Tingkat Pasukan
tTM	: rerata Tingkat Mantra
rTPah	: rerata Tingkat Pahlawan
bPah	: bobot Tingkat Pahlawan
B	: nilai <i>Balancer</i>
bB	: bobot nilai <i>Balancer</i>
TTH	: Tingkat <i>TownHall</i>
bTH	: bobot Tingkat <i>TownHall</i>

Nilai *Balancer* adalah nilai khusus untuk memenuhi kebutuhan pengaturan terhadap tingkat fokus, keterampilan, ataupun keadaan pahlawan (sedang ditingkatkan). Karena dapat saja pemain yang memiliki tingkat pasukan/mantra/pahlawan/*TH* yang tinggi tetapi memiliki kemampuan bermain yang rendah (karena akun yang dibeli), atau sedang sibuk kerja (tidak sempat menyusun taktik serangan), atau sedang meningkatkan pahlawannya, sehingga dibutuhkan nilai untuk mengatur agar pemain tersebut dapat diatur agar menyerang target yang lebih rendah sesuai dengan berkurangnya keefektifannya dalam menyerang. Hal yang sama juga dapat terjadi yaitu pemain yang memiliki kekuatan pasukan/mantra/pahlawan/*TH* yang lebih rendah, tetapi sedang fokus bermain atau memang dimainkan oleh pemain berpengalaman, sehingga keefektifan serangannya tinggi, pemain ini dapat diatur melalui nilai balancernya agar menyerang target yang lebih atas.

Pada setiap perang selesai nilai *balancer* ini akan diatur oleh pengelola klan, sesuai dengan hasil serangannya. Karena dinamika perang di klan, dapat membuat pemain terpaksa harus menyerang kebawah dari target ataupun menyerang ke atas dari target karena berbagai hal (pemain lain tidak menyerang, pemain lari dari perang, dll). Sehingga penilaian dilakukan dengan melihat beberapa hal, untuk pemain yang mengikuti target yang telah ditentukan, dan mengalami kegagalan (bintang 0-1 atau persentasi kerusakan rendah) maka nilai *balancer*-nya akan dikurangi, dan sebaliknya untuk pemain yang mengikuti target dan menghasilkan nilai maksimal (bintang 3, dan persentasi kerusakan 100%) maka nilai balancernya akan ditambah. Untuk pemain yang menyerang dibawah dari targetnya, apabila hasilnya maksimal maka tidak akan dilakukan perubahan pada nilai *balancer*-nya, karena memang sudah seharusnya mendapatkan hasil demikian, tetapi apabila hasilnya tidak maksimal maka nilai *balancer*-nya akan dikurangi. Sebaliknya untuk pemain yang menyerang diatas dari targetnya, apabila hasilnya tidak maksimal, maka nilai *balancer*-nya tidak akan diubah, karena memang *base* yang lebih atas dari target berarti dianggap lebih susah dari target, tetapi apabila hasilnya maksimal maka nilai *balancer*-nya akan ditambah.

Perubahan pada nilai *balancer* akan mempengaruhi target perang selanjutnya. Nilai *balancer* yang ditambah akan membuat pemain akan mendapatkan target yang lebih sulit dan sebaliknya nilai *balancer* yang dikurangi membuat pemain akan mendapatkan target yang lebih mudah.

### Rancangan Algoritma Strategi Perang

Algoritma untuk pengaturan strategi perang dapat dilihat pada baris-baris di Gambar 2. Algoritma menerima masukan berupa *tag* klan yang akan dikelola dan bobot-bobot faktor kekuatan serangan. Menggunakan *COC API* dilakukan permintaan terhadap data perang klan, data ini mencakup status perang saat ini (*preparation, inWar, warEnded*) (baris 1), daftar pemain yang ikut war, dan informasi hasil serangan pertama dan kedua (apabila telah dilakukan). Setelah itu dilakukan pengecekan pengaturan akses data klan, yang dapat diatur dari aplikasi permainan *COC* oleh pemimpin atau wakil pemimpin suatu klan. Program akan menampilkan informasi kesalahan dan tidak menampilkan strategi perang, apabila pengaturan akses data klan tertutup (baris 2-5).

Apabila status perang adalah persiapan (*preparation*), maka diperiksa apakah perang-perang sebelumnya sudah dievaluasi hal ini untuk memastikan bahwa nilai *balancer* sudah diatur dengan performansi pemain pada perang sebelumnya (baris 7-14). Apabila statusnya sudah mulai perang (“*inWar*”) maka dilakukan penilaian tingkat kekuatan serangan terhadap semua pemain yang ikut pada perang berdasarkan rumus (1) (baris 15-26). Data urutan pemain kemudian diurutkan berdasarkan tingkat kekuatan serangan (baris 27), setelah itu pemain diberikan target serang berdasarkan urutan kekuatannya (baris 29-45).

```

Name: Algoritma strategi perang
Input: tagKlan, bPah, bB, bTH
Output: targetPerangTerurutBerdasarkanKekuatanSerangan
begin
01. dataPerangKlan ← getDataPerang(tagKlan)
02. if dataPerangKlan == null
03.   pesanError ← “data klan ini ditutup, hubungi leader/wakil untuk membuka pengaturan data”
04.   return
05. endif
06. status ← getStatusWar(dataPerangKlan)
07. if status == “preparation”
08.   daftarWarBelumDievaluasi
09.   ← cekEvaluasiWarSebelumnya(getWaktuStartPerang(dataPerangKlan), tagKlan)
10.   if daftarWarBelumDievaluasi != null
11.     tampilkan(daftarWarBelumDievaluasi)
12.     pesanError ← “Masih ada war belum dievaluasi”
13.     return
14.   endif
15. elseif status == “inWar” or status == “warEnded”
16.   urutanKekuatanPemain = [ ]
17.   for i ← 1 to getJumlahPemainWar(dataPerangKlan)
18.     tagPemain ← getTagPemain(dataPerangKlan, i)
19.     dataPemain ← getDataPemain(tagPemain)
20.     strength ← getRerataLevelTroopSpell(dataPemain)
21.     hero ← getRerataLevelHero(dataPemain)
22.     levelTH ← getLevelTH(dataPemain)
23.     balancer ← getBalancer(dataPemain)
24.     totalKekuatan ← levelTH*bTH + strength + balancer*bB + hero*bPah
25.     tambah(urutanKekuatanPemain, tagPemain, totalKekuatan)
26.   endfor
27.   urutkanBerdasarkanTotalKekuatan(urutanKekuatanPemain)
28.   dataPerangMusuh ← getDataPerang(getTagKlanMusuh(dataPerangKlan))
29.   daftarTargetPerang = [ ]
30.   for i ← 1 to getJumlahPemainWar(dataPerangKlan)
31.     tagPemain ← getPemain(urutanKekuatanPemain, i)
32.     targetSerang ← tentukanTargetWar(i)
33.     seranganPertama ← getDataSeranganPertama(dataPerangKlan);
34.     seranganKedua ← getDataSeranganKedua(dataPerangKlan);
35.     if dataPerangMusuh == null
36.       tambah(daftarTarget, tagPemain, targetSerang, seranganPertama, seranganKedua)
37.     else
38.       tagBaseMusuhdiSerangPertama ← getTagMusuh(seranganPertama)
39.       tagBaseMusuhdiSerangKedua ← getTagMusuh(seranganKedua)
40.       nomorBaseMusuhdiSerangPertama ← getNomorUrutPemain(dataPerangMusuh)
41.       nomorBaseMusuhdiSerangKedua ← getNomorUrutPemain(dataPerangMusuh)
42.       tambah(daftarTarget, tagPemain, targetSerang, seranganPertama, seranganKedua,
43.         nomorBaseMusuhdiSerangPertama, nomorBaseMusuhdiSerangKedua)
44.     endif
45.   endfor
46.   return daftarTargetPerang
47. endif
end

```

Gambar 2. Algoritma strategi perang

Tergantung kebijakan pengaksesan data pada klan musuh (dibuka atau ditutup) (baris 28), maka dapat ditambahkan informasi mengenai nomor urut *base* yang telah diserang oleh pemain. Hal ini dilakukan dengan mengambil data perang klan musuh, dan nomor urut *base* musuh berdasarkan *tag* pemain musuh (baris 38-44). Informasi ini berguna untuk menilai pemain (mengikuti strategi atau tidak), dan juga untuk informasi penilaian perubahan nilai *balancer*-nya.

### Implementasi Perangkat Lunak

Perangkat lunak diimplementasikan dalam menggunakan bahasa pemrograman *PHP*, dengan memanfaatkan tambahan dukungan interaktifitas skrip dalam bahasa *JavaScript*. Aplikasi akan menggunakan layanan *google chart API* ([developers.google.com/chart](http://developers.google.com/chart)) yang berbasis *javaScript* yakni layanan pembuatan antarmuka visualisasi data. Sedangkan untuk dukungan interaktifitas aplikasi adalah menggunakan *JQuery* ([jquery.com](http://jquery.com)). Khusus untuk halaman utama untuk mendapatkan rancangan yang menarik dan dapat ditampilkan pada berbagai ukuran layar pengguna (komputer, *mobile phone*, dll) maka menggunakan *Bootstrap* ([getbootstrap.com](http://getbootstrap.com)) *CSS framework*. Untuk *server web* adalah menggunakan aplikasi *web Apache*, dengan didukung oleh aplikasi *database server MySQL*.

### Evaluasi

Evaluasi dari usulan aplikasi adalah dinilai menggunakan dua cara, yaitu observasi terhadap kemudahan pengelolaan klan, dan kedua menggunakan indikator data dari *COC API* dan data historis pada sistem. Indikator data yang digunakan yaitu tingkat keaktifan dari total keseluruhan klan dan data hasil perang (menang/kalah). Untuk tingkat keaktifan diukur dari hari pertama kali aplikasi monitoring keaktifan dijalankan hal ini dikarenakan tidak terdapat indikator (ataupun data) yang dapat digunakan sebagai pembanding tingkat keaktifan sebelum aplikasi dipergunakan. Pengukuran hasil perang dilakukan dengan membandingkan rasio kemenangan sejak tanggal aplikasi digunakan, dengan rasio kemenangan sebelum aplikasi digunakan.

### HASIL DAN DISKUSI

Hasil implementasi dari rancangan menunjukkan bahwa rancangan berhasil diimplementasikan untuk memenuhi kebutuhan dalam pengelolaan klan. Kebutuhan yang pertama yang berhasil diimplementasikan adalah kebutuhan penampilan informasi umum klan (Gambar 3), penampilan informasi yang ditampilkan adalah daftar nama pemain, pangkat, jumlah donasi, dan informasi lain. Layanan ini juga mempergunakan informasi berupa gambar tingkat liga pemain sehingga mempermudah pengunjung untuk memahami melalui gambar representasi liga (mis. *Legend*, *Titan*, dll).

Untuk monitoring keaktifan pemain berhasil diimplementasi yaitu menjadi fungsi untuk menghasilkan beberapa fungsi monitoring. Fungsi rekap-tidak-*online* untuk melihat pemain-pemain yang tidak *on* (warna merah) dan atau pemain yang hanya muncul satu kali pada suatu hari (warna biru) (Gambar 4). Fungsi ini digunakan untuk menjalankan kebijakan klan dalam menjamin keaktifan semua pemain, misalnya kebijakan pemain yang tiga hari berturut-turut harus dikeluarkan, atau pemain yang dalam seminggu tidak *online* sebanyak 5 hari (tidak berturut-turut) harus dikeluarkan.

Fungsi lainnya untuk monitoring secara umum adalah peringkat keaktifan pemain dalam klan (Gambar 5). Pengurutan peringkat ini sangat berguna untuk memotivasi pemain-pemain aktif sehingga akan semakin meningkatkan keaktifannya dan juga untuk memonitor pemain-pemain yang tidak aktif (peringkat terbawah dalam keaktifan). Selain fungsi motivasi, fungsi ini untuk



mencari pemain-pemain yang tidak terdeteksi menggunakan rekap tidak-online, karena pemain tersebut *online* setiap hari, namun waktu *online*-nya tidak lama.

1(1)		146		Donasi: 4133	Terima: 2710	4968	AttWin:34
2(2)		185		Donasi: 982	Terima: 510	4851	AttWin:13
3(3)		229		Donasi: 1005	Terima: 248	4704	AttWin:2
4(4)		152		Donasi: 314	Terima: 337	4695	AttWin:16
5(5)		210		Donasi: 794	Terima: 194	4646	AttWin:16
6(6)		217		Donasi: 821	Terima: 571	4581	AttWin:8
7(7)		206		Donasi: 104	Terima: 204	4443	AttWin:0
8(8)		181		Donasi: 149	Terima: 288	4427	AttWin:2

Gambar 3. Informasi umum klan/pemain

Fungsi keaktifan selanjutnya adalah fungsi untuk monitoring waktu *online* pemain, fungsi ini adalah untuk memonitoring waktu-waktu *online* dari pemain yang ditunjukkan pada Gambar 6. Hal ini adalah tambahan informasi untuk melihat waktu-waktu *online* seorang pemain, untuk melihat waktu klan sedang ramai atau sedang sepi. Informasi ini dapat dipergunakan dalam menentukan waktu terbaik untuk memulai perang atau untuk memberikan pengumuman klan.

Fungsi keaktifan individu juga dikembangkan (Gambar 7) yaitu untuk melihat secara rinci tingkat keaktifan pemain diukur mulai dari saat pemain pertama kali masuk klan. Fungsi ini sangat berguna untuk melakukan penilaian keaktifan individu pemain, sehingga membantu dalam pengambilan keputusan, misalnya pemain yang sudah mulai kurang aktif, tetapi merupakan pemain lama sehingga dapat dipertahankan posisinya.

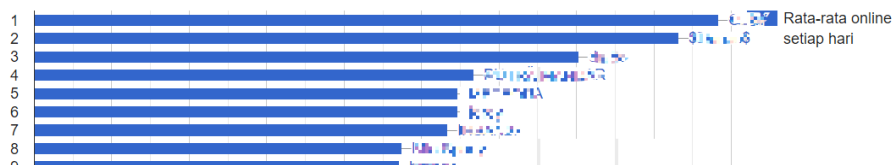
[HARI INI](#) | [03/10/2019](#) | [02/10/2019](#) | [01/10/2019](#) | [30/09/2019](#) | [29/09/2019](#) | [28/09/2019](#) |

**Rekap yang Tidak Online Minggu ini**

Keterangan : Tidak Pernah On, Hanya 1 s.d 10 menit

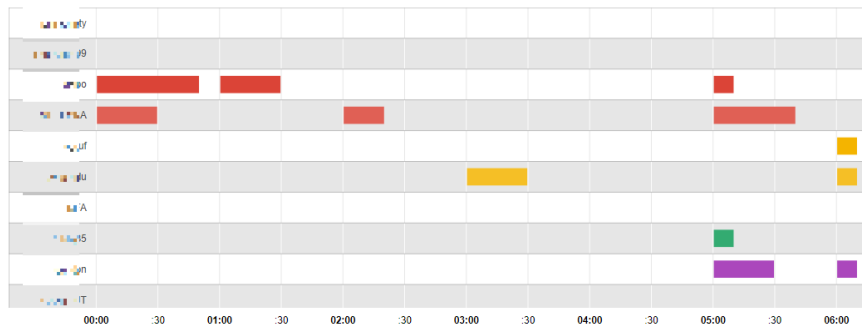
Nama	Tanggal
	03/10/2019, 02/10/2019,
	Sudah keluar klan
	03/10/2019, 02/10/2019, 29/09/2019,
	03/10/2019, 02/10/2019, 01/10/2019, 30/09/2019, 29/09/2019, 28/09/2019,
	03/10/2019, 02/10/2019, 29/09/2019, 28/09/2019,
	03/10/2019, 01/10/2019,
	Sudah keluar klan
	03/10/2019,
	03/10/2019,
	03/10/2019, 02/10/2019, 30/09/2019,
	03/10/2019, 01/10/2019,

Gambar 4. Monitoring pemain tidak *online*

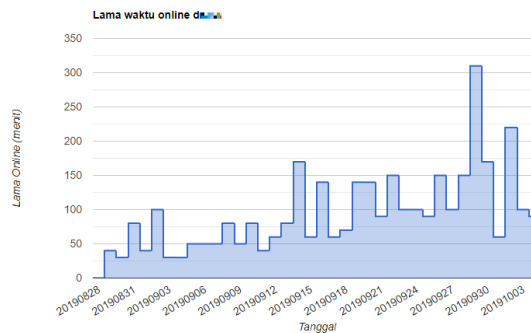


Gambar 5. Peringkat keaktifan pemain

Jam-jam pemain on, Hari Ini



Gambar 6. Visualisasi waktu-waktu *online* pemain



Gambar 7. Informasi keaktifan individu pemain

Untuk layanan pembuatan strategi diimplementasikan sehingga menghasilkan strategi perang, baik perang biasa maupun perang liga (Gambar 8). Pemain dapat menggunakan informasi ini untuk menentukan nomor urut *base* musuh yang harus diserangnya. Fungsi ini dilengkapi dengan informasi mengenai serangan yang telah dilakukan oleh pemain, yaitu informasi serangan pertamanya dan serangan keduanya. Pada informasi serangan, akan terlihat akan terlihat hasil serangan, berupa jumlah bintang (*star*) yang dihasilkan, tingkat kerusakan, dan nomor target musuh yang diserang (informasi ini tergantung pengaturan akses data pada klan musuh perang) (Gambar 8a).

Disediakan juga informasi kekuatan serangan peserta perang (Gambar 9), informasi ini berupa urutan kekuatan serangan pemain yang sedang ikut pada perang yang sedang berlangsung. Informasi kekuatan serangan ini bermanfaat agar pemain dapat menelusuri alasan penetapan target perang bagi dirinya. Informasi ini juga dapat dipergunakan oleh pemain untuk memperbandingkan kekuatan serangannya dengan pemain lain.

Strategi War

Akun	Target	Attack 1	Attack 2
	1-3	no.1, star:1, kerusakan:72%	no.3, star:2, kerusakan:80%
	1-3	no.2, star:1, kerusakan:51%	no.9, star:2, kerusakan:57%
	4-6	no.7, star:2, kerusakan:53%	ki., star:2, kerusakan:85%
	4-6	no.10, star:3, kerusakan:100%	
	7-8	no.4, star:2, kerusakan:54%	no.2, star:2, kerusakan:56%
	7-8	no.8, star:2, kerusakan:73%	

(a)

Strategi War Day-1

Akun	Target
	1
	2
	3
	4
	5
	6
	7
	8
	9

(b)

Gambar 8. Informasi target perang biasa dan perang liga

Kekuatan Attack Peserta War

Akun	TH	Strength	Heroes	Balancer	Total Kekuatan	Level	BK	AQ	GW	Giant	Balloon	Wizard	Healer	Dragon	PEKKA	Hc
	12	6.32	56.67	4	24.65	----	65	65	40	9	8	9	5	7	8	9
	12	6.00	45.33	0	21.07	----	49	65	22	9	8	9	5	7	8	9
	12	6.05	54.00	-12	19.85	----	57	65	40	9	8	9	5	7	8	9
	12	5.77	49.33	-9	19.39	----	56	60	32	9	8	8	5	6	8	7
	12	5.73	33.00	4	19.33	----	32	47	20	7	8	9	5	7	6	5
	12	5.86	43.33	-5	19.28	----	43	65	22	9	8	9	5	6	8	6
	12	5.59	48.33	-8	19.26	----	40	65	40	9	8	9	5	7	8	8
	12	6.00	53.00	-14	19.10	----	54	65	40	9	8	9	5	7	8	9
	12	5.05	35.33	3	18.86	----	20	57	29	8	8	8	5	6	7	6
	12	3.91	22.00	16	18.31	----	24	30	12	8	8	6	5	2	3	3
	12	6.05	48.67	-14	18.28	----	43	65	38	9	8	9	5	5	8	9
	12	4.86	31.67	4	18.20	----	20	56	19	8	8	8	5	5	7	6

Gambar 9. Informasi kekuatan serangan peserta perang

Dalam mengembangkan fungsi-fungsi ini terdapat beberapa kendala yang terjadi yaitu format data yang disediakan oleh *COC API* adalah lebih berfokus pada dukungan terhadap informasi permainan, dan tidak untuk pembuatan sebuah layanan pengaturan strategi perang. Misalnya untuk membuat informasi perang liga (Gambar 8b) adalah tidak terdapat suatu indikator data daftar pemain yang masuk atau tidak masuk pada suatu hari perang (*war-day*), sehingga dalam sistem memerlukan penyimpanan data ini ke dalam basis data sistem (Gambar 10).

Misalnya untuk respon terhadap permintaan data suatu perang liga (*CWL*) dapat dilihat bahwa tidak terdapat perbedaan mengenai posisi klan yang meminta data, apakah sebagai klan *home* atau klan *opponent*. Hal ini membuat program harus melakukan pengecekan secara iteratif untuk mencari informasi khusus klan ini, untuk membuat fungsi pengaturan strategi perang klan.

Hal-hal ini membuat informasi strategi untuk perang liga (Gambar 8b) menjadi tidak dapat diisi dengan informasi seperti tingkat kerusakan ataupun urutan markas musuh yang diserang. Hal ini mempersulit pengelola klan untuk melakukan pengawasan terhadap pelaksanaan strategi oleh peserta perang.

Kesulitan-kesulitan ini kemungkinan karena perancangan layanan *COC API* dibuat berdasarkan informasi yang disediakan pada aplikasi *COC*, dan masih belum memikirkan mengenai kemungkinan pemanfaatannya untuk pengelolaan klan. Hal ini terlihat dari konsistensi antara struktur informasi data respon dengan penampilan informasi pada aplikasi *COC*. Sehingga saran lebih lanjut adalah dalam pengembangan layanan *API* suatu permainan sebaiknya juga memperhatikan penggunaannya bagi berbagai kepentingan pemain pada semua peran yang mungkin, misalnya pemain sebagai individu, pemain sebagai anggota/pengelola suatu kelompok (klan), dll.

```

{
  "state": "inWar",
  "season": "2019-10",
  "clans": [
    {
      "tag": "#P99JRC89",
      "name": "putra pangkep@",
      ...
      "members": [
        {
          "tag": "#2Q2P0C29V",
          "name": "amubasa",
          "townHallLevel": 11
        },
        ...
      ]
    },
    ...
  ]
}

...
{
  "tag": "#LVQYPGGQ",
  "name": "娜美贼团",
  ...
  "members": [
    {
      "tag": "#L2PP0Y80Y",
      "name": "四大才子香奈儿",
      "townHallLevel": 10
    },
    ...
  ]
}

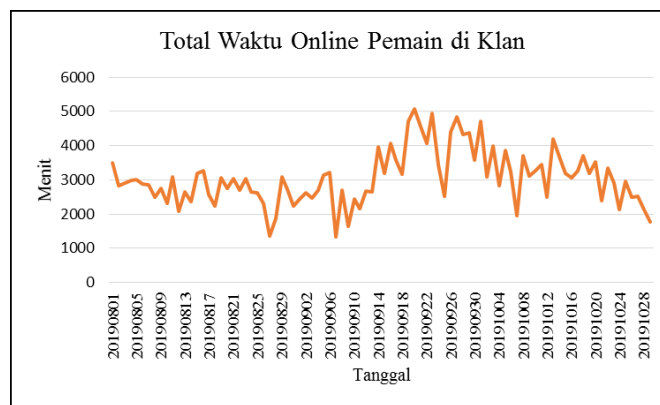
...
"rounds": [
  {
    "warTags": [
      "#2YG2GQPP0",
      "#2YG2GG80G",
      "#2YG2GQ8VJ",
      "#2YG2GQ2RL"
    ]
  },
  {
    "warTags": [ ... ], ...
  }
]
}

```

Gambar 10. Contoh respon data CWL

Evaluasi terhadap nilai kemanfaatan aplikasi dilaksanakan menggunakan observasi terhadap manajemen klan. Dirasakan respon positif dari anggota klan dan juga dari pengelola klan (*coleader* dan *senior*), karena memudahkan dalam memonitor keaktifan pemain dan juga dalam pengaturan klan. Sehingga pengambilan keputusan dalam mengeluarkan pemain menjadi lebih mudah.

Evaluasi terhadap keaktifan pemain dilakukan dengan menggunakan data historis aktifitas pemain. Data historis ini dikumpulkan dari aktifitas yang terdeteksi oleh sistem yaitu dari aktifitas melakukan donasi, aktifitas menerima donasi, melakukan serangan, dan indikator lain. Tidak dapat dilakukan pembandingan terhadap keadaan sebelum menggunakan aplikasi, sehingga yang dilakukan hanya melihat data tersebut, dengan asumsi bahwa seharusnya data tingkat waktu *online* akan meningkat. Hal ini karena semakin dikurangnya (dikeluarkan) pemain yang kurang aktif, dan digantikan dengan pemain baru yang lebih aktif.



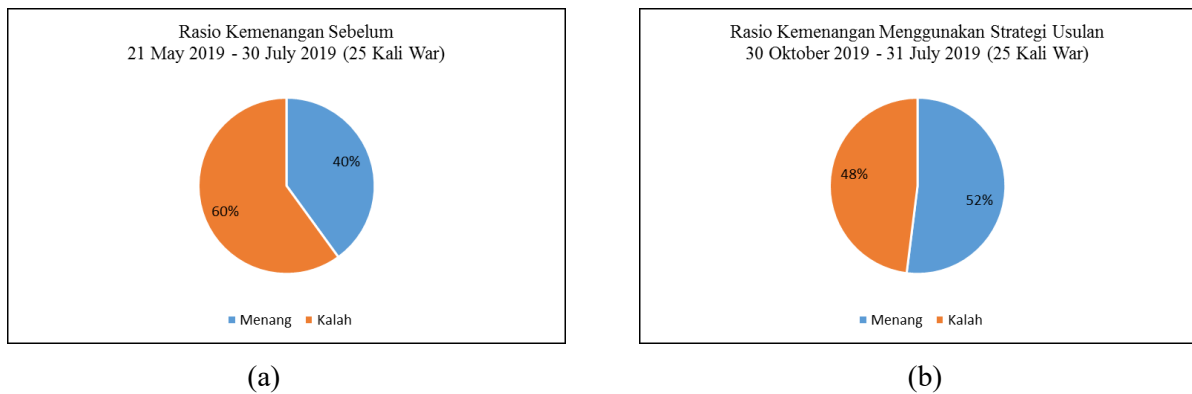
Gambar 11. Total waktu *online* pemain

Hasil yang didapatkan menunjukkan bahwa total tingkat keaktifan pemain berhasil ditingkatkan (Gambar 11). Fluktuasi yang terjadi terhadap total keaktifan pemain adalah aktifitas pengeluaran pemain-pemain yang dianggap kurang aktif dan masuknya pemain baru yang diharapkan lebih aktif. Terdapat sedikit tren penurunan keaktifan pada menjelang akhir Oktober 2019, hal ini karena perubahan mekanisme rekrutmen pemain yang dilakukan oleh *SuperCell* pada *update* Oktober 2019 (digantinya *global chat* dengan fasilitas rekrutmen baru), sehingga membuat klan menjadi kesulitan dalam melakukan rotasi pemain.

Evaluasi terhadap keefektifan perang dilakukan dengan mengambil data hasil perang yang disediakan melalui *COC API*. Rasio kemenangan semenjak pertama kali perang digunakan (31 Juli 2019) sampai saat perang terakhir (24 Oktober 2019) yaitu sebanyak 25 kali perang (Gambar 12b), rasio dibandingkan dengan rasio kemenangan sebanyak 25 kali kemenangan dihitung mundur dari perang terakhir saat tidak menggunakan aplikasi (Gambar 12a). Dari hasil pengukuran didapatkan bahwa setelah menggunakan strategi, rasio kemenangan berhasil meningkat sebanyak 12%, yaitu dari 40% sebelum menerapkan strategi, menjadi 52% (Gambar 12).

Kekurangan dari usulan adalah dari model penilaian kekuatan serangan yang memerlukan diaturnya nilai *balancer* sesuai performansi setiap pemain setiap selesai suatu perang. Hal ini membuat pengelola klan harus menilai satu-per-satu performansi pemain. Penentuan nilai secara otomatis belum diimplementasikan karena masih perlu ditentukan indikator yang terbaik untuk menilai performansi pemain dalam suatu perang, juga karena terdapat kemungkinan klan musuh perang untuk menutup data perangnya sehingga sistem tidak dapat mengambil data detail posisi *base* musuh yang diserang oleh pemain, dan juga penilaian secara otomatis tidak

dapat mengantisipasi kemungkinan pemain yang tiba-tiba melakukan peningkatan terhadap pahlawannya saat strategi terlanjur selesai dibuat.



Gambar 12. Rasio kemenangan perang sebelum dan sesudah menggunakan strategi usulan

## KESIMPULAN

Kesimpulan yang didapatkan adalah penggunaan layanan *COC API* sangatlah membantu dalam melakukan pengelolaan klan di *COC*. Permasalahan-permasalahan utama pada permainan *COC* seperti informasi keaktifan pemain, dan juga dalam pengaturan strategi perang dapat dipecahkan dengan mengembangkan perangkat lunak yang menggunakan layanan *COC API*. Penggunaan fungsi monitoring keaktifan berhasil membuat total waktu *online* pemain menjadi meningkat. Keefektifan metode pengaturan strategi perang juga telah terbukti meningkatkan rasio kemenangan pemain sebesar 12% (dari 40% menjadi 52%).

Perancangan layanan *COC API* masih belum memperhatikan kemungkinannya untuk membantu pengelolaan pemain. Hal ini kemungkinan disebabkan bahwa layanan *API* yang diberikan adalah sebenarnya suatu layanan yang dipakai juga oleh aplikasi *COC*, atau juga memang merupakan rencana pengembang *COC (SuperCell)* untuk mengintegrasikan pengelolaan klan langsung pada aplikasi *COC*.

Pengembangan selanjutnya dengan memperbaiki kekurangan dalam metode ataupun implementasi pengelolaan perang, yaitu dalam penggunaan nilai *balancer* yang harus selalu dinilai oleh pengelola klan, agar dapat diotomatisasi. Pengembangan dapat pula dilakukan dengan mengarahkan fungsi monitoring keaktifan pemain untuk memonitor klan-klan lain untuk mendeteksi pemain-pemain aktif pada klan-klan tersebut, untuk selanjutnya dapat direkrut ke dalam klan sehingga dapat membentuk klan yang semua anggotanya adalah aktif (50 pemain aktif). Eksplorasi juga dapat dilanjutkan dengan eksplorasi formulasi yang berbeda dari strategi perang, dan juga dari pembobotan dari faktor-faktor kekuatan serangan. Dapat juga dilakukan pengembangan untuk membuat aplikasi menjadi berbentuk *framework* aplikasi yang dapat langsung diterapkan untuk pengelolaan klan-klan lain, hal ini untuk membuat kemudahan dalam pengelolaan klannya atau bahkan pengembangan selanjutnya dari aplikasi.

## REFERENSI

- Alanur, S. N., Jamaludin, & Imran. (2016). Permainan tradisional nogarata to kaili sebagai media pendidikan Pkn dalam membangun nilai karakter kewarganegaraan siswa SMP Negeri 13 Palu. *EDU CIVIC*, 4(1), 1–10.
- Aruan, R. S., & Mulyani, H. (2018). Penerapan intelegensia buatan dengan algoritma Negamax pada aplikasi permainan deret lima. *Jurnal Bangkit Indonesia*, 7(1), 1–13.
- Damayanti, A. M. (2015). *Pengaruh game online Clash-of-Clans terhadap perubahan perilaku*

*komunitas COC arek bungkul*. Tesis. Sekolah Tinggi Ilmu Komunikasi Surabaya.

- Dinata, O. (2017). Hubungan kecanduan game online Clash-of-Clans terhadap perilaku sosial (studi kasus game online Clash-of-Clans pada mahasiswa sosiologi Fakultas Ilmu Sosial dan Politik Universitas Riau). *JOM Fisip*, 4(2), 1–15.
- Djawa, Y. A. (2018). Analysis of the jargons used by players of the Clash-of-Clans game , an online game. *Academic Journal of Education Sciences*, 1(1), 28–39. <https://doi.org/10.22341/ajes.000101p028>
- Fahmi, N. E., Syarief, A., & Grahita, B. (2018). Identifikasi pengalaman bermain game mobile (studi kasus game Clash-of-Clans). *Jurnal Sositologi*, 17(2), 246–260. <https://doi.org/10.5614/sostek.itbj.2018.17.2.7>
- Makkarumpa, A. R. (2016). *Pola interaksi antar anggota dalam game online Clash-of-Clans (studi kasus 4 clan Clash-of-Clans) (Thesis)*. Universitas Hasanuddin.
- Malik, R., & Hidayah, N. (2017). Solidaritas virtual antar member game Clash-of-Clans di clan soloken5. *Jurnal Pendidikan Sosiologi*, 1–20.
- Meliani, M. (2017). *Analisis jual beli akun game online Clash-of-Clans dalam perspektif hukum Islam dan hukum positif*. Tesis. Universitas Islam Negeri Raden Intan Lampung.
- Mortensen, S. T., Nygaard, K. K., & Madsen, O. (2019). Outline of an Industry 4.0 Awareness Game. *Procedia Manufacturing*, 31, 309–315. <https://doi.org/10.1016/j.promfg.2019.03.049>
- Nicholas, D., Constantinides, P., & Nandhakumar, J. (2015). How to motivate & engage generation “Clash-of-Clans” at work? Emergent Properties of business gamification elements in the digital economy. In *Proceedings of the International Gamification for Business Conference (IGBC15)*, 13, 1576–1580.
- Preist, C., & Jones, R. (2015). The use of games as extrinsic motivation in education. *Conference on Human Factors in Computing Systems - Proceedings*, 3735–3738. <https://doi.org/10.1145/2702123.2702282>
- Rajanen, M., & Nissinen, J. (2015). A survey of game usability practices in Northern European game companies. *IRIS: Selected Papers of the Information Systems Research Seminar in Scandinavia*, (6).
- Reis, S., Reis, L. P., & Lau, N. (2019). Player engagement enhancement with video games. In *Advances in Intelligent Systems and Computing*, 931, 263–272. [https://doi.org/10.1007/978-3-030-16184-2\\_26](https://doi.org/10.1007/978-3-030-16184-2_26)
- Saputra, Rahmadi. (2016). Hubungan kecanduan game online Clash-of-Clans terhadap prestasi akademik mahasiswa ilmu komunikasi Fakultas Ilmu Sosial dan Politik Universitas Riau. *Ilmu Administrasi Bisnis FISIP Universitas Riau*, 3(2), 1–9.
- Saputra, Rino. (2016). Efek game online terhadap kreatifitas berpikir anak di Kelurahan Kadia Kecamatan Kadia Kota Kendari. *Journal Ilmu Komunikasi UHO*, 1(3), 1–16.
- Syarif, A. S., Aprianti, R. E., & M, U. H. (2011). Pengaruh turn-based strategy game terhadap kemampuan manajemen organisasi pada mahasiswa Fakultas Psikologi Universitas Negeri Makassar. *Jurnal Nalar Pendidikan*, 5(1), 80–85.
- Veronika, U., Arianti, A., Nurnaningsih, & Astuti, P. (2018). Learning English vocabulary through online games: case study of students in 4th grade of state elementary school (SDN) Jombor 01, Sukoharjo, Central Java, Indonesia. *International Journal of Multicultural and Multireligious Understanding*, 5(4), 470. <https://doi.org/10.18415/ijmmu.v5i4.433>
- Vervoort, J. M. (2019). New frontiers in futures games: leveraging game sector developments. *Futures*, 105, 174–186. <https://doi.org/10.1016/j.futures.2018.10.005>
- Yuliasutik, I. (2016). *Pendapat MUI Kota Malang terhadap jual beli account Clash-of-Clans (COC)*. Central Library of Maulana Malik Ibrahim Islamic University of Malang. <https://doi.org/10.3929/ethz-b-000238666>