

Perencanaan Perjalanan Wisata Multi Kota dan Negara dengan Algoritma Cuttlefish

Hendrawan Armanto, *Teknik Informatika Sekolah Tinggi Teknik Surabaya,*
Reynold Kevin, *Teknik Informatika Sekolah Tinggi Teknik Surabaya,*
dan C. Pickerling, *Teknik Informatika Sekolah Tinggi Teknik Surabaya*

Abstrak — Liburan merupakan saat yang paling tepat untuk melakukan perjalanan wisata bersama keluarga. Sebelum melakukan liburan, tentu saja setiap orang akan menyiapkan rencana perjalanan yang paling efektif dan efisien walaupun merupakan hal yang sulit. Meskipun sudah terdapat berbagai paket tur yang ditawarkan bagi para wisatawan, seringkali yang ditawarkan tidak sesuai dengan keinginan dari masing-masing orang karena faktor selera yang berbeda-beda. Oleh karena itu, dibuatlah sebuah penelitian yang digunakan untuk melakukan perencanaan perjalanan wisata multi kota dan negara berbasis aplikasi mobile. Untuk menyelesaikan, penelitian ini akan menggunakan Algoritma Cuttlefish dalam pembuatan jadwal wisata secara otomatis berdasarkan data input yang telah diisikan oleh pengguna sebelumnya. Pada penelitian ini, pengguna dapat melakukan pembuatan perencanaan perjalanan wisata baik secara manual ataupun otomatis. Selain fitur untuk pembuatan perencanaan perjalanan wisata, juga terdapat fitur untuk melakukan pencarian tiket penerbangan dan juga akomodasi tempat tinggal dengan bantuan TravelPayouts API. Pada pembuatan jadwal wisata pun juga sudah dilengkapi dengan pencarian tiket penerbangan dan akomodasi tempat tinggal sesuai dengan input yang diberikan oleh pengguna sebelumnya. Pengguna juga dapat melihat jadwal wisata yang telah terbentuk dalam bentuk Table View, Map View dan PDF. Dengan dibuatnya penelitian ini diharapkan para pengguna dapat membuat sebuah jadwal wisata berdasarkan selera masing-masing. Jadwal wisata yang dibentuk secara otomatis dengan menggunakan Algoritma Cuttlefish tersebut juga dibuat sedemikian rupa hingga tercipta sebuah jadwal wisata yang efisien. Selain itu, pada tahap akhir dari tahap uji coba disebarkan kuesioner kepada responden. Kuesioner bertujuan untuk mengetahui komentar pengguna mengenai hasil akhir dari penelitian yang telah dibuat demi pengembangan untuk kedepannya. Berdasarkan hasil kuesioner, sebanyak 88% responden menyatakan jadwal wisata yang terbentuk sudah cukup baik.

Kata Kunci — Evolutionary Algorithm, Cuttlefish Algorithm, Perencanaan Perjalanan Wisata.

I. PENDAHULUAN

Seiring dengan berjalannya waktu, saat ini terdapat banyak tempat menarik dan dapat dijadikan sebagai objek wisata. Mulai dari tempat wisata alami seperti pegunungan dan pantai sampai tempat wisata buatan seperti

taman bermain hingga tempat-tempat bersejarah yang ada pada daerah tertentu. Hal ini tentu saja akan mendorong para wisatawan untuk ingin mengunjungi berbagai tempat wisata. Selain itu, dengan adanya banyak tempat wisata yang ada, para wisatawan juga semakin memiliki banyak opsi untuk melakukan perjalanan wisata dan tentu saja akan semakin bingung menentukan sebuah rute perjalanan wisata.

Solusi dari masalah inipun sudah sejak lama dipecahkan dengan berbagai macam pilihan tur yang ditawarkan oleh pihak agensi travel sehingga para wisatawan cukup memilih jenis tur yang ada tanpa perlu repot-repot untuk membuat jadwal wisata. Namun, banyak dari pilihan tur yang telah ditawarkan tidak sesuai dengan keinginan masing-masing orang karena faktor selera tiap orang yang berbeda dan juga budget yang dimiliki. Hal ini tentu saja menimbulkan masalah baru. Untuk mempermudah para wisata dalam mengatur perjalanan wisata khususnya pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan, dibentuklah sebuah aplikasi mobile yang digunakan untuk perencanaan perjalanan wisata secara otomatis sesuai dengan keinginan dari pengguna. Selain itu pengguna juga dapat memesan tiket penerbangan dan juga melakukan pencarian terhadap akomodasi tempat tinggal yang diinginkan. Tentunya pembuatan jadwal wisata secara otomatis yang terdapat pada aplikasi tersebut dibentuk dengan bantuan Evolutionary Algorithm [1], yaitu menggunakan Algoritma Cuttlefish [2][3]. Karena sistem ini berbasis pada sebuah aplikasi mobile, sehingga akan dibentuk dengan menggunakan bantuan Vue.js [4], Framework7 [5] yang seluruhnya merupakan bagian dari paket-paket yang terdapat pada Node.js [6].

II. ALGORITMA CUTTLEFISH

Algoritma Cuttlefish merupakan sebuah algoritma yang terinspirasi dari cara binatang sotong untuk bertahan hidup. Algoritma ini akan mensimulasikan mekanisme perubahan warna pada tubuh dari binatang sotong [7]. Algoritma ini ditemukan pada tahun 2013 oleh ilmuwan bernama Adel Sabry Eesa, Adnan Mohsin Abdulazeez Brifciani dan Zeynep Orman [2][3].

Sotong merupakan sebuah binatang laut yang memiliki kemampuan untuk merubah warna kulitnya menjadi sesuai dengan apa yang ada di lingkungan sekitarnya. Sotong akan melakukan mimikri yang merupakan sebuah proses evolusi yang terjadi pada spesies untuk menjadi sama dengan spesies lainnya dengan cara merefleksikan cahaya yang masuk kedalam kulit sotong lalu direfleksikan kembali [8].

Hendrawan Armanto, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Jawa Timur, Indonesia (e-mail: hendrawan@stts.edu)

Reynold Kevin, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Jawa Timur, Indonesia

C. Pickerling, Departemen Teknik Informatika, Sekolah Tinggi Teknik Surabaya, Surabaya, Jawa Timur, Indonesia (e-mail: pickerling@stts.edu).



Gambar. 1. Binatang Sotong

Sotong memiliki 3 jenis komponen kulit yang menumpuk satu sama lain dan nantinya ketiga komponen kulit ini akan saling bekerjasama agar dapat merubah warna kulit sotong menjadi jutaan warna dan motif yang seolah-olah sama persis dengan lingkungan sekitarnya [9]. Berikut adalah 3 jenis komponen kulit yang terdapat pada binatang sotong:

1) *Chromatophores*

Chromatophores merupakan sekumpulan otot dan kantung yang berisikan ribuan pigmen granul. Otot-otot yang terdapat pada chromatophores ini dapat berkontraksi dan relaksi dan menyebabkan kantung dapat membuka atau menutup. Pada saat otot berelaksasi, kantung akan mengecil untuk menyembunyikan pigmen-pigmen yang ada.

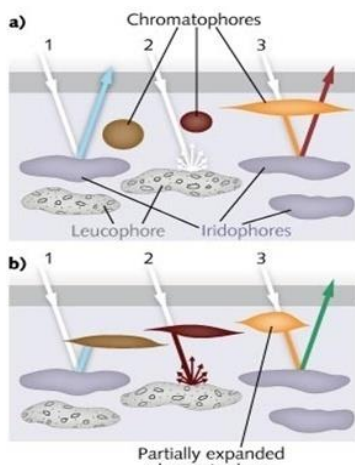
2) *Iridophores*

Iridophores pada sotong akan bekerja untuk menghasilkan warna metalik dan dapat bekerja apabila terkena cahaya. Selain itu, bagian ini juga dapat digunakan binatang sotong untuk membuat bagian kulitnya transparan sehingga seluruh organ dari binatang sotong dapat terlihat.

3) *Leucophores*

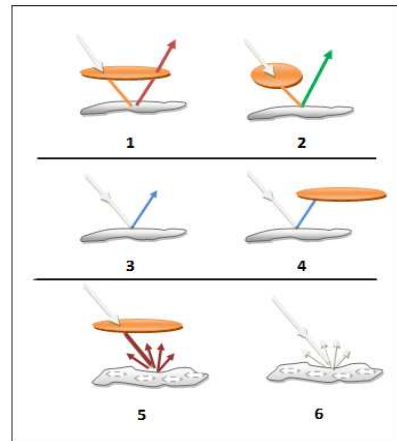
Bagian ini bertanggung jawab untuk menghasilkan sebuah motif bintang-bintang yang terdapat pada binatang sotong. Bagian ini juga merupakan sebuah kumpulan grup-grup sel yang nantinya bisa menyebar dan dapat merefleksikan cahaya yang masuk pada tubuh sotong.

Penampilan atau warna yang terdapat pada binatang sotong ini tergantung pada bagian komponen kulit mana yang terkena cahaya dan nantinya cahaya yang masuk dapat direfleksi baik dari kombinasi antara chromatophores, iridophores, dan leucophores. Proses mekanisme perubah warna pada sotong ini dapat disebut sebagai Global Optimum Solution.



Gambar. 2. Komponen Kulit Sotong

Tujuan dari algoritma ini adalah untuk mendapatkan bentuk dari binatang sotong untuk bersembunyi dengan menyerupai sesuai dengan lingkungannya. Berdasarkan mekanisme perubahan warna kulit yang terjadi pada sotong, Algoritma Cuttlefish akan dibagi menjadi 6 kasus yang mana akan dikumpulkan lagi menjadi 4 grup. Keenam kasus inilah yang nantinya akan mewaliki mekanisme perubahan warna kulit sotong dalam Algoritma Cuttlefish. Untuk setiap kasus, akan memiliki 2 proses utama yaitu reflection dan visibility. Reflection merupakan sebuah proses yang menunjukkan mekanisme untuk memantulkan cahaya yang mengenai pada komponen kulit binatang sotong. Visibility adalah suatu proses yang menunjukkan bahwa binatang sotong memiliki bentuk dan motif yang sama seperti daerah sekitarnya. Untuk setiap individu baru nantinya akan berasal dari hasil perhitungan antara reflection dan visibility.



Gambar. 3. 6 Kasus pada Algoritma Cuttlefish

Langkah pertama yang diharuskan dijalankan ada dengan cara melakukan inisialisasi sebuah populasi dengan setiap individu acak didalamnya serta penginputan nilai untuk masing-masing parameter r_1 , r_2 , v_1 dan v_2 . Kemudian dilanjutkan dengan melakukan proses perhitungan untuk mendapatkan sebuah solusi baru dimana populasi akan dibagi sesuai dengan 4 grup sesuai dengan kasus-kasus yang terdapat pada gambar 3. Untuk grup 1 dan 4 bertanggungjawab sebagai local search, sedangkan pada grup 2 dan 3 bertanggungjawab sebagai global search.

$$newp = reflection + visibility \quad (1)$$

Grup 1 (Kasus 1 dan 2) merupakan proses interaksi antara chromatophores dan iridophores. Mula-mula cahaya akan menuju bagian chromatophores dan akan diteruskan menuju komponen iridophores. Semua cahaya tersebut akan dipantulkan kembali. Meningat sel otot pada chromatophores dapat berkontraksi atau berelaksasi sehingga cahaya pantulan pun dapat kembali pada komponen chromatophores atau tidak. Berikut adalah persamaan yang digunakan untuk mendapatkan solusi dari grup 1:

$$reflection_j = R * G_1[i].Points[j] \quad (2)$$

$$visibility_j = V * (Best.Points[j] - G_1[i].Points[j]) \quad (3)$$

Nilai G_n merupakan grup yang sedang dikerjakan lalu $G_n[i].points[j]$ merupakan individu yang ke- i pada grup n

dan menunjuk pada poin ke j . Sedangkan untuk nilai R pada persamaan 3 merupakan sebuah persamaan yang didapatkan dari nilai acak antara parameter r_2 hingga r_1 atau sebaliknya. Hal ini juga berlaku demikian untuk nilai V yaitu merupakan sebuah persamaan yang didapatkan dari nilai acak antara parameter v_2 hingga v_1 atau sebaliknya. Nilai parameter dari r_1 , r_2 , v_1 dan v_2 ini nantinya didapatkan berdasarkan inputan dari pengguna. Untuk $Best.Points[j]$ merupakan sebuah nilai pada individu terbaik yang menunjuk pada poin ke j .

$$R = random() * (r_1 - r_2) + r_2 \quad (4)$$

$$V = random() * (v_1 - v_2) + v_2 \quad (5)$$

Grup 2 (Kasus 3 dan 4) menyimulasikan komponen kulit iridophores menghasilkan solusi baru berdasarkan refleksi cahaya yang terbaik. Karena dalam hal ini komponen iridophores menghasilkan solusi yang terbaik yaitu menghasilkan refleksi cahaya yang warnanya sama dengan organ tubuh pada binatang sotong sehingga dapat diasumsikan bahwa warna yang masuk dan warna yang dihasilkan dari refleksi cahaya adalah sama dan direpresentasikan sebagai best point. Pada grup ini variabel R akan bernilai 1.

$$reflection = R * Best.Point[j] \quad (6)$$

Grup 3 (Kasus 5) merupakan proses komponen kulit leucophores untuk menghasilkan solusi baru berdasarkan refleksi cahaya sekitar. Untuk menemukan kesamaan antara warna yang masuk dengan warna yang dihasilkan dari refleksi cahaya, dapat diasumsikan bahwa warna yang masuk merupakan best solution dan warna yang dihasilkan dari refleksi cahaya adalah nilai yang terdapat di sekitar best solution. Nilai visibility pada grup ini ditunjukkan pada persamaan 7 dimana nilai AV_{Best} yang merupakan rata-rata nilai pada individu terbaik di setiap poinnya.

$$visibility = V * (Best.Point[j] - AV_{Best}) \quad (7)$$

Grup 4 (Kasus 6) menggunakan komponen leucophores untuk menghasilkan solusi yang acak dari cahaya yang masuk. Dengan adanya solusi yang acak ini, diharapkan dapat menghasilkan sebuah solusi baru yang baik sehingga solusi yang digunakan pada kasus ini adalah sebuah solusi yang acak.

Setelah seluruh perhitungan untuk solusi baru selesai dilakukan pada setiap grupnya, akan dilakukan perubahan nilai best apabila solusi baru memiliki nilai fitness yang lebih kecil dari best (dalam hal ini fitness semakin kecil menandakan individu semakin baik). Hal ini akan dilakukan terus menerus hingga individu yang didapatkan sesuai dengan kondisi atau persyaratan yang telah ditetapkan.

III. VUE.JS

Vue.js [4] merupakan suatu framework yang berbasis Javascript dan digunakan oleh para developer untuk membangun sebuah website yang interaktif. Vue.js didesain untuk berfokus pada view layer saja dan juga sangat mudah diimplementasikan dan diintegrasikan dengan library lain

ataupun dengan project yang sudah ada sebelumnya. Karena framework ini berbasis Javascript, sehingga dalam penggunaannya dibutuhkan aplikasi Node.js untuk melakukan proses instalasi cara termudah.

Pada Vue.js, terdapat sebuah fitur yang digunakan pembuatan komponen-komponen yang nantinya dapat digunakan kembali dan juga fitur data binding yang sangat baik. Selain itu, dengan menggunakan Vue.js struktur kodenya pun juga sudah sangat tertata baik dan structural. Fitur menarik lain yang terdapat pada Vue.js yaitu dapat membuat sebuah aplikasi single-page yang memiliki kapabilitas yang tinggi dengan adanya kombinasi dari library dan tool yang sudah disediakan.

Konsep component based yang diusung oleh Vue.js merupakan sebuah konsep dimana setiap blok layout dianggap sebagai komponen yang independen dan memiliki fungsi sendiri yang terisolir sehingga mudah digunakan ulang oleh setiap halaman website apabila diperlukan. Setiap komponen tersebut nantinya akan disusun menjadi blok-blok yang saling terkait dan membangun sebuah tampilan utuh pada sebuah halaman website.

Ketika menggunakan Vue.js, pasti akan selalu terkait dengan data, methods dan juga props karena 3 hal tersebut merupakan hal utama yang terdapat pada Vue.js. Data dalam Vue.js merupakan sekumpulan variabel yang digunakan oleh HTML template dan dapat dimanipulasi serta dimanfaatkan oleh internal komponen masing-masing dan juga tidak dapat diakses oleh komponen lainnya. Jadi sebuah variabel di dalam suatu komponen Vue.js hanya akan berpengaruh terhadap komponen itu sendiri.

Methods merupakan opsi dalam Vue.js yang berubah objek berisi fungsi-fungsi. Fungsi ini sendiri akan memiliki berbagai tujuan yang berbeda-beda dan bervariasi baik memanipulasi data ataupun melakukan sebuah logika pada sebuah aplikasi. Fungsi yang terdapat dalam methods ini juga bisa dijadikan sebuah Event yang bisa dipanggil dengan mudah di bagian HTML template dari komponen yang bersangkutan.

Props merupakan sebuah parameter yang dapat dilempar oleh sebuah komponen kepada komponen lainnya. Props merupakan satu dari beberapa cara untuk mendistribusikan data dari satu komponen ke komponen lainnya.

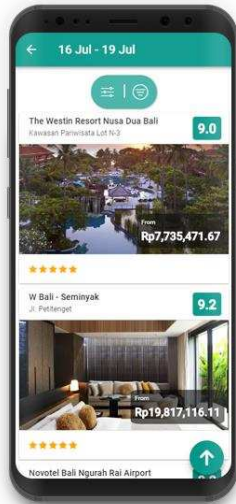
IV. FRAMEWORK7

Framework7 [5] merupakan sebuah framework HTML yang dirilis dibawah lisensi MIT yang bersifat open-source dan terbentuk pada tahun 2014. Framework ini ditujukan untuk pengembangan aplikasi berbasis android maupun iOS yang bersifat hybrid dengan menggunakan HTML, CSS dan Javascript. Framework ini dibentuk dengan tujuan agar para developer dapat membuat sebuah aplikasi mobile tanpa perlu dasar pengetahuan mengenai aplikasi native untuk pembuatan aplikasi mobile itu sendiri.

Meskipun aplikasi berbasis mobile ini dibuat dengan menggunakan HTML, CSS dan Javascript, tampilan dari aplikasi tidak kalah baiknya dengan aplikasi mobile yang dibuat secara native. Jadi dapat dikatakan, Framework7 merupakan sebuah framework website yang sudah menyediakan fitur tampilan responsive untuk aplikasi mobile, sehingga seolah-olah merupakan sebuah aplikasi mobile yang dibuat secara native.

Framework7 ini memang dikhususkan untuk pembuatan aplikasi pada Android maupun iOS saja. Beragam tampilan seperti material design dari google dan komponen-komponen pada iOS juga dapat dipilih agar dapat membuat aplikasi menjadi menarik. Semua itu digunakan dengan menyediakan style-style class pada tag HTML yang sudah dibuat sedemikian rupa menyerupai komponen yang terdapat pada aplikasi mobile serta rangkaian fungsi Javascript yang sudah disediakan oleh Framework7 agar dapat dijalankan layaknya aplikasi mobile.

Framework7 juga sudah didukung dengan bantuan 2 framework Javascript yang dikhususkan untuk pembuatan front-end yaitu React.js dan Vue.js sehingga dapat memperindah tampilan dari desain aplikasi yang telah dibuat. Selain itu sintaks yang terdapat pada Framework7 juga memiliki kemiripan dengan sintaks jQuery. Framework7 juga sudah dilengkapi beragam UI komponen yang mana pengguna hanya perlu melakukan pemanggilan saja untuk menampilkan beragam komponen yang sudah disediakan seperti modal, popup, popover, list view, media list, tab bar, side panel, layout grid, preloader, form, card, button, dll.



Gambar. 4. Contoh Tampilan Framework7

Pada Framework7 juga telah menawarkan fitur yang membuat pengguna dapat merasakan hal yang sama ketika menggunakan aplikasi mobile yang dibuat secara native baik itu dari iOS maupun android seperti swipe back, swipe actions, pull to refresh, infinite scroll, message, dll.

Framework7 sendiri juga sudah menyediakan API yang dapat digunakan untuk pemanggilan fungsi-fungsi seperti menampilkan sebuah popup dan modal dengan menggunakan Javascript dimana dalam penggunaannya memiliki kesamaan dalam menggunakan jQuery. Yang sangat disayangkan dalam penggunaan Framework7 ini adalah dari sisi dokumentasi yang diberikan. Dokumentasi yang diberikan masih belum cukup lengkap karena hanya menjelaskan komponen-komponen serta fitur-fitur yang membuat Framework7 dapat menyerupai aplikasi yang dibuat secara native.

Sedangkan untuk dokumentasi mengenai bagaimana cara mengubah aplikasi dari format HTML dan CSS menjadi format aplikasi mobile baik pada Android maupun iPhone masih belum dimiliki oleh Framework7, sehingga developer harus mempelajari lagi bagaimana penggunaan Apache

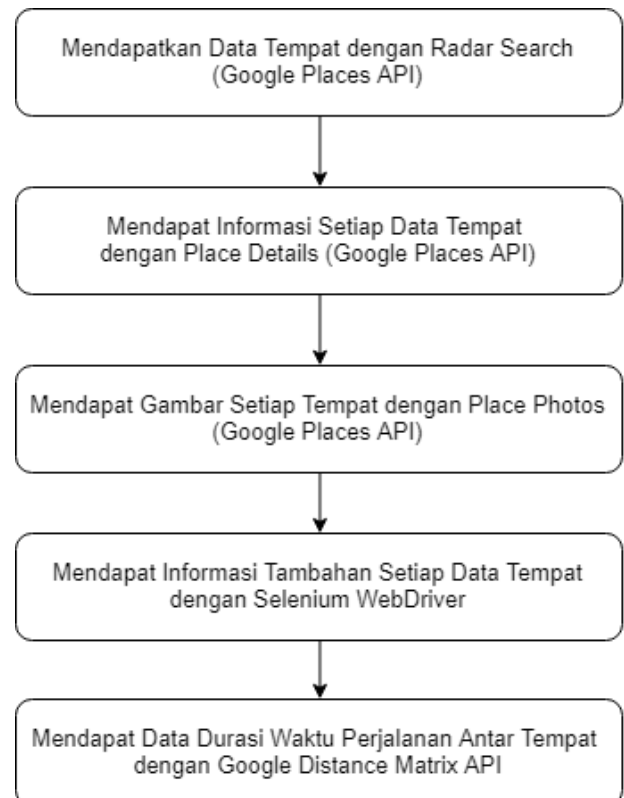
Cordova atau PhoneGap untuk mengubah hasil aplikasi yang berasal dari Framework7 menjadi aplikasi mobile yang ada.

V. DATA

Data tempat yang disediakan pada aplikasi ini merupakan data tempat wisata, tempat makanan serta akomodasi tempat tinggal yang terdapat pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan. Pada pengumpulan mengenai informasi data-data yang digunakan pada aplikasi ini akan digunakan dengan bantuan dari Google Places API [10] dan juga Selenium Webdriver [11] sebagai bantuan utama yang nantinya akan dijelaskan pada subbab selanjutnya.

Seluruh data tempat yang didapatkan nantinya juga hanya berdasarkan hasil pencarian dari Google Places API saja tanpa adanya data tambahan yang dicari secara manual. Data-data yang dihasilkan oleh Google Places API nantinya juga akan disortir lagi karena terdapat beberapa data yang informasinya tidak menunjukkan bahwa tempat tersebut bukan merupakan sebuah tempat wisata, tempat makanan ataupun akomodasi tempat tinggal.

Pada Gambar 5 menunjukkan sebuah alur yang digunakan untuk pencarian data-data tempat yang terdapat pada sistem ini. Proses yang pertama kali harus dilakukan dalam pencarian data-data tempat pada aplikasi ini adalah dengan menggunakan fitur yang terdapat pada Google Places API yaitu Radar Search. Pada tahap ini, akan dilakukan proses pencarian data-data sesuai dengan radius yang telah diinputkan serta tipe-tipe tempat kategori yang diinginkan. Untuk setiap tempat akan dilakukan proses pencarian ini sebanyak 4 kali dengan kategori yang berbeda yaitu untuk mendapatkan data tempat wisata sebanyak satu kali, data tempat makanan sebanyak 2 kali dan akomodasi tempat wisata sebanyak satu kali.



Gambar. 5 Alur Pencarian Data Tempat

Setelah mendapatkan data-data tempat berupa place_id yang berasal dari Google Places API, akan dilanjutkan pada proses selanjutnya yaitu pencarian informasi mengenai setiap hasil data tempat yang sudah didapatkan dengan menggunakan fitur Place Details pada Google Places API. Hasil yang didapatkan dengan menggunakan Place Details adalah informasi seperti nama, alamat, jam buka, dll. Kemudian akan dilanjutkan dengan proses ketiga yaitu mendapatkan gambar tempat dengan menggunakan fitur Place Photos yang terdapat pada Google Places API.

Perlu diketahui bahwa proses pencarian data tempat pada proses pertama hingga ketiga hanya berlaku untuk satu pencarian saja dan setelah seluruh data terkumpul baru dilanjutkan pada proses 4 dan 5 yang menggunakan Selenium Webdriver dan Google Distance Matrix API. [12] Setelah seluruh data tempat terkumpul, akan dilanjutkan dengan proses 4 yaitu mencari informasi tambahan seperti deskripsi singkat, rata-rata durasi menetap pada tempat tersebut dengan menggunakan bantuan Selenium WebDriver.

Setelah seluruh data informasi tambahan untuk setiap tempat wisata selesai didapatkan, akan dilakukan proses sortir secara manual terhadap seluruh data tempat yang sudah didapatkan. Setelah pencarian data selesai dilakukan, proses terakhir yang dilakukan adalah mendapatkan data durasi lama perjalanan setiap tempat yang ada dengan menggunakan Google Distance Matrix API. Untuk setiap proses yang terdapat pada gambar 5 akan dilakukan dengan menggunakan bahasa pemrograman C# untuk menghasilkan kumpulan data tempat yang sudah dikonversikan dalam bentuk CSV.

VI. TRAVELPAYOUTS AFFILIATE PROGRAM

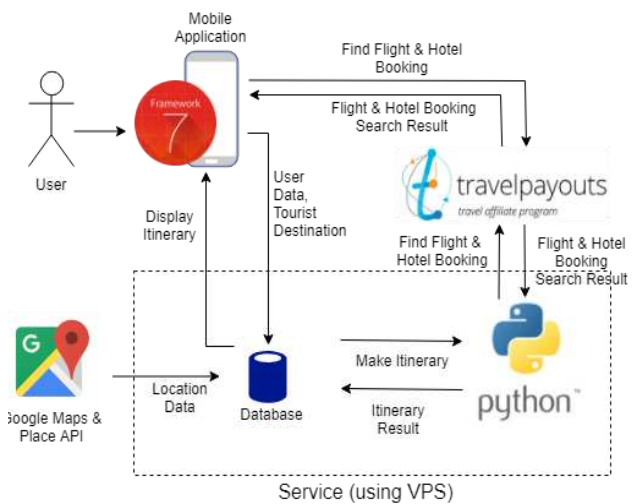
TravelPayouts [13] merupakan sebuah program afiliasi yang bergerak pada bidang travel yang sudah bekerja sama dengan produk-produk seperti JetRadar, Aviasales, Hotellook, dll. Seluruh produk tersebut merupakan sebuah website yang bergerak pada bidang travel namun memiliki fitur utama yang berbeda-beda. JetRadar (Jetradar.com) merupakan salah satu search engine yang terbesar di dunia dan membantu banyak orang dalam membandingkan harga tiket pesawat yang ingin dicari. Aviasales (Aviasales.ru) juga merupakan sebuah search engine yang berguna untuk melakukan pencarian terhadap tiket penerbangan yang berada pada negara Russia dan Ukraina. Sedangkan Hotellook (Hotellook.com) menawarkan fitur untuk melakukan perbandingan harga yang terdapat dalam pencarian akomodasi tempat tinggal yang ada dan tentunya hasil pencarian tersebut berasal dari website travel seperti Booking.com, Agoda.com, Expedia, dll.

Pada program afiliasi TravelPayouts ini telah menyediakan API yang dapat digunakan oleh developer untuk membuat situs pencarian tiket pesawat maupun hotel yang diakses melalui HTTP URL. Jadi dengan menggunakan fitur API ini, developer murni membuat seluruh tampilan dari pencarian tersebut secara manual karena yang hasil yang diberikan akan berupa sebuah JSON yang nantinya akan diubah dalam bentuk tampilan pada sebuah website. Terdapat 5 API yang nantinya dapat digunakan oleh para developer yaitu:

1. API Flight Data Access
2. Affiliate Statistics API

3. Mobile SDK for iOS and Android
4. Flight Search API
5. Hotel Search API

Pada aplikasi ini akan digunakan Flight Search API dan juga Hotel Search API yang digunakan untuk pencarian tiket pesawat serta akomodasi tempat tinggal pada pembuatan jadwal wisata nantinya. Selain itu pada aplikasi ini juga akan menyediakan fitur untuk melakukan pencarian dan pembelian tiket pesawat serta akomodasi tempat tinggal secara independen. Karena aplikasi yang dibuat merupakan sebuah aplikasi mobile yang bersifat hybrid yakni berbasis HTML, sehingga digunakanlah Flight Search API dan Hotel Search API sebagai hasil pencariannya dan untuk desainnya akan dibentuk secara manual seluruhnya. Pada penggunaan akses terhadap Flight Search API dan juga Hotel Search API, akan dibutuhkan request khusus dan dikirimkan pada email aktif yang dimiliki oleh TravelPayouts yang nantinya akan dilakukan verifikasi lebih lanjut apakah request untuk penggunaan akses terhadap kedua API tersebut disetujui oleh pihak TravelPayouts.



Gambar. 6. Alur Sistem

Flight Search API merupakan salah satu API yang disediakan oleh TravelPayouts untuk melakukan hasil pencarian terhadap tiket pesawat dimana dapat diakses melalui HTTP URL dan hasil response yang diberikan dalam bentuk format JSON. Akan terdapat 3 jenis pencarian utama dalam pencarian tiket pesawat nantinya yaitu round trip, one way dan open jaw. Round Trip merupakan sebuah pencarian tiket pesawat dimana akan dilakukan pencarian sekaligus untuk pergi dan pulang. One Way merupakan sebuah pencarian tiket pesawat untuk 1 destinasi saja. Sedangkan Open Jaw merupakan sebuah pencarian tiket pesawat dimana akan dilakukan pencarian lebih dari 1 destinasi yang membentuk suatu rute perjalanan.

Selain adanya fitur untuk melakukan pencarian penerbangan yang terdapat dalam aplikasi ini, juga terdapat sebuah fitur untuk melakukan pencarian terhadap akomodasi tempat tinggal sesuai dengan tempat dan tanggal booking yang dipilih. Pencarian akomodasi tempat tinggal ini juga dapat terwujud dengan bantuan dari TravelPayouts API yaitu Hotel Search API dengan kerjasamanya pada website Hotellook.com. Hotel Search API ini akan memberikan hasil response berupa hasil pencarian terhadap akomodasi tempat tinggal yang berasal dari agensi-agensinya seperti Agoda.com,

Booking.com, Zenhotel.com yang telah bekerja sama dengan Hotellook.

VII. DESAIN SISTEM

Pada sistem ini akan berfokus pada sebuah sistem berbasis aplikasi mobile dimana pengguna dapat membuat sebuah jadwal wisata secara otomatis ataupun manual pada multi kota maupun negara yang terdapat pada 3 negara yaitu Hong Kong, Taiwan dan Korea Selatan. Selain terdapat pembuatan jadwal wisata, sistem ini juga sudah dilengkapi dengan proses pencarian tiket penerbangan serta pencarian akomodasi tempat tinggal dan juga dapat membeli secara langsung. Sistem pencarian baik tiket penerbangan maupun akomodasi tempat tinggal ini juga sudah terintegrasikan dengan sistem pembuatan jadwal wisata serta dapat digunakan secara independen apabila tidak ingin membuat sebuah jadwal wisata yang ada. Selain itu pada sistem ini juga menyediakan sebuah fitur yang digunakan untuk memilih kurs mata uang sehingga para pengguna dapat lebih mudah mengetahui harga yang ada sesuai dengan preferensi yang diinginkan oleh pengguna tersebut.

Pada gambar 6 menunjukkan sebuah alur yang dimiliki oleh sistem dan aplikasi ini akan memiliki nama TripWander. Pada awalnya pengguna dapat membuat sebuah jadwal wisata secara otomatis maupun manual. Pertama-tama pengguna harus mengisi data-data atau informasi yang berkaitan sebelum memulai pembuatan jadwal wisata. Setelah seluruh data telah terisi dan pengguna memutuskan untuk memilih jadwal wisata yang terbuat secara otomatis, maka pengguna harus menunggu hingga jadwal wisata tersebut terbentuk dan akan ditunjukkan melalui notifikasi yang terdapat pada device dari pengguna tersebut.

Jika pengguna memilih untuk membuat jadwal wisatanya sendiri, maka akan dilanjutkan pada halaman selanjutnya untuk mengatur jadwal wisatanya. Hal tersebut juga berlaku pada pengguna yang ingin mengatur ulang jadwal wisata yang telah terbentuk secara otomatis sebelumnya. Selain itu pengguna juga dapat melakukan pencarian terhadap tiket penerbangan serta akomodasi tempat tinggal sesuai dengan tanggal yang diinginkan pada ini yang menggunakan sebuah program afiliasi yang dimiliki oleh TravelPayouts.

Sebelum aplikasi ini terbentuk, seluruh data mengenai tempat-tempat wisata serta akomodasi tempat tinggal yang ada juga sudah didapatkan terlebih dahulu dengan menggunakan bantuan Google Maps API. Pada aplikasi ini juga menggunakan MySQL sebagai database dimana menyimpan seluruh data mengenai informasi pengguna, jadwal wisata, data tempat wisata, data nilai kurs mata uang. Untuk pembuatan aplikasi mobile yang terdapat pada sistem ini menggunakan Framework7 dan Vue.js yang mana merupakan sebuah framework berbasis HTML, CSS dan JavaScript dalam pembuatan aplikasi mobile.

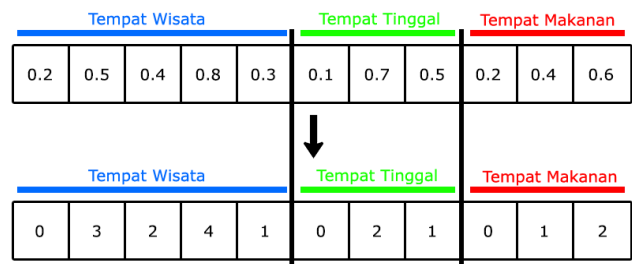
Sedangkan pada backend, akan digunakan Flask [14] yang merupakan sebuah microframework berbasis Python dan digunakan untuk pengaksesan database serta pembuatan sebuah service yang nantinya akan dipanggil oleh aplikasi mobile yang sudah terbuat agar dapat mengakses database yang ada. Selain itu untuk pembuatan jadwal wisata secara otomatis dengan menggunakan Algoritma Cuttlefish juga menggunakan bahasa pemrograman Python. Karena seluruh penggunaan backend

serta pembuatan jadwal wisata yang berbasis Python, maka digunakan sebuah VPS (Virtual Private Server) yang memiliki operating system berbasis Linux dengan versi 14.04. Untuk versi Python yang digunakan adalah Python versi 3.5.4.

VIII. PEMBUATAN JADWAL WISATA OTOMATIS

Sebelum penjelasan tentang bagaimana cara menghitung nilai fitness pada pembuatan jadwal wisata, akan dijelaskan terlebih dahulu mengenai representasi individu yang terdapat pada aplikasi ini. Hal pertama yang dilakukan dalam pembuatan representasi individu adalah dengan membagi masing-masing data tempat wisata, akomodasi tempat tinggal dan tempat makanan yang sudah disortir sesuai dengan budget yang dipilih oleh pengguna ke dalam variabel. Semua data tersebut juga sudah disortir berdasarkan kota yang sedang dipilih.

Kemudian jumlah dimensi yang terdapat pada individu merupakan perpaduan jumlah dari data tempat wisata, akomodasi tempat tinggal dan tempat makanan yang sudah diperoleh dari database. Lalu pada saat proses inialisasi individu baru, akan dilakukan dengan cara mengisi masing-masing dimensi dengan nilai acak antara 0 hingga 1.



Tempat Wisata					Tempat Tinggal			Tempat Makanan		
0.2	0.5	0.4	0.8	0.3	0.1	0.7	0.5	0.2	0.4	0.6
					↓					
Tempat Wisata					Tempat Tinggal			Tempat Makanan		
0	3	2	4	1	0	2	1	0	1	2

Gambar. 7 Ilustrasi Representasi Individu

Setelah seluruh dimensi pada individu baru tersebut telah terisi dengan nilai acak antara 0 hingga 1, nilai yang terdapat pada masing-masing dimensi tersebut akan diubah menjadi indeks yang mana indeks 0 merupakan nilai yang paling kecil dan juga proses pengubahan nilai indeks tersebut akan dibagi menjadi 3 bagian yaitu masing-masing untuk tempat wisata, akomodasi tempat tinggal dan tempat makanan.

Pada gambar 7 merupakan sebuah contoh ilustrasi dari representasi individu yang terdapat dalam pembuatan jadwal wisata. Sebagai contoh terdapat 5 tempat wisata, 3 akomodasi tempat tinggal dan 3 tempat makanan. Lalu jumlah dimensinya akan menjadi 11 pada individu tersebut dan pada masing-masing dimensi akan diisi nilai acak antara 0 hingga 1 dan perlu diketahui pada ilustrasi ini nilai acak memiliki keakuratan 1 angka di belakang koma saja, sedangkan pada implementasinya nilai acak memiliki keakuratan 8 angka di belakang koma.

Setelah masing-masing dimensi telah terisi oleh nilai acak, kemudian akan diubah menjadi indeks sesuai dengan bagiannya dimana indeks terkecil menandakan nilai terendah. Hal ini dilakukan dengan tujuan agar dapat mengetahui tempat mana yang nantinya akan diambil untuk dijadikan jadwal wisata dengan cara pengambilan data sesuai dengan indeks yang terdapat pada dimensi-dimensi tersebut.

Penggunaan Algoritma Cuttlefish pada aplikasi ini hanya sebatas pada 1 kota saja dan nantinya akan dihubungkan pada kota dan negara selanjutnya. Hal ini bertujuan agar

hasil yang didapatkan lebih optimal mengingat banyaknya tempat-tempat yang akan dipilih dan juga penulis masih belum menemukan sebuah metode untuk memicu Algoritma Cuttlefish ini untuk berpindah ke kota atau negara selanjutnya. Meskipun penggunaannya hanya sebatas 1 kota saja, tetapi semuanya sudah terintegrasi agar dapat menghubungkan ke kota dan negara-negara selanjutnya yang ada.

Pada pembuatan jadwal wisata dengan Algoritma Cuttlefish ini juga terdapat beberapa batasan yang digolongkan menjadi soft constraints dan hard constraints. Batasan-batasan soft constraints ini antara lain jam makan dan waktu aktivitas setiap harinya. Sedangkan batasan yang termasuk hard constraints adalah jumlah hari, penerbangan, perpindahan antar kota dan negara, pencarian akomodasi tempat tinggal serta seluruh tempat wisata yang terdapat pada jadwal harus dalam keadaan buka.

Batasan pertama yang dibahas adalah jumlah hari menetap nantinya pada jadwal wisata yang akan dibuat. Pada negara Hong Kong, terdapat 1 kota yang dapat dikunjungi yaitu Hong Kong itu sendiri. Untuk Taiwan, terdapat 5 kota yang dapat dikunjungi yaitu Taipei, Tainan, Taichung, Hsinchu dan Kaohsiung. Sedangkan untuk negara Korea Selatan, terdapat 3 kota yang dapat dikunjungi yaitu Seoul, Busan dan Gyeongju. Untuk setiap negara akan memiliki jumlah minimum hari menetap selama 3 hari dan jumlah hari maksimal yang berbeda-beda yaitu Hong Kong 5 hari, Taiwan 12 hari dan Korea Selatan selama 8 hari.

Penerbangan dalam hal ini berlaku untuk awal keberangkatan dan juga untuk setiap kali berpindah negara. Nantinya tiket pesawat yang akan dipilih merupakan tiket pesawat yang memiliki jam sampai di antara pukul 05.00 hingga 13.00 apabila memungkinkan. Lalu para wisatawan nantinya akan diberikan asumsi waktu selama 60 menit yang digunakan untuk pengambilan barang hingga siap untuk berangkat menuju tempat akomodasi. Apabila wisatawan sampai pada negara tujuan melebihi dari waktu 18.00, maka secara otomatis jadwal wisata akan dilanjutkan keesokan harinya. . Lalu apabila kota yang dikunjungi tidak memiliki bandara, maka akan menuju ke bandara terdekat pada kota yang tersedia. Sebagai contoh, apabila wisatawan hendak berpindah negara dari kota Hsinchu, Taiwan maka wisatawan harus menuju ke bandara yang terdapat pada kota Taipei, Taiwan.

Batasan ketiga merupakan batasan yang terjadi apabila wisatawan akan berpindah kota sehingga batasan ini bersifat opsional. Setiap kali wisatawan akan berpindah kota, maka hal ini selalu terjadi pada pagi hari dimana wisatawan pertama kali menjalankan aktivitas setiap harinya. Untuk transportasi yang ada, akan menggunakan transportasi umum jalur darat sesuai dengan petunjuk yang terdapat pada Google Maps API. Lalu setelah berpindah kota, hal pertama yang dilakukan adalah menuju ke tempat tinggal pada kota yang baru saja dituju untuk melakukan proses check-in yang akan diberi waktu untuk check-in selama 60 menit setiap kali pertama kali mengunjungi tempat akomodasi.

Batasan pindah negara ini hampir sama dengan batasan mengenai pindah kota yaitu bersifat opsional apabila wisatawan mengunjungi lebih dari 2 negara. Wisatawan juga diberikan opsi apabila tidak ingin kembali ke negara asalnya. Pada batasan ini, wisatawan harus sudah sampai pada bandara 90 menit sebelum jam boarding seperti yang tertera pada tiket pesawat. Apabila jam boarding pada tiket

pesawat adalah subuh atau sebelum jam aktivitas dimulai, maka wisatawan juga akan langsung menuju ke bandara 90 menit sebelumnya.

Batasan selanjutnya adalah budget dan berfokus pada pengaruh pembuatan jadwal wisata. Budget yang dimaksudkan ini hanyalah budget yang digunakan dalam pemilihan kualitas akomodasi tempat tinggal yang ada saja karena dalam pengumpulan datanya tidak ditemukan adanya data mengenai biaya yang dikeluarkan untuk melakukan perjalanan dengan transportasi umum dan juga biaya rata-rata yang dikeluarkan untuk masuk pada sebuah tempat wisata. Untuk budget, akan terdapat 2 pilihan yaitu low dan high budget. Low budget merupakan hotel-hotel yang memiliki 3 bintang atau di bawahnya, sedangkan high budget merupakan hotel yang memiliki bintang 4 ke atas.

Pada batasan pencarian akomodasi tempat tinggal sesuai dengan kota yang dituju. Pencarian akomodasi tempat tinggal ini akan didasari pada 2 hal yaitu jarak antara bandara awal atau akhir menuju akomodasi tempat tinggal dan budget. Untuk jarak antara bandara awal atau akhir dengan akomodasi tempat tinggal diusahakan tidak terlalu jauh untuk menghemat waktu perjalanan dan tidak lupa juga dengan batasan-batasan lain yang ada yang nantinya terdapat pada perhitungan fitness.

Pada batasan pemilihan tempat wisata yang nantinya akan dibuat menjadi jadwal perjalanan wisata. Tempat wisata ini dibagi menjadi 4 kategori yaitu must see, culture, nature dan recreation yang digunakan untuk wisatawan dalam pembuatan jadwal sesuai dengan ketertarikannya antara 4 kategori tersebut. Must see merupakan kategori tempat yang dapat dikatakan terkenal pada kotanya dan dalam hal ini tempat wisata tersebut dikategorikan sebagai must see dari jumlah rating yang cukup tinggi yang terdapat pada google. Culture merupakan kategori tempat wisata seperti museum, tempat ibadah, dsb. Nature merupakan kategori tempat wisata yang berhubungan dengan alam seperti air terjun, pantai, pegunungan, dll. Recreation merupakan kategori tempat wisata seperti taman, taman bermain dan juga mall.

Sebelum pembuatan jadwal wisata, para wisatawan akan memberikan nilai terhadap ketertarikannya terhadap 4 kategori yang sudah disediakan. Nilai yang ada akan memiliki rentang -2 hingga 2. Semakin rendah nilai yang diberikan terhadap kategori yang ada maka wisatawan tidak tertarik terhadap kategori yang sedang diberi penilaian dan hal ini juga berlaku kebalikannya yaitu semakin tinggi nilai yang diberikan maka wisatawan tertarik terhadap kategori tersebut. Pada jadwal wisata ini juga dipastikan setiap tempat wisata yang ada harus dalam keadaan buka. Pada pembuatan jadwal wisata ini juga setiap tempat wisata hanya dapat dikunjungi maksimal satu kali saja.

Untuk batasan jam makan, terdapat sebuah asumsi bahwa pada jadwal wisata yang telah terbuat para wisatawan sudah menerima sarapan dari akomodasi tempat tinggal setiap pagi hari. Lalu para wisatawan harus makan 2 kali sehari baik itu makan pada tempat makan atau pada tempat wisata yang terdapat tempat makan di dalamnya. Apabila wisatawan tidak makan sebanyak 2 kali sehari atau makan disaat jam makan telah selesai, maka akan dikenakan penalty pada perhitungan fitness. Untuk jam makan siang akan memiliki rentang antara pukul 12.00 hingga 14.00 sedangkan untuk jam makan malam akan memiliki rentang antara pukul 18.00 hingga 20.00.

TABEL I
CONTOH PERHITUNGAN TOTAL INTEREST

No	Must See		Culture		Nature		Recreation		Total Interest (Nilai Asli)	Keterangan
	B	A	B	A	B	A	B	A		
1	2	6	2	12	2	12	2	12	0(40)	Nilai <i>interest</i> sama
2	2	6	1	6	0	0	-1	-3	9(9)	-
3	-1	-3	-2	-6	-1	-3	-2	-6	0(-18)	Seluruh nilai <i>interest</i> < 0
4	1	3	2	12	2	12	0	0	24(27)	Melebihi batas maksimal

Batasan yang terakhir merupakan jam aktivitas yang akan dilakukan setiap harinya oleh wisatawan. Sebelumnya, wisatawan harus memilih jam aktivitas setiap harinya yaitu untuk pagi hari memiliki rentang antara pukul 07.00 hingga 09.50 pagi. Sedangkan untuk jam malamnya memiliki rentang antara pukul 20.00 hingga 22.50. Nantinya pada hasil jadwal yang telah terbuat akan diusahakan semaksimal mungkin agar para wisatawan menyelesaikan aktivitas per harinya tepat waktu dan akan dikenakan penalty apabila pulang terlalu cepat atau terlalu larut malam.

Perhitungan rumus fitness yang digunakan dalam pembuatan jadwal wisata dengan Algoritma Cuttlefish adalah menggunakan sistem diskon dan penalti. Perhitungan nilai fitness akan menggunakan total waktu secara keseluruhan dari lama perjalanan antar tempat dan juga durasi menetap pada setiap masing-masing tempat dalam satuan waktu detik. Semakin kecil hasil fitness, maka hasil jadwal juga akan semakin baik. Selain total waktu keseluruhan, akan terdapat penalti apabila melanggar batasan-batasan yang sudah dijelaskan sebelumnya. Perlu juga diingat bahwa seluruh perjalanan wisata ini menggunakan transportasi umum.

Setelah seluruh jadwal wisata terbentuk yang tentunya juga sesuai dengan representasi individu dan batasan yang ada, maka akan dilakukan proses untuk perhitungan nilai fitness total. Nilai fitness total akan didapatkan dari jumlah waktu yang berasal dari durasi lama perjalanan dan durasi waktu menetap pada seluruh tempat yang terdapat pada jadwal yang terbentuk. Kemudian juga akan ditambahkan dengan total nilai penalti yang didapatkan serta akan mendapatkan diskon yang berasal dari jumlah tempat yang dikunjungi serta kecocokan antara tempat wisata yang terpilih dengan nilai interest yang telah diisikan oleh pengguna.

$$fitness = place_time + penalty - (place_time * (place_disc + interest_disc)) \quad (8)$$

Pada rumus 8 merupakan sebuah rumus yang digunakan untuk menghitung nilai fitness secara keseluruhan. *place_time* dan *penalty* merupakan sebuah variabel yang berisikan jumlah waktu yang dilalui dalam detik dan jumlah penalti yang didapat karena melanggar batasan-batasan yang terdapat pada pembuatan jadwal otomatis ini. Kemudian setelah kedua variabel tersebut dijumlahkan akan dilakukan pengurangan dengan nilai dari variabel *place_time* yang dikalikan dengan jumlah presentase antara pengaruh jumlah tempat yang dikunjungi dan nilai interest pada masing-masing tempat wisata.

$$place_disc = \frac{1}{place_target_per_day * real_stay_days * 2} * total_place \quad (9)$$

Pada rumus 9 merupakan rumus yang digunakan untuk mendapatkan presentase jumlah tempat pada jadwal yang telah terbentuk. Variabel *place_target_per_day* merupakan target jumlah tempat wisata dan tempat makanan yang diharapkan dalam 1 hari yaitu bernilai sebesar 10. Kemudian seperti yang sudah dijelaskan, variabel *real_stay_days* merupakan jumlah total hari yang terdapat pada jadwal wisata yang terbentuk (akan dikurangkan 1 hari apabila pada saat datang melebihi pukul 18.00). *total_place* merupakan variabel yang berisikan jumlah total tempat wisata dan tempat makanan yang dikunjungi hingga jadwal selesai terbentuk. Pada rumus tersebut, terdapat sebuah pengali yaitu sebesar 2. Hal ini bertujuan agar nilai yang dihasilkan berkisar antara 0 hingga 0.5.

Sebelum melakukan perhitungan terhadap presentase nilai interest, akan terdapat beberapa aturan mengenai masing-masing nilai interest yang berkisar antara -2 sampai 2:

1. Akan terdapat sebuah nilai *interest_multiplier* bernilai 3 yang nantinya digunakan untuk mengalikan setiap nilai interest pada masing-masing interest. Hal ini dilakukan agar pengaruh nilai interest lebih kuat dengan adanya *interest_multiplier*.
2. Akan terdapat sebuah nilai *like_multiplier* bernilai 2 yang merupakan sebuah multiplier apabila nilai interest bernilai positif. Jadi, setelah dilakukan pengalihan terhadap *interest_multiplier* akan dilakukan pengalihan selanjutnya terhadap *like_multiplier* khusus untuk interest yang bernilai positif kecuali untuk kategori interest *must_see*. Hal ini dilakukan agar nilai interest positif akan lebih dominan daripada nilai yang bernilai negatif, sedangkan untuk kategori *must_see* merupakan sebuah kategori yang bukan berdasarkan selera dari pengguna sehingga tidak diperlukan adanya pengalihan terhadap *like_multiplier*.
3. Apabila nilai dari masing-masing interest adalah sama atau seluruh nilai interest kurang dari 0, maka akan dianggap ketertarikan pengguna terhadap masing-masing interest adalah netral sehingga nilai dari masing-masing interest akan diubah menjadi 0. Nilai total interest pada satu tempat wisatanya nantinya akan dibatasi dengan nilai Sesuai dengan rumus 10.

$$total_interest = 2 * 4 * interest_multiplier \quad (10)$$

Pada rumus 10, *total_interest* akan bernilai 24 jika nilai *interest_multiplier* bernilai 3. Hal ini dilakukan untuk

membatasi nilai maksimal dari total interest pada setiap wisata dan memberikan pengaruh nilai yang lebih pada nilai interest yang bernilai positif.

Berdasarkan aturan-aturan yang baru saja dijelaskan, aturan-aturan tersebut dibuat dengan tujuan agar nilai interest yang diisi oleh pengguna memiliki pengaruh yang lebih dominan terutama nilai yang berisikan positif dan juga agar dapat memperbaiki nilai-nilai interest yang telah diisi oleh pengguna apabila memiliki kesamaan nilai pada seluruh interest ataupun seluruh nilai interest yang berisikan kurang dari 0 yang mana sudah terdapat contoh perhitungan total interest pada tabel I.

Setelah mengetahui beberapa aturan yang terdapat pada perhitungan nilai total interest untuk setiap tempat wisata, terdapat sebuah rumus yang digunakan untuk mendapatkan presentase pengaruh tempat wisata terdapat nilai interest. Nantinya presentase ini dapat menjadi sebuah diskon maupun penalti. Dalam hal ini, penalti yang dimaksudkan jika nilai total fitness yang terdapat pada sebuah tempat wisata bernilai negatif dan sebaliknya jika bernilai positif maka akan menjadi sebuah diskon.

$$interest_disc = \frac{sum_list_interest - total_interest_{min}}{total_interest_{max} - total_interest_{min}} - 0.5 \quad (11)$$

Pada rumus 11 menunjukkan sebuah rumus yang digunakan untuk mendapatkan presentase pengaruh kecocokan antara tempat wisata dengan nilai interest yang telah ditentukan oleh pengguna dengan rentang nilai antara -0.5 hingga 0.5. Hal ini dilakukan apabila nilai presentase maksimal untuk diskon terhadap pengaruh jumlah tempat yang bernilai sebesar 0.5 dan juga nilai presentase maksimal untuk nilai interest yang bernilai 0.5, jika digabungkan nilai presentase maksimalnya akan menjadi 1 sehingga menyebabkan kedua faktor tersebut memiliki bobot pengaruh yang sama.

IX. UJI COBA

Ujicoba yang dilakukan antara lain adalah uji coba parameter yang digunakan untuk mengetahui parameter-parameter yang terbaik untuk digunakan dalam pembuatan jadwal wisata. Kemudian juga dilakukan uji coba skenario yang digunakan untuk mengetahui sebuah hasil jadwal wisata yang dibentuk secara otomatis beserta dengan deskripsi tentang fitness yang ada. Pertama, Uji coba yang dilakukan adalah uji coba parameter dengan menggunakan seluruh data input wisata yang sama yaitu berupa 1 kota saja.

TABEL II
PENGARUH JUMLAH ITERASI TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
1000	200	0.55	-0.55	1	-1	1128817.20
5000	200	0.55	-0.55	1	-1	723584.87
10000	200	0.55	-0.55	1	-1	612637.52
25000	200	0.55	-0.55	1	-1	286305.10
50000	200	0.55	-0.55	1	-1	252804.40
75000	200	0.55	-0.55	1	-1	252804.40

Kota yang digunakan adalah Kota Taipei dan perjalanan wisata dimulai selama 3 hari yaitu dari tanggal 18 Mei 2018 hingga 20 Mei 2018. Pada uji coba ini juga diasumsikan

akan melakukan perjalanan pada 1 negara saja lalu kembali ke kota asal, sehingga untuk jam kedatangan yang ada adalah pukul 10.15 dan jam keberangkatan untuk pulang ke negara asal adalah pukul 21.15. Selain itu pada uji coba ini juga tidak digunakan interest atau dapat dikatakan seluruh nilai interest yang ada bernilai 0.

Pada waktu untuk memulai serta mengakhiri aktivitas adalah mulai dari pukul 08.00 hingga 21.30 setiap harinya. Pada uji coba ini, parameter yang akan diuji coba adalah jumlah iterasi, jumlah populasi, nilai parameter *r1*, *r2*, *v1* dan *v2*. Untuk uji coba ini pun juga sudah menggunakan random seed yang berasal dari library pada Python yang bertujuan untuk menghasilkan nilai acak yang sama meskipun dilakukan pada waktu yang berbeda.

Berdasarkan hasil uji coba untuk masing-masing parameter yang terdapat pada Algoritma Cuttlefish untuk pembuatan jadwal otomatis dan telah tercantum mulai dari tabel II hingga tabel VII, berhasil didapatkan nilai parameter terbaik yang digunakan pada pembuatan jadwal wisata dengan Algoritma Cuttlefish adalah jumlah iterasi sebesar 50000, jumlah populasi sebesar 300, nilai parameter *v1* dan *v2* sebesar 1 dan -1 serta nilai parameter *r1* dan *r2* yang bernilai sebesar 0.5 dan -0.5.

TABEL III
PENGARUH JUMLAH POPULASI TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	50	0.55	-0.55	1	-1	296317.30
50000	100	0.55	-0.55	1	-1	206155.92
50000	150	0.55	-0.55	1	-1	529473.50
50000	200	0.55	-0.55	1	-1	252804.40
50000	250	0.55	-0.55	1	-1	114071.50
50000	300	0.55	-0.55	1	-1	111680.63
50000	350	0.55	-0.55	1	-1	115408.88
50000	400	0.55	-0.55	1	-1	126658.43

TABEL IV
PENGARUH PARAMETER V TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.55	-0.55	0.25	-0.25	150273.00
50000	300	0.55	-0.55	0.5	-0.5	122142.37
50000	300	0.55	-0.55	0.75	-0.75	135548.20
50000	300	0.55	-0.55	1	-1	111680.63
50000	300	0.55	-0.55	1.25	-1.25	123374.60
50000	300	0.55	-0.55	1.5	-1.5	127177.60
50000	300	0.55	-0.55	2	-2	133491.00

TABEL V
PENGARUH PARAMETER R TERHADAP PERBUATAN JADWAL WISATA DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.25	-0.25	1	-1	145588.77
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.75	-0.75	1	-1	114336.42
50000	300	1	-1	1	-1	133681.22
50000	300	1.25	-1.25	1	-1	119116.15
50000	300	1.5	-1.5	1	-1	114336.42
50000	300	2	-2	1	-1	114336.42

TABEL VI
PENGARUH PARAMETER R1 TERHADAP PERBUATAN JADWAL WISATA
DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.25	-0.5	1	-1	115747.12
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.75	-0.5	1	-1	117234.52
50000	300	1	-0.5	1	-1	114336.42
50000	300	1.25	-0.5	1	-1	144618.8
50000	300	1.5	-0.5	1	-1	117804
50000	300	1.75	-0.5	1	-1	114336.42
50000	300	2	-0.5	1	-1	119116.15

TABEL VII
PENGARUH PARAMETER V1 TERHADAP PERBUATAN JADWAL WISATA
DENGAN ALGORITMA CUTTLEFISH

Jumlah Iterasi	Jumlah Populasi	R1	R2	V1	V2	Nilai Fitness
50000	300	0.5	-0.5	0.25	-1	126823.35
50000	300	0.5	-0.5	0.5	-1	127916.23
50000	300	0.5	-0.5	0.75	-1	136507.20
50000	300	0.5	-0.5	1	-1	109656.00
50000	300	0.5	-0.5	1.25	-1	112376.60
50000	300	0.5	-0.5	1.5	-1	158921.87
50000	300	0.5	-0.5	1.75	-1	132779.33
50000	300	0.5	-0.5	2	-1	128078.90

Uji coba berikutnya adalah uji coba skenario. Pada uji coba ini akan dijabarkan secara lengkap mulai dari akomodasi tempat tinggal yang dipilih dan juga perhitungan nilai fitness beserta dengan data input awal sesuai dengan yang terdapat pada tabel VIII. Setelah hasil jadwal terbentuk yang berasal dari data input yang terdapat pada tabel VIII, telah didapatkan nilai fitness sebesar 73271.84 dan terdapat sebuah hasil jadwal wisata yang dapat dilihat pada tabel IX

TABEL IX
JADWAL WISATA UJI COBA SKENARIO

No	Nama	Jenis	Waktu	Interest
Hari Pertama – 13 September 2018				
1	Hong Kong International Airport	Bandara	13.50 – 14.50	-
2	Mandarin Oriental, Hong Kong	Hotel	15.40 – 16.40	-
3	Keung Kee Meat Shop	Tempat Makan	17.00 – 17.45	-
4	Central-Mid-Levels Escalators	Tempat Wisata	18.07 – 19.07	must_see
5	Bubba Gump	Tempat Makan	19.49 - 21.19	-
6	Mandarin Oriental, Hong Kong	Hotel	22.01 – 22.01	-
Hari Kedua – 14 September 2018				
7	Mandarin Oriental, Hong Kong	Hotel	08.00 – 08.00	-
8	Lung Fu Shan Country Park	Tempat Wisata	08.18 – 09.03	recreation
9	Dr Sun Yat-sen Museum	Tempat Wisata	09.18 – 10.48	culture
10	Airport Core Programme Exhibition Centre	Tempat Wisata	11.56 – 12.26	recreation
11	Main Street Corner Café	Tempat Makan	13.42 – 15.42	-
12	Peak Tower	Tempat Wisata	17.12 – 18.42	must_see, recreation
13	Yardbird	Tempat Makan	19.29 – 21.29	-
14	Mandarin Oriental, Hong Kong	Hotel	21.43 – 21.43	-
Hari Ketiga – 15 September 2018				
15	Mandarin Oriental, Hong Kong	Hotel	08.00 – 08.00	-
16	Hong Kong Avenue of Comic Stars	Tempat Wisata	08.16 – 09.46	recreation
17	Noon Day Gun	Tempat Wisata	10.11 – 12.11	culture
18	ABC Kitchen	Tempat Makan	12.41 – 13.26	-
19	Soho, Hong Kong	Tempat Wisata	13.45 – 15.15	must_see, recreation
20	Mandarin Oriental, Hong Kong	Hotel	15.28 – 16.28	-
21	Hong Kong International Airport	Bandara	17.19 – 17.19	-

dengan mayoritas tempat wisatanya memiliki interest recreation dan must_see.

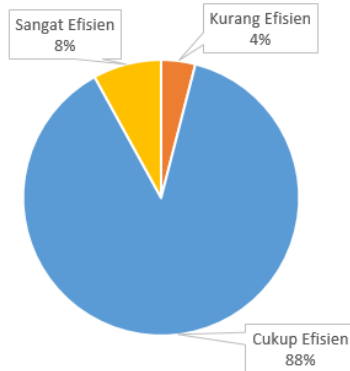
Dari 8 tempat wisata yang akan dikunjungi pada jadwal wisata untuk uji coba 2 ini, dapat dilihat terdapat 2 tempat wisata yang memiliki interest culture, 1 tempat wisata yang memiliki interest must_see, 3 tempat wisata yang memiliki interest recreation dan 2 tempat wisata yang memiliki interest must_see, recreation. Sehingga dapat disimpulkan terdapat 3 dari 8 data atau 37.5% data yang memiliki tipe interest must_see, 2 dari 8 data atau 25% data yang memiliki tipe interest culture, 5 dari 8 data atau 62.5% data yang memiliki tipe interest recreation dan tidak terdapat 1 pun tempat wisata yang memiliki tipe interest nature.

TABEL VIII
DATA INPUT UJI COBA SKENARIO

Tujuan	Hong Kong (3 Hari) – Hong Kong
Jumlah Wisatawan	Dewasa : 2
Tanggal Keberangkatan	12 September 2018
Kota Asal	Surabaya, Indonesia (Kembali)
Interest	Must See : 1 Culture : 0 Nature : -2 Recreation : 2
Budget	High
Waktu Aktivitas	08.00 – 21.30

Selain uji coba skenario, juga telah dilakukan uji coba berupa kuesioner yang dilakukan pada 25 responden. Sebanyak 88% responden mengatakan bahwa hasil jadwal wisata secara otomatis yang terbentuk sudah cukup efisien dan sesuai dengan keinginan pengguna dan terdapat 8% yang merasa bahwa hasil jadwal wisata yang terbentuk

sudah sangat efisien. Sedangkan terdapat 4% responden yang berpendapat bahwa hasil jadwal wisata yang terbentuk masih belum cukup efisien dan untuk visualisasi data tersebut sudah terdapat pada gambar 8. Dari tanggapan responden tersebut, dapat disimpulkan bahwa hasil jadwal wisata yang terbentuk secara otomatis sudah cukup efisien, namun masih harus ditingkatkan kembali dari sisi perhitungan fitness dan dari data tempat yang disediakan.



Gambar. 8 Hasil Uji Coba Kepuasan Pengguna Terhadap Hasil Jadwal

X. KESIMPULAN

Dari hasil uji coba yang dilakukan, Algoritma Cuttlefish mampu menghasilkan sebuah jadwal perjalanan wisata yang sudah diintegrasikan dengan pencarian tiket penerbangan dan akomodasi tempat tinggal dengan bantuan TravelPayouts API.

Berdasarkan hasil uji coba terhadap pengaruh jumlah iterasi, jumlah populasi, nilai parameter $v1$, $v2$, $r1$ dan $r2$, maka pada penerapan Algoritma Cuttlefish dalam pembuatan jadwal wisata menggunakan nilai parameter 50000 pada jumlah iterasi, 300 pada jumlah populasi, 1 dan -1 pada $v1$ dan $v2$, 0.5 dan -0.5 pada $r1$ dan $r2$.

Database mengenai tempat baik tempat wisata, tempat makanan maupun akomodasi tempat tinggal serta data mengenai durasi waktu perjalanan sangat berpengaruh dalam proses perencanaan perjalanan wisata ini, semakin akurat dan baik data yang berhasil dikumpulkan maka semakin baik pula hasil jadwal yang berhasil terbentuk mengingat seluruh data yang berhasil dikumpulkan berasal dari data yang terdapat pada Google Maps saja.

Berdasarkan hasil uji coba kuesioner yang diberikan pada responden, sebanyak 88% responden menyatakan jadwal wisata yang terbentuk sudah cukup baik serta 8% responden menyatakan bahwa jadwal wisata yang dibentuk sudah sangat baik dan 4% responden sisanya menyatakan bahwa jadwal wisata yang ada masih kurang baik.

DAFTAR PUSTAKA

- [1] D. Simon, *Evolutionary Optimization Algorithms, Biologically Inspired and Population-Based Approaches to Computer Intelligence*, J. Wiley, and Sons, Inc, 2013.
- [2] A. S. Eesa, A. M. Abdulazeez, Z. Orman, Cuttlefish Algorithm – A Novel Bio-Inspired Optimization Algorithm, *International Journal of Scientific and Engineering Research*, Vol. 4, Issue 9, September, 2013.
- [3] A. S. Eesa, A. M. Abdulazeez, Z. Orman, A New Tool for Global Optimization Problems- Cuttlefish Algorithm, *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering* Vol. 8, No. 9, 2014.

- [4] O. Filipova, *Learning Vue.js 2*, Packt Publishing Ltd, 2016, pp. 53-65.
- [5] <https://framework7.io/>.
- [6] Eduonix Learning Solutions, *Learn Node.js by Building 6 Projects: Build six practical and instructive Node.js projects*, Packt Publishing Ltd, 2018.
- [7] <http://www.thecephalopodpage.org/>.
- [8] R. T. Hanlon, and J. B. Messenger, *Cephalopod Behavior*, Cambridge: Cambridge University Press, 1996.
- [9] R. A. Cloney, and S. L. Brocco, Chromatophore organs, reflector cells, iridocytes and leucophores in cephalopods, *Am. Zool.* 1983.
- [10] https://developers.google.com/places/web-service/intro?hl=en_US.
- [11] Z. Zhan, *Selenium WebDriver Recipes in C#: Second Edition*, Apress, 2015.
- [12] <https://developers.google.com/maps/documentation/distance-matrix/intro#DistanceMatrixRequests>
- [13] <https://www.travelpayouts.com/>
- [14] G. C. Hillar, *Building RESTful Python Web Services*, Packt Publishing Ltd, 2016, pp.189-301.

Hendrawan Armanto lahir di Surabaya, Indonesia, pada tahun 1986. Menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 2008. Menyelesaikan studi masternya pada program studi Teknologi Informasi STTS. Minat penelitian adalah bidang Artificial Intelligent, Evolutionary Algorithm, dan Game Development.

Reynold Kevin lahir di Kota Surabaya, Jawa Timur, Indonesia pada tahun 1996. Saat ini sebagai mahasiswa tingkat akhir dengan jurusan S1 Teknik Informatika di STTS Surabaya. Risetnya berfokus pada Evolutionary Algorithm dan pembuatan aplikasi mobile hybrid.

C. Pickerling lahir di Surabaya, Indonesia, pada tahun 1986. Menyelesaikan studi S1 di program studi Teknik Informatika STTS pada tahun 2008. Menyelesaikan studi masternya pada program studi Teknologi Informasi STTS pada tahun 2016. Minat penelitian adalah pada bidang ilmu software engineering dan evolutionary algorithm.