

# *Penerapan Algoritma Divide and Conquer untuk Estimasi Home-Range Hewan*

Haifa Fadhila Ilma 13516076

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

haifadhila.irma@gmail.com

**Abstrak**—Divide and Conquer adalah salah satu strategi algoritma untuk penyelesaian berbagai masalah. Salah satu penerapan dari strategi ini adalah algoritma *quickhull* sebagai penyelesaian permasalahan *convex hull*. *Convex hull* itu sendiri merupakan sebuah poligon yang disusun oleh subset dari sebuah himpunan sedemikian sehingga tidak ada titik dari himpunan tersebut yang berada diluar poligon. Konsep ini dapat digunakan untuk mengestimasi *home-range* atau daerah jelajah dan tempat tinggal suatu spesies hewan tertentu dengan menggunakan sampel kumpulan titik yang pernah disinggahi oleh spesies hewan tersebut.

**Keywords**—*convex; quickhull; divide; conquer; hewan; home; range;*

## I. PENDAHULUAN

Salah satu sifat makhluk hidup yang alami adalah kemampuannya beradaptasi di berbagai kondisi. Hewan pun tidak luput dari sifat tersebut. Hewan seringkali mengubah lingkungan hidup serta wilayah tempat tinggalnya sebagai peningkatan kemampuan dan kesempatan untuk bertahan hidup. Cara yang paling baik untuk mengobservasi dan memahami sifat alami serta perilaku sosial hewan adalah dengan membedakan antara bagaimana perilaku tertentu muncul, dengan perilaku alamiah yang dimilikinya (dapat dianalisis berdasarkan sifat turun temurun yang sudah ada).

*Home range* hewan adalah daerah dimana suatu jenis hewan tertentu hidup dan menghabiskan sebagian besar waktu mereka didalamnya. Konsep *home range* ini dapat digunakan sebagai salah satu sarana untuk melakukan observasi terhadap hewan karena hal tersebut dapat merepresentasikan interaksi antara hewan dan lingkungannya. Seperti yang telah disebutkan diatas, perilaku sosial hewan juga dapat dilihat dengan menggunakan analisis sifat berdasarkan keberadaannya didalam *home range*. Misalnya, seekor hewan yang terpisah dari kawanannya atau berada di daerah diluar *home range* nya akan menunjukkan perilaku yang berbeda dari perilaku alamiah yang dimilikinya.

Analisis perilaku dan kebutuhan hewan untuk menggiatkan perlindungan spesies hewan di habitatnya, atau untuk sekedar bahan pembelajaran, dapat dilakukan salah satunya dengan mengetahui *home range* hewan tersebut. Salah satu metode penentuan *home range* adalah meneliti kumpulan titik yang

pernah disinggahi oleh hewan tertentu dan membuat poligon terkecil yang mungkin dibentuk dari kumpulan titik tersebut. Poligon yang dapat merepresentasikan daerah *home range* tentunya perlu mencakup seluruh titik yang ada. Oleh karena itu, diperlukan perhitungan dan metode khusus untuk membentuk poligon tersebut agar estimasi daerah yang dihasilkan lebih optimal dan lebih merepresentasikan *home range* yang sebenarnya dari spesies hewan tertentu.

Pada makalah ini, penulis menjelaskan teori-teori dasar algoritma *divide and conquer*, *convex hull*, algoritma *quickhull* sebagai penerapannya, serta teori dasar tentang *home-range* hewan yang akan digunakan untuk menganalisis dan membahas penerapan algoritma *divide and conquer* dalam menentukan *home range* itu sendiri.

## II. LANDASAN TEORI

### A. Algoritma Divide and Conquer

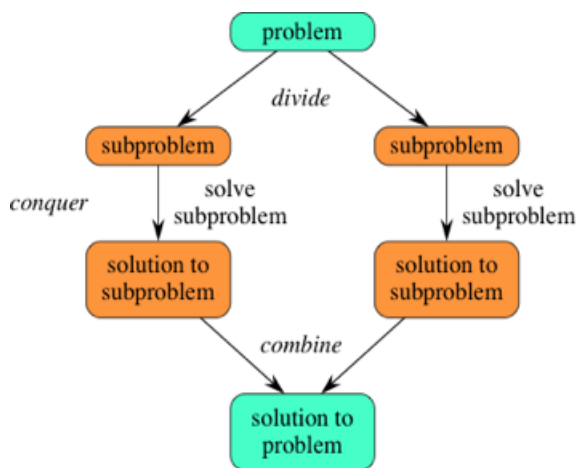
*Divide and Conquer*, atau biasa disingkat menjadi algoritma D&C pada ranah ilmu komputer merupakan salah satu strategi algoritma penyelesaian masalah yang menggunakan konsep rekursif dalam implementasinya. Nama dari strategi algoritma ini berasal dari kata-kata berbahasa inggris yaitu *divide* yang artinya ‘membagi’, dan *conquer* yang artinya ‘menaklukkan’.

Sesuai dengan namanya, strategi dari algoritma ini adalah membagi masalah-masalah yang ada menjadi beberapa bagian yang sama rata atau sejenis sampai bagian-bagian tersebut cukup sederhana untuk diselesaikan secara langsung. Kemudian, bagian-bagian masalah yang telah diselesaikan tersebut digabung kembali untuk membentuk sebuah solusi lengkap atas masalah awal yang diberikan. Karena penyelesaian masalah dilakukan secara rekursif, bagian-bagian permasalahan tentunya harus lebih kecil dari masalah awal. Pembagian harus ditentukan dan dilakukan sedemikian rupa sehingga dapat diselesaikan dengan lebih sederhana dan memiliki basis penyelesaian.

Algoritma ini secara umum memiliki tiga tahapan, yaitu *divide*, *conquer*, dan *combine*. Berikut penjelasan lebih rinci dari ketiga tahapan tersebut:

- *Divide*: Tahap ini membagi masalah utama menjadi bagian-bagian masalah sejenis yang lebih kecil hingga cukup sederhana untuk diselesaikan.
- *Conquer*: Tahap ini akan menyelesaikan tiap sub-masalah, yang telah dibagi di tahap divide, secara rekursif dengan bagian terkecil dijadikan basis dan diselesaikan dengan cara yang telah ditentukan dan menghasilkan solusi untuk tiap-tiap bagian yang diselesaikan.
- *Combine*: Tahap ini akan menggabungkan solusi dari bagian masalah untuk menjadi solusi utama dari masalah awal (masalah utama yang belum terpisah-pisah).

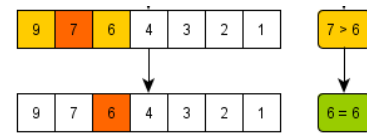
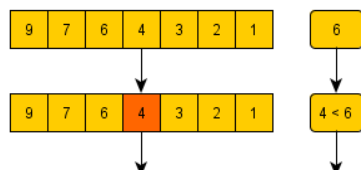
Berikut merupakan representasi dari tahapan algoritma *Divide and Conquer*:



**Gambar 2.1:** Tahapan Algoritma *Divide and Conquer*  
Sumber: <https://www.khanacademy.org>

Algoritma ini dapat diimplementasikan di berbagai jenis cara penyelesaian masalah, seperti contoh-contoh berikut ini:

- *Binary Search*: Ini merupakan salah satu algoritma pencarian elemen numerik pada array terurut membesar. Di setiap tahapannya, algoritma akan membandingkan nilai masukan (angka yang dicari) dengan angka yang berada di tengah-tengah array. Jika angka tersebut sesuai, maka kembalikan indeks tengah array. Jika tidak, lakukan perbandingan, jika angka masukan lebih besar dari nilai tengah, maka pencarian dilakukan secara rekursif di sisi kanan elemen tengah array. Jika angka masukan lebih kecil, maka akan dilakukan di sebelah kiri. Berikut contoh *binary search*:



**Gambar 2.2:** Contoh *Binary Search*  
Sumber: <http://www.programming-algorithms.net>

- *Quick Sort*: Ini merupakan salah satu algoritma pengurutan elemen numerik pada array. Pada awalnya, algoritma akan memilih satu elemen pivot, kemudian menyusun agar elemen yang lebih kecil dari pivot tersebut berada di sebelah kiri pivot, dan elemen yang lebih besar berada di sebelah kanan. Kemudian untuk sisi kiri dan kanan pivot, dipisahkan dan dilakukan kembali cara yang sama seperti cara awal hingga semua elemen terurut.

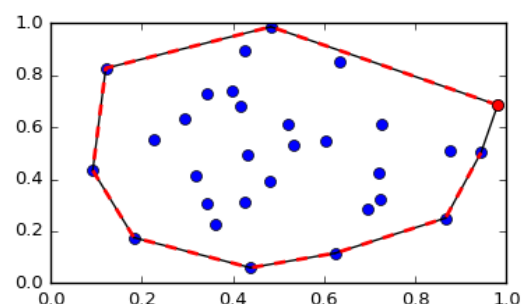
Poin-poin diatas adalah dua contoh dari berbagai penerapan algoritma *divide and conquer*. Untuk penyelesaian masalah yang akan dibahas pada makalah ini, kita akan menggunakan konsep algoritma *quick sort* yang dikembangkan untuk pemrosesan kumpulan titik membentuk *convex hull*, algoritma ini dinamakan algoritma *quickhull* yang akan dibahas pada bagian selanjutnya.

## B. Convex Hull

*Convex Hull* merupakan suatu poligon konveks yang dibentuk dari kumpulan titik yang merupakan bagian (*subset*) dari sebuah himpunan titik. Poligon yang konveks adalah poligon yang memiliki garis batas luar poligon (penghubung antar titik-titik sudut dari bidang datar) yang tidak dipotong garis lain. Artinya, seluruh sudut dalam dari poligon harus kurang dari  $180^\circ$ .

Maka dari itu, *convex hull* adalah satu poligon yang dibentuk dari sub-himpunan titik dimana dengan garis batas yang terbentuk, tidak ada satupun titik dari himpunan awal yang berada diluar poligon tersebut (semua titik berada didalam poligon atau tepat berada di garis batas). Selain itu, poligon ini juga harus berbentuk konveks.

Berikut merupakan contoh *convex hull*:



**Gambar 2.3:** Contoh *Convex Hull*  
Sumber: <https://docs.scipy.org>

Pembentukan *convex hull* dari sebuah himpunan titik dapat dilakukan dengan memanfaatkan persamaan garis pada bidang dan membagi kumpulan titik tersebut berdasarkan garis yang dibuat untuk kemudian diproses lebih lanjut dengan algoritma *quickhull*.

Konsep *convex hull* ini bukan hanya terdapat pada bidang datar saja (dua dimensi), melainkan juga pada bidang tiga dimensi atau lebih. Jadi, kumpulan titik yang ada pada bidang tiga dimensi akan dibentuk menjadi sebuah benda tiga dimensi yang konveks dan sesuai dengan sifat-sifat *convex hull* yang telah disebutkan diatas.

Contoh penggunaan *convex-hull* dalam kehidupan sehari-hari dan dalam sarana pengembangan teknologi adalah untuk *pattern recognition*, pemrosesan citra, berbagai aplikasi statistika, sistem informasi geografi, membantu dalam pembuatan diagram fase di ilmu fisika dan kimia, dan sebagai estimasi sederhana penentuan *home range* hewan yang akan dibahas pada makalah ini.

### C. Algoritma Quickhull

Algoritma *quickhull* adalah salah satu implementasi strategi *divide and conquer* yang didesain untuk menyelesaikan permasalahan *convex hull*. Algoritma ini akan menghasilkan sub-himpunan yang berisi kumpulan titik yang merupakan bagian dari himpunan titik yang lebih besar. Sub-himpunan titik ini akan membentuk poligon konveks sesuai dengan syarat-syarat sebuah *convex hull*.

Algoritma ini sejenis dengan algoritma *quicksort*, dimana namanya pun diturunkan dari nama algoritma ini, karena memang strategi dan metode penyelesaiannya sejenis dengan metode pengurutan di algoritma *quicksort*. Bedanya, algoritma *quickhull* memanfaatkan garis tengah yang dibentuk oleh titik dengan jarak terjauh di salah satu sumbu orientasi yang ditentukan (misal: sumbu x atau sumbu y), bukan hanya elemen tengah saja. Kemudian, himpunan titik lainnya dibagi ke dua sisi garis dengan memanfaatkan konsep persamaan garis pada bidang, lalu proses awal kembali dilakukan pada kedua bagian titik tersebut.

Berikut adalah tahapan pada algoritma *quickhull* pada bidang dua dimensi dengan acuan titik awal terjauh berdasarkan sumbu x:

1. Cari titik pada himpunan titik dengan nilai koordinat x paling minimum ( $x_{min}$ ), dan juga titik dengan nilai koordinat x paling maksimum ( $x_{max}$ ).
2. Bentuk garis yang menghubungkan kedua titik ini (misal garis G). Garis ini akan membagi himpunan titik menjadi dua bagian. Kemudian tiap bagian ini masing-masing akan diproses dengan cara yang sama oleh tahap berikutnya.
3. Untuk satu bagian, cari suatu titik pada bagian himpunan ini yang memiliki jarak paling maksimum dengan garis G (misal titik T<sub>max</sub>). Kemudian, titik ini akan membentuk segitiga dengan titik  $x_{min}$  dan  $x_{max}$ . Titik didalam segitiga tidak akan diproses lagi karena ia akan berada didalam poligon hasil dan tidak mungkin membentuk garis *convex hull*.

4. Kemudian, fokus pada garis yang menghubungkan titik T<sub>max</sub> dan  $x_{min}$ , serta garis yang menghubungkan titik T<sub>max</sub> dan  $x_{max}$ . Kedua garis ini merupakan dua garis baru (seperti garis yang dihasilkan di tahap pertama) dan kumpulan titik yang berada di luar garis-garis tersebut adalah sub-himpunan titik baru yang akan diproses seperti tahap nomor 3.
5. Ulangi langkah nomor 3 dan 4 pada setiap garis dan sub-himpunan titik yang terbentuk secara terpisah sampai tidak ada titik yang berada diluar garis yang terbentuk, kemudian masukkan kumpulan titik akhir ini (T<sub>max</sub> pada tiap sub-himpunan akhir) kedalam himpunan titik *convex-hull*.

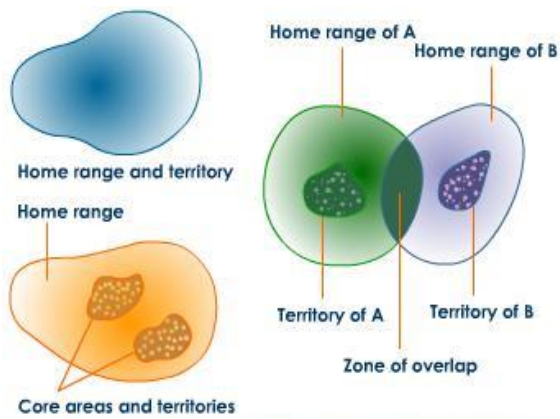
### D. Home Range

*Home range*, atau bisa disebut juga ‘daerah jelajah’ yang dimiliki oleh hewan merupakan suatu hal yang mendasar ada pada bidang ekologi dan perilaku hewan. Konsep dasar dari daerah jelajah atau *home range* ini adalah bagian dari peta kognitif hewan beserta lingkungannya yang terus diperbarui oleh spesies hewan itu sendiri. Pengertian lain dari *home range* adalah suatu area dimana spesies hewan tertentu menetap dan berpindah-pindah atau melakukan mobilitasi secara periodik dalam melaksanakan kelangsungan hidup setiap harinya.

*Home range* adalah wilayah dimana hewan menghabiskan sebagian besar waktu dan kegiatannya disana, maka dari itu, hewan dengan jenis tertentu akan lebih mungkin ditemukan dengan mengetahui *home range*-nya. Di daerah jelajah ini, hewan akan melakukan kegiatan sehari-harinya, seperti mencari makanan dan minuman, membentuk kawanan, sebagai tempat berlindung dan istirahat, dan tempat ia mencari pasangan serta berkembangbiak. Salah satu contoh representasi yang sesuai untuk mendeskripsikan daerah jelajah ini adalah kita sebagai manusia. Kita memiliki tempat-tempat dan daerah mobilitasi kegiatan sehari-hari, misalnya ke sekolah, supermarket, taman kota, dan pastinya ke rumah. Tempat-tempat ini penting untuk memenuhi kebutuhan dan kelangsungan hidup. Daerah sekitar tempat-tempat tersebut dapat dikatakan sebagai *home-range* kita.

Hewan berpindah-pindah tempat untuk berbagai alasan seperti yang telah dijelaskan diatas. Beberapa jenis hewan berpindah ke tempat yang jauh berdasarkan pergantian musim, yang dinamakan migrasi. Memiliki *home range* merupakan hal yang penting bagi hewan untuk bertahan hidup karena mereka akan familiar dengan daerah jelajah mereka sendiri. Mereka akan lebih mudah mengetahui dimana sumber makanan dan minuman mereka, dimana ancaman luar berada, mengetahui dimanakah tempat istirahat dan berlindung yang terbaik bagi mereka, dan akan sadar jika ada perubahan atau serangan di daerah jelajah mereka. Hewan merasa familiar dengan *home range* sebagaimana kita merasa familiar dengan lingkungan tempat tinggal kita. Bahkan, beberapa hewan sangat terikat dengan *home range* mereka dan memungkinkan untuk kembali lagi ke daerah awal ketika terpisah dengan jarak yang cukup jauh.

Berikut adalah contoh *home range* beserta perbedaannya dengan teritori hewan:



**Gambar 2.3:** Home Range dan Teritori Hewan  
Sumber: <https://www.tutorvista.com>

Dapat dilihat pada gambar diatas, *home range* antar spesies hewan dapat tumpang tindih karena memang secara umum beberapa hewan hidup bersamaan dengan sumber pangan dan kebutuhan lingkungan yang hampir sama pula. Pada gambar diatas, *home range* spesies A bertumpang tindih pada area tertentu dengan *home range* spesies B. Selain itu, didalam *home range* hewan terdapat area yang dipersempit lagi yang dinamakan teritori dan area inti dari hewan tersebut.

Teritori dan *home range* adalah kedua hal yang berbeda. Area teritori berada didalam *home range* dan umumnya lebih kecil dan tidak tumpang tindih dengan *home range* spesies lain karena area ini cenderung dikuasai dan dilindungi dari pihak luar untuk menjaga keamanan dan menghindari ancaman dari luar. Biasanya teritori ini digunakan sebagai tempat hewan istirahat rutin, membuat sarang (sarang adalah area yang lebih sempit lagi dari teritori), atau digunakan saat musim perkawinan hingga membesarkan keturunannya agar tetap aman dari gangguan.

*Home range* spesies hewan dapat berbeda-beda tergantung jenis hewan. Berikut faktor-faktor yang mempengaruhinya:

- Distribusi makanan dan perlindungan: Jika distribusi makanan dan perlindungan terlalu menyebar dan tidak teratur, maka *home range* juga kerap berubah-ubah secara periodik, begitupun sebaliknya.
- Keberadaan musuh atau spesies lain: Kedatangan musuh atau spesies lain dapat mempengaruhi perubahan *home range* spesies yang terdapat di wilayah tersebut. Jika tidak bisa dipertahankan, *home range* mungkin saja mengecil atau berpindah.
- Jenis dan ukuran tubuh hewan: Pada umumnya, hewan yang memiliki ukuran tubuh besar memiliki *home range* yang besar pula karena perpindahannya lebih signifikan. Selain itu, jika ukurannya sama, hewan karnivora cenderung memiliki *home range* yang lebih besar daripada herbivora dan omnivore. Hewan jantan dan hewan yang lebih tua akan memiliki *home range* yang

lebih besar daripada hewan betina dan hewan muda.

Pada dasarnya, *home range* di alam bebas cenderung dinamis karena dipengaruhi dari berbagai factor. Tetapi, *home range* spesies tertentu dapat diestimasi dari keberadaan hewan-hewan tersebut di suatu daerah.

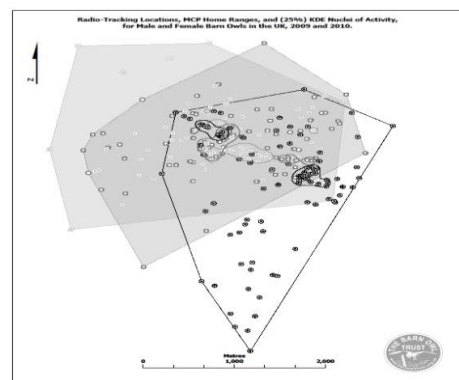
### III. PENERAPAN KONSEP CONVEX HULL PADA PENENTUAN HOME-RANGE

Penentuan *home range* atau daerah jelajah dari suatu spesies hewan bergantung pada berbagai faktor lingkungan dan kebutuhan hewan itu sendiri. Daerah persebaran hewan ini dapat tergantung dari distribusi spasial wilayah, kualitas sumber daya alam dan hubungannya dengan kebutuhan pangan dan tempat tinggal hewan, habitat natural hewan dan perilakunya terhadap perubahan lingkungan, dan aspek-aspek lain mengenai keadaan lingkungan dan sifat natural hewan agar dapat dilihat bagaimana hewan melihat lingkungannya sendiri.

Seperti yang telah dijelaskan sebelumnya, analisis perilaku dan kebutuhan hewan adalah hal yang perlu dilakukan, bukan hanya untuk studi penelitian, melainkan juga bentuk pemeliharaan lingkungan dan makhluk hidup. Analisis perilaku dan kebutuhan hewan ini salah satunya dapat dilakukan dengan mengetahui daerah jelajah atau *home range*.

Salah satu pendekatan sederhana dalam menentukan *home range* suatu jenis hewan adalah mengobservasi pergerakannya secara periodik. Hal ini dapat dijadikan sumber data titik-titik lokasi suatu jenis hewan di wilayah tertentu. Kita dapat mengamati, mengambil sampel dan mencatat titik posisi suatu jenis hewan yang ia temukan dalam kurun waktu tertentu dan dalam jumlah tertentu. Kemudian, kumpulan titik hasil pengamatan tersebut dapat dipetakan dan dapat terlihat persebaran hewan.

Melihat persebaran hewan saja belum cukup untuk menentukan *home range*. Perlu ditentukan dan digambarkan *range* wilayah tertentu sebagai daerah pasti *home range* hewan. Untuk menentukan hal tersebut, kumpulan titik ini akan diproses membentuk sebuah bidang datar dengan sisi lebih dari 2 (poligon) yang akan mencakup seluruh titik hasil pengamatan didalamnya. Poligon ini perlu berbentuk konveks agar memaksimalkan daerah yang tercakup. Poligon ini bernama *convex hull*.



**Gambar 3.3:** Contoh Home Range Convex Hull  
Sumber: <https://www.barnowltrust.org.uk>

Gambar diatas merupakan contoh convex hull dalam penentuan daerah *home range*. Pada gambar tersebut, ditunjukkan hasil penelitian *home range* dari burung hantu *barn jantan* dan betina di Inggris pada tahun 2009 dan 2010. Dapat dilihat bahwa *home range* tersebut merupakan sebuah convex hull yang dibentuk dari sekumpulan titik. Pada gambar tersebut juga ditunjukkan lingkaran-lingkaran kecil yang merupakan teritori spesies burung hantu tersebut, dan dua *home range* spesies burung hantu lainnya yang tumpang tindih.

#### IV. IMPLEMENTASI ALGORITMA DIVIDE AND CONQUER UNTUK MENENTUKAN HOME RANGE HEWAN

Seperti yang telah dijelaskan pada bagian sebelumnya, penentuan *home range* suatu jenis hewan dapat dilakukan dengan pengamatan titik-titik jenis hewan tersebut di wilayah tertentu dan melakukan pemrosesan kumpulan titik tersebut menjadi sebuah *convex hull*. *Convex hull* inilah yang menjadi hasil *home range* hewan tersebut. Titik-titik yang dikumpulkan tentunya harus cukup untuk merepresentasikan persebaran hewan tersebut, agar *home range* yang terbentuk juga lebih sesuai dengan daerah jelajah hewan yang sebenarnya.

Pembentukan *convex hull* dari himpunan titik ini dapat dilakukan dengan menggunakan salah satu implementasi algoritma *divide and conquer* yang bernama algoritma *quickhull*. Algoritma ini memanfaatkan garis tengah yang dibentuk oleh titik dengan jarak terjauh di salah satu sumbu orientasi yang ditentukan (misal: sumbu x atau sumbu y). Kemudian, himpunan titik lainnya dibagi ke dua sisi garis dengan memanfaatkan konsep persamaan garis pada bidang. Berikut merupakan *pseudocode* untuk lebih menjelaskan algoritma *quickhull* dalam pembentukan *convex hull*:

```
{ PROGRAM QUICK HULL}

arrPoint <- Array of Input Points
convexHull <- Array of Solved Points

Procedure getConvexHull (arrPoint):
  1. Sort ( semua titik pada arrPoint )
     Berdasarkan absis. Jika absis sama,
     berdasarkan ordinat

  2. P1 <- arrPoint[Idx pertama]
     P1 adalah titik minimum

  3. Pn <- arrPoint[Idx terakhir]
     Pn adalah titik maksimum

  4. Masukkan P1 dan Pn kedalam array convexHull

  5. Panggil fungsi
     quickHull(P1,Pn,arrTitikdiKananGaris) Proses
     titik sebelah kiri garis P1-Pn

  6. Panggil fungsi
     quickHull(P1,Pn,arrTitikdiKananGaris) Proses
     titik sebelah kanan garis P1-Pn

  7. return convexHull
```

```
Procedure quickHull (P1,Pn,arrTitik):
  1. if (Masih ada titik di sisi kanan/kiri garis
     P1-Pn):
     - titikMax <- Titik dengan jarak terjauh
       dari garis P1-Pn
     - Masukkan titikMax kedalam array
       convexHull

  { Memproses secara rekursif titik yang berada
  disebelah kiri atau kanan garis yang dibentuk P1-
  titikMax dan titikMax-Pn }

  2. Panggil fungsi
     quickHull (P1,titikMax,arrTitikBaru)

  3. Panggil fungsi
     quickHull (titikMax,Pn,arrTitikBaru)

Main Program:
  1. Inisialisasi array arrPoint dan array
     convexHull
  2. Panggil prosedur getConvexHull()
  3. Gambar seluruh titik arrPoint dan garis yang
     menghubungkan titik-titik di array
     convexHull
```

Pada *pseudocode* ini, pertama-tama kita mendeklarasikan *arrPoint* sebagai array yang berisi himpunan titik hasil pengamatan, dan *convexHull* sebagai array yang berisi himpunan bagian titik yang nantinya membentuk *convex hull*.

Kemudian, kita melakukan pemanggilan prosedur *getConvexHull()*. Pada prosedur ini, pertama kita akan mengurutkan (*sort*) titik di *arrPoint* berdasarkan absisnya. Jika absis sama, pengurutan dilakukan berdasarkan ordinatnya. Kemudian, kita akan memasukkan titik paling minimum (indeks pertama pada *arrPoint*) ke P1 dan memasukkan titik paling maksimum (indeks terakhir) ke Pn. P1 dan Pn sudah pasti titik pembentuk *convex hull*, maka kedua titik tersebut dimasukkan ke dalam array *convexHull*. Selanjutnya, pisahkan kedua kumpulan titik di kanan garis P1-Pn dan di kiri garis menjadi dua bagian, dan panggil fungsi *quickHull()* dengan masukan P1, Pn, dan array titik baru untuk masing-masing bagian. Pemrosesan akan dilanjutkan secara rekursif di fungsi tersebut dan terakhir, kembalikan array *convexHull* sebagai hasil akhir.

Pada fungsi *quickHull()*, dilakukan pengecekan apakah masih ada titik di sisi kanan atau kiri garis masukan (P1-Pn), jika ya, maka masukkan titik dengan jarak terjauh dari garis P1-Pn ke variabel *titikMax*, dan masukkan *titikMax* tersebut ke array *convexHull*. Selanjutnya, pisahkan lagi kedua kumpulan titik tersebut berdasarkan garis yang dibentuk oleh P1-*titikMax* dan Pn-*titikMax*, lalu panggil kembali fungsi *quickHull()* secara rekursif dengan masukan kedua titik dan array yang baru.

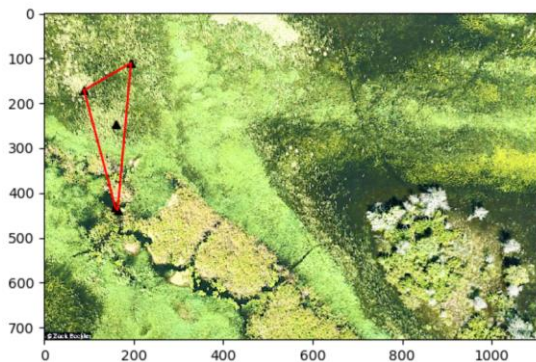
Lalu, di program utama, akan dilakukan pengisian *arrPoint* dengan data titik pengamatan, diikuti pemanggilan fungsi *getConvexHull()* serta menggambar *convex hull* dari kumpulan titik yang dihasilkan.



## V. HASIL UJI IMPLEMENTASI ALGORITMA *QUICKHULL*

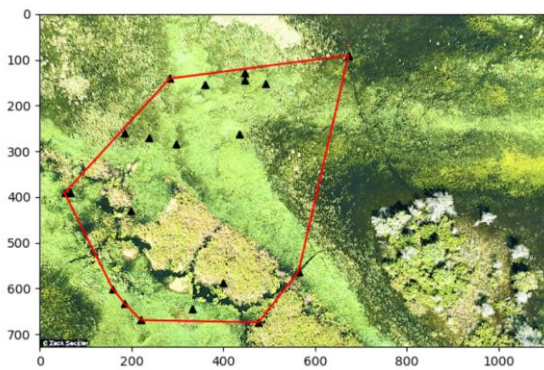
Berikut merupakan beberapa hasil pengujian program yang merupakan implementasi dari algoritma *quickhull* untuk membentuk *convex hull* dari berbagai variasi jumlah titik.

### 1. Hasil uji dengan masukan 5 titik



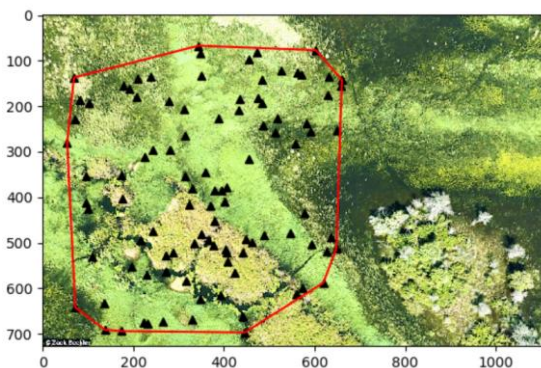
**Gambar 5.1:** Hasil uji program QuickHull dengan masukan 5 titik  
Sumber: Hasil kompilasi program milik penulis

### 2. Hasil uji dengan masukan 20 titik



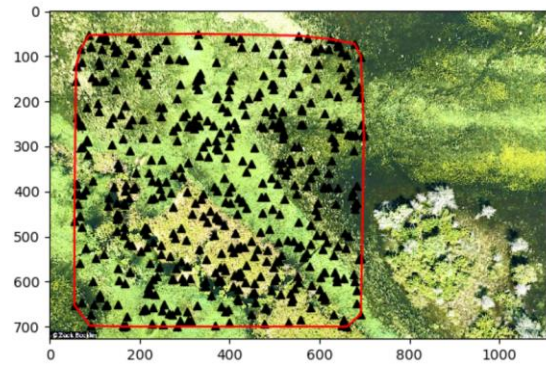
**Gambar 5.2:** Hasil uji program QuickHull dengan masukan 20 titik  
Sumber: Hasil kompilasi program milik penulis

### 3. Hasil uji dengan masukan 100 titik



**Gambar 5.3:** Hasil uji program QuickHull dengan masukan 100 titik  
Sumber: Hasil kompilasi program milik penulis

### 4. Hasil uji dengan masukan 500 titik



**Gambar 5.4:** Hasil uji program QuickHull dengan masukan 500 titik  
Sumber: Hasil kompilasi program milik penulis

Pada keempat hasil uji ini, dapat ditunjukkan hasil yang berbeda pada tiap jumlah masukan yang berbeda. *Convex hull* yang dihasilkan akan lebih merepresentasikan distribusi titik jika masukan titik lebih banyak.

## VI. KESIMPULAN

Penentuan dan estimasi *home range* dari suatu spesies hewan dapat dilakukan dengan pembentukan *convex hull* dari sekumpulan titik hasil pengamatan dimana hewan-hewan tersebut ditemukan. Pembentukan *convex hull* dapat dilakukan dengan melakukan pemrosesan kumpulan titik menggunakan strategi algoritma *divide and conquer*, lebih tepatnya algoritma *quickhull*. Semakin banyak jumlah masukan titik hasil pengamatan, *home range* dari *convex hull* yang dihasilkan juga akan lebih merepresentasikan persebaran hewan yang ada.

## REFERENSI

- [1] Pebbie, GAIBlog: "Convex Hull 2D" dalam <https://pebbie.wordpress.com/2009/09/29/convex-hull-2d/> [diakses 12 Mei 2018 pukul 16:00].
- [2] Cillperqueen: "Home Range and Territory" dalam <https://cillperqueen.wordpress.com/2010/06/25/home-range-and-territory/> [diakses 12 Mei 2018 pukul 16:05].
- [3] Roger A. Powell, Michael S. Mitchell: "What is a Home Range?" dalam Journal of Mammalogy, Volume 93, Issue 4, Hlm. 948-958.
- [4] Encyclopedia Britannica: "Animal Social Behaviour" dalam <https://www.britannica.com/topic/animal-social-behaviour/The-how-and-why-of-social-behaviour> [diakses 12 Mei 2018 pukul 15:30].
- [5] KhanAcademy: "Divide and Conquer Algorithms" dalam <https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms> [diakses 12 Mei 2018 pukul 18:25].
- [6] GeeksforGeeks: "Divide and Conquer / Set 1 (Introduction)" dalam <https://www.geeksforgeeks.org/divide-and-conquer-introduction/> [diakses 12 Mei 2018 pukul 20:30].
- [7] GeeksforGeeks: "Quickhull Algorithm for Convex Hull" dalam <https://www.geeksforgeeks.org/quickhull-algorithm-convex-hull/> [diakses 13 Mei 2018 pukul 11:29].
- [8] Wildlife Journal Junior: "Home Range: Roam to Roam, Moving, No Place Like Home" dalam <http://www.nhptv.org/wild/homerange.asp> [diakses 13 Mei 2018 pukul 14:05].

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 13 Mei 2018

A handwritten signature in black ink, appearing to be 'Haifa' followed by a stylized flourish.

Haifa Fadhila Ilma 13516076