|  |  |
| --- | --- |
|  |  |
|  | **MODUL 4 PERKULIAHAN** |
|  |  |
|  | **Proyek Data Science** |
|  |  |
|  | **Studi kasus penerapan algoritma data science** |
|  |  |

# Pembahasan

**Pertemuan 4**

**Studi kasus penerapan algoritma data science**

Sebelum kita membahas contoh penerapan Data Science pada dunia perindustrian, kita perlu tau dulu definisi apa itu Data Science. Data Science merupakan disiplin ilmu dalam memanfaatkan data berskala besar, baik terstruktur maupun tidak terstruktur, guna untuk membuat keputusan yang tepat.

Kini Data Science menjadi satu dari sekian istilah paling populer dalam dunia perindustrian. Data Science pun menjadi teknologi yang berperan besar di dalamnya. Secara tidak sadar, hampir semua perusahaan menerapkan Data Science. Apa saja sih contohnya? Yuk, kita bahas bersama!

1. E-Commerce

Saat ini orang-orang lebih banyak berbelanja secara digital atau biasa kita kenal dengan marketplace. Hampir semua orang menggunakan marketplace, penjual online secara otomatis akan menyesuaikan etalasenya berdasarkan profil data pembeli. Dengan mengubah tata letak halaman dan menyesuaikan produk jualannya secara otomatis dan real-time. Beberapa online shop juga menyesuaikan harga berdasarkan profil penghasilan konsumen, atau disebut dengan harga yang dipersonalisasi. Teknologi ini merupakan salah satu aplikasi penerapan ilmu Data Science.

2. Fraud Detection

Mendeteksi penipuan adalah bagian terpenting dari segala industri keuangan. Data Science dan AI adalah kedua ilmu yang sering digunakan disini. Bahkan kerusakan dan gangguan kecil akan dapat menyebabkan kerugian finansial. Analisis prediksi real-time membantu dalam peningkatan deteksi penipuan dan juga keamanan cyber. Dengan bantuan Data Science, perusahaan dapat menyediakan layanan keuangan yang lebih efektif. Teknologi ini membantu untuk mengidentifikasi potensi transaksi penipuan yang dilakukan di setiap aktivitas. Dan ini juga akan membantu untuk memblokir sesi atau akun jika terdeteksi ada aktivitas keuangan yang tidak biasa.

3. Dynamic Pricing

Penetapan dynamic pricing sangat berguna dalam industri travel. Penetapan harga dinamis ini digunakan oleh perusahaan dengan menggunakan data untuk mensegmentasikan pelanggan atau konsumen secara akurat. Setiap kelompok pelanggan kemudian dapat ditawari produk dengan harga berbeda. Penawaran ini didasari pada informasi yang dihasilkan oleh Data Science dan berbagai faktor lainnya.

Salah satu contoh perusahaan yang menerapkannya adalah Airbnb. Airbnb menggunakan ilmu Data Science dan algoritma dynamic pricing yang berfokus pada harga. Algoritma ini memperhitungkan berbagai macam kategori. Seperti lead time, review properti dan fasilitas yang disediakan. Algoritma ini juga dapat digunakan oleh pemilik properti untuk menentukan biaya per malam secara otomatis.

Contoh 1:

*Suprayogi, Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry* **121**

**Penerapan Algoritma Genetika *Traveling Salesman Problem with Time* *Window:* Studi Kasus Rute Antar Jemput *Laundry***

**Dwi Aries Suprayogi1, Wayan F. Mahmudy2**

Program Studi Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya

Jl. Veteran, Malang 65145

E-mail: 1[dwiariess@yahoo.co.id,](mailto:dwiariess@yahoo.co.id) 2[wayan\_firdaus@yahoo.com](mailto:wayan_firdaus@yahoo.com)

Masuk: 20 Juni 2014; Direvisi: 18 Juli 2014; Diterima: 21 Juli 2014

***Abstract.*** *Routing optimization is a widely discussed problem in the area of computer**science research. An example of the problem is laundry pickup service. In this case, each customer has reserved time to receive their order. Unlike the conventional traveling salesman problem (TSP) which goal is to choose the shortest path, in this case the precise arrival time each customer must be considered. The best Solution to the problem is achieved by combining solutions (chromosomes) to produce new solutions by using genetic operators (selection, crossover and mutation). A set of experiments is performed to find the best combination of parameters such as crossover probability, mutation probability, population size, and number of generation. From the test results the best combination crossover probability is 0,4 and mutation probability is 0,6. The optimum number of generation is 2000. The parameter can be used to obtain solution for serve all customer with different time window.*

***Keywords****: Traveling salesman problem, time windows, genetic algorithm, optimum route,**laundry pickup service.*

***Abstrak****. Optimasi pemilihan rute merupakan masalah yang banyak dibahas pada**penelitian ilmu komputer. Antar jemput laundry dengan pelanggan yang memiliki waktu khusus untuk menerima barang adalah salah satu contoh kasus pemilihan rute. Penghitungan rute tercepat memegang peranan penting karena harus tepat waktu dan semua pelanggan dapat dilayani. Berbeda dengan traveling salesman problem (TSP) konvensional yang bertujuan untuk meminimalkan jarak, kasus ini juga harus dipertimbangkan waktu ketersediaan setiap pelanggan. Pencarian solusi untuk permasalahannya adalah dengan mengkombinasikan solusi-solusi (kromosom) untuk menghasilkan solusi baru dengan menggunakan operator genetika (seleksi, crossover dan mutasi). Untuk mencari solusi terbaik digunakan beberapa kombinasi probabilitas crossover dan mutasi serta ukuran populasi dan ukuran generasi. Dari hasil pengujian kombinasi probabilitas crossover yang terbaik adalah 0,4 dan mutasi adalah 0,6 sedangkan untuk ukuran generasi optimal adalah 2000. Dari nilai-nilai parameter ini didapatkan solusi yang memungkinkan untuk melayani semua pelanggan dengan time window masing - masing.*

***Kata Kunci****: Traveling salesman problem, time windows, algoritma genetika, rute terbaik,**antar jemput landry.*

**1. Pendahuluan**

Usaha *laundry* adalah usaha yang bergerak dibidang jasa cuci dan setrika. Keberadaan jasa cuci mencuci dan setrika sudah menjadi bagian dari kebutuhan hidup manusia. Berkembangnya bisnis *laundry* kiloan menjadikan persaingan di sektor ini menjadi semakin ketat. Untuk menjaga agar usaha ini tidak sepi oleh pelanggan setiap penyedia jasa *laundry* memiliki ciri khas dan cara promosi masing-masing, seperti menyediakan jasa antar jemput cucian (Anonimous, 2013). Pelanggan *laundry* ada yang mempunyai waktu tertentu untuk mengambil dan menerima cucian mereka. Melihat dari keadaan ini maka parameter waktu yang akan dihabiskan dijalan dapat diperkirakan rute mana yang akan diambil oleh sopir antar jemput cucian agar bisa datang di tempat sesuai dengan waktu yang di inginkan pelanggan. Untuk

menyelesaikan masalah ini maka dibuatlah suatu sistem yang memperhitungkan jarak antar pelanggan dengan melalui rute tercepat manggunakan konsep *Vahicle Routing Problem with* *Time Windows* (VRPTW) yang merupakan sebutan bagi VRP dengan kendala tambahan berupaadanya *time windows* pada masing-masing pelanggan (Gambardella, 1999). Waktu ketersediaan pada masing-masing pelanggan dapat berbeda satu sama lain dan dinyatakan dalam selang waktu berupa batas waktu awal sampai akhir pelayanan pada pelanggan tersebut.

Salah satu contoh pencarian rute tercepat yaitu pemilihan rute yang dipilih sopir pengirim barang untuk sampai pada tujuan dengan tepat waktu. Setiap daerah tujuan pengiriman tersebut harus dikunjungi satu kali, kemudian kembali lagi ke tempat awal. Permasalahan tersebut dikenal sebagai *Travelling Salesman Problem* (TSP). salah satu bentuk pengembangan TSP yang lebih rumit yang melibatkan dua variabel atau lebih adalah TSP-TW yaitu pencarian rute optimal yang mempertimbangkan waktu total waktu perjalanan, dan waktu ketersediaan pelanggan (Gambardella, 1999).

Beberapa penelitian sebelumnya untuk menyelesaikan VRPTW telah diajukan seperti “Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya” (Purwanto, 2005) dengan menggunakan *graph,* bisa dilakukan pencarian rute terpendek dikota Surabaya. “Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner” (Widodo, 2010) membahas tentang jarak terpendek dari tujuan wisatawan yang ingin melakukan wisata kuliner dengan menggunakan algoritma genetika. “*Vehicie Routing Optimization Probiem with Time-windows* *and its Solution by Genetic Aigorithm*” oleh(Chen, 2013) meneliti tentang suatu optimasimasalah VRP dengan *time windows* menggunakan algoritma genetika. Pada penelitian ini digunakan algoritma genetika dengan harapan akan diperoleh optimasi rute perjalanan yaitu kondisi dimana terjadi kombinasi terbaik untuk jalur yang akan dilalui dan waktu perjalanan yang cepat, serta semua pelanggan dapat terlayani.

**2. Tinjauan Pustaka**

**2.1. *Vahicle Routing Problem with Time Windows***

*Vahicle Routing Problem* adalah suatu masalah pencaian jalur yang akan dilalui dengantujuan mencari rute yang paling cepat atau pendek. *Vahicle Routing Problem with Time* *Windows* (VRPTW) yang merupakan sebutan bagi VRP dengan kendala tambahan berupaadanya *time windows* pada masing-masing pelanggan. Waktu ketersediaan pada masing-masing pelanggan dapat berbeda dan dinyatakan dalam selang waktu berupa batas waktu awal sampai akhir pelayanan pada pelanggan tersebut (Gambardella, 1999).

**2.2. Algoritma Genetika**

Algoritma generika pertama kali dikembangkan pada tahun 1975 oleh Jhon Hollan dari Unversitas Michigan(Nugraha, 2008). Algoritma genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi yang dikemukakan oleh Charles Darwin. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. “Hanya individu-individu yang kuat yang mampu bertahan”. Algoritma genetika mungkin tidak selalu mencapai hasil yang terbaik, tetapi seringkali memecahkan masalah dengan cukup baik. Algoritma genetika merepresentasikan suatu solusi permasalahan sebagai kromosom. Terdapat beberapa aspek penting dalam algoritma genetika antara lain defenisi fungsi *fitness*, defenisi dan implementasi representasi genetika, defenisi dan implementasi operasi genetika. Ketiga aspek diatas sangat mendukung kinerja algoritma genetika (Nugraha, 2008). Jumlah populasi solusi yang besar adalah keunggulan algoritma genetika.

**2.2.1. Nilai *Fitness***

*Fitness* adalah nilai yang dimiliki oleh masing-masing individu untuk menentukantingkat kesesuaian individu tersebut dengan criteria atau tujuan (obyektif) permasalahan yang ingin dicapai (Mahmudy,2014). Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Fungsi tujuan untuk sistem optimasi rute antar jemput

*laundry* menggunakan Algoritma genetika dapat ditunjukkan pada persamaan Persamaan 1 danPersamaan 2 dibawah ini (Mahmudy, 2013a).

*NilaiFitne ss* 1

*f x*

dimana:

*f x* (*cij* ) *pi*

Keterangan:

*  adalah waktu tempuh dari titik i ke titik j.
*  merupakan penalti jika pelanggan dilayani diluar jadwal.

(1)

(2)

**2.2.2. *Crossover***

*Crossover* adalah mekanisme yang dimiliki algoritma genetika dengan menggabungkandua kromosom sehingga menghasilkan anak kromosom yang mewarisi ciri-ciri dasar dari parent *crossover* bekerja membangkitkan *offspring* baru dengan mengganti sebagian informasi dariparents (Mahmudy, 2013a). Pada penelitian ini digunakan metode *crossover* PMX dikarenakan dengan metode ini bisa mencegah adanya gen ganda pada suatu individu. Langkah – langkah metode ini adalah (Azmi, 2011): (1) Tentukan dua posisi pada kromosom dengan aturan acak. *Substring* yang berada dalam dua posisi ini dinamakan daerah pemetaan. (2) Tukar dua *substring* antar induk untuk menghasilkan anak. (3) Tentukan hubungan pemetaan di antara duadaerah pemetaan. (4) Tentukan kromosom keturunan mengacu pada hubungan pemetaan.

**2.2.3. Mutasi**

Proses mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk mengganti gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal (Zukhri, 2004). Sehingga bisa disimpulkan bahwa mutasi akan meningkatkan variasi populasi. penelitian ini digunakan *reciprocal exchange* mutasi dengan memilih dua posisi secara random, kemudian menukar kedua posisi tersebut. Penggunaan metode ini dikarenakan sangat mudah dan sederhana untuk diimplementasikan dan hasil dari proses mutasi tidak akan terdapat gen yang sama pada anaknya.

**2.2.4. Seleksi**

Proses seleksi adalah proses untuk menyaring calon generasi yang baru. Induk yang baik akan mampu untuk menghasilkan anak yang baik. Semakin tinggi nilai *fitness* dari suatu individu maka semakin besar kemungkinannya untuk terpilih (Widodo, 2010). Kemampuan algoritma genetika untuk memproduksi kromosom yang lebih baik secara progresif tergantung pada penekanan selektif (*selective pressure*) yang diterapkan ke populasi. Penekanan selektif dapat diterapkan dalam dua cara. Cara pertama adalah membuat lebih banyak kromosom anak yang dipelihara dalam populasi dan memilih hanya kromosom-kromosom terbaik bagi generasi berikutnya. Walaupun orang tua dipilih secara acak, metode ini akan terus menghasilkan kromosom yang lebih baik. Cara lain menerapkan penekanan selektif adalah memilih orang tua yang lebih baik ketika membuat keturunan baru. Dengan metode ini, hanya kromosom sejumlah populasi yang akan disimpan untuk generasi selanjutnya. Metode untuk seleksi yang sering digunakan antara lain adalah seleksi roda roulet (*roulette wheel selection*), seleksi ranking (*rank* *selection*), elitis dan seleksi turnamen (*tournament selection*).

**2.2.4.1. Metode Seleksi *Roulette Wheel***

Metode seleksi *roulette wheel* merupakan metode yang paling sederhana serta paling banyak digunakan, dan sering juga dikenal dengan nama *stochastic sampling with* *replacement*. Pada metode ini, orangtua dipilih berdasarkan nilai *fitness*nya, semakin baik nilai

*fitnes*snya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosomdiletakkan pada sebuah roda roulet, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitnes*snya (Wati, 2011).

**2.2.4.2. Metode Seleksi Elitis**

Metode Seleksi Elitis memilih individu-individu untuk dipakai pada generasi selanjutnya didasarkan pada urutan nilai *fitness*. Semakin tinggi nilai *fitness*nya maka individu tersebut akan dipertahankan (Wati, 2011). Proses seleksi dilakukan dengan mengurutkan semua kromosom pada satu generasi lalu diambil sebanyak ukuran populasi yang diinginkan.

**3. Metodologi Penelitian**

Terdapat dua jenis data antar pelanggan yang digunakan pada penelitian ini yaitu data pelanggan yang berjumlah 20 pelanggan untuk dataset 1 dan 30 pelanggan untuk dataset 2. Data jarak antar pelanggan didapatkan dari *google maps* dan data waktu ketersediaan pelanggangan dibangkitkan secara acak dari pukul 07.00 sampai pukul 16.00. Contoh data jarak antar pelanggan bisa dilihat pada Tabel 1 dan untuk contoh data waktu ketersediaan pelanggan bisa dilihat pada Tabel 2.

**Tabel 1. Contoh data jarak antar pelanggan**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Tujuan** | **0** | **1** | **2** | **3** | **4** | **5** |
|  | **0** | 0 | 7 | 13 | 12 | 8 | 9 |
|  | **1** | 7 | 0 | 8 | 6 | 6 | 6 |
|  | **2** | 13 | 8 | 0 | 8 | 13 | 12 |
|  | **3** | 12 | 6 | 8 | 0 | 10 | 12 |
|  | **4** | 8 | 6 | 13 | 10 | 0 | 15 |
|  | **5** | 9 | 6 | 12 | 12 | 15 | 0 |

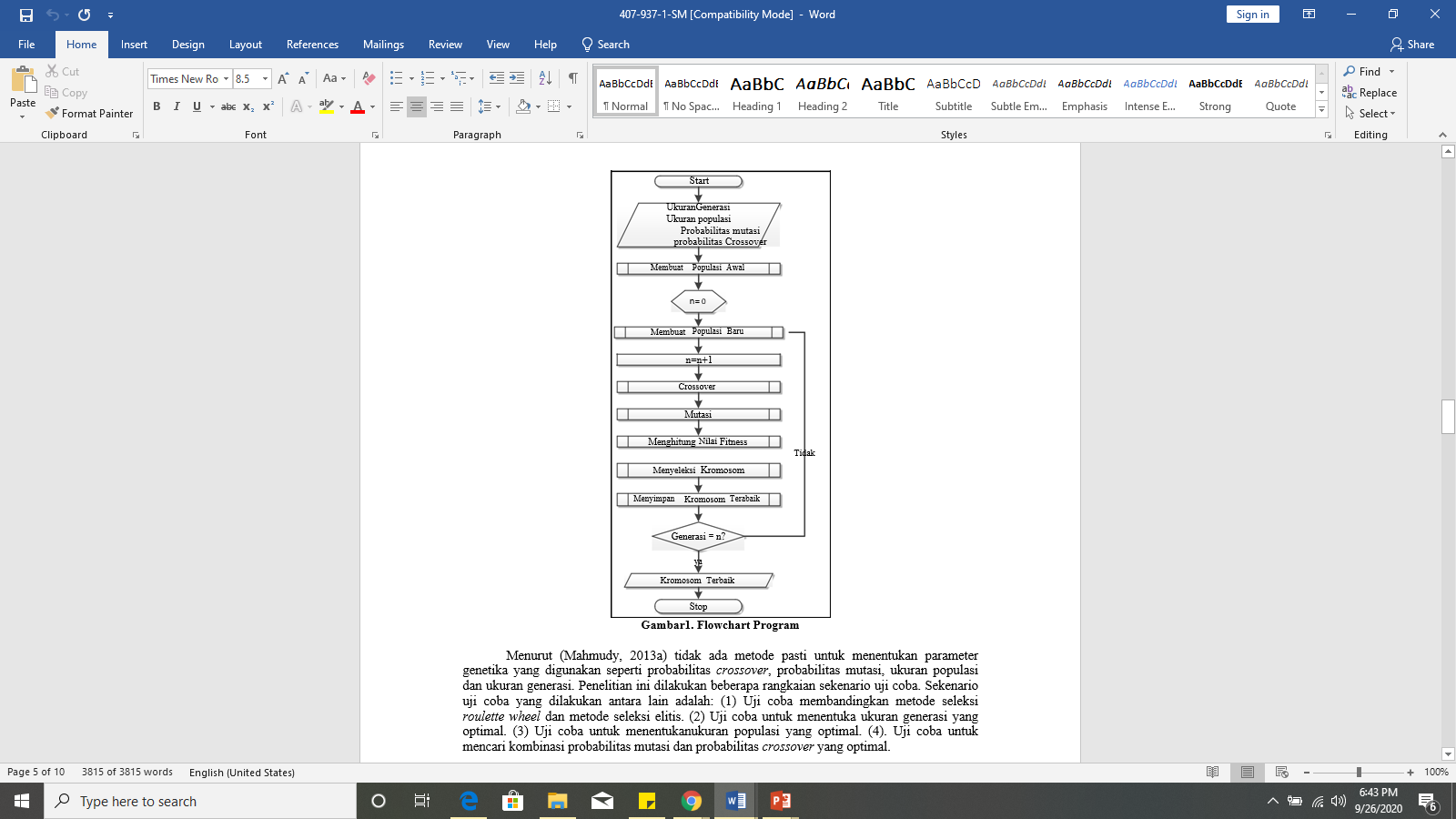
**Tabel 2. Contoh data waktu ketersediaan pelanggan**

|  |  |  |
| --- | --- | --- |
|  | **Tujuan** | **Ketersediaan Waktu** |
|  | 0 | 08.00-16.00 |
|  | 1 | 08.00-14.00 |
|  | 2 | 14.00-16.00 |
|  | 3 | 12.00-14.00 |
|  | 4 | 09.00-13.00 |
|  | 5 | 08.00-09.00 |

Proses pencarian rute tercepat dengan algoritma genetika adalah seperti pada *flowchart* [Gambar1.](#page5) Proses pertama adalah dengan menginisialisasikan parameter awal yaitu: (1) Memasukkan tujuan dan waktu ketersediaan untuk masing-masing tujuan. (2) Ukuran individu pada setiap populasi. (3) Ukuran generasi. (4) Probabilitas *crossover*. (5) Probabilitas mutasi.

Setelah menginisialisasikan parameter awal proses selanjutnya adalah membangkitkan populasi awal dengan panjang kromosom adalah banyaknya tujuan yang akan dituju, banyaknya populasi sesuai dengan jumlah individu yang telah diinisialisasi sebelumnya. Setelah mendapatkan populasi awal langkah selanjutnya adalah reproduksi dengan cara melakukan *crossover* dan mutasi. Pada proses *crossover* dan mutasi populasi diambil untuk dijadikansebagai calon induk. Pemilihan induk dilakukan secara random untuk menghasilkan anak sebanyak probabilitas *crossover* dan mutasi.

Proses selanjutnya adalah dengan menghitung nilai *fitness* semua kromosom dari semua proses pada generasi ini. Proses perhitungan nilai *fitness* dengan menggunakan Persamaan 1 dan Persamaan 2. setelah semua kromosom dihitung nilai *fitness*nya dilanjutkan dengan menyeleksi kromosom untuk diproses pada generasi selanjutnya dan menyimpan kromosom dengan nilai *fitness* terbaik. Pemilihan kromosom terbaik dilakukan dengan cara membandingkan nilai *fitness* terbaik pada setiap generasi. Proses seleksi melibatkan seluruh kromosom dari generasiawal dan kromosom hasil dari proses *crossover* dan mutasi. Hasil akhir dari algortima genetika adalah menampilkan kromosom yang memiliki nilai *fitness* tertinggi dari semua generasi.



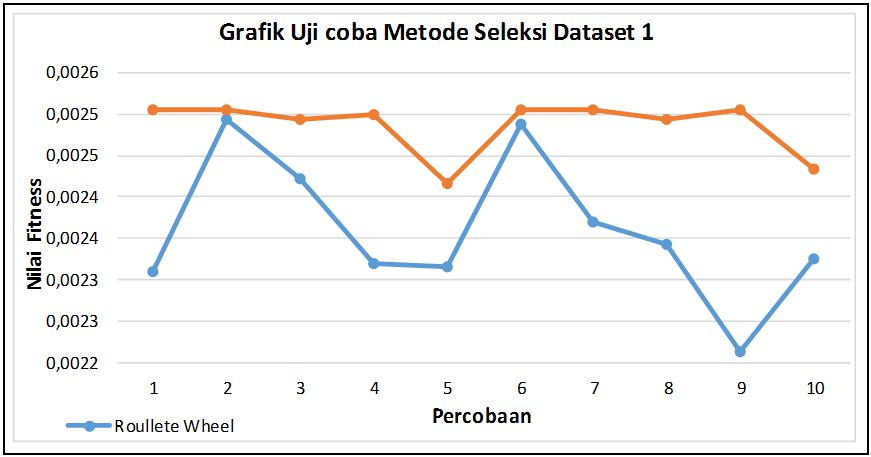
Menurut (Mahmudy, 2013a) tidak ada metode pasti untuk menentukan parameter genetika yang digunakan seperti probabilitas *crossover*, probabilitas mutasi, ukuran populasi dan ukuran generasi. Penelitian ini dilakukan beberapa rangkaian sekenario uji coba. Sekenario uji coba yang dilakukan antara lain adalah: (1) Uji coba membandingkan metode seleksi *roulette wheel* dan metode seleksi elitis. (2) Uji coba untuk menentuka ukuran generasi yangoptimal. (3) Uji coba untuk menentukanukuran populasi yang optimal. (4). Uji coba untuk mencari kombinasi probabilitas mutasi dan probabilitas *crossover* yang optimal.

**5. Hasil dan Pembahasan**

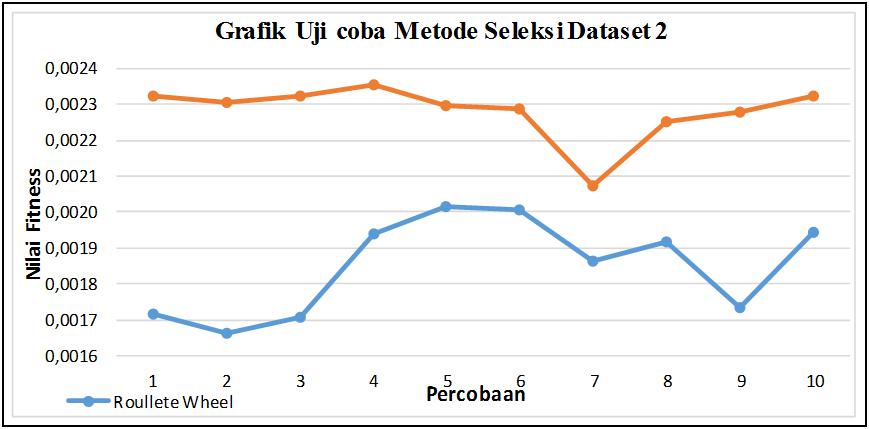
**5.1. Hasil dan Analisa Uji Coba Perbandingan Metode Seleksi *Roulette wheel* dan Elitis** Uji coba yang pertama dilakukan pengujian metode seleksi elitis dan metode seleksi

*roulette wheel*. Setiap metode seleksi dilakukan 10 kali percobaan dengan ukuran populasiadalah 40 populasi, ukuran generasi adalah 1000 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,5. Hasil uji coba bisa dilihat pada Gambar 2 untuk dataset 1, dan Gambar 3 untuk dataset 2.

Grafik menunjukan bahwa dengan 10 kali percobaan dengan dataset1 nilai *fitness* yang dihasilkan metode seleksi *roulette wheel* selalu berada dibawah nilai *fitness* percobaan dengan menggunakan metode seleksi elitis. Pada Gambar 2 bisa dilihat metode seleksi elitis cenderung menghasilkan nilai *fitness* yang stabil dibandingkan dengan metode seleksi *roulette wheel*. Tidak jauh berbeda dengan 10 kali percobaan pada dataset2 pada Gambar 3 nilai *fitness* yang dihasilkan metode seleksi *roulette wheel* selalu berada dibawah nilai *fitness* percobaan dengan menggunakan metode seleksi elitis. Pada Gambar 3 bisa dilihat metode seleksi elitis cenderung menghasilkan nilai *fitness* yang stabil dibandingkan dengan metode seleksi *roulette wheel*.



**Gambar 2. Grafik Perbandingan Metode Seleksi dataset 1**

****

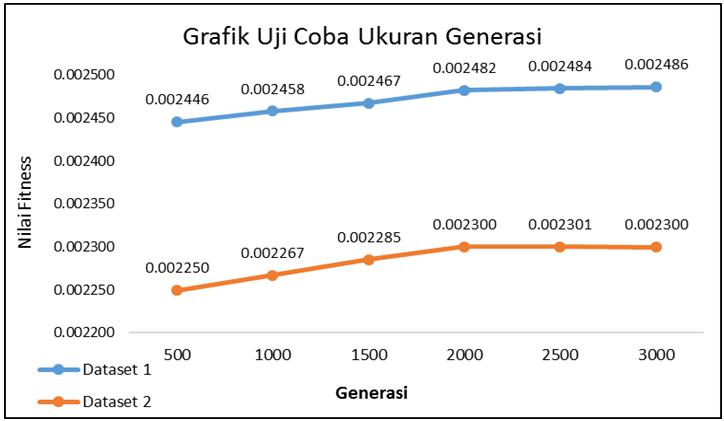
**Gambar 3.Grafik Perbandingan Metode Seleksi dataset 2**

Uji coba menggunakan dua jenis dataset yang berbeda bisa disimpulkan bahwa seleksi dengan menggunakan metode elitis lebih sesuai untuk masalah *traveling salesman problem* dengan *time windows* karena pada metode seleksi elitis semua individu yang bagus langsung dipilih menjadi indukan yang akan di proses pada generasi selanjutnya, sedangkan dengan metode *roulette wheel* individu yang belum bagus masih punya kesempatan untuk menjadi induk untuk generasi selanjutnya, sehingga nilai *fitness* yang dihasilkan pada setiap percobaan cenderung tidak stabil dan tidak optimal.

**4.2. Hasil dan Analisa Uji Coba Ukuran Generasi**

Uji coba yang kedua dilakukan pencarian ukuran generasi yang optimal. Setiap ukuran generasi dilakukan 20 kali percobaan dengan ukuran populasi adalah 40 populasi, ukuran generasi adalah kelipatan 500 mulai dari 500 sampai 3000 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,3. Metode seleksi yang digunakan adalah elitis.

Pada [Gambar 4](#page7) bisa dilihat bahwa ukuran generasi berpengaruh terhadap hasil dari proses algoritma genetika. Nilai terendah terdapat pada generasi 500 yaitu ukuran generasi terendah pada percobaan ini dikarenakan algoritma genetika belum memproses secara optimal karena kurangnya generasi. Ukuran generasi yang optimal untuk *traveling salesman problem* *with time windows* adalah 2000 generasi, dikarenakan dengan ukuran generasi 2000 sampai3000 generasi, nilai *fitness* yang dihasilkan tidak berbeda jauh dan cenderung membentuk garis lurus.



**Gambar 4. Grafik rata-rata nilai fitnes tiap generasi dataset 1**

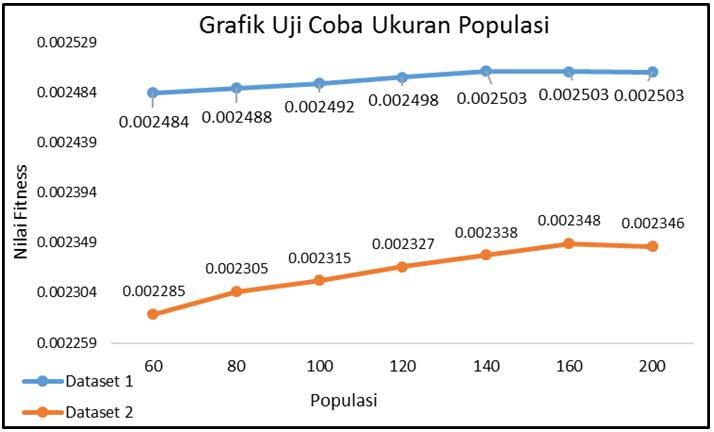
Terlalu banyak ukuran generasi belum tentu membuat algoritma genetika menjadi lebih optimal. Selain waktu proses yang menjadi lebih lama hasil nilai *fitness* yang dihasilkan juga belum tentu jauh lebih baik dari generasi yang lebih rendah. Ukuran generasi yang tinggi akan mengakibatkan proses evolusi semakin sering dilakukan. Pada setiap satu generasi, akan dilakukan proses rekombinasi yang terdiri atas *crossover* dan mutasi. Sehingga semakin banyak generasi maka proses rekombinasi akan semakin sering dilakukan. Tentunya hal ini akan berpengaruh pula terhadap individu – individu baru yang dihasilkan. Semakin banyak melakukan proses *crossover* dan mutasi maka individu – individu baru yang dihasilkan akan semakin bervariasi dan memungkinkan pula bervariasinya nilai *fitness* yang dihasilkan. Dengan begitu akan memberikan peluang yang besar untuk mendapatkan nilai *fitness* yang baik.

**5.3. Hasil dan Analisa Uji Coba Ukuran Populasi**

Uji coba yang ketiga dilakukan pencarian ukuran populasi yang optimal. Setiap ukuran populasi dilakukan 20 kali percobaan. Ukuran populasi adalah 60 sampai 200 populasi dengan kelipatan 40 populasi, ukuran generasi adalah 1500 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,3. Metode seleksi yang digunakan adalah elitis.Hasil uji cobabisa dilihat pada [Gambar 5.](#page8)

Pada [Gambar 5](#page8) bisa dilihat kenaikan signifikan rata-rata nilai *fitness* untuk 20 kali percobaan pada *dataset* 1 mulai dari ukuran populasi 60 sampai dengan 140 populasi, namun untuk ukuran populasi 140 sampai 200 populasi sudah tidak terjadi perubahan yang cukup signifikan dan pada grafik cenderung membentuk garis lurus. 140 populasi merupakan ukuran populasi yang optimal untuk masalah *TSPTW* pada dataset 1. Untuk dataset 2 bisa dilihat kenaikan signifikan rata-rata nilai *fitness* untuk 20 kali percobaan mulai dari ukuran populasi 60 sampai dengan populasi 160 populasi, namun pada ukuran populasi 160 sampai 200 populasi sudah tidak terjadi perubahan yang cukup signifikan dan pada grafik cenderung membentuk garis lurus. 160 populasi merupakan ukuran populasi yang optimal untuk masalah *traveling* ***salesman problem with*** *time windows* pada dataset 2. Dilihat dari perbedaan ukuran populasi

yang optimum dari hasil percobaan dengan dua jenis *dataset* yang berbeda maka bisa disimpulkan bahwa perbedaan dataset yang digukanan dapat mempengaruhi nilai *fitness* dan parameter genetika optimum yang dihasilkan oleh algoritma genetika.



**Gambar 5. Grafik hasil uji coba ukuran populasi dataset 1**

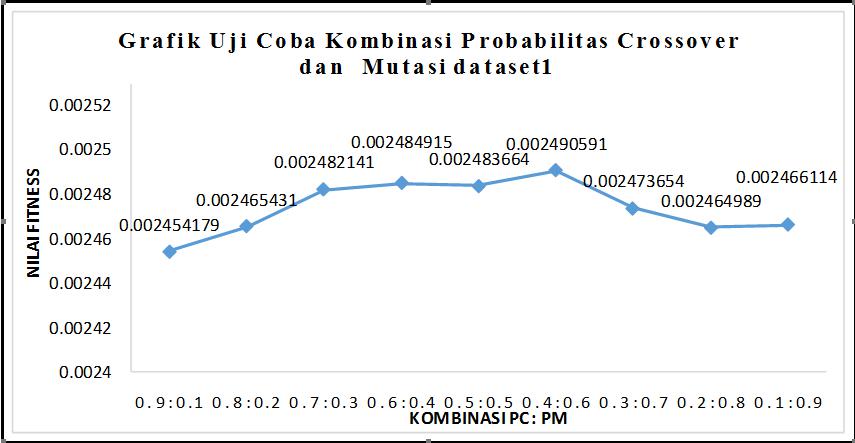
Semakin tinggi ukuran populasi maka berpengaruh pada rata-rata nilai *fitness* yang didapatkan namun semakin tinggi ukuran populasi juga berrpengaruh pada waktu pemrosesan algoritma genetika akan semakin lama, sehingga menentukan berapa ukuran populasi yang optimal sangat diperlukan. Ukuran populasi yang optimal untuk setiap *dataset* dapat berbeda-beda seperti pada grafik [Gambar 5.](#page8) Ukuran populasi yang optimal untuk *dataset* 1 adalah 140 populasi berbeda dengan *dataset* 2 yaitu 160 populasi, hal ini dapat disebabkan karena ukuran populasi yang kecil dimana variasi individu-individu di dalamnya sedikit. Individu yang akan terpilih sebagai kandidat *parent* akan memiliki variasi terbatas, dan anak yang dihasilkan bisa jadi memiliki sifat yang mirip dengan nilai *fitness* yang hampir dekat (tidak mengalami improvisasi).

**5.4. Hasil dan Analisa Uji Coba Kombinasi Probabilitas *Crossover* dan Mutasi**

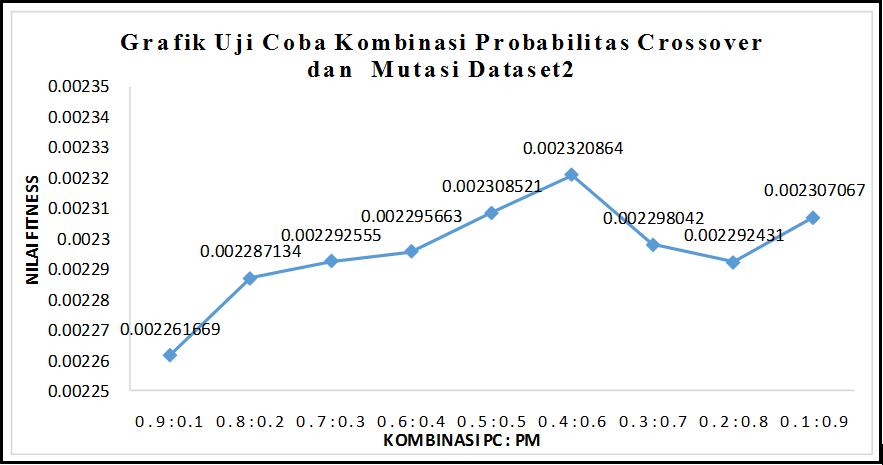
Uji coba yang keempat dilakukan pencarian kombinasi probabilitas *crossover* dan mutasi yang optimal. Penentuan kombinasi tingkat crossover dan mutasi sangat penting dilakukan untuk memperoleh solusi yang baik (mendekati optimum). Tingkat crossover yang terlalu besar (dan tingkat mutasi kecil) akan menghilangkan kesempatan algoritma genetika untuk mengeksplorasi daerah pencarian baru. Pada kondisi sebaliknya (tingkat crossover besar, tingkat mtasi kecil), algoritma genetika tidak mampu mengeksploitasi daerah optimum lokal (Mahmudy, 2012) (Mahmudy, 2013b).

Setiap kombinasi probabilitas *crossover* dan mutasi dilakukan 20 kali percobaan. Ukuran populasi adalah 40 populasi, ukuran generasi adalah 1500 generasi dan kombinasi probabilitas *crossover* dan mutasi adalah 0,1 : 0,9 sampai 0,9 : 0,1.Metode seleksi yang digunakan adalah elitis. Hasil uji coba bisa dilihat pada grafik Gambar 6 untuk dataset 1, dan Gambar 7 untuk dataset 2.

Berdasarkan Gambar 6 dan Gambar 7 rata-rata nilai *fitness* terbesar terdapat pada kombinasi probabilitas *crossover* dan probabilitas mutasi 0,4:0,6 dengan rata-rata nilai *fitness* adalah 0,00249 untuk dataset 1, dan 0,00232 untuk dataset 2. Berdasarkan Gambar 6 dan Gambar 7 bisa dilihat rata-rata nilai *fitness* semakin ketengah maka grafik semakin memuncak dengan kombinasi probabilitas *crossover* 0,4 dan probabilitas mutasi sebagai puncak tertingginya. Berarti kombinasi probabilitas *crossover* dan probabilitas mutasi yang terbaik untuk menyelesaikan masalah *TSP-TW* dalah 0,4:0,6.



**Gambar 6. Grafik hasil uji coba kombinasi probabilitas *crossover* dan mutasi dataset 1**

****

**Gambar 7. Grafik hasil uji coba kombinasi probabilitas *crossover* dan mutasi dataset 2**

Perbedaan nilai probabilitas *crossover* dan mutasi mempengaruhi nilai *fitness* yang dihasilkan. Semakin besar nilai probabiltas *crossover* dan probabilitas mutasi maka peluang individu yang mengalami proses *crossover* dan mutasi akan semakin besar dan akan semakin banyak individu – individu baru yang dihasilkan. Dengan demikian nilai *fitness* yang dihasilkan juga semakin bervariasi, sehingga peluang untuk mendapatkan individu dengan nilai *fitness* yang baik juga akan semakin besar.

**6. Kesimpulan**

Kesimpulan yang dapat diambil dari penelitian ini adalah menggunakan ukuran generasi, ukuran populasi, probabilitas *crossover* dan mutasi dan metode seleksi yang tepat algoritma genetika dapat diimplementasikan untuk menyelesaikan permasalahan antar jemput *laundry*. Metode seleksi Elitis lebih baik dan lebih stabil dari pada metode seleksi *roulette wheel*. Ukuran generasi yang optimal adalah 2000 generasi dengan Probabilitas *crossover* yangoptimal adalah 0,4 dan probabilitas mutasi yang optimal adalah adalah 0,6. Perbedaan *dataset*

yang digunakan dapat mempengaruhi nilai optimal parameter algoritma genetika. Dari nilai-nilai parameter ini didapatkan solusi yang memungkinkan untuk melayani semua pelanggan dengan *time* *window* masing–masing,

Untuk pengembangan penelitian selanjutnya dapat dikembangkan untuk menyelesaikan permasalahan pencarian rute perjalanan tercepat dengan menambahkan faktor seperti data kepadatan jalan dan data lampu merah pada jalan yang dilalui yang dapat mempengaruhi waktu tempuh perjalanan serta diambil secara *real time*. Pada kasus seperti ini diperlukan metode yang lebih baik untuk menjamin bahwa solusi yang mendekati optimum bisa tercapai. *Hibridisasi* algoritma genetika dengan metode lain merupakan satu pilihan yang terbukti efektif pada beberapa kasus kompleks (Mahmudy, 2013b).

**Referensi**

Anonimous. 2013. *Antar Jemput Laundry Kiloan*. “*Dari Jasa Antar Jemput Secuter*”, (Online), [(http://antarjemputsecuter.wordpress.com,](http://antarjemputsecuter.wordpress.com/) diakses 19 April 2014)

Azmi, N., Jamaran, I., Arkeman, Y., & Mangunwidjaya, D.2011. “Penjadwalan Pesanan Menggunakan Algoritma Genetika Untuk Tipe Produksi Hybrid And Flexible Flowshop Pada Industri Kemasan Karton”. *Jurnal Teknik Industri Vol 1 No 7*.

Chen, T., & Zhou, G. 2013. “Vehicie Routing Optimization Probiem With Time-Windows And Its Soiution By Genetic Aigorithm”. *Journal of Digital Information Management* *Volume 11 Number 2 April 2013*: ZhejiangGongshangUniversity.

Gambardella, L. M., Taillard, E., & Agazzi, G. 1999. “*A Multiple Ant Colony System For* *Vehicle Routing Problems With Time Windows”*. New Ideas in Optimization :McGraw-Hill, London.

Mahmudy, WF, Marian, RM & Luong, LHS 2012, 'Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part II: optimization', *International Conference on Control, Automation and Robotics,* *Singapore*, 12-14 September 2012, World Academy of Science, Engineering andTechnology, pp. 706-710,

Mahmudy, F. W. 2013a.”*Algoritma Evolusi”.*2013a.Malang: Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Mahmudy, WF, Marian, RM & Luong, LHS. 2013b, 'Real coded genetic algorithms for solving flexible job-shop scheduling problem – Part II: optimization', *Advanced Materials* *Research, vol. 701, pp. 364-369.*

Mahmudy, WF, Marian, RM & Luong, LHS .2014, 'Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing system', *ECTI Transactions on Computer and Information Technology* *(ECTI‐CIT), vol. 8, no. 1, pp. 80-93.*

Nugraha, I. 2008.”*Algoritma Genetik Untuk Optimasi Penjadwalan Kegiatan Belajar* *Mengajar”*. Institut Teknologi Bandung. Institut Teknologi Bandung.

Purwanto, Y., Purwitasari, D., & Wibowo, W. A. 2005. “Implementasi Dan Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya”. *Jurnal Penelitian dan Pengembangan* *Telekomunikasi (Edisi 10).*Surabaya:ITS.

Wati, A. W. 2011. “Penerapan Algoritma Genetika Dalam Optimasi Model Dan Simulasi Dari Suatu Sistem”. *Jurnal Keilmuan Tehnik Industri (Edisi 1 Nomor 2),* Jakarta : Universitas Trisakti.

Widodo, AW & Mahmudy, WF 2010. 'Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner', *Kursor, vol. 5, no. 4, pp. 205-211.*

Zukhri, Z. 2004. “Penyelesaian Masalah Penugasan dengan Algoritma Genetika”. *Seminar* *Nasional Aplikasi Teknologi Informasi,* Yogyakarta : Universitas Islam Indonesia.

Contoh 2 :

**PERANCANGAN APLIKASI QR *CODE* SEBAGAI MEDIA INFORMASI PENGENALAN SATWA KEBUN BINATANG BERBASIS WEBSITE**

**Muhammad Dimas Setiadi1) dan Neny Rosmawarni 2)**

**Program Studi Sistem Informasi, F.TSI - Institut Sains dan Teknologi Nasional**

**Jl. Moh. Kahfi II, Bhumi Srengseng Indah, Jagakarsa, Jakarta Selatan 12640**

**Telp. (021) 7270090, Fax. (021) 7866955, website : https://www.istn.ac.id**

**E-mail : prodisi.istn@yahoo.com, sistem\_informasi@istn.ac.id**

**1)** **setiadidimas43@gmail.com, 2) dan neny@istn.ac.id**

***ABSTRACT***

*With the development of technological advances, the internet is a means of infrastructure not only for communicating or interacting, in Indonesia alone there are 78 million people who use the internet. QR Code or Quick Respone is a matrix Code that was first discovered by denso wave in 1994, this QR Code has become the world's trending in this modern and digital era. One of the advantages of QR Code is that QR Code is universal and has little risk of error, zoo is a means for just either recreation or picnic. The zoo itself usually consists of various kinds of animals and primates, the purpose of this research is to make it easier for zoos to introduce their animals by utilizing this QR Code technology so that it can simplify and streamline the work of zoo staff and make it easier for visitors to recognize animal species so that development technology can be utilized properly.*

*Keywords : Internet, QR Code, Zoo*

**ABSTRAK**

Dengan berkembangnya kemajuan teknologi, internet merupakan sarana prasarana bukan hanya untuk berkomunikasi maupun berinteraksi, di indonesia sendiri ada 78 juta jiwa yang menggunakan internet. QR *Code* atau Quick Respone adalah kode matriks yang pertama kali di temukan oleh denso wave tahun 1994, QR *Code* ini sudah menjadi tranding dunia di jaman yang modern dan digital ini salah satu keunggulan QR *Code* adalah QR *Code* bersifat universal dan beresiko sedikit kesalahan, kebun binatang merupakan sarana untuk sekedar baik berekreasi ataupun piknik. Kebun binatang sendiri biasanya terdiri dari berbagai macam satwa dan primata, tujuan dilakukannya penelitian ini adalah untuk mempermudah kebun binatang dalam memperkenalkan satwa-satwanya dengan memanfaatkan teknologi QR *Code* ini sehingga dapat mempermudah dan mengefisienkan perkerjaan petugas kebun binatang dan mempermudah pengunjung dalam pengenalan satwa ragunan sehingga perkembangan teknologipun dapat dimanfaatkan dengan baik.

Kata Kunci : Internet, QR *Code*, Kebun binatang

1. **PENDAHULUAN**

**Latar Belakang**

Di zaman yang sudah modern ini penggunaan internet di dunia sudah sangat banyak, internet ini sangatlah mempermudah manusia dalam sekedar mencari informasi, berkomunikasi, dll. Menurut info dari CNN Indonesia data terbaru yang di rilis oleh Perserikatan Bangsa Bangsa (PBB), mencatat saat ini ada 3,9 miliar orang atau lebih dari setengah populasi dunia telah meggunakan internet. Angka ini untuk pertama kalinya melebihi Setengah populasi di dunia.(CNN Indonesia, 2019)

Dalam dunia internet kita menggenal istilah website, merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan infomasi teks, gambar diam atau gerak, animasi, suara, atau gabungan dari semuanya, baik bersifat statis maupun dinamis. (Bekti, 2015)

Dalam mengakses sebuah website kita hanya perlu membutuhkan koneksi internet dengan smartphone, PC, laptop. dengan begitu kita bisa mengakses dimanapun dan kapanpun selagi memiliki syarat diatas, dalam website pun kita bisa dengan mencari informasi, promosi, komunikasi dll.

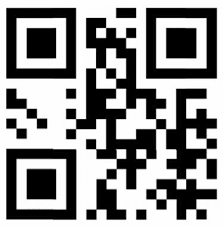
QR *Code* adalah salah satu jenis kode matriks atau kode batang dua dimensi yang dikembangkan Denso Wave, Denso Wave adalah sebuah divisi di perusahaan Denso Corperation Jepang, QR *Code* pertama kali di publikasikan pada tahun 1994. Sedangkan perangkat yang digunakan untuk membaca QR *Code* disebut QR *scanner*, atau pemindai QR. (Thonky, 2015)

Kebun binatang (sering disingkat bonbin, dari kebon binatang) atau taman margasatwa adalah tempat hewan dipelihara dalam lingkungan buatan, dan dipertunjukkan kepada publik. Selain sebagai tempat rekreasi, kebun binatang berfungsi sebagai tempat pendidikan, riset, dan tempat konservasi untuk satwa terancam punah. Binatang yang dipelihara di kebun binatang sebagian besar adalah hewan yang hidup di darat, sedangkan satwa air di pelihara di akuarium. Dengan adanya kebun binatang ini diharapkan spesies – spesies yang sedang terancam punah bisa kembali berkembang biak dan tidak menjadi punah. Ini juga menjadi salah satu tempat wisata yang jarang sekali sepi pengunjung, banyak orang yang pergi kesana sekedar piknik dan berekreasi semata, pada umumnya ketika kita datang ke kebun binatang kita akan di suguhkan dengan papan informasi di setiap kandang hewan yang sudah di sediakan oleh petugas kebun binatang, dengan adanya aplikasi QR *Code* Kebun Binatang Ragunan ini di harapkan bisa mempermudah dalam mencari informasi hewan yang kita butuhkan. Agar penulisan ini lebih terarah pembahasannya, maka perumusan masalah dalam penelitian ini adalah : Bagaimana cara memindai QR *Code* pada website, Bagaimana QR *Code* mampu mempermudah pengenalan satwa ragunan, dan Bagaimana QR *Code* dapat mempermudah petugas kebun binatang ragunan. Adapun tujuan dari penelitian ini, yaitu: Merancang dan membangung aplikasi pemindai QR *Code* berbasis web, Mengimplementasikan QR *Code* untuk pengenalan satwa ragunan, dan Mengimpelentasikan aplikasi dalam menginput data hewan ragunan.

1. **TINJAUAN PUSTAKA**

***Quick Respone Code* (QR *Code)***

*Quick Respone Code* atau yang sering disingkat QR *Code* merupakan sebuah *barCode* dua dimensi yang diperkenalkan oleh perusahaan jepang denso wave pada tahun 1994. Jenis bar*Code* ini awalnya digunakan untuk melacak persediaan dibagian manufaktur kendaraaan dan sekarang sudah digunakan dalam berbagai industri perdagangan dan jasa. Pada dasarnya bahwa QR *Code* dikembangkan sebagai suatu kode yang memungkinkan isinya untuk dapat diterjemahkan dengan kecepatan tinggi. QR *Code* terdiri dari sebuah untaian kotak persegi yang disusun dalam suatu pola persegi yang lebih besar, yang disebut sebagai modul. Gambar 2.1 berikut ini, menunjukan gambaran dari sebuah QR *Code*.



Gambar 2.1 QR *Code* (Roni, H., Dinda, A.M., & Nuha, H.K. (2020)

**Karakteristik QR *Code***

Karakteristik dari QR *Code* yaitu dapat menampung jumlah data yang besar. Secara teori sebanyak 7089 karakter numerik maksimum data dapat tersimpan di dalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol linier) dan pembacaan kode dengan cepat. QR *Code* juga memiliki kelebihan lain baik dalam hal untuk kerja dan fungsi (Ariadi, 2020). Berikut ini merupakan kelebihan unjuk kerja dan fungsi yang dimiliki oleh QR *Code*.

1) Pembacaan data dari segala arah (360 derajat)

Pembacaaan kode matriks dengan menggunakan sensor kamera CCD (*Charge Coupled Device*) dimana data akan memindai baris per baris dari citra yang ditangkap dan kemudian disimpan dalam memori dengan menggunakan perangkat lunak tertentu *finding pattern* akan dikenali dan posisi simbol dideteksi.

2) Ketahanan terhadap penyimpanan simbol

Simbol matriks 2 dimensi akan rentan terhadap penyimpanan bentuk ketika ditempatkan pada permukaan yang tidak rata, sehingga sensor pembaca menjadi miring karena sudut antara sensor CCD dan simbol matriks 2 dimensi ini telah berubah. Untuk memperbaiki penyimpanan ini, QR *Code* memiliki perata pola (*Alignment pattern*) yang menyusun dengan jarak yang teratur dalam suatu daerah.

3) Fungsi pemulihan data

QR *Code* mempunyai empat tingkatan koreksi eror (7%, 15%, 25%, dan 30%) di dalam mengendalikan kerusakan yang akibatnya kotor ataupun rusak. QR *Code* memanfaatkan algoritma Reed-Solomon yang tahan terhadap kerusakan tinggi.

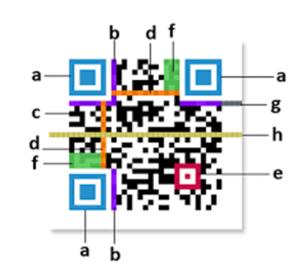
4) Kemampuan enkode karakter kanji dan kana jepang

QR *Code* berkembang pesat di negara jepang. Hal ini menyebabkan perkembangan QR *Code* untuk dapat menerima input data berupa karakter yang non-alfabetis. Ketika pembuatan QR *Code* dengan inputan berupa huruf jepang, maka data tersebut akan di ubah kedalam bentuk binear 16 bit (2 byte) untuk karakter tunggal, sendangkan untuk gabungan karakter akan di enkode dalam binear 13 bit.

5) Fungsi linking pada simbol

QR *Code* juga memiliki kemampuan dapat dipecah menjadi beberapa bagian dengan maksimum pembagiannya 16 bagian. Dengan fungsi linking ini, maka QR *Code* dicetak pada daerah yang tidak terlalu luas untuk sebuah QR *Code* tunggal.

**Anatomi *QR Code***



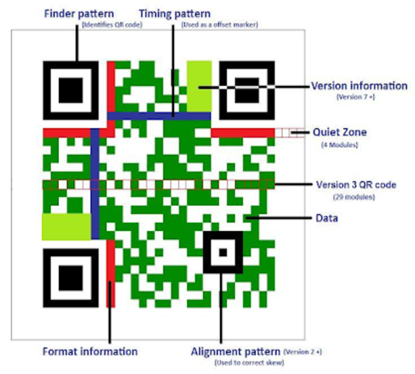
Gambar 2.2 Anatomi QR *Code* (Roni, H., Dinda, A.M., & Nuha, H.K 2020)

Beberapa penjelasan anatomi QR *Code* menurut (Ariadi, 2020) adalah sebagai berikut:

1. Finder Pattern berfungsi untuk identifikasi letak QR *Code*.
2. Format Information berfungsi untuk informasi *error correction level* dan *mask pattern*.
3. Data berfungsi untuk menyimpan data yang di kodekan.
4. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat QR *Code*, berbentuk modul hitam.
5. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan QR *Code* terrutama distorsi non linier.
6. *Version Information* adalah versi dari sebuah QR *Code*.
7. *Quiet Zone* merupakan bagian kosong.
8. QR *Code Version* adalah versi dari QR *Code* yang digunakan.

**Struktur QR *Code***

QR *Code* memiliki bagian-bagian struktur yang akan penulis jelaskan dalam gambar 2.3 dibawah ini :



Gambar 2.3 Struktur QR *Code* (Roni, H., Dinda, A.M., & Nuha, H.K 2020)

Berikut ini merupakan penjelasan istilah-istilah yang berkenaan dengan gambar QR *Code* di atas:

1. *Finding pattern* merupakan pola untuk mendeteksi posisi QR *Code.*
2. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat QR *Code*, berbentuk modul hitam putih bergantian
3. *Version Information* adalah versi dari sebuah QR *Code*, versi terkecil adalah 1 (21 x 21) modul dan versi terbesar adalah 40 (117 x 117) modul.
4. *Quiet Zone* merupakan daerah kosong di bagian terluar QR *Code* yang mempermudah mengenali pengenalan QR *Code* oleh sensor CCD
5. QR *Code Version* adalah versi dari QR *Code* yang digunakan pada contoh gambar, versi yang digunakan adalah versi 3(29 x 29) modul.
6. Data merupakan daerah tempat data tersimpan atau dikodekan
7. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpanan QR *Code* terutama distorsi non linear
8. *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*
9. **METODOLOGI PENELITIAN**

Beberapa metode yang dilakukan untuk meneliti dan menggembangkan aplikasi dalam menyelesaikan permasalahan, adapun metode tersebut adalah :

1. Metode pengumpulan data

Metode pengumpulan data merupakan teknik atau cara yang dilakukan untuk mengumpulkan data. Metode menunjuk suatu cara sehingga dapat diperlihatkan penggunaanya melalui :

a. Studi Literatur

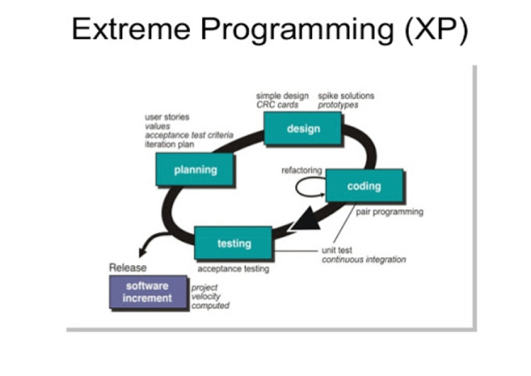
Studi literatur adalah pengumpulan data melalui buku-buku, jurnal, paper, dan bacaan-bacaan yang ada kaitannya dengan QR *Code* dan beberapa refrensi lainya untuk menunjang pencapaian tujuan tugas akhir.

b. Observasi

Observasi adalah pengumpulan data dan mencatat hal-hal yang berhubungan dengan QR *Code* secara lengkap dan sistematis.

2. Metode pembangunan perangkat lunak

Pada kasus ini Metode pengembangan sistem yang dipilih *Extreme Programming* (XP). *Extreme Programming* (XP) dipilih karena perangkat lunak yang akan dibuat tidak terlalu kompleks dan tergolong perangkat lunak berskala kecil dan juga membutuhkan waktu pengembangan yang tidak terlalu lama, *Extreme Programming* mencakup beberapa aturan dalam prakteknya, yang terdiri dari *planning, design, Implementation / Coding* dan *Test*



Gambar 3.1 Skema *Extreme Programming* (researchgate.net)

Adapun penjelasan dari tahapan metode pengembangan sistem *Agile Extreme Programming* adalah :

1. Planning / Perencanaan

Pada tahapan ini perencanaan terhadap *software* yang diinginkan mengacu pada *user* *stories*. *User* *stories* menggambarkan fitur dan fungsi yang dibutuhkan terhadap *software* tersebut. Ketika semua *user* *stories* telah ditentukan, *developer* akan menentukan lama pengerjaan untuk tiap-tiap *user stories*. Perencanaan dapat dilakukan apabila sudah mengetahui batasan masalahnya apa saja

Pada bagian *Planning / Perencanaan* ini saya membuat analisa terlebih dahulu dengan mengamati analisa yang sudah berjalan lalu membuat analisa usulan, pada analisa usulan ini saya membuatnya dengan tahap pembuatan *Flowchart, Use case, Activity Diagram, Class Diagram, Sequence Diagram* Menggunakan Draw.io

2. *Design*

Proses *desain* pada XP mengikuti prinsip KIS (*Keep It Simple*). Desain akan berisikan semua implementasi dari *stories* tanpa ada pengurangan maupun penambahan. Desain yang memiliki fungsi tambahan tidak disarankan. XP menggunakan CRC(*Class-Responsibility Collaborator*) *cards* untuk mengidentifikasi dan mengorganisasikan kelas berorientasi objek yang berkaitan dengan proses pengembangan perangkat lunak. Jika terdapat kesulitan untuk melakukan desain terhadap *stories*, XP menyarankan untuk membuat *prototype* dari desain tersebut. Hal ini disebut sebagai *spike solution, prototype* nantinya akan diimplementasikan dan dievaluasi.

3. *Coding*

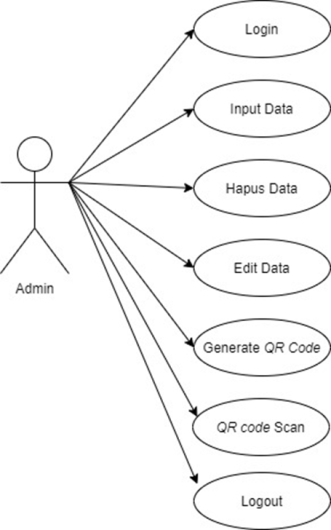
Pada tahap ini, proses pengembangan tidak langsung melakukan implementasi terhadap desain yang telah dibuat. Pembuatan unit test untuk tiap-tiap stories yang nantinya akan diimplementasikan. Saat unit *test* selesai dibuat, pengembang lebih baik fokus terhadap apa yang akan diimplementasikan untuk melewati unit test. Tahap ini akan mengacu pada desain sebelumnya. Karena pembuatan *unit test* dilakukan terlebih dahulu maka implementasi desain sebaiknya dibuat untuk melewati *unit test* yang dibuat.

4. *Testing*

Tahap ini akan menggunakan unit *test* yang sebelumnya telah dibuat. Karena pembuatan dari unit *test* adalah pendekatan utama dari XP.

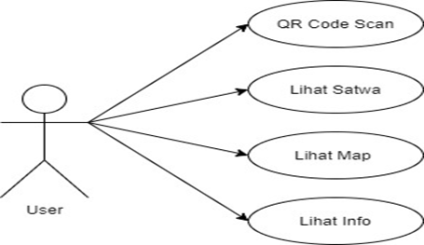
Pada tahap ini penulis menggunakan metode yaitu *black box*, untuk mengetahui algoritma yang saya gunankan telah berjalan sesuai dengan logika saya dan aplikasi *scanner* dapat berjalan dengan baik.

**Diagram** ***Use Case***



Gambar 3.2 Diagram *Use Case* Admin Yang Di Usulkan

Pada usulan *use case diagram* admin ini penulis ingin menjelaskan bahwa di aplikasi ini admin memegang hal penting ya itu menginput, mengedit, menghapus berbaagai macam data yang akan di sajikan kepada user / pemakai aplikasi.

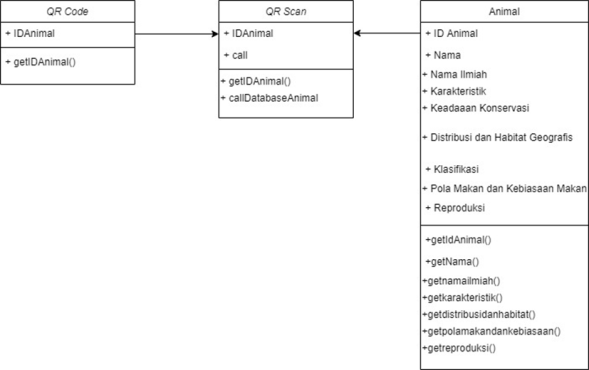


Gambar 3.3 Diagram *Use Case User* Yang Di Usulkan

Pada usulan *use case diagram user* ini penulis ingin menjelaskan bahwa *user* pada aplikasi ini dapat meng-*scan* semua QR *Code* yang telah di daftarkan oleh admin dan setelah meng-*scan* QR tersebut maka informasi yang di butuhkan akan keluar.

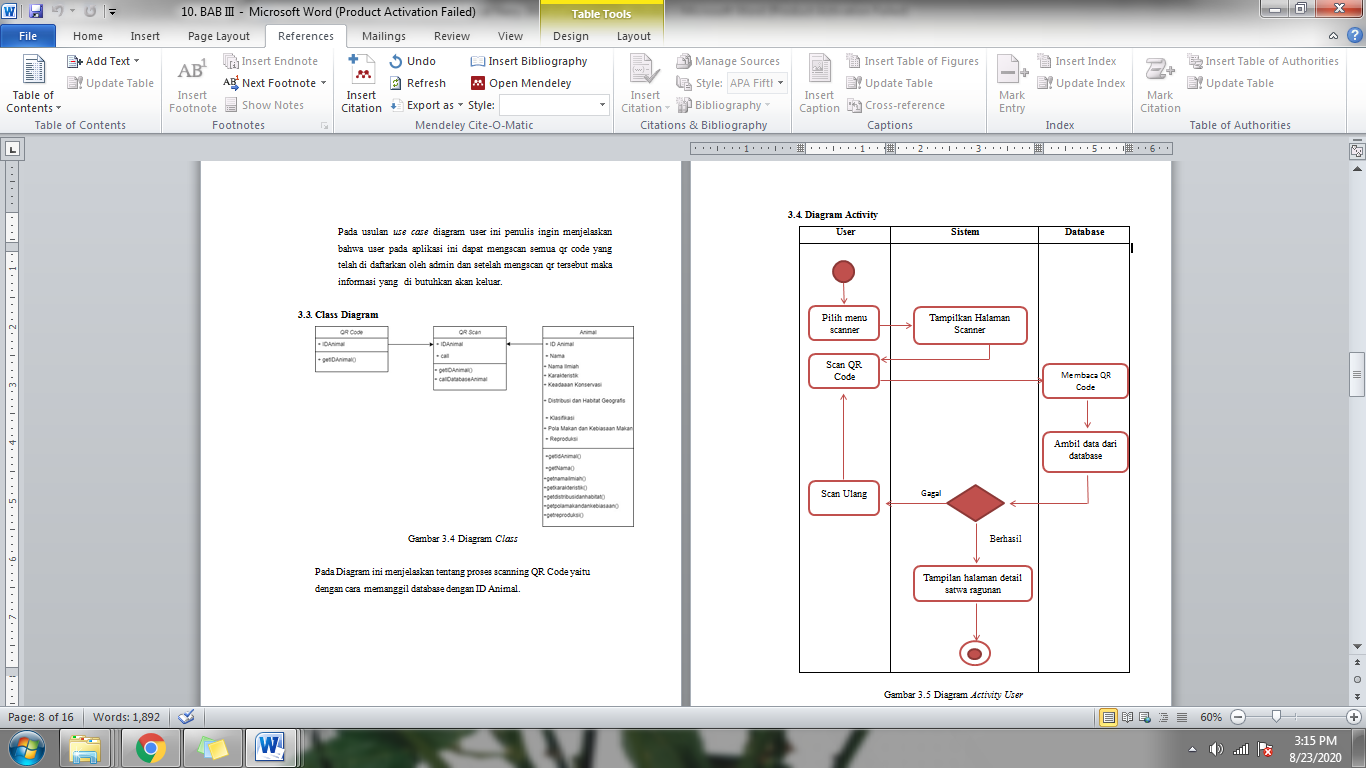
***Class Diagram***

Pada Diagram ini menjelaskan tentang proses *scanning* QR *Code* yaitu dengan cara memanggil *database* dengan ID Animal.



Gambar 3.4 Diagram Class

**Diagram *Activity***



Gambar 3.5 Diagram Activity User

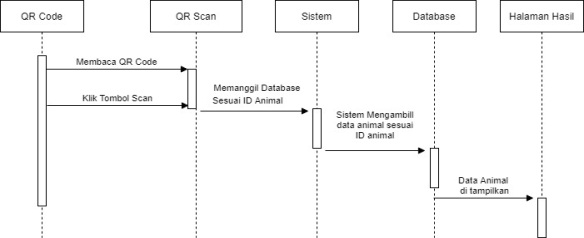
Pada Diagram *Activity User* yang penulis buat yaitu menjelaskan *logic* yang penulis gunakan dalam proses scanning qr *Code* dan pengambilan data dari *database*.

**Diagram *Sequence***



Gambar 3.6 Diagram Sequence User

Pada Diagram Sequence yang penulis buat ini menjelaskan tentang cara user menggunakan aplikasi.



Gambar 3.7 Diagram *Sequence* QR *Code*

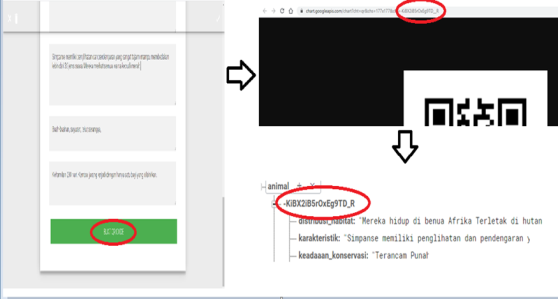
Pada Diagram Sequence yang penulis buat ini menjelaskan cara kerja QR scan terhadap QR *Code*.

**Metode QR *Code***

Pada aplikasi ini penulis memakai metode QR *Code* yaitu QR De*Code*r. QR *DeCoder/EnCoder library* adalah sebuah library yang memungkinkan anda untuk membuat (menyandikan) gambar QR *Code* atau, membaca (de*Code*) gambar yang mengandung satu atau lebih QR *Code*, kode sumber terlampir terbuat dari 2 solusi QR *Code* yaitu QR *Code* En*Code*r dan QR *Code* *deCoder*. QR *Code* ini membutuhkan NET Framework minimal (NET462) dan NET standard (netstandard2.0).

QR De*Code*r mengubah gambar QR *Code* Bitmap menjadi array byte array. Jika gambar QR *Code* berisi Nilai Tugas ECI opsional, dapat diambil kembali seperti yang tercantum di bawah ini. Untuk mendekode Bitmap yang mengandung satu atau lebih gambar QR *Code*.

Buat objek QR De*Code*r. Objek ini dapat digunakan kembali. Jika Anda ingin membuat banyak Kode QR, cukup gunakan kembali objek ini. Tidak ada inisialisasi atau membuang persyaratan. Dan panggil metode Image De*Code*r.

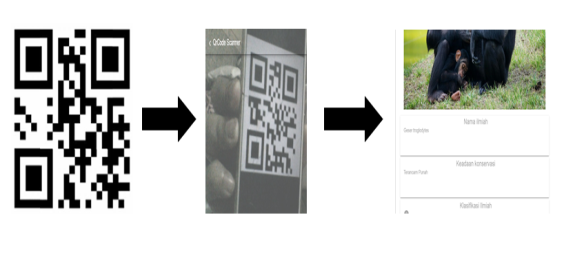


Pada Gambar 3.8 ini menjelaskan bagaimana cara mengenerate QR *Code*.

yaitu setelah admin menginput data hewan lalu mengcreate QR *Code* yang berisi kode unik lalu setelah melakukan proses scanning sistem membaca QR *Code* dan melakukan pemanggilan *database*.

**Perancangan Aplikasi QR *Code***

Pada perancangan aplikasi QR *Code* Kebun Binatang menjelaskan bagaimana cara kerja alur sistem, disaat aplikasi QR *Code* Kebun Binatang sedang digunakan. Sistem ini memberikan informasi satwa Kebun Binatang.



Gambar 3.9 Alur Kerja Sistem QR *Code*

Pada gambar ini menjelaskan cara kerja aplikasi yaitu mulai dari scanning QR *Code* hinggal menunjukan hasil dari scanning QR *Code*.

1. **HASIL DAN PEMBAHASAN**

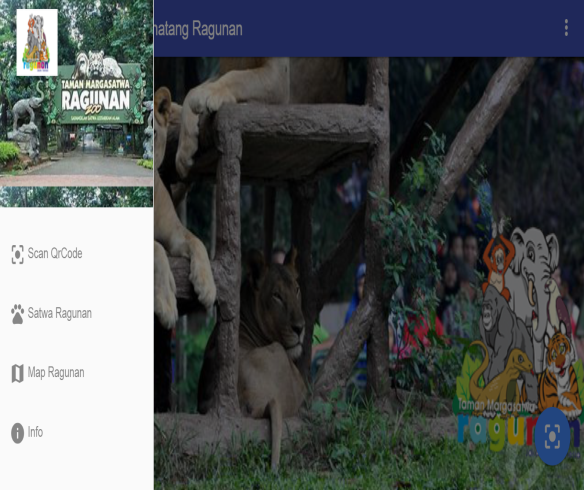
**Hasil**

Tampilan aplikasi pada sistem akademik, sebagai berikut :



Gambar 4.1 Tampilan Beranda

Pada halaman *home,* pada halaman ini menjelaskan bentuk awal dari aplikasi scan qr *Code* ragunan ketika baru di buka oleh pengunjung.



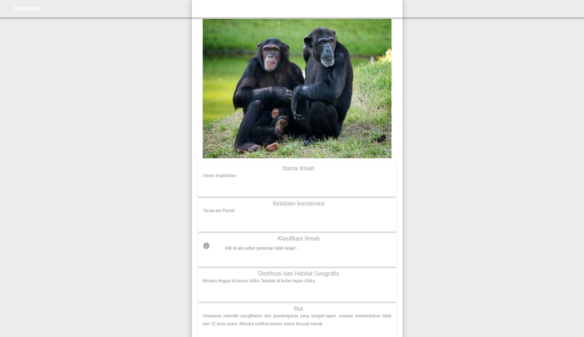
Gambar 4.2 Fiktur Fiktur Pada Web

Pada Halaman Web ini kita di perlihatkan fiktur fiktur yang tersedia di menu home pada aplikasi QR *Code* web pada kebun bintang ragunan yang nantinya bisa di akses oleh para pengunjung.



Gambar 4.3 Halaman *Scan* QR *Code*

Pada halaman ini pengunjung dapat melakukan scan QR *Code* yang di sediakan oleh pihak ragunan di tempat tempat yang sudah di sediakan di depan kandang hewan ragunan**.**



Gambar 4.4 Halaman Hasil *Scanner*

Pada halaman ini menampilkan hasil dari scanner *QR Code.*

**Pengujian Perangkat Lunak**

Pengujian perangkat lunak yang digunakan untuk mengetahui kelayakan program aplikasi untuk digunakan. Metode yang dilakukan dalam pengujian adalah *black box* *testing*. yaitu pengujian yang berfokus pada persyaratan fungsional perangkat lunak tanpa memperdulikan *coding* program. Pengujian ini untuk menentukan sejauh mana sistem yang ada dapat memenuhi keingan *user*.

• Prosedur Pengujian

Pengujian yang dilakukan adalah *black box testing* yaitu pengujian yang terfokus pada apakah unit program memenuhi kebutuhan yang disebutkan dalam spesifikasi. Pada *black box testing* cara pengujiannya hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit tersebut sesuai dengan proses bisnis yang diinginkan.

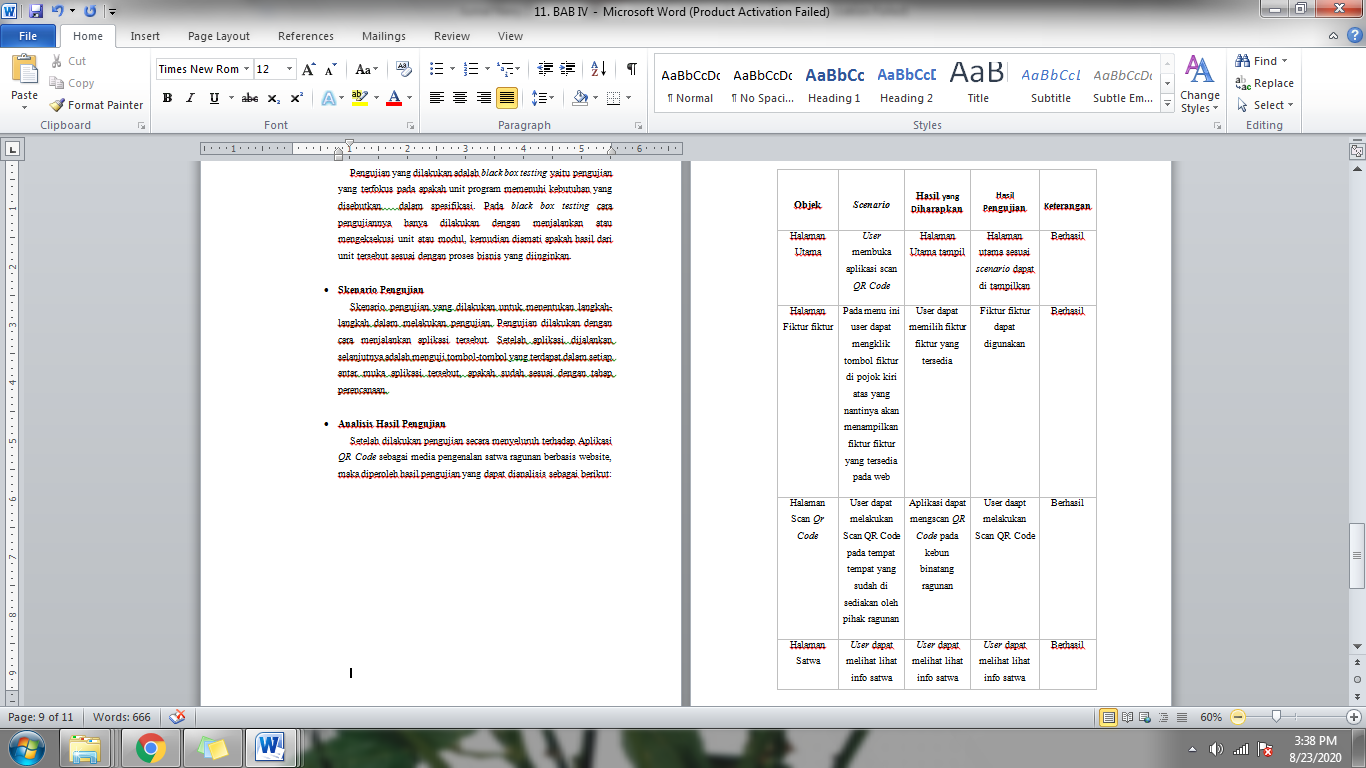
• Skenario Pengujian

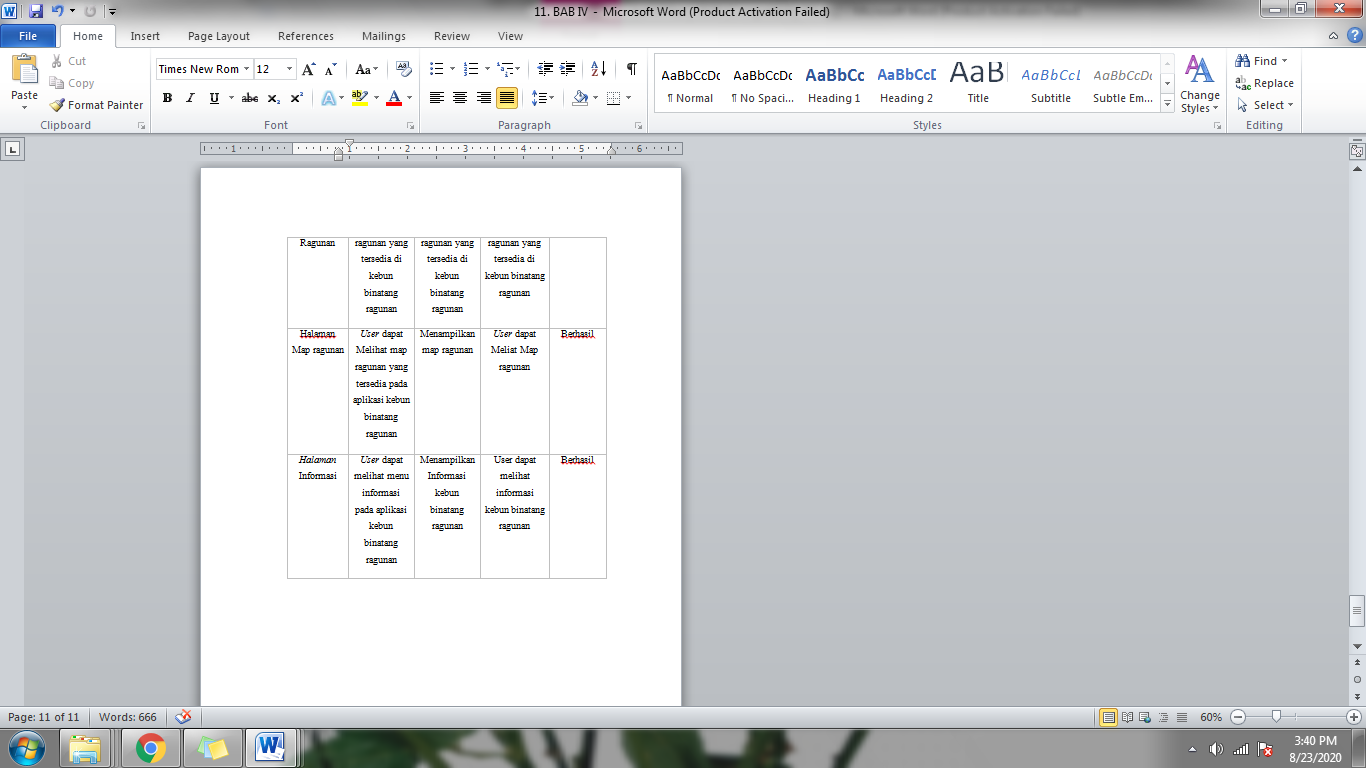
Skenario pengujian yang dilakukan untuk menentukan langkah-langkah dalam melakukan pengujian. Pengujian dilakukan dengan cara menjalankan aplikasi tersebut. Setelah aplikasi dijalankan selanjutnya adalah menguji tombol-tombol yang terdapat dalam setiap antar muka aplikasi tersebut, apakah sudah sesuai dengan tahap perencanaan.

• Analisis Hasil Pengujian

Setelah dilakukan pengujian secara menyeluruh terhadap Aplikasi QR *Code* sebagai media pengenalan satwa ragunan berbasis website, maka diperoleh hasil pengujian yang dapat dianalisis sebagai berikut:

TABEL 4.1 Tabel Hasil uji coba Black box





1. **SIMPULAN DAN SARAN**

**5.1 Simpulan**

Berdasarkan hasil dan analisis dan pengujian yang dilakukan pada bab sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut :

1. Dengan adanya aplikasi ini, diharapkan dapat menghemat biaya operasional taman marga satwa.
2. Dengan adanya aplikasi ini, memudahkan pengunjung taman margasatwa ragunan dalam melihat informasi hewan.
3. Dengan adanya aplikasi ini, petugas taman margasatwa ragunan hanya perlu menginput data-data hewan ke dalam web dan membuat QR *Code*.
   1. **Saran**

Berdasarkan simpulan di atas maka dapat diusulkan beberapa saran demi menunjang efektifitas dari aplikasi :

1. Diharapkan pengelola taman marga satwa dapat menerapkan dan melakukan sosialisasi terhadap penggunaan aplikasi ini kepada para pengunjung.
2. Diharapkan Aplikasi ini dapat dikembangkan lagi sehingga aplikasi ini menjadi lebih baik untuk kedepannya.

**DAFTAR PUSTAKA**

1. Abdulloh, Rohi (2018) 7 in 1 Pemrograman Web Untuk Pemula.
2. Bekti, Bintu Humairah. (2015). Mahir Membuat Website dengan Adobe Dreamweaver CS6, CSS dan JQuery, Yogyakarta: ANDI.
3. Buana, I Komang Setia, (2014). Jago pemrograman PHP. Dunia Komputer, Jakarta Indonesia.
4. Ginanjar, Cipto Prasetyo, (2014). Implementasi *QR Code* Dan Algortima *Vigenere* Pada Sistem Laporan Kehilangan Kendaraan Bermotor.
5. Guy, E. Blelloch., (2017). *An introduction to Reed-Solomon Code: Principle, architecture and implementation.*
6. Ibnu, M.S. (2020) Otodidak Web Programming: Membuat Website Edutainment.
7. Indrajani, S.Kom, MM., (2011). Perancangan Basis Data Dalam Allin1. Jakarta Elex Media Komputerindo.
8. Romney, Marshall B. dan Steinbart, (2015). Sistem Informasi Akuntansi, Edisi 13, alihbahasa : Kikin Sakinah Nur Safira dan Novita Puspasari, Salemba Empat, Jakarta.
9. Roni, H., Dinda, A.M., & Nuha, H.K. (2020). Aplikasi Inventory Barang Menggunakan *QR Code.*
10. Sanjaya. (2015). Model Pengajaran Dan Pembelajaran, Bandung: CV Pustaka Setia.
11. Silvia, A. F., Haritman, E., & Mulyadi, Y. (2014). Rancangan Bangun Akses Kontrol Pintu Gerbang Berbasis Andorid
12. Thonky. (2015). *QR Code Tutorial.*
13. Solichin, Achmad (2016) Pemrograman Web Dengan PHP dan MySQL
14. [www.firebase.google.com/dosc](http://www.firebase.google.com/dosc) diakses pada tanggal 15 Juli 2020 jam 18:52
15. [www.json.org](http://www.json.org) diakses pada tanggal 15 Juli 2020 jam 19:24
16. <https://www.researchgate.net/figure/XP-Software-Process-Model-5_fig1_336724102>
17. di akses pada tanggal 16 Juli 2020

**Ucapan Terima Kasih :**

Dalam kesempatan ini penulis mengucapkan terima kasih kepada Pimpinan Institut Sains dan Teknologi Nasional dan seluruh jajarannya, Kepala Pusat Informasi Fasilitas dan Komunikasi (Taman Marga Satwa) Kebun Binatang Rangunan Jakarta.

Quiz Pertemuan 4

Buatlah 1 contoh studi kasus penerapan algoritma data science yang menurut kalian kasus tersebut belum pernah ada dan menarik untuk di teliti. Jelaskan bagaimana konsepnya dan perkiraan solusinya!

# Daftar Pustaka

[1] “Data Scientist: The Sexiest Job of The 21st Century” Diakses 01 April 2019 dari https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

[2] “50 Best Jobs in America 2019” Diakses 01 April 2019 dari https://www.glassdoor.com/List/Best-Jobs-in-America-LST\_KQ0,20.htm

[3] “Tech in Asia Jobs” Diakses 02 Desember 2018 dari https://www.techinasia.com/jobs

[4] “Kalibrr: Where Jobs Find You” Diakses 02 Desember 2018 dari https://www.kalibrr.com/

[5] “Data Scientist Earning More Than CAs, Engineers” Diakses 01 April 2019 dari https://timesofindia.indiatimes.com/india/Data-scientists-earning-more-than-CAs-engineers/articleshow/52171064.cms

[6] “5 Facts About Software Engineers, Like Which One Gets Paid The Most” Diakses 01 Apil 2019 dari https://learning.linkedin.com/blog/tech-tips/the-american-city-that-pays-software-engineers-the-most–and-oth

[7] “Top 10 Reasons Why You Should Learn Data Analytics” Diakses 01 April 2019 dari https://bigdata-madesimple.com/10-reasons-why-you-should-learn-data-analytics/

[8] “5 Reasons Why Everybody Should Learn Data Analytics” Diakses 01 April 2019 dari https://www.sas.com/en\_au/insights/articles/analytics/5-reasons-why-everybody-should-learn-data-analytics.html

[9] “Permintaan Tenaga Data Scientist Melonjak, Jadikan Profesi Ini Kian Menjanjikan” Diakses 10 Desember 2018 dari https://id.techinasia.com/talk/profesi-data-scientist-menjanjikan

[10] “Go-Jek Buka Kantor Data Science di Singapura, Apa Alasannya?” Diakses 01 April 2019 dari https://www.liputan6.com/tekno/read/2998722/go-jek-buka-kantor-data-science-di-singapura-apa-alasannya

[11] Introduction to Data Science. A Python Approach to Concepts,Techniques and Applications. Laura Igual, Santi Segui. Tahun

2017.

[12] Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk from The Frontline. O'Reilly. 2014.

[13] Jure Leskovek, Anand Rajaraman and Jerey Ullman. Mining of Massive Datasets. v2.1, Cambridge University Press. 2014. (free online)

[14] Mohammed J. Zaki andWagner Miera Jr. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press. 2014.

[15] https://anaktik.com/data-science/

[16] https://anaktik.com/skill-data-scientist/