KURAWAL Jurnal Teknologi, Informasi dan Industri *Volume 2 Nomor 2 - Oktober 2019*
***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

RANCANG BANGUN APLIKASI REKOMENDASI PARSEL MENGGUNAKAN *DIFFERENTIAL EVOLUTION*

Fika Handani[1)], Hendry Setiawan[2)], Paulus Lucky Tirma Irawan[3)]
Teknik Informatika Universitas Machung, Villa Puncak Tidar N-1 Malang
email : fika.hdn@gmail.com [1)], hendry.setiawan@machung.ac.id[2)], paulus.lucky@machung.ac.id[3)]

Abstraksi

Knapsack Problem adalah permasalahan dimana seseorang dihadapkan pada optimasi pemilihan objek yang dimasukan kedalam wadah dengan kapasitas terbatas. Rekomendasi untuk memilih barang berdasarkan kategori dan jenis adalah salah satu hal yang termasuk dalam kategori knapsack problem. Mulia Jaya Minimarket adalah salah satu supermarket yang melayani pesanan parsel. Masalah yang sering terjadi adalah kurangnya variasi untuk menentukan kombinasi item pada parsel. Maka dibutuhkan aplikasi yang dapat membantu memberikan rekomendasi parsel secara otomatis. Aplikasi ini dibuat dengan menerapkan algoritma differential evolution. Solusi untuk rekomendasi diimplementasikan dalam bentuk vector. Setiap vector akan dihitung menggunakan nilai fitness dengan mempertimbangkan kualitas dan budget yang diberikan. Proses ini akan dihentikan ketika telah mendapatkan vector terbaik dengan Batasan iterasi tertentu. Hasil uji coba menunjukan bahwa aplikasi ini memiliki tingkat akurasi budget sebesar 0,99820 dan rata-rata nilai fitness terbaik 0,99501 dengan nilai crossover sebesar 50.

Kata Kunci :
Differential Evolution, Rekomendasi, Knapsack Problem

Abstract

Knapsack problem is a problem where people are faced to an optimization in the selection of objects that are inserted into a limited capacity container. Recommendations for selecting items, parcel's categories and items are one of the problems categorized as knapsack problem. Mulia Jaya Minimarket is one of the supermarkets that serve parcel orders. The problem that often occurs is the lack of initiative in variety of items in the parcel. So, it needs an application to give recommendations on parcel that considering on budget and quality. In order to meet this need, an application is made to provide parcel recommendations automatically. This application is created by implementing a differential evolution algorithm. Solution for recommendation is implemented in vector form. Each vector will calculate the fitness value with the criteria of quality and budget maximization. This process will be carried out until it reaches the best vector that is able to provide the best recommendations. The evaluation result shows that the application has a level of accuracy budget 0,99820 and average of best fitness is 0,99501 with value of crossover is 50.

Keywords :
Differential evolution, recommendation, knapsack problem

1. PENDAHULUAN

Knapsack merupakan optimasi pengangkutan barang atau disebut juga optimasi kombinatorial. *Knapsack problem* adalah salah satu masalah bagaimana cara menentukan pemilihan barang dari sekumpulan barang di mana setiap barang tersebut menmpunyai berat dan *profit* masing-masing, sehingga dari pemilihan barang tersebut didapatkan *profit* yang maksimum. Diah (2010). Tujuan *Knapsack problem* adalah agar mendapatkan keuntungan

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media transportasi tersebut. Dalam teori algoritma, persoalan *Knapsack* termasuk kedalam kelompok *NP-complete*.

Rekomendasi parsel merupakan salah satu permasalahan yang tergolong *knapsack problem*. Kendala yang selama ini dihadapi oleh pemilik toko atau swalayan maupun pemesan adalah kombinasi dari berbagai jenis barang, sesuai dengan jenis parsel serta harga untuk parsel tersebut. Untuk menentukan kombinasi parsel bisa dilakukan secara manual dengan pertimbangan harga dan jenis barang yang diinginkan. Akan tetapi jika terdapat aplikasi yang bisa membantu pemesan atau pemilik toko untuk mendapatkan rekomendasi parsel, akan lebih memudahkan dan mampu memberikan beberapa alternatif dari sistem pada aplikasi.

Mulia Jaya Minimarket yang berlokasi di jalan parangargo No. 24B RT 09 RW 02 Wagir, Kabupaten Malang merupakan salah satu swalayan yang menjual berbagai produk keperluan sehari-hari. Seperti sembako, peralatan kecantikan, mainan anak-anak dan barang-barang penunjang yang lain. Selain menjual barang kebutuhan sehari-hari, swalayan ini juga melayani pemesanan parsel untuk keperluan hari raya, hantaran dan keperluan parsel yang lain. Swalayan ini mempunyai banyak data barang dengan jenis yang berbeda-beda. Jumlah data yang ada di minimarket ini terdapat kurang lebih 10.000 item barang dengan kategori yang berbeda-beda. Dengan data yang banyak menyebabkan kombinasi barang di dalam parsel bisa bermacam-macam dengan harga yang berbeda-beda. Kendala yang selama ini dihadapi oleh pemilik toko atau swalayan maupun pemesan adalah kombinasi dari berbagai jenis barang, sesuai dengan jenis parsel serta harga untuk parsel tersebut. Untuk menentukan kombinasi parsel bisa dilakukan secara manual dengan pertimbangan harga dan jenis barang yang diinginkan. Akan tetapi jika terdapat aplikasi yang bisa membantu pemesan atau pemilik toko untuk mendapatkan rekomendasi parsel, akan lebih memudahkan dan mampu memberikan beberapa alternatif dari sistem pada aplikasi.

Permasalahan pemilihan rekomendasi parsel pernah dilakukan dalam penelitian oleh Wardojo (2011) Universitas Ma Chung dengan judul Pemilihan Parsel Makanan Menggunakan Algoritma *Harmony Search*. Aplikasi yang dibangun pada penelitian ini menggunakan Visual Basic dengan *database* MySQL. Dalam penelitian ini tidak mempertimbangkan jenis parsel seperti yang akan dilakukan dalam penelitian ini.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

2. TINJAUAN PUSTAKA

2.1 Pengertian Knapsack Problem

Knapsack merupakan optimasi pengangkutan barang atau disebut juga optimasi kombinatorial. *Knapsack problem* adalah salah satu masalah bagaimana cara menentukan pemilihan barang dari sekumpulan barang di mana setiap barang tersebut menmpunyai berat dan *profit* masing-masing, sehingga dari pemilihan barang tersebut didapatkan *profit* yang maksimum. Diah (2010). Tujuan *Knapsack problem* adalah agar mendapatkan keuntungan yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media transportasi tersebut. Dalam teori algoritma, persoalan *Knapsack* termasuk kedalam kelompok *NP-complete*.

Knapsack problem merupakan masalah dimana orang dihadapkan pada persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam wadah yang memiliki keterbatasan ruang atau daya tampung. Dengan adanya optimasi dalam pemilihan benda yang akan dimasukkan ke dalam wadah tersebut diharapkan dapat menghasilkan pilihan yang optimum. Benda-benda yang dimasukkan ini masing-masing memiliki nilai yang digunakan untuk menentukan prioritasnya dalam pemilihan tersebut. Nilainya dapat berupa tingkat kepentingan, harga barang, atau yang lainnya. Wadah yang dimaksud disini juga memiliki nilai konstanta yang merupakan nilai pembatas untuk benda-benda yang akan dimasukkan ke dalam wadah tersebut sehingga harus diambil sebuah cara memasukkan benda-benda tersebut ke dalam wadah sehingga menghasilkan hasil optimum tetapi tidak melebihi kapasitas atau batas yang telah ditetapkan

2.2 Pengertian Parsel

Parsel adalah pemberian yang dibungkus atau sebuah paket yang dibungkus Kusno (2015). Parsel biasanya diberikan kepada seseorang ketika menjelang hari raya seperti idul fitri, natal sebagai bentuk ungkapan kekerabatan. Saat ini banyak sekali bentuk parsel yang dapat kita jumpai seperti karangan bunga, bahan pecah belah, kue kering, kue basah, *snack*, sirup atau bentuk yang lain. Parsel yang diberikan kepada orang lain merefleksikan diri atau institusi kita sehingga dalam memberikan parsel kepada orang lain membutuhkan pilihan yang selektif.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

2.3 *Algoritma Optimasi*

Algoritma optimasi dapat didefinisikan sebagai algoritma atau metode *numeric* untuk menemukan nilai *x* sedemikian sehingga menghasilkan (*fx*) yang bernilai sekecil atau sebesar mungkin untuk suatu fungsi *f* yang diberikan, yang mungkin disertai dengan beberapa batasan pada *x*. Disini, *x* bisa berupa skalar atau vektor dari nilai-nilai kontinyu atau diskrit Suyanto (2010).

Algoritma optimasi sedikit berbeda dengan algoritma pencarian (*search algorithm*). Pada algoritma pencarian terdapat suatu kriteria tertentu yang menyatakan apakah elemen xi merupakan solusi atau bukan. Sebaliknya pada algoritma optimasi mungkin tidak terdapat kriteria tersebut, melainkan hanya fungsi-funsgi objektif yang menggambarkan bagus tidaknya suatu konfigurasi yang diberikan. Algoritma optimasi bisa dikatakan sebagai generalisasi dari algoritma pencaran atau dengan kata lain, algoritma pencarian adalah kasus khusus dari algoritma optimasi.

2.4 *Differential Evolution*

Algoritma *Differential Evolution* (DE) adalah salah satu algoritma metaheuristik yang diperkenalkan oleh Storn dan Price pada tahun 1997. Algoritma DE termasuk dalam keluarga *Evolutionary Algorithm* (EA) yaitu *evolutionary population-based algorithm* dimana prinsip dan filosofi algoritmanya meniru perilaku evolusi biologi. Teknik optimasi praktis umumnya harus memenuhi 3 persyaratan [7]. Pertama, metode harus menemukan nilai global optimum. Kedua, konvergensi harus cepat. Ketiga, program harus memiliki minimal parameter kontrol sehinggal akan lebih mudah untuk digunakan. Ketiga persyaratan tersebut yang mendasari munculnya DE. DE memiliki keunggulan yaitu konsep yang sederhana, implementasi yang mudah dan konvergensi yang cepat, walaupun kinerja DE sangat tergantung dari parameternya [8]. Langkah-langkah algoritma DE tersebut dapat dijabarkan sebagai berikut.

1. Inisialisasi Populasi: Sebelum melakukan inisialisasi titik populasi, perlu dilakukan penentuan batas atas (ub) dan batas bawah (lb). Berikutnya adalah membangkitkan bilangan acak untuk setiap parameter j dari vektor i pada iterasi g. Misal nilai inisial (g = 0) dapat diwakili dengan notasi sebagai berikut:

$$\chi_{j,i,0} = lb_j + rand_j(0,1)\,(ub_j - lb_j) \quad (1)$$

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Bilangan *random* dibangkitkan dengan fungsi *rand*(), dimana bilangan yang dihasilkan terletak pada rentang (0,1).

Mutasi: Setelah tahapan inisialisasi, DE akan melakukan mutasi dan kombinasi populasi awal untuk menghasilkan populasi dengan ukuran N vektor percobaan. Mutasi dilakukan dengan cara menambahkan perbedaan dua vektor terhadap vektor ketiga dengan cara random. Berikut ini adalah persamaan yang menunjukkan bagaimana membentuk vektor mutan, $\upsilon_{i,g}$:

$$\upsilon_{i,g} = \chi_{r0,g} = F.(\chi_{r1,g} - \chi_{r2,g}) \qquad (2)$$

Dimana $r0$, $r1$, $r2$ merupakan indeks acak, bilangan integer yang berbeda.

2. *Crossover*: Pada tahap ini DE menyilangkan setiap vektor, $X_{i,g}$ dengan vektor mutan, $V_{i,g}$ untuk membentuk vektor hasil persilangan yaitu $U_{i,g}$. Persamaan untuk vektor uji adalah sebagai berikut:

$$u_{i,g}=u_{j,i,g}= \begin{cases} vj,i,g & if(rand)j(0,1) \leq Cr,\ or\ j = jrand) \\ xj,i,g & sebaliknya \end{cases} \qquad (3)$$

Probabilitas *crossover*, $Cr \in (0,1)$ merupakan nilai yang didefinisikan untuk mengendalikan fraksi nilai parameter yang disalin dari mutan.

3. *Seleksi*: Terdapat 2 tahapan dalam proses evolusi yang menggunakan seleksi yaitu sebagai berikut :

   a. *Parent selection:* Vektor yang terpilih ditandai dengan nilai fungsi terbaik dan probabilitas seleksi tertinggi. Metode ini dalam memberikan probabilitas seleksi membutuhkan tambahan asumsi tentang bagaimana menggambarkan nilai fungsi tujuan menjadi probabilitas.

   b. *Survivor selection:* Pada metode ini, untuk mengetahui apakah vektor menjadi anggota generasi g + 1, maka vektor uji $u_{i,g}$ dibandingkan dengan vektor target $\chi_{i,g}$. Jika vektor $u_{i,g}$ mempunyai fungsi tujuan yang lebih kecil daripada $\chi_{i,g}$ maka $u_{i,g}$ akan menggantikan posisi $\chi_{i,g}$ dalam populasi pada generasi berikutnya. Bila

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

sebaliknya maka nilai $\chi_{i,g}$ yang lama dipertahankan. Persamaan tersebut dapat ditunjukkan sebagai berikut:

$$X, i, g+1 \begin{cases} ui, g & if(f(ui,g) \leq f(xi,g) \\ xi, g & sebaliknya \end{cases} \quad (4)$$

Proses akan diulang sampai *stopping criterion* tertentu dicapai.

3 HASIL DAN PEMBAHASAN

3.1 Analisis dan Perancangan Sistem

3.1.1 Identifikasi Masalah

Banyaknya jumlah data barang yang dapat dijadikan alternatif dalam penjualan parsel menjadi salah satu kendala yang dialami di Mulia Jaya Minimarket. Alternatif untuk variasi yang dapat direkomendasikan secara manual juga sangat minim sehingga dibutuhkan sebuah aplikasi yang dapat memberikan rekomendasi parsel secara otomatis dengan mempertimbangan budget dan kualitas barang berdasarkan prioritas yang dimasukkan. Pada tahapan ini juga dibedakan jenis parsel menjadi dua yaitu jenis parsel sembako atau jenis parsel makanan ringan dan minuman, dimana setiap jenis parsel akan memiliki kategori barang lainnya.

3.1.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini sejumlah kurang lebih 3.000 item yang terbagi menjadi dua jenis parsel. 990 item barang adalah jenis parsel sembako dan 1.836 adalah parsel dengan jenis makanan ringan dan minuman.

Data tersebut kemudian dikelompokan kembali berdasarkan kategori barang sesuai dengan jenis parsel. Untuk jenis parsel sembako data dibagi menjadi 361 mie instan, 141 minyak goreng, 21 gula pasir, 30 beras, 184 bumbu dapur, 197 tepung, 88 kopi, 14 teh dan 146 susu. Sedangkan untuk jenis parsel makanan ringan dan minuman data dikelompokan menjadi 178 wafer, 285 biskuit, 65 sirup, 175 permen, 191 susu siap minum, 221 minuman, 171 coklat, 150 keripik, 370 snack dan 30 sereal. Data tersebut diidentifikasi lagi dengan membedakan kualitas barang, dimana kualitas barang terbagi menjadi 3 yaitu kualitas baik, kualitas sedang dan kualitas biasa. Kualitas ini ditentukan oleh pemilik toko pada data barang yang tersedia.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Data yang akan dioperasikan dengan menggunakan algoritma differential evolution adalah id barang. Selain itu sistem memberikan fitur untuk menentukan jumlah barang yang boleh sama. Jumlah barang ini nantinya juga akan diproses dengan menggunakan algoritma differential evolution untuk mendapatkan hasil yang lebih baik untuk mendapatkan rekomendasi parsel.

3.1.3 Perencanaan Model

Sistem yang dibangun akan memberikan akses *user* untuk memberikan nilai budget, jenis parsel yang diinginkan dan kategori barang yang dipilih. Sehingga sistem akan menerima 3 input dari *user*. Kemudian sistem akan mengakses database yang berisi nilai parameter *differential evolution* seperti nilai *crossrate*, jumlah maksimal iterasi dan jumlah vektor yang dibangkitkan.

Nilai *crossrate* dalam sistem akan berada antara 0 sampai dengan 1. Nilai *crossrate* akan digunakan pada proses *crossover*. Jika nilai random *crossover* lebih kecil dari pada nilai *crossover* yang ditetapkan, maka individu terpilih adalah hasil inisialisasi dan jika nilai *crossover* yang dihasiklak lebih besar dari nilai *crossover* yang ditetapkan, maka individu terpilih adalah hasil mutasi.

Jumlah maksimal iterasi adalah penentu cepat atau lambatnya sebuah rekomendasi. Semakin besar jumlah iterasi yang diberikan dalam program maka akan semakin lama program memberikan hasil rekomendasi, karena program harus melakukan perulangan lebih lama. Demikian juga sebaliknya, jika nilai maksimal yang diberikan semakin sedikit, maka hasil rekomendasi yang diberikan juga akan semakin cepat hasil rekomendasi yang diberikan. Akan tetapi dalam memberikan nilai maksimum iterasi, disarankan untuk tidak memberikan nilai yang terlalu kecil. Nilai yang terlalu kecil untuk nilai maksimum iterasi maka hasil rekomendasi yang diberikan tidak akan mencapai hasil maksimal.

Jumlah vektor adalah jumlah individu yang akan dibangkitkan. Satu vektor akan mewakili satu hasil rekomendasi yang diberikan. Semakin banyak jumlah vektor yang dibangkitkan akan semakin banyak alternatif rekomendasi yang diberikan. Akan tetapi semakin banyak jumlah vektor yang dibangkitkan, maka akan semakin lama pula jalannya aplikasi dalam memberikan hasil rekomendasi.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3.1.4 Inisialisasi, Mutasi dan *Crossover* pada Sistem

Proses pertama yang dilakukan dalam algorima adalah inisialisasi, dimana proses inisialisasi adalah membangkitkan individu baru secara acak. Dimana jumlah individu yang akan dibangkitkan akan sama dengan jumlah vektor yang disimpan dalam *database*. Proses berikutnya yang dilakukan adalah proses mutasi. Proses mutasi pada sistem dilakukan dengan menggunakan persamaan sebagai berikut :

$$v_{i,g} = x_{r_o,g} + F(x_{r_1,g} - x_{r_2,g}) \quad (5)$$

Dimana:

$v_{i,g}$: nilai hasil mutasi

$x_{r_o,g}$: id barang ke g pada vektor terpilih r0

$x_{r_1,g}$: id barang ke g pada vektor terpilih r1

$x_{r_2,g}$: id barang ke g pada vektor terpilih r2

Setelah menyelesaikan proses mutasi, individu akan dilakukan proses *crossover*. Proses *crossover* akan membangkitkan terlebih dahulu nilai antara 0-100. Jika nilai *random* yang dihasilkan lebih kecil dari pada nilai *crossrate* pada sistem, maka individu yang terpilih adalah individu dari hasil inisialisasi. Sebaliknya, jika nilai random yang dihasilkan lebih besar dari pada nilai *crossrate* pada sistem, maka individu yang terpilih adalah individu hasil mutasi.

Proses terakhir yang dilakukan pada setiap iterasi adalah proses seleksi. Proses seleksi dilakukan dengan cara menurutkan individu dengan nilai fitness total terbesar hingga terkecil. Semakin besar nilai *fitness* yang dimiliki oleh individu maka individu tersebut dipilih sebagai hasil terbaik dari rekomendasi.

3.1.5 Hasil Rekomendasi

Hasil rekomendasi yang diberikan akan dipertimbangkan dengan selisih antara budget dan rekomendasi biaya yang diberikan oleh sistem. Semakin kecil selisih biaya yang dihasilkan oleh sistem, maka hasil tersebut dianggap sebagai hasil rekomendasi terbaik. Hasil rekomendasi akan dipilih 5 individu dengan nilai *fitness* terbaik pada iterasi terakhir. Sehingga dalam sistem rekomendasi nantinya, *user* akan diberikan 5 pilihan parsel dengan item yang berbeda.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3.1.6 Perhitungan Nilai Fitness

Seperti yang sudah dipaparkan pada penjelasan sebelumnya bahwa nilai *fitness* total akan mempertimbangkan 2 faktor yaitu pilihan kualitas barang yang ditentukan dan selisih antara budget dengan hasil rekomendasi yang diberikan.

Nilai *fitness* pertama adalah *fitness* yang mempertimbangkan pilihan kualitas barang. Jika barang yang dipilih memiliki urutan prioritas awal, maka barang pada kategori tersebut akan dipilihkan item yang memiliki kualitas terbaik. *Fitness* pertama dirumuskan sebagai berikut

$$fitness1 = \sum \quad bobot\ kualitas \times bobot\ prioritas \tag{6}$$

Sedangkan perhitungan nilai *fitness* kedua adalah mempertimbangkan selisih antara budget yang ditentukan dengan biaya hasil rekomendasi sistem. Semakin kecil selisih antara biaya hasil rekomendasi dengan budget yang ditentukan, maka akan semakin baik nilai *fitness* yang didapatkan. *Fitness* kedua dirumuskan sebagai berikut

$$fitness1 = 1 - (|budget - perolehan\ harga| : budget) \tag{7}$$

Setelah mendapatkan nilai *fitness* 1 dan 2, masih-masih nilai *fitness* dikali dengan 0,5 untuk mendapatkan *range* nilai antara 0 – 1 dan kemudian menjumlahkan kedua nilai tersebut. Hasil *fitness* terbaik adalah hasil yang memiliki nilai mendekati angka 1 (maksimum).

3.1.7 Kriteria Pemberhentian Iterasi

Iterasi akan dihentikan ketika sudah mencapai jumlah iterasi maksimum yang ditentukan pada sistem. Pada tahapan uji coba nantinya, akan dilakukan beberapa kali pengujian jumlah iterasi untuk mendapatkan jumlah iterasi yang cukup baik untuk mendapatkan hasil rekomendasi yang terbaik.

Tabel 1 Hasil Uji Coba Nilai *Crossrate* 50

Nilai Crossrate	Budget	Rekomendasi	Akurasi Biaya	Rata-rata fitness
50	Rp200.000	Rp201.800	0,99102	0,95210
50	Rp500.000	Rp499.500	0,99804	0,99900
50	Rp1.000.000	Rp1.004.700	0,99832	0,99530
50	Rp1.500.000	Rp1.507.900	0,99473	0,99473
Rata-rata			0,99553	0,99501

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3.2 Hasil Uji Coba

3.2.1 Uji Coba parameter terbaik differential evolution

Untuk menjalankan optimasi, maka parameter-parameter dari *differential evolution* perlu dilakukan uji coba. Uji coba dilakukan dengan cara menguji nilai *crossrate* dan jumlah iterasi. Nilai *crossrate* yang disarankan untuk menjalankan aplikasi rekomendasi ini adalah 50, dengan perolehan nilai *fitness* terbaik yaitu 0,99501 setelah dilakukan 4 kali pengujian dengan memberikan input pada nilai yang berbeda. Hasil tersebut dapat dilihat pada table 1 tentang hasil uji coba nilai *crossrate*.

Jumlah iterasi yang direkomendasikan pada aplikasi rekomendasi ini adalah 50. Hasil tersebut didapatkan dari hasil pengujian dengan menggunakan jumlah iterasi sebesar 20, 50, 100, 200 dan 500 iterasi. Dengan memberikan nilai jumlah maksimal iterasi sebesar 50, nilai *fitness* terbaik yang didapatkan sudah cukup stabil dengan perolehan lebih dari 90% ketika sampai pada iterasi ke 15 dengan perolehan nilai uji coba sebesar 0,99740. Sehingga jumlah maksimal iterasi cukup sampai 50 untuk mendapatkan hasil rekomendasi parsel yang baik.

3.2.2 Hasil Uji Coba Akurasi Biaya

Uji coba akurasi biaya dilakukan untuk menguji hasil dari rekomendasi parsel berdasarkan budget saja tanpa mempertimbangkan kualitas barang yang terpilih didalamnya. Hasil Uji coba akurasi biaya dilakukan bersamaan dengan uji coba *crossrate* hanya saja tanpa mempertimbangkan kualitas barang terpilih. Uji coba dilakukan dengan memberikan input budget sebesar Rp. 200.000, Rp. 500.000, Rp. 1.000.000 dan Rp. 1.500.000. Berdasarkah hasil uji coba yang dilakukan, maka didapatkan nilai akurasi sebesar 0,98804. Hasil tersebut dapat dikatakan adalah hasil yang baik karena memiliki tingkat akurat lebih dari 90% menuju target budget yang diberikan.

3.2.3 Hasil Uji Coba Nilai Ekstrem

Ujicoba nilai ekstrem dilakukan untuk menguji keadaan ekstrem ketika budget yang diberikan kecil akan tetapi *user* memilih banyak item parsel didalamnya dan sebaliknya ketika user memberikan nilai budget yang besar akan tetapi memilih sedikit item parsel didalamnya. Uji coba nilai ekstrem dilakukan dengan memberikan input nilai jumlah vektor sebanyak 15, iterasi maksimum sebanyak 50 dan memasangkan nilai *crossrate* sebesar 50. Hasil uji coba pada nilai ekstrem masih belum dapat berjalan dengan baik. Dengan memberkan input budget sebesar Rp. 200.000 dan memilih 20 item barang, hasil rekomendasi yang didapatlan sebesar Rp. 769.300 dengan perolehan nilai *fitness* terbaik -1,84650. Hal ini menunjukan bahwa aplikasi yang dibangun masih belum bisa digunakan pada keadaan ekstrem.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

4 SARAN DAN KESIMPULAN

Saran yang dapat digunakan untuk pengembangan yaitu aplikasi dapat dikembangkan dengan menambahkan bentuk barang, berat dan kemasan yang dapat digunakan untuk mengemas parsel yang akan direkomendasikan.

5 DAFTAR PUSTAKA

[1] Diah, Kartina, 2010,'Penyelesaian Knapsack Problem Menggunakan Algoritma Genetika', *Politeknik Caltex Riau Pekanbaru*

[2] Irawan, Doddy, 2013,'Analisis Pemberian Parcel Kepada Pegawai Negeri Sipil Sebagai Grafitasi Berdasarkan Undang-undang Pemberantasan Tindak Korupsi',*Hukum Pidana Fakultas Hukum, Bandar Lampung*

[3] Krismiaji, 2010. Sistem Informasi Akuntansi edisi ketiga. Yogyakarta: Unit Penerbit dan Percetakan Sekolah Tinggi Ilmu YKPN.

[4] Kusno, Gustaaf, 2005. Salah Kaprah Penyebutan Istilah Parcel. (http:/www.kompasiana/Salah-Kaprah-Penyebutan-Istilah-Parce-oleh-Gustaaf-Kusno-Kompasiana.com.html) diakses tanggal 14 Maret 2018

[5] Price, et al, 2005,'Metaheuristic Evolutionary Algorithm Based Search'

[6] Sandi, Kosasi, 2013,'Penyelesaian Bounded Knapsack Problem Menggunakan Dynamic Programming', *STMIK Pontianak*

[7] Setemen, Koman, 2010,'Implementasi Algoritma Genetika Pada Knapsack Problem Untuk Optimasi Pemilihan Buah Kemasan Kotak',*Universitas Udayana Bali*

[8] Surpriana, Wayan, 2016,'Optimalisasi Penyelesaian Knapsack Problem dengan Algoritma Genetika',*Universitas Udayana Indonesia*

[9] Wardojo, Kevin, 2011,'Rancang Bangun Aplikasi Pemilihan Barang Untuk Pembuatan Bingkisan Dengan Menggunakan Algoritma Harmony Search',*Universitas Ma Chung, Malang*