

PENGELOMPOKAN POLIGON UNTUK PERMASALAHAN 2D IRREGULAR STRIP PACKING BERDASARKAN CONVEX HULL DAN BOUNDING BOX

Fetty Tri Anggraeny¹, Nanik Suciati², Anny Yuniarti³

¹ Jurusan Teknik Informatika, Fakultas Teknologi Industri, UPN “Veteran” Jawa Timur

^{2,3} Jurusan Teknik Informatika, Fakultas Teknologi Informasi,

Institut Teknologi Sepuluh Nopember

Email: ¹ fetty10@mhs.if.its.ac.id, ² nanik@if.its.ac.id, ³ anny@if.its.ac.id

Abstrak – Strip packing problem (SPP) merupakan permasalahan peletakan sekumpulan objek ke dalam sebuah kontainer persegi dengan panjang minimum. Objek dapat berbentuk regular (persegi, lingkaran, segitiga, dsb) dan irregular (poligon), sedangkan kontainer berbentuk persegi dengan lebar tetap dan panjang tak hingga. Dalam penelitian ini mengusulkan pengelompokan polygon berdasarkan convex hull dan bounding box untuk menggabungkan beberapa polygon menjadi sebuah polygon baru yang lebih besar. Uji coba menggunakan dataset DAGLI, DIGHE1, FU, JAKOBS2, MAO dan MARQUES menunjukkan bahwa pengelompokan berdasarkan parameter convex hull dan bounding box dapat mengurangi jumlah poligon dengan rata-rata 37%.

Kata kunci: 2D Irregular Strip Packing Problem, Pengelompokan Polygon, Convex Hull, Bounding Box.

1. PENDAHULUAN

Strip Packing Problem (SPP) merupakan permasalahan peletakan sekumpulan objek ke dalam sebuah kontainer persegi. Objek dapat berbentuk regular (persegi, lingkaran, segitiga, dsb) dan *irregular* (poligon), sedangkan kontainer umumnya memiliki lebar tetap dan panjang tertentu. Hal yang penting dalam kasus SPP adalah tidak boleh terjadi tumpang tindih antar objek dalam kontainer. Metode untuk mendeteksi terjadinya tumpang tindih antara lain metode piksel/raster, fungsi-*D*, *No Fit Polygon* (NFP), dan fungsi-phi (Bennel dkk, 2008). Poligon yang diolah dapat berbentuk convex maupun non-convex. Kerumitan pembentukan NFP muncul jika salah satu atau kedua poligon berbentuk non-convex (Bennel dkk, 2012). Seperti halnya ketika kita ingin mengepak barang, barang berukuran kecil akan ditempatkan pada satu kotak agar mudah dibawa dan ditata. Bennel dkk (2012) menggunakan parameter *convex hull* dan *bounding box* sebagai dasar pengepakan poligon secara langsung ke dalam kontainer. Dalam penelitian ini kedua parameter tersebut digunakan sebagai dasar penggabungan poligon dengan bentuk poligon baru yang lebih padat dan mendekati bentuk

persegi. Seperti halnya dalam pengolahan citra digital, sebelum sebuah citra siap untuk diproses inti harus dilakukan *preprocessing* seperti pengurangan noise, deteksi tepi, perbaikan kontras dan sebagainya. Pengelompokan poligon dalam penelitian ini juga berfungsi sebagai *preprocessing* sebelum penataan di kontainer.

Penelitian ini membahas pengelompokan poligon berdasarkan parameter *convex hull* dan luasan persegi. Dengan menggabungkan beberapa poligon menjadi poligon yang lebih besar diharapkan dapat meningkatkan efisiensi kontainer.

2. METODOLOGI PENELITIAN

Strip packing problem (SPP) merupakan salah satu topik dalam *Cutting Stock Problem* (CSP) dimana dalam proses pema hanya memperhatikan variable panjang dari kontainer, sedangkan lebar kontainer adalah tetap (Kendall, 2000). Dalam *irregular SPP* terdapat poligon $\mathcal{P} = (P_1, \dots, P_n)$, r_i menyatakan himpunan orientasi untuk poligon P_i . Dalam penelitian ini batasan orientasi yang digunakan adalah 180° untuk semua poligon, sehingga

$r_1 = \dots = r_n = \{0, 180\}$. Kontainer C

berbentuk persegi dengan ukuran lebar W dan panjang L ($C(W,L)$). Penggabungan poligon didasarkan pada Minkowski sum (penjelasan lebih lanjut lihat subbab 2.4). Jika vektor translasi poligon P_i adalah $v_i = (v_{i1}, v_{i2})$, maka poligon hasil translasi $P_i \oplus v_i = \{p + v_i | p \in P_i\}$. Irregular SPP secara formal dirumuskan sebagai berikut:

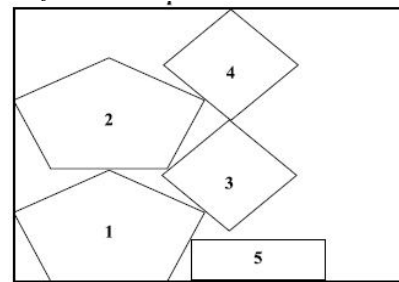
$$\begin{aligned} & \text{minimize} && L && (1) \\ & \text{subject to} && (P_i(\eta_i) \oplus v_i) \cap (P_j(\eta_j) \oplus v_j) = \emptyset, \\ & && (P_i(\eta_i) \oplus v_i) \subseteq C(W, L) \\ & && L \in \mathbb{R}_+, \eta_i \in \mathcal{O}, x_i \in \mathbb{R}^2, 1 \leq i, j \leq n. \end{aligned}$$

2.1 Penyediaan Dataset

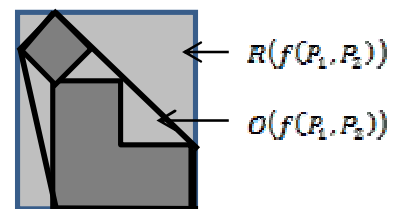
Dataset yang digunakan untuk uji coba adalah dataset dalam permasalahan *cutting and packing problem*. Dataset dapat diperoleh dari ESICUP (*Euro SpecialInterest Group on Cutting and Packing*), lihat Tabel 1. Setiap dataset disimpan dalam format *.xls dan/atau *.xml. Dataset mencakup lebar poligon, jumlah tipe poligon, jumlah poligon tiap tipe dan titik-titik pembentuk poligon. Dataset merupakan data peneliti yang tergabung dalam ESICUP. Adapun rotasi setiap dataset adalah variasi dataset yang digunakan oleh peneliti. Contoh isi dataset dapat dilihat pada Tabel 1, dataset Fu terdiri dari 12 poligon dengan 11 tipe poligon (poligon no.1 dan no.2 adalah sama), rata-rata jumlah vertex poligon 3,58, lebar kontainer 38, dan rotasi yang digunakan adalah 0, 90, 180, dan 270. Tetapi pada prinsipnya rotasi yang digunakan dapat disesuaikan dengan tujuan. Misal jika container adalah gulungan kain dan poligon adalah pola potongan baju, maka rotasi yang digunakan adalah 180. Karena dalam industri garmen untuk menghasilkan produk yang baik tidak boleh melawan serat kain.

2.2 Peletakan Poligon dengan Strategi Bottom-Left

Strategi peletakan *bottom-left* merupakan pendekatan sederhana dari 2D CPP untuk meminimalkan panjang kontainer yang dibutuhkan dengan meletakkan potongan di paling kiri-bawah (Downsland dkk, 2002). Gambar 1 menunjukkan cara kerja peletakan *bottom-left*, nampak bahwa sebelum poligon diletakkan ke dalam kontainer, poligon diurutkan berdasarkan luas poligon mulai luasan terbesar sampai terkecil (*descending*). Setelah pengurutan, poligon diletakkan satu persatu ke dalam kontainer dengan aturan prioritas peletakan paling bawah dan paling kiri, dengan tetap memperhatikan tumpang tindih. Poligon 3 tidak bisa diletakkan di atas poligon 2 karena ruang yang tersisa tidak bisa menampung poligon 3. Sehingga diletakkan di lokasi lain dengan tetap menggunakan konsep *bottom-left*.



Gambar 1. Strategi Peletakan *Bottom-left*



Gambar 2. Sisa material dalam area *convex hull* dan luasan persegi.

Tabel 1. Informasi Dataset

Dataset	Jumlah Tipe Poligon	Jumlah Poligon	Rata-rata vertex	Lebar Kontainer	Rotasi
Dagli	10	30	6,30	60	0, 180
Dighel	16	16	3,87	100	0
Fu	11	12	3,58	38	0, 90, 180, 270
Jakobs2	23	25	5,35	70	0, 90, 180, 270
Mao	9	20	9,22	2550	0, 90, 180, 270
Marques	8	24	7,37	104	0, 90, 180, 270

Sumber: ESICUP

2.3 Pengelompokan poligon

Dua poligon P_1 dan P_2 , *convex hull* dan *bounding box* penggabungan $f(P_1, P_2)$ dinotasikan sebagai $O(f(P_1, P_2))$ and $R(f(P_1, P_2))$, lihat Gambar 1. Rumusan berikut digunakan untuk mengukur sisa material dalam area *convex hull* dan luasan persegi:

$$U_{COV}^{f(P_1, P_2)} = 1 - \frac{\text{Area}(O(f(P_1, P_2))) - \text{Area}(P_1) - \text{Area}(P_2)}{\text{Area}(R(f(P_1, P_2)))}$$

(2)

$$U_{REC}^{f(P_1, P_2)} = 1 - \frac{\text{Area}(R(f(P_1, P_2))) - \text{Area}(O(f(P_1, P_2)))}{\text{Area}(R(f(P_1, P_2)))}$$

Untuk mendapatkan hasil penggabungan poligon, harus ditetapkan posisi terbaik suatu poligon terhadap poligon yang lain. Penggabungan yang baik adalah yang tidak menghasilkan sisa material yang tidak perlu, hal ini dapat tercapai dengan mencari posisi dimana kedua poligon saling bersentuhan menggunakan NFP. Jika ingin meminimumkan sisa material, penggabungan dengan minimum *convex hull* adalah pilihan terbaik. Tetapi jika penampung/kontainer berbentuk persegi, maka minimum *bounding box* juga harus dipertimbangkan. $U_{COV}^{f(P_1, P_2)}$ mengukur sisa material dalam *convex hull*, jika bernilai kecil menghasilkan penggabungan padat. $U_{REC}^{f(P_1, P_2)}$ mengukur sisa material antara *bounding box* dengan *convex hull*, jika bernilai kecil maka penggabungan mendekati bentuk persegi. Sehingga rumusan perhitungan utilitas total ($U_W^{f(P_1, P_2)}$) adalah:

$$U_W^{f(P_1, P_2)} = wU_{REC}^{f(P_1, P_2)} + (1 - w)U_{COV}^{f(P_1, P_2)} \quad (3)$$

Semakin besar nilai bobot (w), maka utilitas *bounding box* (U_{REC}) memiliki prioritas lebih besar daripada utilitas *convex hull* (U_{COV}). Sedangkan nilai θ dijadikan batasan pengelompokan, apakah suatu poligon dapat digabungkan dengan poligon dalam kluster yang sudah terbentuk.

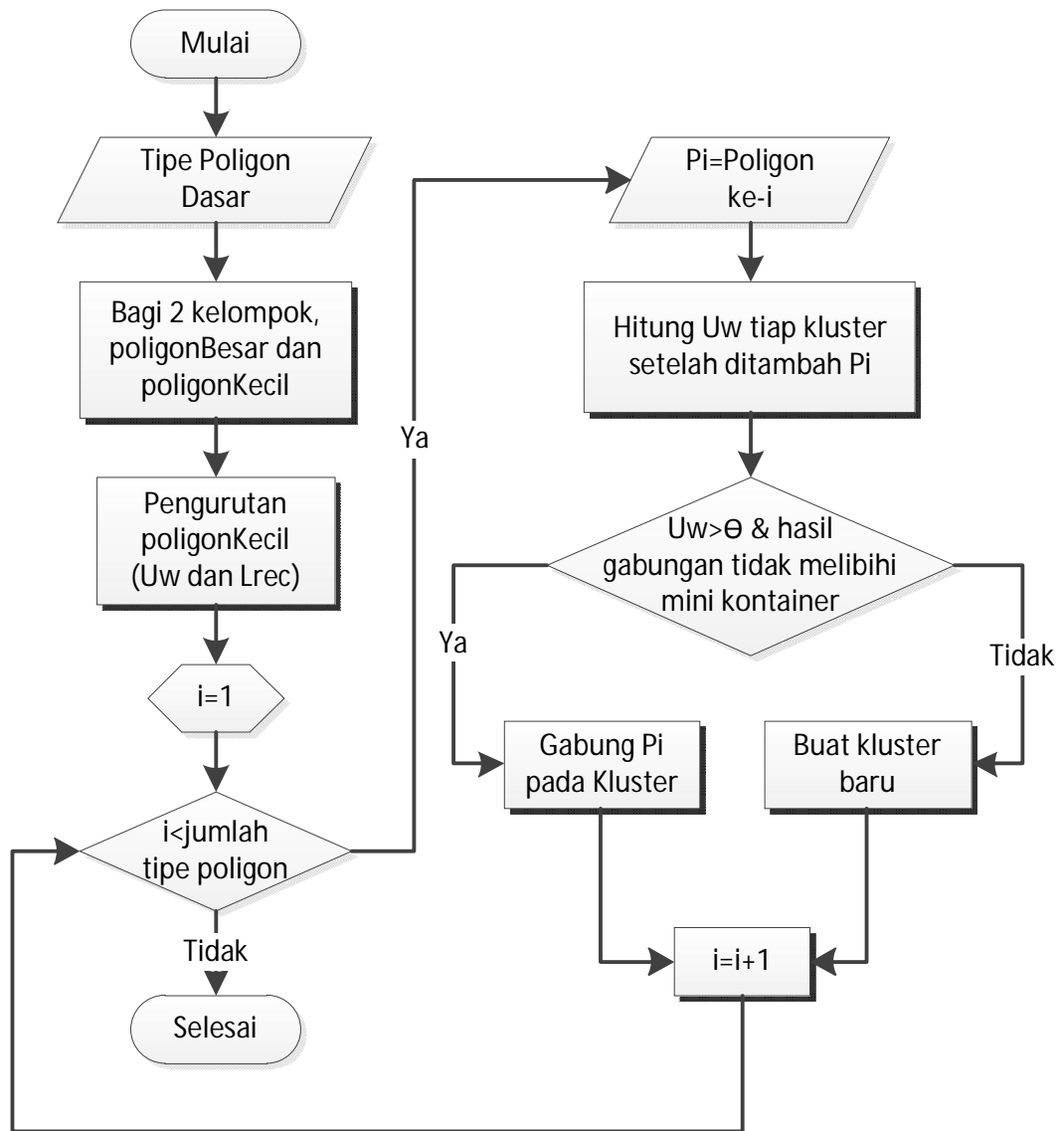
Hasil penggabungan poligon tidak boleh melebihi mini kontainer berukuran setengah dari lebar kontainer ($\frac{1}{2}W$). Hal ini dilakukan agar ruang

kosong yang terbentuk saat peletakan dalam kontainer tidak terlalu luas. Batasan ukuran persegi kelompok poligon dipilih tidak terlalu besar dan tidak terlalu kecil, hal ini agar penataan dalam kontainer lebih mudah. Poligon dibagi menjadi 2 grup, yaitu poligon besar dan poligon kecil, berdasarkan ukuran sisi luasan persegi. Pengelompokan hanya dilakukan pada poligon dengan luasan persegi kecil, sehingga dapat mengurangi total jumlah poligon. Parameter *convex hull* digunakan agar setiap kelompok poligon memiliki ruang sisa yang minimal. Jika suatu poligon akan digabungkan dalam satu kelompok, maka dilakukan perhitungan *convex hull* untuk menentukan posisi pengelompokan dan dilakukan pengecekan luasan persegi dari kelompok poligon. Jika ternyata hasil pengelompokan melebihi batas luasan persegi dan lebih kecil dari threshold (θ), maka dilakukan pengecekan terhadap kelompok poligon lain. Jika tidak ada satupun kelompok poligon yang bisa menampung, maka dibuatkan kelompok poligon baru. Proses pengelompokan poligon dapat dilihat pada Gambar 3.

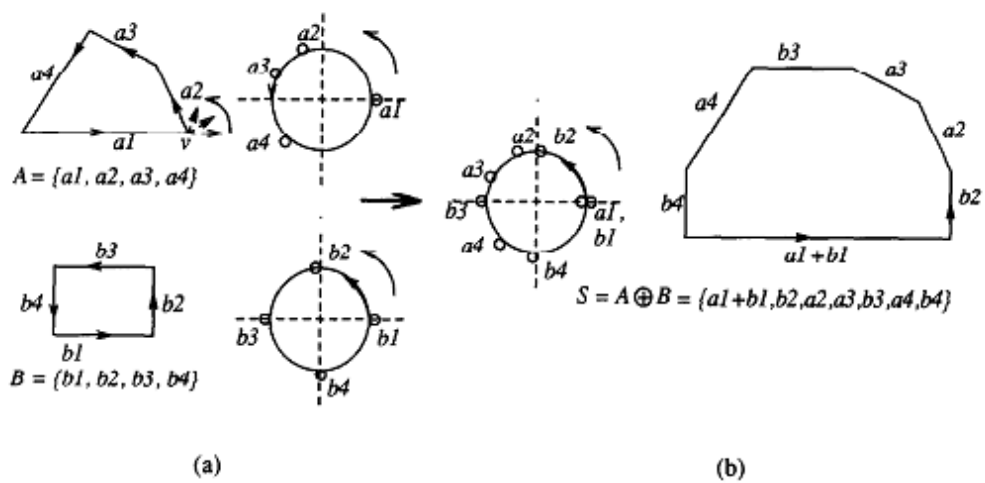
2.4 No-fit Poligon (NFP)

Dalam permasalahan SPP terdapat lebih dari satu potongan bentuk, baik regular maupun *irregular*, yang harus disusun dalam 1 (satu) kontainer dengan tujuan optimasi. Ketika menyusun potongan bentuk dapat menyebabkan kondisi tumpang tindih, menyentuh, atau terpisah.

Ghosh (1991) mengembangkan teorema penambahan *boundary* untuk poligon konveks dan non-konveks. Dasar teorema menggunakan diagram *slope* untuk merepresentasikan *Minkowski sum*. Gambar 4 menjelaskan teorema yang diusulkan Ghosh untuk membangun *Minkowski sum*. Setiap poligon diberi label berlawanan arah jarum jam, kemudian didata kemiringan *edge*, direpresentasikan ke dalam diagram *slope*. Jika dua poligon berbentuk *convex* maka diagram *slope* kedua poligon diurutkan sehingga menghasilkan *Minkowski sum* (Gambar 4 (b)). Tetapi jika salah satu dari berbentuk *non-convex* maka perlu dilakukan proses penelusuran lebih lanjut. Algoritma pembentukan NFP lebih detil dapat dilihat pada algoritma Minkowski dan Minkpos.



Gambar 3. Diagram Alur Pengelompokan Poligon



Gambar 4. Algoritma Penambahan Boundary, (a) Dua Poligon Konveks dan Diagram Slope, (b) Hasil Penggabungan Diagram Slope.

```

Begin Minkowski(A,B)
Ubah B menjadi -B
Mulai dari titik paling bawah, beri
    label edge berlawanan arah
jarum jam
    Cek apakah A dan -B memiliki
    turning point, jika ada maka
    poligon convex
    If A dan -B convex then
        [S,tS] = MergeSlope(A, -B)
    If A convex dan -B non-convex then
        [S,tS] = Minkpos(-B,A)
    If A non-convex dan -B convex then
        [S,tS] = Minkpos(A,-B)
    If A dan -B non-convex then
        convB=convhull(-B)
        [S,tS] = Minkpos(A,convB)
    End If
End

Begin Minkpos(A,B)
Langkah 1. Buat MS = MergeSlope(A,B).
Langkah 2. Set i=1, k=1, arah=1,
S(k)=A(i),
    tanda(k)=1, j=posisi 'a1' pada
    MS; i=i+1;
Langkah 3. While MS(i) ≠ 'a1'
    If arah=1 then j = j+1;
    Else j = j-1;
    r = MS(j);
    If r ∈ B then
        k=k+1; S(k)=r;
tanda(k)=arah;
    If r ∈ A then
        k=k+1; S(k)=r;
tanda(k)=1;
    If r turning point A then
        arah=-1*arah;
    i=i+1;
    End while.
Langkah 4. Set j=2, next=2,
direction=1,
SEQ(j)=B(1), SEQ(1)=edge A
sebelum B(1) pada S, i=indeks
B(1) di S, i=i+1.
While semua edge A dan B belum
digunakan
r=S(i);
If r ∈ A then
    j=j+1; SEQ(j)=r;
    tSEQ(j)=tanda(i);
    If r turning point A then
arah=-1*arah; next=next+arah;
    If r ∈ B then
j=j+1; SEQ(j)=r;

```

```

tSEQ(j)=tanda(i);
next=next+arah;
i=i+1;
End while.
End

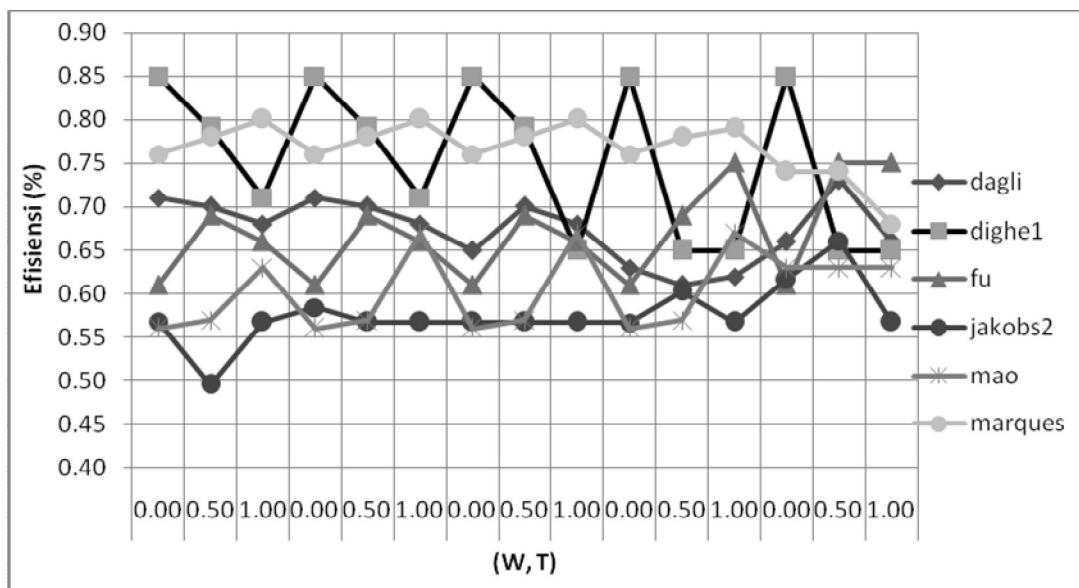
```

3. HASIL DAN PEMBAHASAN

Uji coba bertujuan untuk mencari kombinasi nilai bobot (w) dan threshold (θ) terbaik dalam menghasilkan poligon baru hasil pengelompokan. Untuk mengukur kualitas pengelompokan digunakan efisiensi kontainer pada tahapan inialisasi. Semakin besar efisiensi semakin padat penggunaan kontainer.

Gambar 5 menunjukkan bahwa setiap dataset memiliki efisiensi terbaik pada variasi w dan θ berbeda-beda. Dari hasil uji coba diperoleh rata-rata efisiensi untuk setiap bobot adalah 67,38%, 67,62%, dan 67,16%. Dan rata-rata efisiensi untuk setiap threshold adalah 67,4%, 68,11%, 67,35%, 66,27%, 67,81%. Berdasarkan hasil perhitungan rata-rata efisiensi menunjukkan nilai parameter terbaik adalah $w=0,5$ dan $\theta=0,8$.

Hasil pengelompokan dengan menggunakan nilai bobot (w) 0,5 dan threshold (θ) 0,8 ditampilkan pada Gambar 4, dapat disimpulkan bahwa pengelompokan berdasarkan *convex hull* dan *bounding box* dapat mengurangi jumlah poligon dengan rata-rata persentase pengurangan jumlah poligon sebesar 37%. Waktu pengelompokan sangat dipengaruhi oleh jumlah poligon kecil, yaitu poligon yang dapat masuk ke dalam mini kontainer dan diproses dalam pengelompokan poligon. Dataset JAKOBS2 membutuhkan waktu pengelompokan paling lama, hal ini dikarenakan dataset tersebut memiliki jumlah jenis poligon paling banyak dibanding yang lain. Dataset MAO dan MARQUES meskipun memiliki jumlah jenis poligon lebih sedikit daripada FU dan DIGHE1, tetapi MAO dan MARQUES memiliki poligon yang berbentuk convex (lihat Tabel 2). Sehingga dapat ditarik kesimpulan bahwa waktu pengelompokan sangat dipengaruhi jumlah poligon kecil dan bentuk poligon, convex atau non-convex. Kemudian hasil pengelompokan poligon diletakkan ke dalam kontainer menggunakan algoritma *bottom-left*, dan menghasilkan efisiensi rata-raya sebesar 68,43% (Tabel 3).



Gambar 4. Efisiensi kontainer menggunakan variasi nilai w dan θ .

Tabel 2. Hasil pengelompokan dengan $w=0,5$ dan $\theta=0,8$.

Dataset	\sum jenis poligon dasar	\sum total poligon	\sum jenis poli convex	\sum jenis poli non-convex	Jum poli kecil	\sum cluster dasar	\sum total cluster	% Pengurangan jumlah poligon	Rata-rata jumlah poligon dalam 1 kluster	Waktu komputasi (s)
Dagli	10	30	7	3	9	5	15	50	2	85
Dighe1	16	16	15	1	13	9	11	31,25	1,4545	110,4475
Fu	12	12	11	0	11	7	8	33,33	1,5	77,348
Jakobs2	23	25	14	9	23	10	11	56	2,2724	2066
Mao	9	20	3	6	9	8	18	10	1,1111	321,2
Marques	8	24	3	5	8	6	14	41,67	1,7143	325
Rata-rata % pengurangan jumlah poligon								37		

Tabel 3. Efisiensi dan panjang kontainer dengan parameter pengelompokan $w=0,5$ dan $\theta=0,8$.

Dataset	Panjang layout	% Efisiensi layout
dagli	72	70,44
dighe1	126,4	79,11
fu	41	69,34
jakobs2	34	56,76
mao	2576	57,22
marques	89	77,72
Rata-rata		68,43

4. KESIMPULAN

Pengelompokan berdasarkan *convex hull* dan *bounding box* mampu mengurangi jumlah poligon sebesar 37%, tetapi bentuk poligon gabungan yang dihasilkan lebih kompleks. Peletakan kelompok poligon ke dalam kontainer mencapai efisiensi rata-rata sebesar 68,43%.

Kelompok poligon yang dihasilkan dari proses pengelompokan mayoritas berbentuk *non-convex* (Gambar 5). Proses pembentukan NFP jika salah satu/dua poligon berbentuk *non-convex* memiliki kompleksitas algoritma lebih tinggi daripada NFP poligon *convex*. Selain itu, NFP poligon *nonconvex* memiliki

bentuk yang lebih kompleks dan tersusun atas banyak *vertex*. Dengan semakin banyaknya titik pembentuk NFP, maka waktu yang dibutuhkan untuk mencari posisi yang tepat bagi setiap poligon lebih lama. Rendahnya efisiensi disebabkan oleh proses peletakan kelompok poligon. Peletakan kelompok poligon ke dalam kontainer menggunakan algoritma *bottom-left* tidak cukup untuk menghasilkan *layout* dengan kepadatan maksimum (Gambar 6). *Bottom-left* fokus pada meletakkan kelompok poligon ke dalam kontainer dengan posisi paling kiri dan paling bawah, tidak memperhatikan apakah posisi tersebut meningkatkan efisiensi atau bahkan menurunkan efisiensi kontainer.

Untuk meningkatkan efisiensi kontainer dapat dilakukan beberapa pengembangan dari metode yang diusulkan antara lain memperbaiki proses pengelompokan poligon agar menghasilkan kelompok poligon dengan kualitas lebih baik, misal dengan mempertimbangkan kecocokan sisi poligon. Selain itu proses peletakan kelompok poligon menggunakan konsep *bottom-left* tidak mampu menghasilkan pengepakan yang padat. Dengan menambahkan parameter utilitas *bounding box*

kontainer pada tahap peletakan ke dalam kontainer diharapkan dapat meningkatkan kepadatan pengepakan.

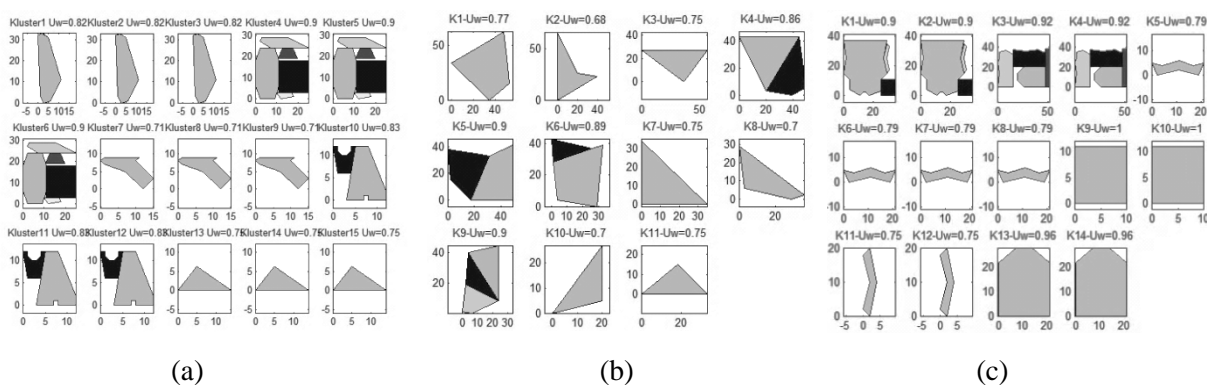
5. PUSTAKA

Bennel, J.A., Oliveira J.F (2008), “The geometry of nesting problem: A tutorial”, *European Journal of Operational Research*, Vol. 184, Hal. 397-415.

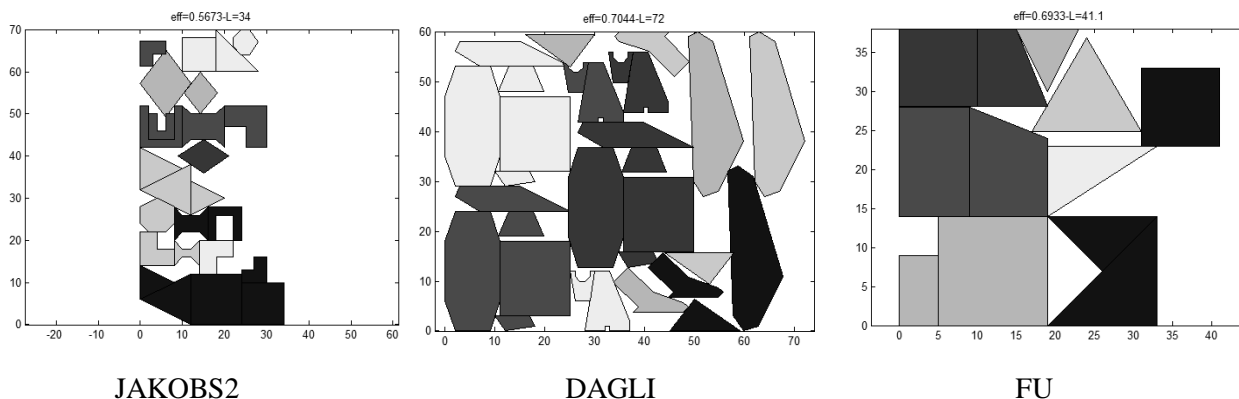
Bennell, J.A., Han, W., Zhao, X. and Song, X.(2012) “Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints”, *University of Southampton Institutional Research Repository*, Southampton, GB, University of Southampton.

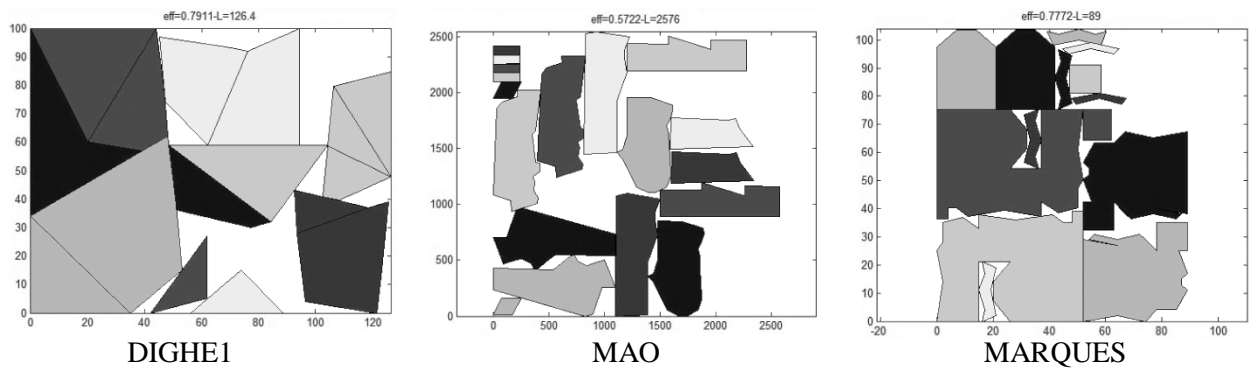
Dowland, K.A., Vaid, S., dan Dowland, W.B. (2002), “An algorithm for polygon placement using a bottom-left strategy”, *European Journal of Operational Research*, Vol. 141, Hal.371–381.

Ghosh, P.K. (1991), “An algebra of polygons through the notion of negative shapes”, *CVGIP: Image Understanding*, Vol. 54 No. 1, Hal. 119–144.



Gambar 5. Hasil pengelompokan poligon dengan parameter pengelompokan $w=0,5$ dan $\theta=0,8$, (a) dataset DAGLI, (b) dataset DIGHE1, dan (c) dataset MAO.





Gambar 6. Hasil peletakan kelompok poligon menggunakan algoritma bottom-left dengan parameter pengelompokan $w=0,5$ dan $\theta=0,8$.