

Optimasi Strategi Hunt/Target pada Permainan Papan Battleship dengan Metode Greedy

Muhammad Akmal Pratama - 13515135

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

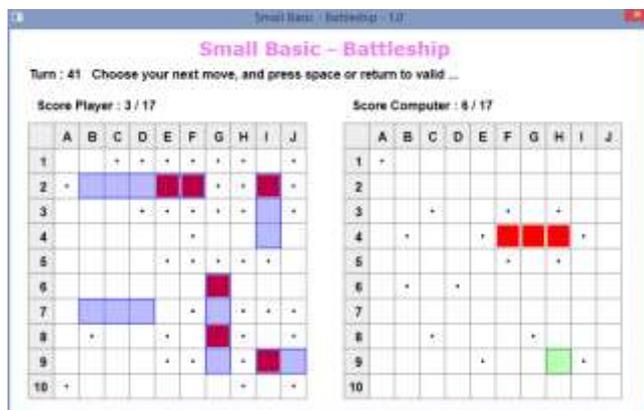
m.akmalpratama@students.itb.ac.id

Abstract—Dalam kondisi tertentu, kita kerap kali dihadapkan dengan persoalan yang menuntut kita untuk mengambil keputusan. Namun, adakalanya keputusan yang kita ambil berdampak krusial sehingga setiap langkah yang hendak diambil ada baiknya dievaluasi dengan cermat agar keputusan tersebut menuntun kita ke hasil yang diharapkan. Hal itu juga berlaku dalam papan permainan Battleship. Dalam makalah ini, fokus akan dibahas mengenai strategi permainan Battleship dalam pemilihan lokasi tembak target kapal melalui pendekatan greedy, mengingat semua batasan yang ada, dengan harapan solusi optimum global dapat diraih dengan mengambil solusi minimum lokal.

Keywords—*Battleship; greedy; papan permainan; strategi; Hunt/Target*

I. PENDAHULUAN

Battleship merupakan permainan papan yang dimainkan sebanyak dua orang. Permainan ini diduga berasal dari permainan asal Perancis L'Attaque yang dimainkan selama Perang Dunia I. Pada saat itu, permainan ini dimainkan menggunakan kertas dan pensil. Kemudian barulah pada tahun 1967, Milton Bradley membuat mainan versi plastiknya. Semenjak saat itu, permainan Battleship terus berkembang dan sudah diadaptasi dalam bentuk video game.



Gambar 1.1 Battleship versi digital

(sumber: <https://gallery.technet.microsoft.com/Small-Basic-Game-Battleship-4760e6f3>)

Permainan Battleship ini sendiri memiliki banyak variasi. Dalam makalah ini, pembahasan dikerucutkan hanya pada versi

Milton Bradley 1990. Permainan terbagi menjadi dua fase, yaitu fase persiapan dan fase penyerangan. Saat fase persiapan, tiap pemain menempatkan kapalnya secara rahasia dan mengatur penempatan sedemikian rupa sehingga posisi kapal menjadi sulit ditebak. Saat fase penyerangan, pemain secara bergantian menyerang daerah lawan sampai salah satu diantara kedua pemain kehilangan semua kapalnya.

Pada awal permainan, masing-masing pemain menyiapkan selembar kertas yang berisi dua buah matriks berukuran 10x10. Matriks pertama berisi posisi kapal-kapal anda. Tiap kapal dapat ditempatkan secara vertikal ataupun horizontal tapi tidak diagonal. Kapal dapat ditempatkan bersebelahan tetapi tidak tumpang tindih. Setiap kapal memiliki ukuran tertentu sesuai dengan sel pada matriks. Berikut jenis kapal beserta ukurannya

Tipe	Ukuran
Aircraft Carrier	5x1
Battleship	4x1
Submarine	3x1
Cruiser	3x1
Destroyer	2x1

Gambar 1.2 Spesifikasi kapal versi Milton Bradley

Setelah semua kapal selesai diletakkan, selanjutnya adalah fase penyerangan. Urutan pemain mana yang mulai terlebih dahulu ditentukan secara acak. Kemudian pemain, secara bergiliran, menyebutkan koordinat mana ia ingin menembakkan pelurunya. Koordinat peluru yang telah ditembakkan dicatat di matriks pertama dan kedua dengan diberi tanda silang. Apabila lawan berhasil menebak koordinat yang menjadi posisi kapal pemain, maka pemain tersebut menyebutkan "Hit". Jika tidak, maka katakan "Miss". Untuk kapal yang berhasil tertembak dapat dibedakan tandanya, dengan tanda silang berwarna merah misalnya.

Tujuan dari permainan ini adalah anda diminta untuk menenggelamkan semua kapal lawan anda sebelum kapal anda ditenggelamkan terlebih dahulu. Permainan berakhir ketika semua kapal anda tenggelam atau kapal lawan anda tenggelam.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma greedy merupakan metode yang umum digunakan untuk persoalan optimasi. Greedy dapat diartikan tamak, rakus, loba. Prinsip dari algoritma greedy, yaitu mengambil sebanyak mungkin apa yang bisa diambil saat ini[1].

Cara kerja algoritma greedy adalah dengan memecahkan masalah langkah demi langkah. Pada setiap langkah, kita membuat pilihan optimum lokal dengan harapan bahwa langkah selanjutnya akan mengarah ke solusi optimum global. Perlu.

Algoritma greedy disusun oleh elemen-elemen berikut:

1. Himpunan kandidat (C)
Berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi (S)
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi (*selection function*)
Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi kelayakan (*feasible*)
Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi obyektif
Fungsi obyektif, yaitu fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Dengan kata lain, algoritma greedy melibatkan pencarian sebuah himpunan bagian S, dari himpunan kandidat C. Dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi obyektif.

Skema umum algoritma Greedy

```
function greedy(input C: himpunan_kandidat) →  
  himpunan_kandidat  
{ Mengembalikan solusi dari persoalan optimasi  
  dengan algoritma greedy  
  Masukan: himpunan kandidat C  
  Keluaran: himpunan solusi yang bertipe  
  himpunan_kandidat  
}  
Deklarasi  
  x : kandidat  
  S : himpunan_kandidat  
  
Algoritma:  
  S ← {} { inisialisasi S dengan kosong }
```

```
while (not SOLUSI(S)) and (C ≠ {} ) do  
  x ← SELEKSI(C) { pilih sebuah  
  kandidat dari C}  
  C ← C - {x} { elemen himpunan  
  kandidat berkurang satu }  
  if LAYAK(S ∪ {x}) then  
    S ← S ∪ {x}  
  endif  
endwhile  
{ SOLUSI(S) or C = {} }  
  
if SOLUSI(S) then  
  return S  
else  
  write('tidak ada solusi')  
endif
```

Perlu diingat bahwa solusi yang diselesaikan tidak menjamin selalu menghasilkan solusi optimum global, tetapi suboptimum atau pseudooptimum. Alasannya adalah algoritma greedy tidak mengevaluasi secara menyeluruh terhadap semua alternatif solusi yang ada sebagaimana pada pencarian exhaustive search. Selain itu juga, terdapat beberapa fungsi seleksi yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin menghasilkan algoritma yang optimal. Jadi pada sebagian masalah, algoritma greedy tidak selalu berhasil memberikan solusi yang optimal.

B. Paritas

Paritas adalah istilah matematika yang menjelaskan tentang sifat-sifat inklusif bilangan bulat dalam satu dari dua kategori, yaitu ganjil dan genap. Sebuah bilangan bulat dikatakan genap apabila terbagi sama besar oleh 2 dan ganjil jika tidak terbagi sama besar[2]. Sebagai contoh bilangan genap -4, 0, dan 8. Khususnya 0 merupakan bilangan genap[3]. Contoh lain bilangan ganjil adalah -5, 3, 9, 73.

Definisi formal dari bilangan genap adalah bilangan bulat dalam bentuk $n = 2k$, dimana k adalah bilangan bulat, [4] kemudian dapat pula disimpulkan bahwa bilangan ganjil adalah bilangan bulat dalam bentuk $n = 2k + 1$.

Himpunan dari bilangan genap dan ganjil dapat didefinisikan sebagai berikut :

- $Genap = \{2k : k \in \mathbb{Z}\}$
- $Ganjil = \{2k + 1 : k \in \mathbb{Z}\}$

Hukum berikut dapat diuji dengan sifat keterbagian. Hukum tersebut adalah kasus khusus dari aturan dalam aritmatika modular, dan umum digunakan untuk mengecek jika persamaan sepertinya benar dengan mengecek paritas dari kedua sisi persamaan.

Penjumlahan dan pengurangan

- $genap \pm genap = genap$
- $ganjil \pm genap = ganjil$
- $ganjil \pm ganjil = genap$

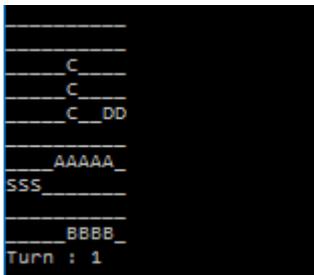
III. IMPLEMENTASI

Dalam fase penyerangan, masing-masing pemain secara bergiliran melakukan penembakan ke daerah lawannya. Terdapat berbagai strategi yang dapat dilakukan untuk menghancurkan kapal, diantaranya yang paling terkenal adalah strategi *Hunt/Target*. Namun, sebelum pembahasan dikhususkan ke arah tersebut, perlu diketahui juga beberapa strategi umum yang menjadi dasar untuk perkembangan bagi strategi yang lebih lanjut.

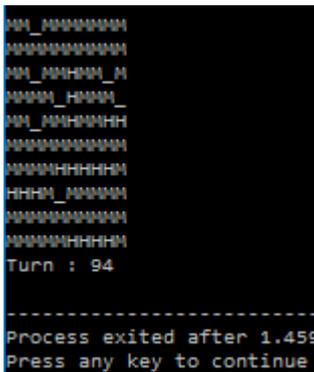
A. Strategi Acak

Strategi ini merupakan metode yang naif digunakan untuk menebak lokasi sel kapal. Pemilihan secara acak didasari atas ketidaktahuan pemain akan posisi kapal lawan. Strategi ini umumnya digunakan oleh para pemain yang belum paham konsep dari peletakan kapal.

Pemilihan sel secara acak merupakan metode yang sangat buruk. Kemungkinan terburuknya adalah apabila posisi sel terakhir kapal yang masih ada dipilih terakhir. Hal ini berarti dibutuhkan paling tidak seratus kali tembakan/giliran untuk menghancurkan semua kapal atau dengan kata lain membombardir semua area yang ada di arena.



Gambar 3.1 Keadaan awal permainan



Gambar 3.2 Keadaan akhir permainan dengan Strategi Acak

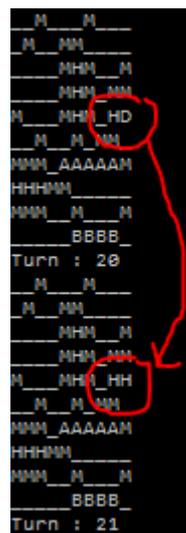
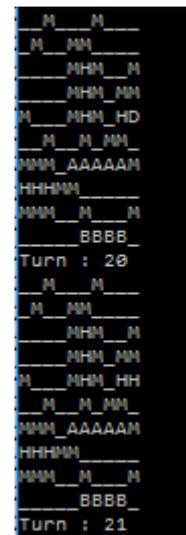
Dapat dilihat bahwa setidaknya hampir dibutuhkan lebih dari 90 tembakan untuk menenggelamkan semua kapal yang ada dengan strategi secara acak. Oleh sebab itu, strategi ini sangat buruk untuk digunakan dalam permainan Battleship.

B. Strategi Hunt/Target

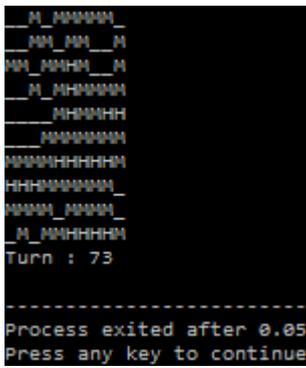
Strategi yang sedikit lebih baik lagi adalah *Hunt/Target*. Pada awalnya, pencarian kandidat kapal dilakukan seperti pada Strategi Acak. Pemilihan ini disebut Hunt mode. Apabila tembakan mengenai salah satu bagian kapal, maka mode pemilihan berganti Target mode.

Dalam Target mode, kandidat sel untuk ditembak berikutnya adalah sel bagian atas, bawah, kanan, dan kiri. Kandidat sel ini disimpan dalam stack kemudian untuk setiap sel yang kena berikutnya dilakukan hal yang sama.

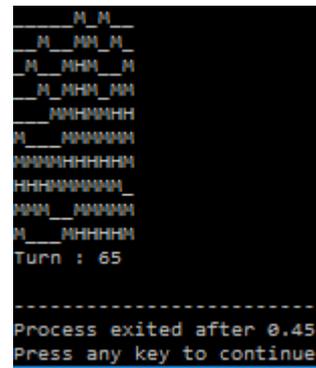
Hal ini didasari atas struktur kapal yang berupa sel $n \times 1$ sehingga apabila salah satu bagian kapal telah tertembak maka cukup dicek ke empat arah tersebut dan penembakan akan dijamin kena.



Gambar 3.3 Mode Target mengincar kapal Destroyer begitu salah satu bagian terkena



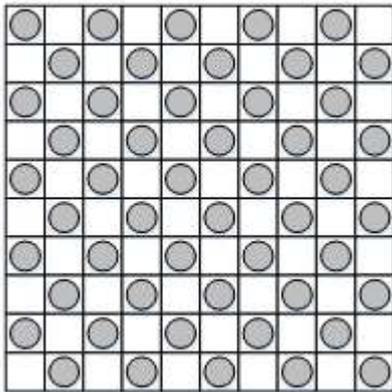
Gambar 3.4 Keadaan akhir permainan dengan *Hunt/Target*



Gambar 3.7 Keadaan akhir permainan dengan *Hunt/Target* Paritas

C. Strategi *Hunt/Target* dengan Paritas

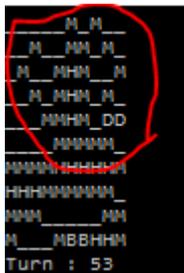
Seperti strategi *Hunt/Target* pada yang sudah dijelaskan sebelumnya, namun dengan perbaikan pada metode Hunt. Pemilihan sel yang menjadi lokasi didasarkan pada prinsip paritas.



Gambar 3.5 Penerapan strategi dengan paritas

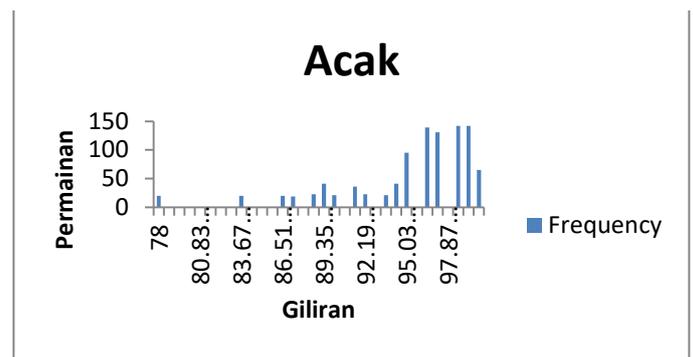
(sumber: <https://frankdecaire.blogspot.co.id/2015/01/game-algorithms.html>)

Setiap sel dalam matriks diklasifikasikan ganjil dan genap secara berselang-seling, seperti halnya papan catur. Pemilihan tetap dilakukan secara acak, namun pilihan hanya dibatasi paritas tertentu saja, yakni sel kategori ganjil atau genap. Selanjutnya, mode target dilakukan seperti biasa.



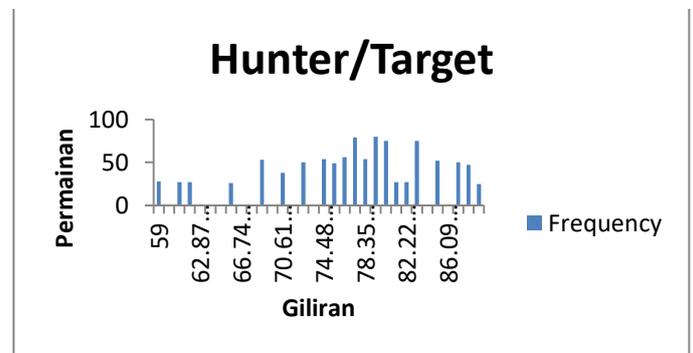
Gambar 3.6 Efek paritas terhadap mode Hunt

IV. ANALISIS



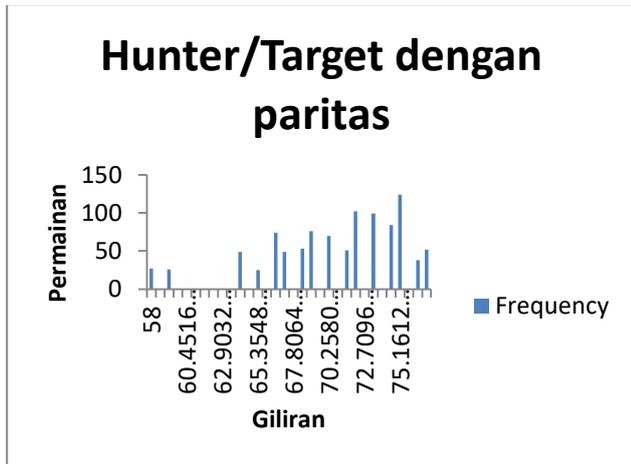
Histogram 1 Jumlah permainan yang diselesaikan dengan dengan banyaknya giliran pada strategi acak

Berdasarkan perbandingan hasil grafik yang telah didapat dari ketiga buah strategi yang telah dibahas. Strategi acak memiliki kemungkinan yang sangat buruk untuk menang dengan dibutuhkan lebih dari 90 giliran untuk memenangkan pertandingan sebanyak 100 kali menurut Histogram 1.



Histogram 2 Jumlah permainan yang diselesaikan dengan dengan banyaknya giliran pada strategi hunter/target

Pada strategi hunter/target, kemungkinan menang jauh lebih baik dengan kebutuhan menang untuk lebih dari 50 kali dibutuhkan giliran 70 sampai 80 kali dan kemenangan terbaik dicapai dengan 59 giliran saja



Histogram 3 Jumlah permainan yang diselesaikan dengan dengan banyaknya giliran pada strategi hunter/target dengan paritas

Strategi hunter/target dengan paritas memiliki peluang yang lebih baik walaupun hanya sedikit dibanding dengan tanpa paritas. Meskipun secara statistik hanya unggul lebih sedikit, namun dalam praktek permainan battleship, hal ini cukup berpengaruh mengingat permainan dilakukan pergiliran sehingga kesalahan sedikit saja dalam memilih kandidat sel ingin diserang dapat memberikan ruang gerak bagi lawan untuk melakukan serangan balik.

V. KESALAHAN UMUM

Kesalahan umum yang sering terjadi adalah pemilihan sel tembak secara acak dan diikuti oleh pemilihan sel secara acak berikutnya meskipun sebagian lokasi kapal sudah diketahui. Hal ini sangat sia-sia, sebab dengan menghancurkan kapal yang sudah diketahui posisinya terlebih dahulu membantu kita mengurangi ruang pencarian sel yang menjadi kandidat posisi kapal. Selain itu juga, apabila sudah diketahui sisa kapal lawan tinggal satu dan sudah diketahui pula posisi sel yang ditempatinya, maka tidak perlu melakukan pencarian secara acak lagi. Pencarian sebaiknya difokuskan kepada sel yang bersisian dengan sel yang sudah diketahui lokasi kapalnya saja.

Kesalahan berikutnya adalah dengan tidak memperhatikan konsep paritas pada sel yang akan menjadi sasaran tembak atau melakukan paritas tetapi secara parsial. Padahal, dengan menerapkan konsep paritas saja dapat meminimumkan ruang cukup banyak.

VI. KESIMPULAN

	Acak	HT	HT-P
Mean	95.001	76.845	70.452
Median	96	78	71.5
Modus	99	79	75

Tabel 1 Rangkuman deskripsi statistik ketiga strategi

Dari tabel 1, sudah dapat terlihat dengan jelas bahwa strategi Hunter/Target dengan paritas memiliki sedikit keunggulan dibanding strategi-strategi lainnya. Ada pun penambahan heuristik memilih sel kategori ganjil/genap membantu iap mengurangi ruang pencarian sebab setiap kapal pasti memiliki bagian pada sel kategori ganjil atau genap. Selain itu mode target yang digunakan memastikan setiap bagian kapal yang tertembak, akan segera “dihabisi” sisa bagian kapal lainnya.

VII. LAMPIRAN

Strategi Acak

Mean	95.001
Standard Error	0.14626138
Median	96
Mode	99
Standard Deviation	4.625190957
Sample Variance	21.39239139
Kurtosis	2.532896621
Skewness	1.596203592
Range	22
Minimum	78
Maximum	100
Sum	95001
Count	1000
Largest(1)	100
Smallest(1)	78
Confidence Level(95.0%)	0.287014771

Tabel 2 Deskriptif statistik strategi acak

Hunter/Target

Mean	76.854
Standard Error	0.241261
Median	78
Mode	79
Standard Deviation	7.629344
Sample Variance	58.20689
Kurtosis	-0.07939

Skewness	-0.63082
Range	30
Minimum	59
Maximum	89
Sum	76854
Count	1000
Largest(1)	89
Smallest(1)	59
Confidence Level(95.0%)	0.473437

Tabel 3 Deskriptif statistik strategi Hunter/Target

<i>Hunter/Target Paritas</i>	
Mean	70.452
Standard Error	0.144563
Median	71.5
Mode	75
Standard Deviation	4.571498
Sample Variance	20.89859
Kurtosis	0.323369
Skewness	-0.83208
Range	19
Minimum	58
Maximum	77
Sum	70452
Count	1000
Largest(1)	77
Smallest(1)	58
Confidence Level(95.0%)	0.283683

Tabel 4 Deskriptif statistik strategi Hunter/Target dengan paritas

UCAPAN TERIMAKASIH

Pada kesempatan ini, penulis ingin mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas segala nikmat yang diberikan sehingga penulis dapat makalah ini. Penulis juga ingin mengucapkan terima kasih kepada dosen mata kuliah Matematika Diskrit Bapak Dr.Ir. Rinaldi Munir, MT. atas bimbingan dan ilmu yang didapat selama kuliah berlangsung. Terakhir, penulis ingin mengucapkan terima kasih kepada semua pihak yang turut berkontribusi dalam pengerjaan makalah ini baik secara langsung maupun tidak langsung.

REFERENCES

- [1] Munir, Rinaldi. 2013. *Strategi Algoritmik*. Bandung:Penerbit ITB
- [2] A.V.Vijaya & Dora Rodriguez, *Figuring Out Mathematics*, Pearson Education India, pp. 20-21, ISBN 9788131703571
- [3] Bóna, Miklós (2011), *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory*, World Scientific, p. 178, ISBN 9789814335232
- [4] Bassarear, Tom (2010), *Mathematics for Elementary School Teachers*, Cengage Learning, p. 198, ISBN 9780840054630.
- [5] <http://www.datagenetics.com/blog/december32011/> diakses 14 Mei 2017

PERNYATAAN

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

Bandung, 19 Mei 2017



ttd
Muhammad Akmal Pratama
13515135