Penerapan Algoritma *Greedy* dalam Pembuatan Klasemen Kompetisi

Muhammad Rizky W. / 13511037
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511037@std.stei.itb.ac.id

***Abstract*— Setiap hari kita berhadapan dengan banyak informasi. Salah satu bentuk informasi yang paling sering dilihat adalah dalam bentuk tabel. Khususnya bagi pecinta olahraga, informasi dalam bentuk tabel pasti sering terlihat dalam bentuk klasemen sebuah liga, contohnya klasemen Liga Primer Inggris, Liga Indonesia, dll. Banyak situs olahraga yang menyediakan *space* agar para pengunjungnya bisa melihat klasemen tersebut. Namun terkadang informasi dalam klasemen tersebut kurang memuaskan, dimana hanya jumlah poin dari setiap tim yang terlihat. Sementara pengunjung juga ingin mengetahui berapa kali sebuah tim menang, imbang atau kalah. Dalam paper ini saya akan mencoba untuk menggunakan algoritma *greedy* untuk menentukan jumlah menang, imbang dan kalah sebut tim dengan hanya mengetahui jumlah pertandingan yang telah tim tersebut lakukan.**

***Index Terms*—klasemen, olahraga, *greedy*.**

I. INTRODUCTION

Informasi merupakan salah satu hal yang paling penting dalam kehidupan manusia sehari – hari. Memasuki masa informasi, informasi telah menjadi bagian yang tidak terpisahkan bagi manusia pada umumnya. Bentuk penyampaian informasi itu sendiri juga beragam. Ada yang berupa esai, laporan, jurnal, gambar, bahkan suara juga bisa mengandung informasi. Namun salah satu cara penyampaian informasi yang paling banyak dipakai adalah dalam bentuk sebuah tabel.

Menurut Kamus Besar Bahasa Indonesia, tabel adalah daftar berisi ikhtisar sejumlah (besar) data, biasanya berupa kata-kata dan bilangan yang tersusun secara bersistem, urut ke bawah dan deret tertentu dengan garis pembatas sehingga dapat dengan mudah disimak. Penggunaan tabel dapat memudahkan pembacanya untuk mendapat informasi yang mereka inginkan dari tabel tersebut. Karenanya, tabel merupakan salah satu cara penyampaian yang paling banyak digunakan. Ada banyak tabel yang digunakan dalam kehidupan sehari – hari, bahkan jadwal pelajaran juga bisa termasuk tabel. Namun banyak tabel yang hanya digunakan untuk pribadi karena informasi yang terkandung di dalamnya hanya berguna bagi beberapa orang termasuk pribadi itu sendiri, karenanya hanya sedikit tabel yang biasa dilihat oleh banyak orang sekaligus dikarenakan informasi yang ada di dalamnya berguna bagi banyak orang. Salah satu tabel yang memenuhi kriteria tersebut adalah tabel klasemen sebuah kompetisi olahraga.

Sebuah klasemen merupakan sebuah bagian penting yang dimiliki oleh hampir semua kompetisi olahraga yang menggunakan sistem liga. Informasi mengenai klasemen dapat membuat banyak pencinta olahraga membeli koran/majalah olahraga hanya untuk melihat dimana posisi tim kesayangannya di klasemen saat ini, berapa jarak timnya dengan tim – tim lain di kompetisi tersebut. Bagi yang tidak ingin mengeluarkan uang juga bisa melihat informasi klasemen tersebut di situs web tertentu. Namun klasemen yang ditampilkan oleh web terkadang tidak terlalu lengkap, sehingga yang ditampilkan hanyalah nama tim, jumlah pertandingan yang telah dilalui dan nilai tim tersebut. Tidak ada detil berapa kali tersebut telah menang, imbang atau kalah. Karenanya, saya ingin mencoba untuk membuat sebuah program dimana program tersebut dapat melengkapi jumlah menang, imbang dan kalah yang telah dilalui oleh sebuah tim dalam sebuah liga, berdasarkan input jumlah pertandingan yang telah dilalui dan poin masing – masing tim. Algoritma yang akan digunakan pada program ini adalah algoritma greedy

II. LANDASAN TEORI

A. Klasemen

Sebuah klasemen terdiri dari beberapa bagian, tergantung dari sistem kompetisi yang diwakili oleh klasemen tersebut. Umumnya, klasemen terdiri dari nama tim/individu yang mengikuti kompetisi, jumlah pertandingan yang telah dilalui oleh masing – masing tim/individu tersebut, dan jumlah poin yang didapat oleh masing – masing tim/individu. Pada beberapa olahraga seperti sepakbola atau bola basket, jumlah pertandingan ditulis karena tidak semua tim telah melalui jumlah pertandingan yang sama dikarenakan perbedaan jadwal masing – masing tim. Namun ada juga klasemen sebuah kompetisi yang tidak menyertakan jumlah pertandingan yang telah dilalui karena semua tim/individu yang terlibat pasti mengikuti semua pertandingan yang telah diagendakan dan

jadwalnya disamakan, contohnya kompetisi *motorsport*. Klasemen juga dapat berbeda dari sistem olahraga tersebut. Misalnya, pada olahraga basket tidak ada pertandingan yang diakhiri dengan hasil imbang, karenanya hanya ada hasil menang dan kalah pada klasemen bola basket, sedangkan pada klasemen sepakbola ada hasil imbang.

B. Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang memcahkan masalah secara langkah per langkah. Pada setiap langkah, algoritma greedy akan mengambil solusi optimum pada langkah tersebut. Secara kasarnya, prinsip dari algoritma *greedy* adalah *"take what you can get now"*. Algoritma ini tidak mempedulikan langkah berikutnya dan hanya mengambil solusi terbaik pada langkah yang sedang dilalui, dengan harapan bahwa dengan mengambil solusi optimal lokal pada setiap langkah maka akan didapat solusi optimal global.

Pada dasarnya, algoritma *greedy* terdiri dari 5 elemen, yaitu:

1. Himpunan kandidat, C, yang berisi elemen – elemen yang bisa menjadi solusi optimum lokal
2. Himpunan solusi, S, berisi elemen yang telah menjadi solusi optimum lokal
3. Fungsi seleksi (selection function), fungsi yang memilih kandidat yang memungkinkan untuk menjadi solusi optimum lokal.
4. Fungsi kelayakan (feasible), fungsi yang memeriksa apakah kandidat yang telah dipilih sebagai solusi optimum lokal bisa menjadi bagian dari solusi optimum maksimum, dimana kandidat yang telah terpilih tidak akan melanggar parameter yang telah disepakati sebelumnya.
5. Fungsi obyektif, fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Dengan kata lain, algoritma greedy adalah algoritma yang mengumpulkan sebuah Himpunan solusi (S), dimana S merupakan himpunan bagian dari Himpunan Kandidat (C), dimana S harus memenuhi beberapa kriteria tertentu. Terkadang solusi yang diberikan oleh algoritma *greedy* bukanlah solusi optimum global dikarenakan algoritma *greedy* tidak memikirkan alternatif solusi selain solusi optimum lokal. Keuntungan penggunaan algoritma ini adalah kecepatan eksekusi yang baik dalam mencari Solusi yang diinginkan.

Berikut ini adalah 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
```

Gambar 1. Skema umum algoritma greedy

III. PEMBAHASAN

Pada klasemen yang menggunakan sistem menang-kalah-imbang seperti sepakbola, jumlah pertandingan minimal yang mereka lalui adalah n-1, dimana n adalah jumlah tim yang mengikuti kompetisi tersebut, sedangkan jumlah maksimal pertandingannya beragam, ada kompetisi yang mengharuskan setiap tim bertemu tim lainnya 2 kali, sehingga jumlah pertandingan yang harus dilalui setiap tim menjadi 2(n – 1). Ada juga kompetisi yang mengharuskan setiap tim bertemu tim lainnya 3 kali, sehingga jumlah pertandingan yang harus dilalui setiap tim menjadi 3(n – 1).

Jumlah menang dan kalah pada sebuah klasemen juga berhubungan. Idealnya, ketika semua pertandingan pada kompetisi tersebut telah dilalui, jumlah menang semua tim pasti sama dengan jumlah kalah semua tim, dan jumlah imbang pasti genap. 2 kriteria ini akan menjadi parameter yang menentukan apakah program telah menemukan solusi dari input yang diberikan. Contoh klasemen dapat dilihat pada gambar dibawah ini:

Liga Champions UEFA

Grup A

Pos	Tim	P	M	S	K	GM	GK	+/-	Pts
1	Atlético Madrid	6	4	1	1	14	3	+11	13
2	Juventus	6	3	1	2	7	4	+3	10
3	Olympiakos Piraeus	6	3	0	3	10	13	-3	9
4	Malmö FF	6	1	0	5	4	15	-11	3

Gambar 2. Contoh klasemen kompetisi Liga Champions

Bisa dilihat, jumlah kemenangan pada grup itu sama dengan jumlah kekalahannya, yaitu 11, sementara jumlah hasil imbang pada grup itu bisa dibagi dengan 2 yaitu 2.

Pada persoalan kali ini, ada beberapa batasan yang diberikan yaitu:

1. Klasemen yang diberikan adalah klasemen dimana semua tim telah menyelesaikan semua pertandingan, dengan jumlah pertandingan adalah $2(n - 1)$
2. Setiap tim yang menang diberikan nilai 3, dan hasil imbang akan memberikan nilai 1 bagi kedua tim, sementara tim yang kalah tidak mendapat nilai
3. Input yang diberikan adalah jumlah tim (n), nama tim, dan poin masing – masing tim

Algoritma *greedy* akan mencari jumlah hasil menang, imbang, dan kalah yang telah dialami oleh masing – masing tim. Dengan tahapan sebagai berikut:

1. Mencari jumlah hasil menang, imbang dan kalah setiap tim
2. Memeriksa apakah klasemen telah memenuhi kriteria yang telah disepakati (jumlah imbang genap dan jumlah menang = jumlah kalah)

1. Mencari jumlah hasil menang, imbang dan kalah setiap tim

Program akan mencoba untuk mencari jumlah menang terlebih dahulu dengan pendekatan *greedy*. Pada awalnya, jumlah menang, imbang dan kalah setiap tim diisi dengan nilai 0. Kemudian program akan memeriksa ke setiap tim berapa selisih poin yang telah dikumpulkan dengan poin seharusnya.

Misalnya, saat inisialisasi poin yang telah dikumpulkan tim A adalah 0, karena jumlah menang, imbang dan kalah diiniasialisai dengan nilai 0. Sementara poin seharusnya adalah 12 dan pertandingan yang telah dilalui adalah 6. Karena selisih poin seharusnya dan poin yang dikumpulkan >= 3, maka jumlah menang tim A dapat ditambah 1. Hal itu terus dilakukan sampai selisih poin seharusnya dan poin yang telah dikumpulkan mencapai < 3 dan > 0. Ketika itu terjadi, jumlah menang tim A tidak bisa ditambah lagi karena nilai poin yang dikumpulkan tidak boleh lebih besar dari poin seharusnya. Namun karena kedua poin tersebut belum sama (selisih masih lebih besar dari 0), maka yang ditambah dari tim A adalah jumlah hasil imbang yang mereka dapat. Hal itu terus dilakukan hingga jumlah poin yang telah dikumpulkan sama dengan poin seharusnya. Jika itu terjadi, maka jumlah kalah tim A diisi dengan jumlah pertandingan – jumlah menang – jumlah imbang. Pada tahap ini, seharusnya jumlah menang – imbang – kalah yang didapat oleh tim A adalah 4 – 0 – 2.

2. Memeriksa apakah klasemen telah memenuhi kriteria yang telah disepakati

Setelah jumlah hasil menang, imbang dan kalah semua tim telah terisi, program akan memeriksa apakah program telah menemukan solusi yang memenuhi kriteria, yaitu jumlah menang dan kalah yang sama dan jumlah imbang harus genap. Apabila kedua kriteria tersebut tidak terpenuhi, maka program akan kembali memeriksa setiap tim. Kali ini pemeriksaan yang dilakukan sedikit berbeda.

Kedua kriteria yang diberikan saling berhubungan, apabila jumlah imbang tidak genap, maka jumlah menang keseluruhan pasti tidak sama dengan jumlah kalah keseluruhan. Pemeriksaan dilakukan dengan melibatkan perbandingan antara jumlah menang dan jumlah kalah keseluruhan.

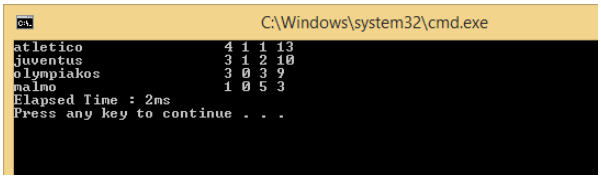
Apabila jumlah kalah keseluruhan lebih besar, maka program akan memeriksa setiap tim apakah jumlah menang dan jumlah imbang mereka bisa diubah. Hal ini bisa diketahui dengan melihat jika jumlah menang dan imbang mereka diubah, apakah akan melebih jumlah pertandingan yang telah mereka lalui, dan poin yang didapat tidak berubah.

Pada contoh sebelumnya, kita telah mengetahui bahwa jumlah menang – imbang – kalah oleh tim A adalah 4 – 0 – 2. Namun bisa saja ada solusi lain yang lebih tepat untuk jumlah menang – imbang – kalah yang didapat oleh tim A. Berdasarkan batasan, poin yang didapat jika sebuah tim memenangkan pertandingan adalah 3, sedangkan jika mereka mendapat hasil imbang maka nilai yang didapat adalah 1. Ini berarti nilai 1 kali kemenangan = 3 kali hasil imbang. Dengan kata lain, kita dapat mengurangi kemenangan sebuah tim, menambah 3 kali hasil imbang dan poin yang didapat tetap sama. Dalam hal ini, jumlah menang yang didapat tim A berkurang menjadi 3, dan jumlah imbang yang didapat bertambah menjadi 3, sedangkan jumlah kekalahan yang didapat tim A menjadi 0. Walaupun jumlah menang berkurang 1, namun secara keseluruhan jumlah kalah berkurang 2, sehingga jarak antara jumlah menang dan jumlah kalah keseluruhan menjadi berkurang. Hal ini terus dilakukan ke setiap tim hingga jumlah menang dan jumlah kalah keseluruhan sama.

Sedangkan jika jumlah menang keseluruhan lebih besar, maka program akan menambah jumlah menang sebesar 1, mengurangi jumlah imbang sebanyak 3, dan menambah jumlah kalah sebanyak 2, kebalikan dari apa yang dilakukan sebelumnya. Namun jumlah menang – imbang – kalah setiap tim tetap tidak boleh melebihi jumlah pertandingan yang telah mereka lalui, dan jumlah poin yang telah dikumpulkan juga tetap sama dengan jumlah poin seharusnya.

3. Hasil implementasi

Algoritma *greedy* ini telah diimplementasikan dalam bahasa Java. Dengan menggunakan tabel klasemen Liga Champion Eropa 2014-2015 sebagai sampel, didapat dari 8 grup Liga Champion, program dapat menghasilkan solusi yang sesuai pada ke-8 grup tersebut. Rata – rata waktu yang dibutuhkan oleh program tersebut untuk menemukan solusi adalah 2 ms. Satu grup Liga Champion Eropa terdiri dari 4 tim. Salah satu contoh dari program dapat dilihat di gambar dibawah, dengan input berdasarkan contoh grup Liga Champion pada gambar 1.



Gambar 3. Contoh hasil *output* program

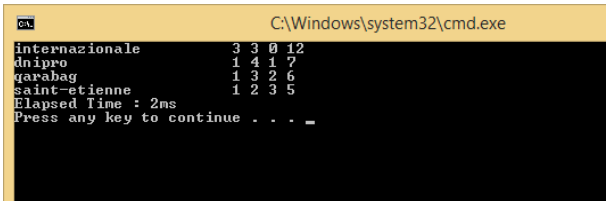
Namun saat digunakan sampel lain, yaitu tabel klasemen Liga Europa 2014 – 2015, program menghasilkan hasil yang berbeda pada salah satu tabel grup. Gambar dibawah ini merupakan tabel grup yang benar:

Grup F

Pos		Tim	P	M	S	K	GM	GK	+/-	Pts
1		Internazionale	6	3	3	0	6	2	+4	12
2		Dnipro Dnipropetrovsk	6	2	1	3	4	5	-1	7
3		Qarabağ	6	1	3	2	3	5	-2	6
4		AS Saint-Étienne	6	0	5	1	2	3	-1	5

Gambar 4. Grup F Liga Europa

Dan berikut ini merupakan hasil output program berdasarkan input Grup F Liga Europa. Perhatikan pada hasil tim Dnipro dan Saint-Etienne dimana hasil *output* berbeda dengan hasil yang diinginkan



Gambar 5. Hasil *output* Grup F Liga Europa

Hal ini dapat terjadi dikarenakan program memeriksa tim yang paling atas terlebih dahulu. Apabila setelah tim teratas diubah dan masih belum menghasilkan solusi, maka tim dibawahnya akan dirubah juga. Jika pada tim ketiga dan keempat program telah menghasilkan solusi yang sesuai dengan kriteria, maka program akan mengeluarkan *output* solusi tersebut, karena program mengenali *output* ini sebagai solusi yang memenuhi kedua kriteria, yaitu jumlah menang keseluruhan sama dengan jumlah kalah keseluruhan (6) dan jumlah imbang bisa dibagi dengan 2 (12). Sedangkan pada kenyataannya hasil yang diinginkan berbeda dengan hasil yang didapat oleh program, karena pada dasarnya solusi ini sudah memenuhi kriteria. Hasil pada output ini juga menunjukkan bahwa kriteria yang ada belum cukup untuk menghasilkan *output* klasemen yang sesuai dengan yang diinginkan.

IV. KESIMPULAN

Sebuah klasemen adalah bagian yang tidak terpisahkan bagi sebuah kompetisi. Klasemen yang tidak lengkap dapat dilengkapi dengan menggunakan algoritma *greedy*. Namun pada makalah ini Kriteria solusi masih kurang memadai untuk menemukan solusi yang diinginkan. Program dapat menemukan solusi, namun masih ada peluang dimana ada lebih dari 1 solusi yang memenuhi kriteria solusi, dan bisa saja solusi yang program pilih tidak sesuai dengan solusi yang diinginkan dikarenakan kriteria yang kurang tersebut.

REFERENSI

[1] Cormen, H Thomas, Charles E Leiserson, and Ronald L Rivest. "Introduction to Algorithms". Boston: Massachussets Institue of Technology Press. 1990

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, 4 Mei 2015

ttd

Muhammad Rizky W. / 13511037

```
import java.util.Scanner;

public class champgroup {
	private int team;
	private int poinmenang = 3;
	private int poinseri = 1;
	private int poinkalah = 0;
	private int match;
	private int[][] grupangka;
	private String[][] gruph2h;

	public void setteam(int a) {
		team = a;
	}

	public int getteam() {
		return team;
	}

	public void setmatch(int a) {
		match = a;
	}

	public int getmatch() {
		return match;
	}

	public void buildgrupangka(int a) { //build grupangka
		grupangka = new int[a][5];
		for (int i = 0; i < a; i++) {
			grupangka[i][0] = i;
			for (int j = 1; j < 5; j++) {
				grupangka[i][j] = 0;
			}
		}
	}

	public void setgrupangka(int a, int b, int c) { //setter grupangka
		grupangka[a][b] = c;
	}

	public int getgrupangka(int a, int b) { //getter grupangka
		return grupangka[a][b];
	}

	public void modwin(int a, int b) {
		if (b == 1) {
			setgrupangka(a,1,getgrupangka(a,1) + 1);
		} else if (b == 0) {
			setgrupangka(a,1,getgrupangka(a,1) - 1);
		}
	}

	public void moddraw(int a, int b) {
		if (b == 1) {
			setgrupangka(a,2,getgrupangka(a,2) + 1);
		} else if (b == 0) {
			setgrupangka(a,2,getgrupangka(a,2) - 1);
		}
	}

	public void modlose(int a,int b) {
		if (b == 1) {
			setgrupangka(a,3,getgrupangka(a,3) + 1);
		} else if (b == 1) {
			setgrupangka(a,3,getgrupangka(a,3) + 1);
		}
	}
```

```
	public int pointeam(int a) {
		return getgrupangka(a,4);
	}

	public int winteam(int a) {
		return getgrupangka(a,1);
	}

	public int drawteam(int a) {
		return getgrupangka(a,2);
	}

	public int loseteam(int a) {
		return getgrupangka(a,3);
	}

	public int calcpointeam(int a) {
		return (getgrupangka(a,1) * poinmenang + getgrupangka(a,2) * poinseri);
	}

	public void isigrupangka() {
		boolean filled = false;
		while (!isgrupangkafinished()) {
			if (!filled) {
				for (int i = 0; i < team; i++) {
					while (pointeam(i) > calcpointeam(i)) {
						if (pointeam(i) > calcpointeam(i)) {
							if ((pointeam(i) - calcpointeam(i)) >= poinmenang) {
								modwin(i,1);
							} else if ((pointeam(i) - calcpointeam(i)) > 0 && (pointeam(i) -
calcpointeam(i) < poinmenang)) {
								moddraw(i,1);
							}
						}
					}
					setgrupangka(i,3,getmatch()-winteam(i)-drawteam(i));
				}
				filled = true;
			} else if (sumwin() < sumlose()) {
				for (int i = 0; sumwin() < sumlose(); i++) {
					if (canchange(i)) {
						modwin(i,0);
						for (int j = 0; j < poinmenang; j++) {
							moddraw(i,1);
						}
					}
					setgrupangka(i,3,getmatch()-winteam(i)-drawteam(i));
				}
			} else if (sumwin() > sumlose()) {
				for (int i = 0; sumwin() > sumlose(); i++) {
					if ((drawteam(i) - poinmenang) >= 0) {
						modwin(i,1);
						for (int j = 0; j < poinmenang; j++) {
							moddraw(i,0);
						}
					}
					setgrupangka(i,3,getmatch()-winteam(i)-drawteam(i));
				}
			}
		}
	}

	public boolean isgrupangkafinished() {
		if ((sumwin() != 0) && (sumdraw() != 0) && sumlose() != 0) {
			return iswinlosesame() && isdrawgenap();
		} else return false;
	}
```

```
	public boolean canchange(int i) {
		int a = winteam(i) - 1 + drawteam(i) + poinmenang;
		if ((winteam(i) - 1 + drawteam(i) + poinmenang) <= match) {
			return true;
		} else return false;
	}

	public int sumwin() {
		int a = 0;
		for (int i = 0; i < team; i++) {
			a = a + winteam(i);
		}
		return a;
	}

	public int sumdraw() {
		int a = 0;
		for (int i = 0; i < team; i++) {
			a = a + drawteam(i);
		}
		return a;
	}

	public int sumlose() {
		int a = 0;
		for (int i = 0; i < team; i++) {
			a = a + loseteam(i);
		}
		return a;
	}

	public boolean iswinlosesame() {
		 if (sumwin() == sumlose()) {
			return true;
		} else return false;
	}

	public boolean isdrawgenap() {
		if (sumdraw() % 2 == 0) {
			return true;
		} else return false;
	}

	public void printgrup() {
		for (int i = 0; i < team; i++) {
			if (getgruph2h(i,0).length() > 7) {
				System.out.println(getgruph2h(i,0) + "		" + getgrupangka(i,1) + " "+
getgrupangka(i,2) + " "+ getgrupangka(i,3) + " "+ getgrupangka(i,4));
			} else if (getgruph2h(i,0).length() <= 7) {
				System.out.println(getgruph2h(i,0) + "			" + getgrupangka(i,1) + " "+
getgrupangka(i,2) + " "+ getgrupangka(i,3) + " "+ getgrupangka(i,4));
			}
		}
	}

	public void buildgruph2h(int a) { //build gruph2h
		gruph2h = new String[a][a];
		for (int i = 0; i < a; i++) {
			gruph2h[i][i] = "N";
		}
	}

	public void setgruph2h(int a, int b, String c) { //setter gruph2h
		gruph2h[a][b] = c;
	}
```

```
	public String getgruph2h(int a, int b) { //getter gruph2h
		return gruph2h[a][b];
	}

	public static void main (String[] args) {
		champgroup test = new champgroup();
		Scanner scanner = new Scanner(System.in);
		System.out.println("Masukkan jumlah tim");
		int x = scanner.nextInt();
		test.setteam(x);
		test.setmatch((test.getteam()-1)*2);
		test.buildgrupangka(test.getteam());
		test.buildgruph2h(test.getteam());
		Scanner scan = new Scanner(System.in);
		for (int i = 0; i < test.getteam(); i++) {
			System.out.println("Masukkan nama tim ke-" + i);
			String a = scan.nextLine();
			test.setgruph2h(i, 0, a);
		}
		for (int j = 0; j < test.getteam(); j++) {
			System.out.println("Masukkan nilai tim ke-" + j);
			int b = scanner.nextInt();
			test.setgrupangka(j, 4, b);
		}
		long startTime = System.nanoTime();
		test.canchange(0);
		test.isigrupangka();
		test.printgrup();
		long endTime = System.nanoTime();
		long duration = endTime - startTime;
		System.out.println("Elapsed Time : " + duration/1000000 + "ms");
	}
}
```