

# Implementasi algoritma pohon merentang merentang minimum (MST) pada permainan

Herlambang Tri Nugroho<sup>1</sup>

Sisworo<sup>2</sup>

Darmawan Satyananda<sup>2</sup>

Universitas Negeri Malang

E-mail : [callme\\_lampard@yahoo.com](mailto:callme_lampard@yahoo.com)

[Sisworo.fmipa@um.ac.id](mailto:Sisworo.fmipa@um.ac.id)

[Darmawan.satyananda.fmipa@um.ac.id](mailto:Darmawan.satyananda.fmipa@um.ac.id)

**ABSTRAK:** Permainan pohon merentang minimum adalah permainan berkonsep matematika. Konsepnya diambil dari salah satu matakuliah matematika S1. Permainannya cukup simpel. Yaitu pemain dihadapkan pada beberapa titik secara acak dimana masing-masing titik mempunyai bobot antar titik yang berbeda. Pemain harus menghubungkan semua titik dengan sisi dengan syarat sisi tidak boleh membentuk lintasan tertutup (sikel) dan semua titik terhubung. Jawaban pemain selanjutnya akan dibandingkan dengan dua algoritma. Yaitu algoritma Kruskal dan algoritma Prim.

Algoritma Kruskal maupun algoritma Prim adalah algoritma pada pohon merentang minimum. Kedua algoritma ini sebenarnya mempunyai tujuan yang sama, yaitu mencari bobot total minimum dari suatu pohon. Pohon adalah graf terhubung yang tidak memuat sikel. Perbedaan algoritma ini hanya pada proses pencarian. Algoritma Prim dimulai dari sebarang titik. Kemudian mencari titik lain yang belum terpilih dan terhubung dengan titik terpilih yang memiliki bobot paling kecil. Sedangkan algoritma Kruskal mengurutkan sisi yang terpendek terlebih dahulu. Kemudian mengambil urut dari sisi yang terpendek dengan syarat sisi tidak boleh membentuk sikel.

Dari proses uji coba dan analisis data, algoritma Prim dan algoritma Kruskal sudah menjadi suatu teorema, oleh karena itu kedua algoritma ini menghasilkan jawaban semimumum mungkin. Input permainan ini adalah titik-titik dan sisi. Selama proses bermain, sisi yang dipilih pemain tidak boleh membentuk sikel dan memuat semua titik. Output permainan berupa minimum spanning tree.

**Kata Kunci:** Algoritma Kruskal dan Prim, Pohon Merentang Minimum.

**ABSTRACT:** The Minimum Spanning Tree Games is a game that use a mathematics concept. That concept is taken from one of the mathematics lesson. This game is simply enough. The player is faced to various of point and every point have various distance with another point. The player must connect every point with line and with condition the line cannot form close circuit (cycle), and all of point are connected. After that, the player's answer will be compared with two algorithms. That algorithms are Kruskal algorithm and Prim algorithm.

Kruskal algorithm and Prim algorithm are algorithm on the minimum spanning tree. The both of them actually have same purpose, that purpose is

- 
1. Herlambang Tri Nugroho adalah Mahasiswa Program Studi Matematika Universitas Negeri Malang
  2. Dr. Sisworo, M.Si, dan Darmawan Satyananda, S.T, M.T, adalah Dosen Universitas Negeri Malang

to find the total weight of the minimum spanning tree. Tree is a connected graph that has no cycles. The process of making the game, starts on making label of vertices. The vertices has weight that presented on the table. The way to play the game is connect the vertices with choose the edge with condition has no cycle and vertices is connected.

From the process of the program experiment and analysis, both of Prim algorithm and Kruskal algorithm are theorem, therefore they have minimum result. The input of game is vertices and edge. While on the process, the drawn edge must not be a cycle and graph must be connected. The output of program is minimum spanning tree..

**Keywords:** Kruskal and Prim Algorithm, Minimum Spanning Tree.

Permainan pohon merentang adalah permainan yang memakai konsep algoritma pada pohon merentang minimum. Ada beberapa algoritma pohon merentang minimum. Diantaranya algoritma Kruskal dan algoritma Prim. Kedua algoritma ini bertujuan mencari bobot total minimum dari suatu pohon merentang. Prinsip kedua algoritma ini dipakai dalam pembuatan sekaligus aturan permainan.

Pada penelitian ini yang berjudul “Implementasi Algoritma Pohon Merentang Minimum pada Permainan”, akan dibuat suatu permainan yang memakai konsep algoritma pohon merentang minimum. Pemain akan dihadapkan pada beberapa titik secara acak dengan bobot sisi yang bervariasi. Pemain harus menghubungkan semua garis dengan syarat graf yang terbentuk tidak memuat siklus dan graf terhubung, seperti definisi tree.

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah dalam penelitian ini adalah “Bagaimana langkah-langkah membuat permainan yang memakai konsep algoritma pohon merentang minimum, Bagaimana mengetahui efektivitas Algoritma Kruskal dan Prim, dan Menerapkan permainan ini pada Borland Delphi 7?”

Tujuan penelitian ini adalah untuk mengembangkan permainan berbasis matematika khususnya pada pohon merentang minimum dan mengembangkan permainan yang lebih mengasah otak ketimbang permainan yang lebih menonjolkan kecepatan tangan.

## **METODE**

Pada proses pembuatan permainan dibutuhkan beberapa

### **1. Perancangan Program**

Pada bagian ini penulis ingin menjelaskan rancangan awal program. Rancangan awal program mengacu pada struktur game pada umumnya. Berikut poin-poin dalam rancangan awal program.

1. Input Program (game).
2. Cara Pemain (user) memainkan atau menjalankan game.
3. Output Program (game).

### 1.1 Input Program

Pada saat pemain membuka program, pemain harus mengisi atau menekan suatu tombol untuk memainkan game. Pertama pemain harus memilih tingkat kesulitan atau level terlebih dahulu, dimaksudkan agar program dapat menentukan jumlah titik yang akan ditampilkan. Berikut penjelasan perbedaan masing-masing level.

1. Level Easy menampilkan 5 titik.
2. Level Medium menampilkan 7 titik.
3. Level Hard menampilkan 9 titik.

Kemudian pemain menekan tombol mulai sebagai tanda bahwa game telah dimulai dan pada bagian setengah layar sebelah kiri menampilkan titik dalam posisi acak. Jika sebelum game dimulai pemain belum memilih level maka game tidak bisa dijalankan dan tidak ada tampilan titik.

### 1.2 Cara Pemain Memainkan Game

Ketika pemain sudah memilih level dan menekan tombol mulai, pemain akan dihadapkan pada tampilan titik secara acak tadi. Pada bagian kiri layar juga terdapat tabel untuk menampung jarak antar titik. Fungsi tabel ini sekaligus untuk menentukan garis yang ingin dilukis pemain. Karena tujuan game mencari pohon merentang minimum, maka garis yang dilukis si pemain harus graf yang berbentuk pohon (*tree*).

### 1.3 Output Program

Setelah pemain melukis garis yang tidak membentuk siklus dan semua titik terhubung, jawaban si pemain akan dibandingkan dengan kunci jawaban. Kunci jawaban diambil dari algoritma Prim atau Kruskal yang menghasilkan nilai paling minimum. Misalnya algoritma Prim menghasilkan nilai yang lebih minimum dari algoritma Kruskal, maka kunci jawaban diambil dari hasil algoritma Prim. Setelah proses perbandingan si pemain tinggal menekan tombol skor. Skor pemain akan ditampilkan pada memo.

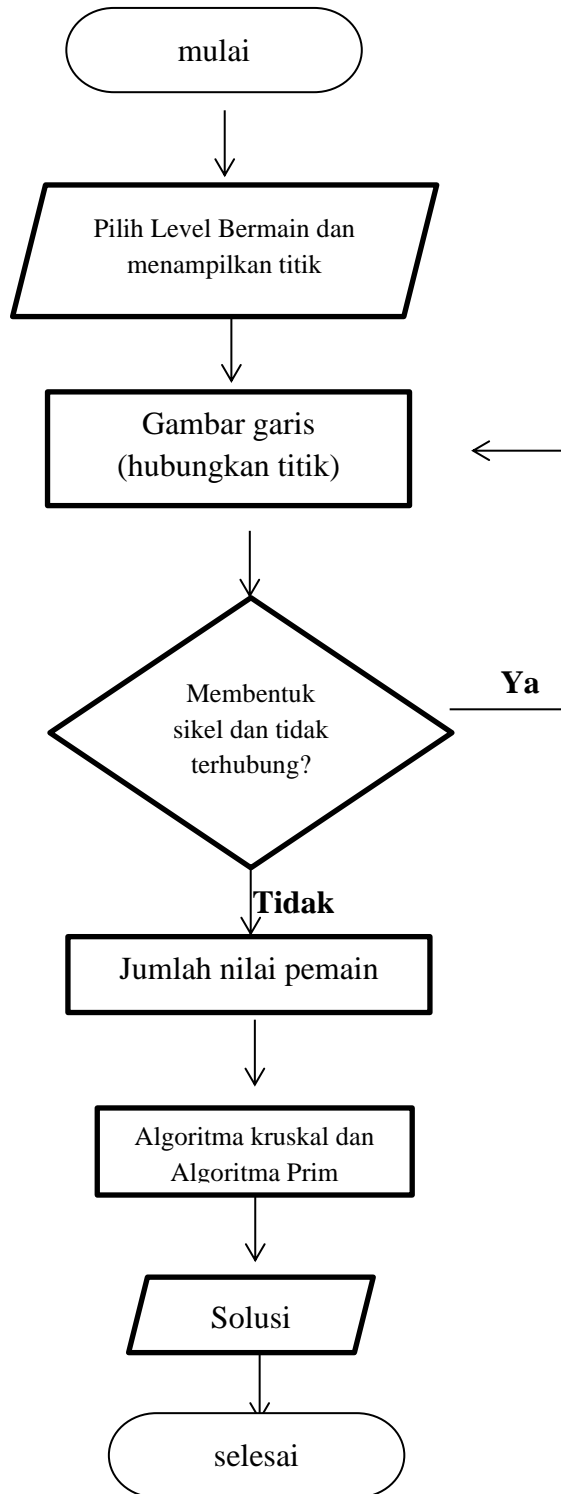
## 2. Flowchart Program

Flowchart adalah alur pemrograman yang akan dibuat. Flowchart berbentuk simbol-simbol dimana setiap simbol memiliki arti dan kegunaan masing-masing dan dengan menggunakan flowchart kita bisa lebih mudah untuk menjelaskan alur dari algoritma yang digunakan. Flowchart program yang akan dibuat ada pada Gambar 3.1.

Penjelasan *flowchart* pada Gambar 3.1 adalah sebagai berikut.

1. Pertama buka program atau game.
2. Pilih level bermain dan menampilkan titik.
3. Hubungkan titik-titik dengan garis dengan cara menekan tombol pada tabel jarak.
4. Jika garis yang terbentuk membentuk siklus atau graf masih belum terhubung maka koreksi penggambaran garis.

5. Jika garis yang terbentuk sudah tidak membentuk sikel atau sudah terhubung maka jumlahkan bobot dari garis yang dipilih si pemain.
6. Bandingkan jawaban si pemain dengan hasil minimum yang diperoleh dari algoritma kruskal atau prim (hasil kedua algoritma diambil yang paling minimum tetapi jika kedua algoritma menghasilkan kunci jawaban yang sama maka diambil salah satu).
7. Solusi disini adalah skor pemain yang diperoleh dari perhitungan kunci jawaban (hasil dari algoritma Prim atau Kruskal yang menghasilkan nilai paling minimum) dibagi jawaban si pemain dikali 100. Skor pemain dibulatkan menjadi bilangan bulat.



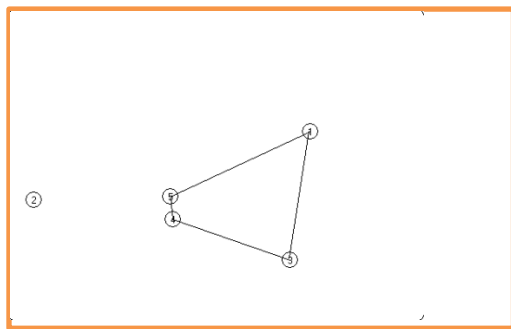
**Gambar 3.1** Struktur Flowchart

### 2.1 Proses Seleksi jawaban Pemain

Pada bagian ini jawaban akan dikoreksi apakah pemilihan garis si pemain membentuk siklus atau graf belum terhubung. Jika jawaban pemain membentuk siklus atau graf masih belum terhubung akan diberi peringatan langsung pada program.

### 2.2 Pengecekan Sikel atau Tidak

Pada proses seleksi jawaban pemain, garis yang dipilih pemain akan dicek terlebih dahulu. Apakah garis yang dipilih pemain membentuk siklus atau tidak. Jawaban pemain dianggap membentuk siklus apabila graf yang terbentuk membentuk siklus. Misalnya pada ilustrasi berikut ini. Perhatikan Gambar 3.2, apabila titik yang kita pilih pertama adalah titik 1. Lintasan yang terbentuk adalah 1-3-4-5-1. Karena dari titik 1 garis bisa kembali ke titik 1 lagi, maka graf tersebut membentuk siklus.



**gambar 3.2** *Graf Membentuk Sikel dan Belum Terhubung*

### 2.3 Pengecekan Graf Terhubung atau Tidak

Graf dinyatakan tidak terhubung apabila garis tidak memuat semua titik. Pada Gambar 3.2 titik 1,3,4, dan 5 sudah terhubung garis, tetapi titik 2 belum terhubung garis. Pemain tidak bisa mendapatkan skor jika titik yang dipilihnya masih belum terhubung semua.

### 2.4 Proses Mencari Kunci Jawaban

Pencarian kunci jawaban menggunakan dua algoritma yaitu algoritma Kruskal dan Algoritma Prim.

#### 2.4.1 Algoritma Prim

Setelah data bobot antar titik sudah disimpan pada StringGrid, kemudian menentukan pohon rentang dari graf yang terbentuk menggunakan algoritma prim. Agar tidak mengubah isi matriks bobot mtBobot, digunakan matriks yang diberi nama mtC. Bila elemen mtBobot bernilai 0 maka elemen mtC pada posisi yang bersesuaian diisi dengan nilai yang sangat besar (dalam

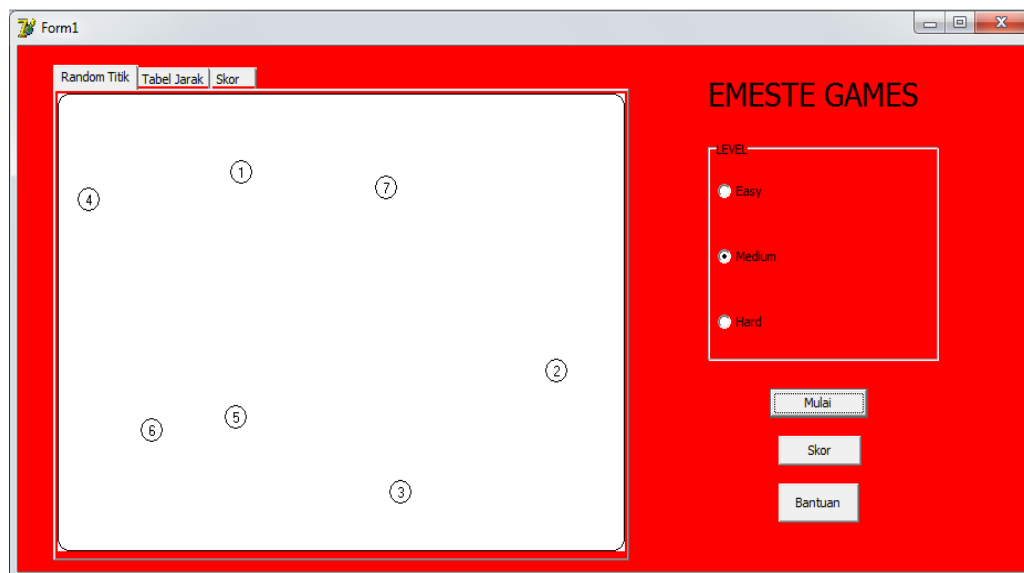
hal ini adalah 9999). Setelah menentukan isi himpV dengan semua titik dalam graf, berikutnya adalah iterasi untuk mencari sisi terkecil. Sisi  $(u, v)$  yang didapatkan adalah bila  $u$  ada di himpU dan  $v$  ada di himpV-himpU dan memiliki bobot terkecil. Panjang total algoritma Prim diambil dari sisi  $(u,v)$  yang terpilih.

#### 2.4.2 Algoritma Kruskal

Pada algoritma Kruskal dibuat array yang menampung urutan sisi dari yang terkecil sampai terbesar. Prinsipnya sama dengan seleksi sikel di atas. Sisi  $(u,v)$  yang didapatkan adalah sisi terkecil dari pengurutan tadi. Kemudian mengambil sisi terkecil berikutnya dengan syarat tidak membentuk sikel. Panjang total algoritma Kruskal diambil dari sisi  $(u,v)$  yang terpilih.

### 3. Rancangan Tampilan Program

Langkah selanjutnya adalah membuat rancangan tampilan program. Pada tampilan program terdapat nama permainan, level bermain, tombol mulai, tombol skor, bantuan, tampilan titik secara acak, tampilan tabel jarak sekaligus tombol untuk melukis garis, dan tampilan skor pemain pada memo.



Gambar 3.3 Tampilan Awal Program

Penjelasan dari tampilan rancangan program adalah sebagai berikut.

- A. Pada bagian level berisi 3 kategori, diantaranya easy, medium, dan hard. Pemain atau user harus memilih salah satu dari ketiga level tersebut.
- B. Tampilan titik secara acak muncul ketika tombol mulai diklik.

- C. Tampilan tabel jarak. Tabel ini selain memberikan informasi mengenai bobot atau jarak antar titik juga sekaligus menjadi tombol untuk menggambar atau membuat garis yang dipilih pemain.
- D. Tombol mulai berfungsi menampilkan random titik sebagaimana disebutkan pada sebelumnya. Sebelum menekan tombol mulai pemain harus memilih level bermain.
- E. Tombol skor ini berfungsi menampilkan hasil akhir atau nilai dari si pemain tersebut. Tentu tombol ini berfungsi jika pemain sudah membuat garis dengan tepat (tidak membentuk sikel dan terhubung).
- F. Pada bagian memo, berisi tampilan skor pemain setelah dibandingkan dengan kunci jawaban.
- G. Tombol bantuan berfungsi memberikan informasi secara khusus mengenai program tujuannya sebagai bantuan apabila user/pemain mengalami kesulitan selama memainkan game.

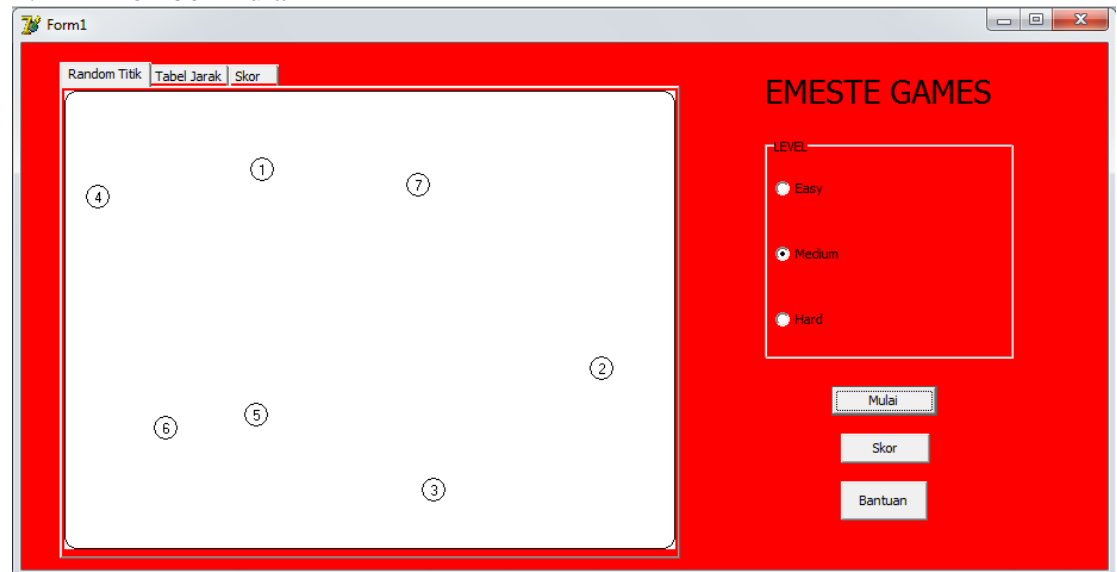
## 1. Hasil dan Pembahasan

### 1.1 Ujicoba Program

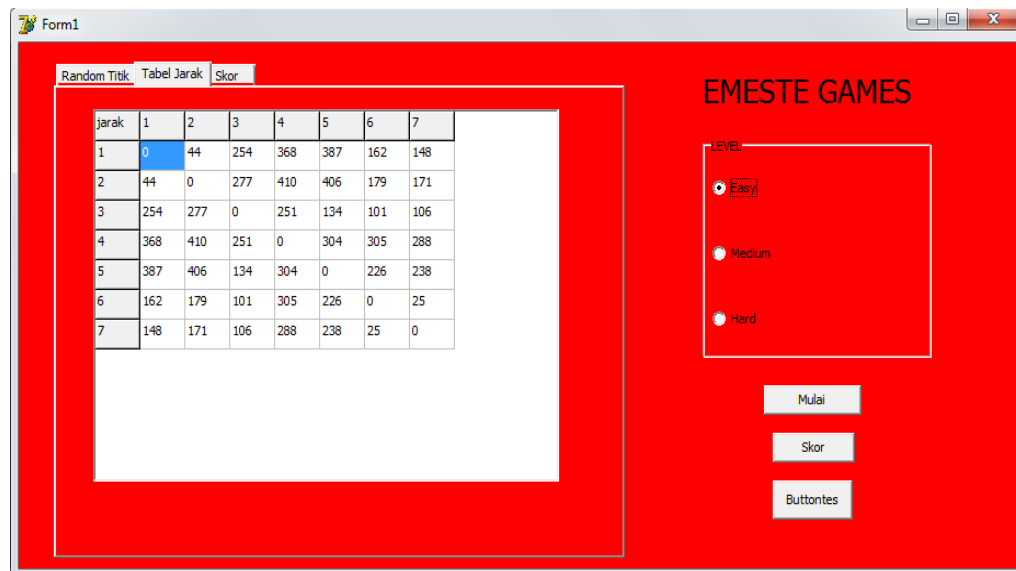
Pada tahap ini program akan diuji coba apakah program sudah berjalan dengan baik. Apakah semua tombol berfungsi dengan baik, begitu juga dengan hasil apakah sudah terisi dengan benar sesuai soal dan jawaban si pemain. Kemudian terdapat syarat- syarat sebelum jawaban si pemain dihitung, yaitu gambar garis si pemain tidak boleh membentuk lintasan tertutup (sikel) dan semua titik terhubung.

#### 1.1.1 Pengecekan Fungsi Tombol

##### 1. Tombol Mulai



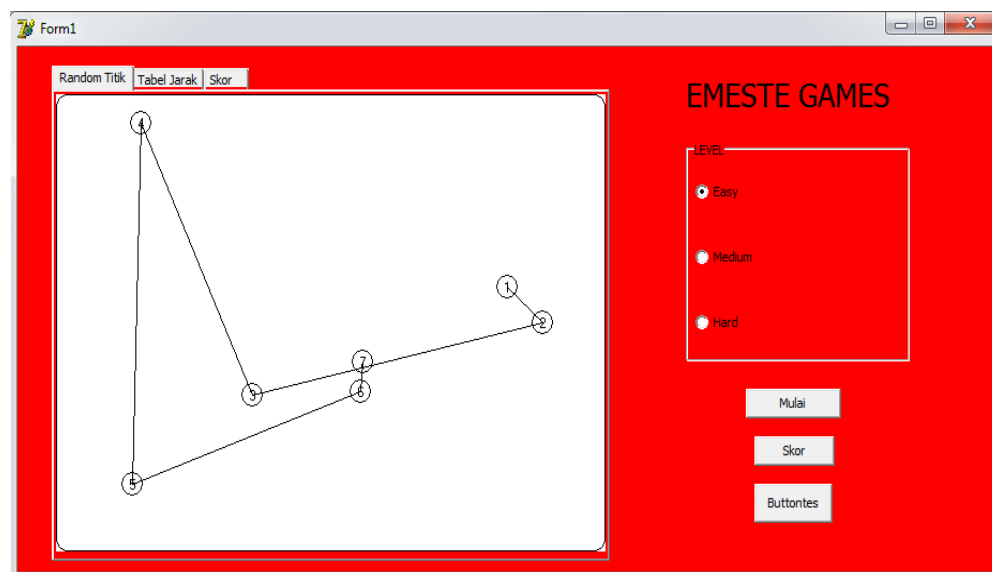
**gambar 4.1** Gambar ketika menekan tombol mulai



Gambar 4.2 Gambar tabel jarak

Terlihat pada Gambar 4.1 dan Gambar 4.2 bahwa tombol mulai sudah berfungsi dengan baik sebagaimana mestinya.

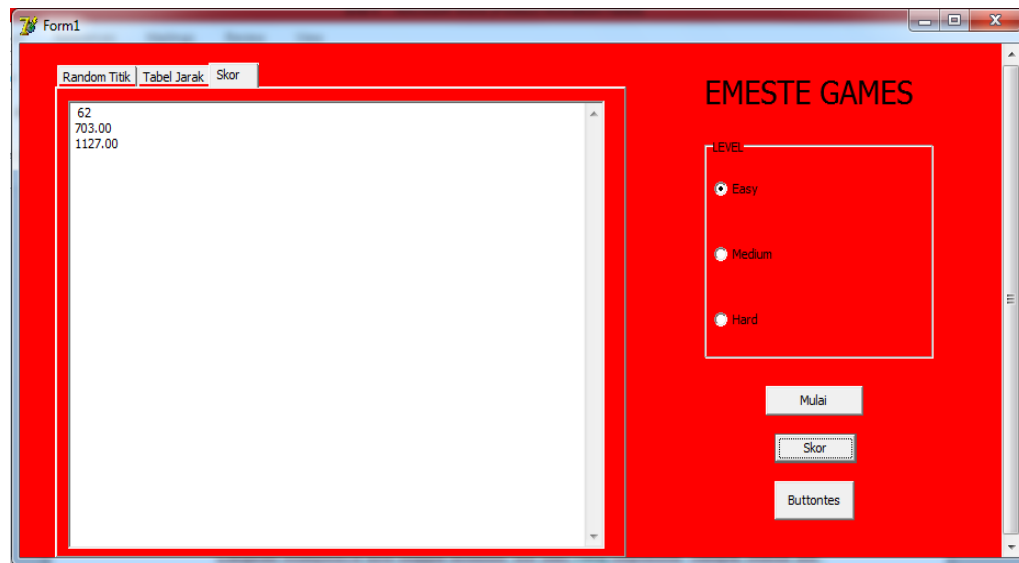
## 2. Tombol pada grid.



gambar 4.3 Gambar sisi

Pada Gambar 4.3 menunjukkan bahwa tombol grid sudah berfungsi dengan baik. Garis dapat dilukis dengan menekan sel pada grid .

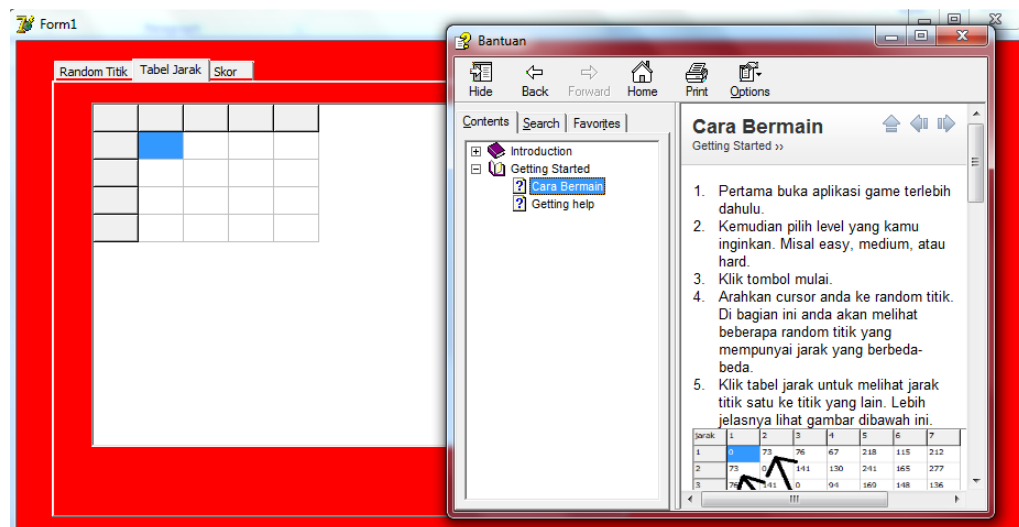
### 3. Tombol skor



**gambar 4.5** Gambar hasil permainan

Ketika pemain menekan tombol skor akan muncul kunci jawaban, jawaban pemain, dan skor pemain. Data ini ditampilkan pada memo skor. Terlihat Gambar 4.5 menunjukkan skor pemain, kunci jawaban, dan jawaban pemain.

### 4. Tombol Bantuan



**gambar 4.6** Tampilan Bantuan

#### 4.5 Analisis Hasil

Analisis yang telah dilakukan, pertama prinsip dari algoritma Kruskal dapat digunakan untuk mengecek apakah pemilihan garis dari pemain membentuk siklus dan graf belum terhubung. Jika pemilihan garis tersebut masih membentuk siklus dan tidak terhubung maka pemain harus memilih garis tadi dari awal. Dari kedua algoritma yang dipakai baik Kruskal maupun Prim keduanya sama-sama menjadi teorema, sehingga kemungkinan jawaban yang dihasilkan akan semimumum mungkin.

## PENUTUP

### Kesimpulan

Dari pembahasan algoritma pohon merentang minimum yang dikembangkan dalam suatu permainan dapat disimpulkan sebagai berikut

1. Dalam pengembangan game pohon merentang minimum dibutuhkan beberapa langkah diantaranya.
  - Pembuatan random titik sekaligus jarak atau bobot antar titik tersebut.
  - Seleksi pemilihan garis pemain.
  - Membandingkan hasil algoritma prim dan algoritma kruskal.
  - Membandingkan kunci jawaban dengan nilai pemain.
2. Konsep pohon rentang minimum dapat dikembangkan pada suatu permainan.
3. Permainan pohon rentang minimum dapat menjadi alternatif game yang selama ini hanya mengandalkan sisi kecepatan tangan.

### Saran

Melengkapi *software* dengan fasilitas lain yang lebih lengkap sehingga permainan menjadi lebih menarik dan bervariasi misalnya menambahkan warna pada titik, membuat cara tersendiri untuk menggambarkan garis, memberikan animasi atau gambar yang mudah dicerna oleh pemain.

## DAFTAR RUJUKAN

Kadir, Abdul. 2006. *Dasar Pemrograman Delphi*. Yogyakarta: ANDI Yogyakarta.

Munir, Rinaldi. 2008. *Diktat Kuliah IF2091 Struktur Diskrit*. Program Studi Teknik Informatika, Sekolah Tinggi Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Prima, Pudy. 2010. *Membandingkan kemangkusan Algoritma Prim dan Algoritma Kruskal Dalam Pemecahan Masalah Pohon Merentang Minimum*. Progam Studi Teknik Informatika, Sekolah Tinggi Teknik Elektro dan Informatika. Institut Teknologi Bandung.

Satyananda, Darmawan. 2014. *Struktur Data*. Progam Studi Matematika, Fakultas MIPA, Universitas Negeri Malang.