



Perancangan Aplikasi Kata Lintas: Teka-teki Silang Dengan Menerapkan Aplikasi Fisher Yates Shuffle

Fahmi Ardiyansa

Program Studi Teknik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamangaraja No.338, Medan, Sumatera Utara, Indonesia

Email: fahmiardiyansa4gmail.com

Abstrak - TTS (Teka teki silang) merupakan suatu permainan kata yang biasanya berbentuk serangkaian ruang-ruang kosong berbentuk kotak berwarna hitam dan putih, tujuan dari permainan ini adalah untuk mengisi kotak-kotak putih dengan huruf, membentuk kata atau frasa tertentu, dengan menyelesaikan petunjuk yang mengarah ke jawaban tertentu. Jenis permainan teka-teki silang juga saat ini sudah berkembang salah satunya teka-teki silang kata lintas, pada game ini kumpulan huruf di acak dan pengguna akan mencari suatu kata dari huruf yang di acak tersebut. Namun, untuk pembuatan aplikasi permainan kata lintas masih belum tidak diketahui algoritma pengacakan apa yang digunakan untuk mengacak huruf-huruf tersebut sehingga bisa menjadi sebuah huruf acak yang akan disusun oleh pengguna. Setelah melihat beberapa penelitian terdahulu, maka memilih Algoritma Fisher Yates Shuffle yang diterapkan menggunakan Aplikasi Visual Basic sebagai solusi dirasa sangatlah tepat, dimana algoritma ini akan menginisialisasi huruf terlebih dahulu menjadi angka kemudian huruf yang sudah diinisialisasikan tersebut diacak menggunakan algoritma, dan pengguna menebak rangkaian huruf yang sudah diacak sebelumnya.

Kata Kunci: Teka-teki Silang; Visual Studio; Fisher Yates Shuffle

Abstract - Crossword puzzle is a word game which is usually in the form of a series of empty spaces in the form of black and white squares, the objective of this game is to fill the white boxes with letters, form certain words or phrases, by completing clues that lead to a specific answer. The type of crossword puzzle game is currently developing, one of which is a cross word crossword puzzle, in this game the collection of letters is random and the user will look for a word from the random letters. However, it is still not known what randomization algorithm was used to make the crossword game application to scramble the letters so that they can become random letters that will be compiled by the user. After seeing some previous research, choosing the Fisher Yates Shuffle Algorithm which is applied using the Visual Basic Application as a solution is considered very appropriate, where this algorithm will initialize letters first to numbers then the initialized letters are randomized using an algorithm, and the user guesses the series of letters that are used has been scrambled before.

Keywords: Crossword Puzzle; Visual Studio; Fisher Yates Shuffle

1. PENDAHULUAN

TTS (Teka teki silang) merupakan suatu permainan kata yang biasanya berbentuk serangkaian ruang-ruang kosong berbentuk kotak berwarna hitam dan putih. Tujuan dari permainan ini adalah untuk mengisi kotak-kotak putih dengan huruf, membentuk kata atau frasa tertentu, dengan menyelesaikan petunjuk yang mengarah ke jawaban tertentu. Kotak yang berwarna hitam (atau warna lainnya) biasanya di gunakan untuk memisahkan kata atau frasa yang berbeda. Saat ini banyak media pembelajaran yang digunakan dengan menerapkan teknologi, salah satunya melalui permainan (game). Contoh dari permainan yang dapat menambah wawasan dan ilmu pengetahuan yaitu permainan teka teki silang (TTS). Selain seru, permainan ini juga melatih kemampuan mengingat dan memahami.

Jenis permainan teka-teki silang juga saat ini sudah berkembang, diantaranya teka-teki silang kata lintas, pada game ini kumpulan huruf di acak dan pengguna akan mencari suatu kata dari huruf yang di acak tersebut. Permainan ini juga dapat mengasah kemampuan berfikir dari otak, dengan sering memainkan permainan kata lintas ini bisa melatih pemainnya untuk berfikir lebih aktif dan menambah wawasan tentang kata-kata tertentu. tetapi yang menjadi permasalahannya adalah pada aplikasi kata lintas tersebut masih tidak diketahui algoritma pengacakan apa yang digunakan untuk mengacak huruf-huruf tersebut sehingga bisa menjadi sebuah huruf acak yang akan disusun oleh pengguna.

Pada tahun 2012, Ade-Ibijola dan Abejide Olu, merancang Aplikasi A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures. Aplikasi ini Membuat suatu aplikasi untuk simulasi permainan kartu secara virtual dengan Algoritma fisher yates dan Domain Specific Data Structures. Dari 54 buah kartu yang ditarik ke tempat secara acak, pengacakan kartu dapat berjalan sesuai alur. Aplikasi ini bisa di terapkan dalam desktop dengan desain.[1]

Dari penelitian sebelumnya yang dilakukan oleh Mhd Arief Hasan pada tahun 2017 dengan judul "Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru disimpulkan bahwa dalam menerapkan algoritma Fisher-Yates yang digunakan pada aplikasi Computer Based Testing dapat mengacak soal pada setiap peserta ujian sehingga dalam pelaksanaan ujian setiap mahasiswa dalam menjawab soal memiliki nomor yang sama tetapi bentuk soal yang berbeda.[2]

Berdasarkan pemaparan masalah di atas maka dilakukan penelitian dengan menerapkan algoritma Fisher Yates Shuffle yang akan diterapkan pada permainan kata lintas : teka teki silang

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Untuk mendukung kelancaran penelitian ini, maka dilakukan tahapan penelitian sebagai berikut:

- a. Studi pustaka
Merupakan bentuk penelitian yang dilakukan berdasarkan kepustakaan atau buku-buku, jurnal, dan sumber-sumber sejenis (tertulis maupun dokumen) lain yang mempunyai hubungan dengan masalah yang sedang dibahas
- b. Analisa dan Perancangan
Analisa digunakan setelah data-data yang diperoleh dari studi pustaka kemudian dilakukan analisa sesuai dengan permasalahan penulis dalam penelitian ini
- c. Perancangan dan Implementasi
Tahap ini merupakan perancangan serta membangun aplikasi dan mengimplementasikan algoritma Fisher Yates Shuffle guna mendapatkan gambaran hasil akhir terhadap aplikasi Kata Lintas : Teka-Teki Silang. Selanjutnya akan mengimplementasikan algoritma yang akan digunakan sesuai hasil analisa dan perancangan aplikasi pada tahap sebelumnya.
- d. Pengujian
Pada tahap ini dilakukan Pengujian dari teori atau data yang telah dirancang dan diimplementasikan guna menguji kebenarannya.
- e. Dokumentasi
Pada tahap ini seluruh kegiatan dalam pembuatan sistem didokumentasikan kedalam bentuk tulisan berupa laporan penelitian.

2.2 Fisher Yates Shuffle

Fisher-Yates Shuffle (dinamai berdasarkan penemunya, Ronald Fisher dan Frank Yates) digunakan untuk mengubah urutan masukan yang diberikan secara acak. Permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, metode dasar yang diberikan untuk menghasilkan permutasi acak dari angka 1-N. Algoritma ini dinyatakan bisa karena permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, hal ini dibuktikan dengan percobaan mengacak satu set kartu yang dilakukan secara berulang [3].

Pemakaian Fisher Yates Shuffle bisa melalui dua cara yaitu *original method* dan *modern method*. Menurut Pavel Micka, *original method* dipublikasikan pada tahun 1938, pada metode ini dilakukan dengan cara penarikan secara berulang dari unsur daftar masuk kemudian menuliskannya ke daftar keluaran kedua pendekatan ini dilakukan oleh manusia dengan secara kertas dan sebuah pensil. Richard Durstenfiel pada tahun 1964 *Modern method* dikenal karna lebih optimal dibandingkan dengan *Original Method* algoritma yang *modern* berbeda dari yang sebelumnya, sangat komputasi dan matematis proses angka terakhir akan dipindahkan ke angka yang ditarik keluar dan menggubah angka yang ditarik keluar menjadi angka akhir yang tidak ditarik lagi untuk setiap kali penarikan dan berlanjut untuk iterasi berikutnya hal ini dilakukan dalam $O(1)$ waktu dan ruang dengan demikian waktu dan ruang kompleksitas algoritmanya $O(n)$ yang optimal permutasi yang akan dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama hal ini di buktikan dengan percobaan mengacak suatu set kartu yang dilakukan berulang-ulang seperti yang bisa dilihat pada tidak jenuh apabila bermain dikarenakan mendapatkan soal yang sama.

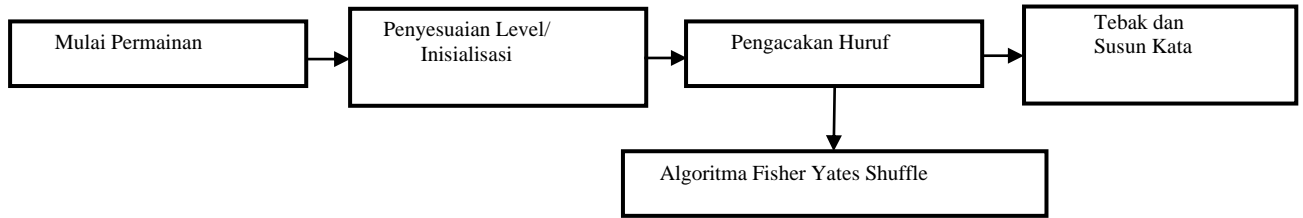
Fisher-Yates Shuffle terdapat beberapa tahap dalam proses pengacakannya. Adapun tahapan tersebut adalah:

- a. Tentukan nilai n
- b. Pilih angka acak (x) dimana $1 \leq x \leq n$
- c. Tukar posisi (x) dengan angka terakhir pada range $1 - n$
- d. Pindahkan angka x ke list array
- e. Atur ulang nilai n , dimana $n = n - 1$
- f. Jika n masih memenuhi syarat $n > 0$ maka kembali lakukan proses pilih angka acak (x) dimana $1 \leq x \leq n$ (proses b)
- g. Jika $n = 0$ maka pengacakan telah selesai di lakukan

3. HASIL DAN PEMBAHASAN

3.1 Analisa Proses Permainan Kata Lintas

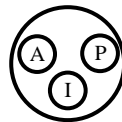
Pada permainan lintas kata ini dimulai disaat pengguna menjalankan aplikasi yang dirancang. Disaat pengguna membuka aplikasi terdapat pemilihan menu yang dapat dipilih oleh pengguna, apabila pengguna memilih memulai permainan maka pengguna akan memulai permainan lintas kata : teka-teki silang melalui level terendah yaitu level 1, dimana setiap level memiliki perbedaan, level 1 terdapat 3 huruf yang diacak dan, kemudian level 2 terdiri dari 4 huruf yang diacak, dan level 3 yang menjadi level tertinggi terdiri atas 5 huruf untuk dicak dan ditebak oleh pengguna. kemudian huruf diacak menggunakan algoritma *Fisher Yates Shuffle*, dimana penerapan algoritma ini dimulai dengan mengganti huruf yang diinisialisasi dengan angka guna menerapkan algoritmanya. Setelah huruf diacak lalu huruf yang teracak ditampilkan untuk kemudian pengguna harus mengisi kolom yang disediakan untuk menebak kata apa yang bisa dihasilkan dari huruf yang sebelumnya telah diacak menggunakan algoritma *Fisher Yates Shuffle*.



Gambar 1. Prosedur Permainan Lintas Kata

3.1.1 Inisialisasi Huruf Berdasarkan Huruf

Sebelum melakukan penerapan algoritma Fisher Yates Shuffle dalam permainan kata lintas : teka teki silang, terlebih dahulu dimulai dengan menginisialisasi huruf yang akan digunakan pada setiap level dipertandingan guna mempermudah penerapan algoritma.



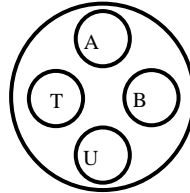
Gambar 2. Kata Level 1

Penginisialisasian yang dilakukan berdasarkan kata level 1 yang terdiri dari 3 huruf, dimana kata yang digunakan adalah kata: API

A: diinisialkan dengan angka 1 pada penerapan algoritma

P: diinisialkan dengan angka 2 pada penerapan algoritma

I: diinisialkan dengan angka 3 pada penerapan algoritma



Gambar 3. Kata Level 2

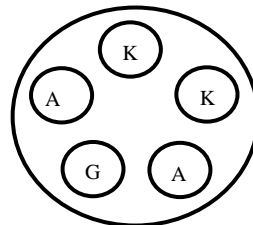
Penginisialisasian yang dilakukan berdasarkan kata level 2 yang terdiri dari 4 huruf, dimana kata yang digunakan adalah kata: BATU

B: diinisialkan dengan angka 1 pada penerapan algoritma

A: diinisialkan dengan angka 2 pada penerapan algoritma

T: diinisialkan dengan angka 3 pada penerapan algoritma

U: diinisialkan dengan angka 4 pada penerapan algoritma



Gambar 4. Kata Level 3

Penginisialan dilakukan berdasarkan kata level 3 yang terdiri dari 5 huruf: KAGAK

K: diinisialkan dengan angka 1 pada penerapan algoritma

A: diinisialkan dengan angka 2 pada penerapan algoritma

G: diinisialkan dengan angka 3 pada penerapan algoritma

A: diinisialkan dengan angka 4 pada penerapan algoritma

K: diinisialkan dengan angka 5 pada penerapan algoritma

3.1.2 Penerapan Algoritma Fisher Yates Shaffle

Pada penerapan algoritma dimulai dengan mengurutkan terlebih dahulu huruf yang sudah diinisialisasi dengan angka sebelumnya. Dimana proses penerapan dilakukan pada setiap level. Level 1:

Tabel 1. Putaran Level 1

| Range(n) | Roll(x) | Scratch | Hasil (Array) |
|----------|---------|---------|---------------|
| | | 1,2,3 | |

Untuk putaran pertama pada level 1 maka menggulirkan angka acak dari 1 hingga 3, lalu menukar angka tersebut dengan angka yang berada diposisi terakhir.

Tabel 2. Pengacakan Huruf Level 1

| Range | Roll | Scratch | Hasil |
|-------|------|---------|-------|
| 1-3 | 2 | 1,3 | 2 |
| 1-2 | 3 | 1 | 2,3 |
| 1-1 | 2 | 0 | 2,3,1 |

Berdasarkan pengacakan yang dilakukan menggunakan algoritma *Fisher Yates Shuffle* diperoleh hasil pengacakan dengan urutan: 2,3,1 dimana urutan ini akan diterapkan pada aplikasi. Level 2:

Tabel 3. Pengacakan Huruf Level 2

| Range (n) | Roll (x) | Scratch | Hasil (Array) |
|-----------|----------|---------|---------------|
| | | 1,2,3,4 | |
| 1-4 | 2 | 1,4,3 | 2 |
| 1-3 | 2 | 1,3 | 2,4 |
| 1-2 | 2 | 1 | 2,4,3 |
| 1-1 | 3 | 0 | 2,3,4,1 |

Berdasarkan pengacakan yang dilakukan menggunakan algoritma *Fisher Yates Shuffle* pada level 2 diperoleh hasil pengacakan dengan urutan: 2,4,3,1 dimana urutan ini akan diterapkan pada aplikasi. Level 3:

Tabel 4. Pengacakan Huruf Level 3

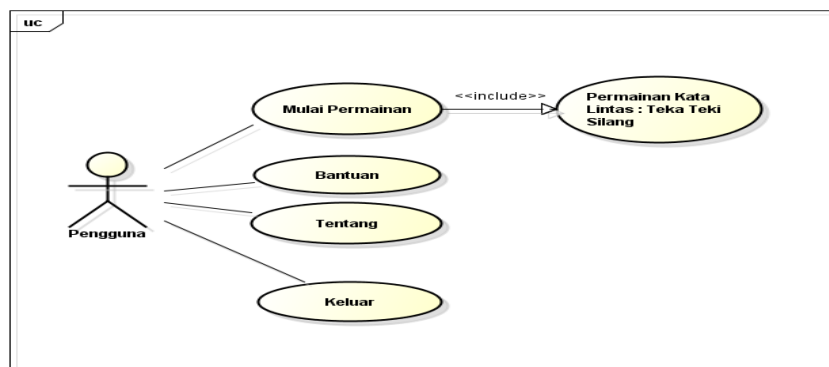
| Range (n) | Roll (x) | Scratch | Hasil (Array) |
|-----------|----------|-----------|---------------|
| | | 1,2,3,4,5 | |
| 1-5 | 3 | 1,2,5,4 | 3 |
| 1-4 | 2 | 1,4,5 | 3,2 |
| 1-3 | 2 | 1,5 | 3,2,4 |
| 1-2 | 2 | 1 | 2,4,3,5 |
| 1-1 | 3 | 0 | 2,4,3,5,1 |

Berdasarkan pengacakan yang dilakukan menggunakan algoritma *Fisher Yates Shuffle* pada level 3 diperoleh hasil pengacakan dengan urutan: 2,4,3,5,1 dimana urutan ini akan diterapkan pada aplikasi. Dari proses pengacakan yang telah dilakukan kemudian angka diubah kembali menjadi huruf dikarenakan angka tersebut merupakan inisial dari huruf yang diacak, huruf tersebutlah yang akan dijadikan soal untuk menebak kata pada aplikasi kata lintas : Teka Teki Silang.

3.2 Perancangan

a. Use Case Diagram

Pada *Use Case Diagram* pengguna memiliki beberapa pilihan didalam menu utama melalui penggambaran ini pengguna dapat langsung mengklik mulai permainan lalu langsung masuk ke permainan kata silang, jika pengguna mengklik menu bantuan maka pengguna akan diberikan tampilan tentang penjelasan cara permainan, apabila pengguna mengklik Tentang maka pengguna akan diarahkan keform tentang perancang aplikasi, dan pengguna juga bisa keluar dari aplikasi dengan mengklik pilihan keluar.

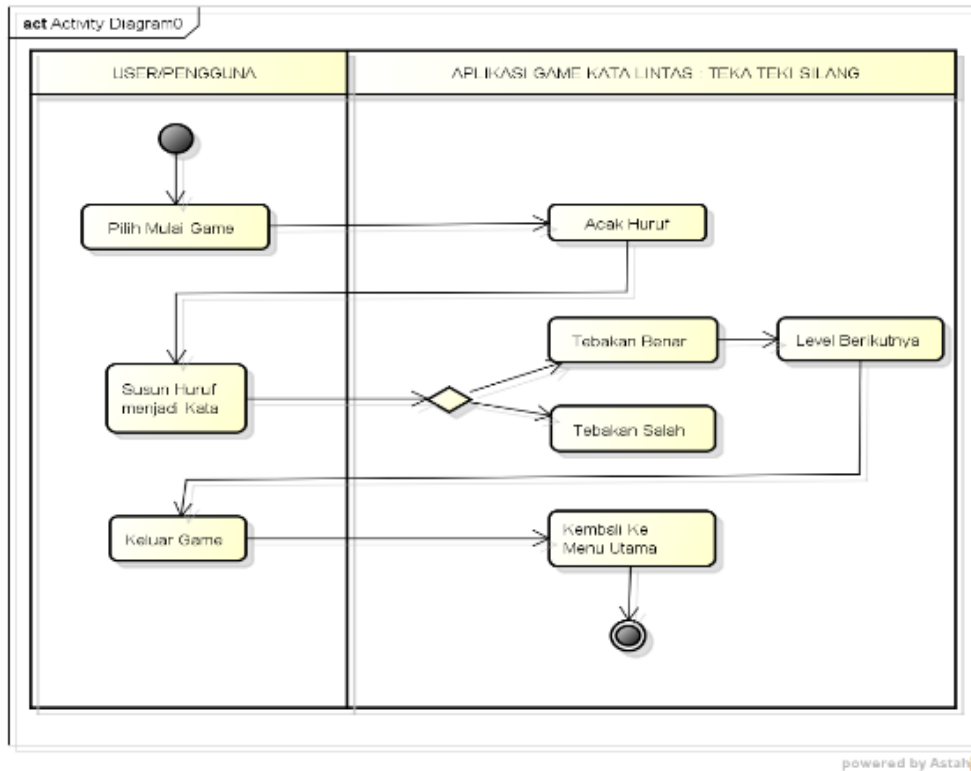


Gambar 5. Use Case Diagram Aplikasi Kata Silang

b. *Activity Diagram*

1. Activity Diagram Mulai Game

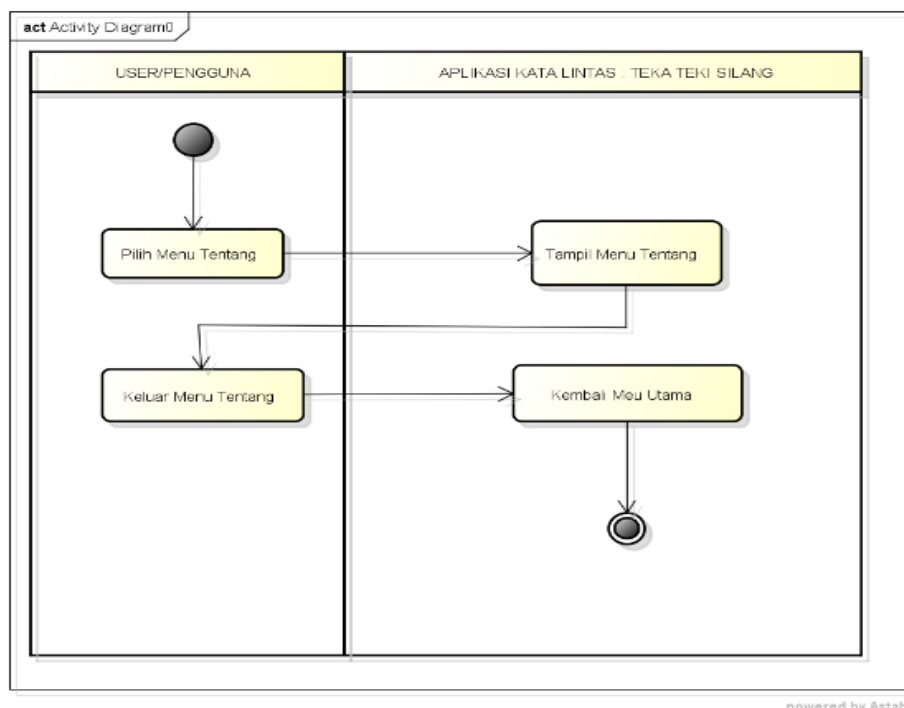
Dimulai pada saat *user* memilih menu Mulai *Game* lalu tampil *form* untuk Mulai *Game*, selanjutnya *user* menyusun huruf untuk mebuat sebuah kata dengan mengklik button huruf yang disediakan.



Gambar 6. Activity Diagram Mulai Game

2. Activity Diagram Tentang

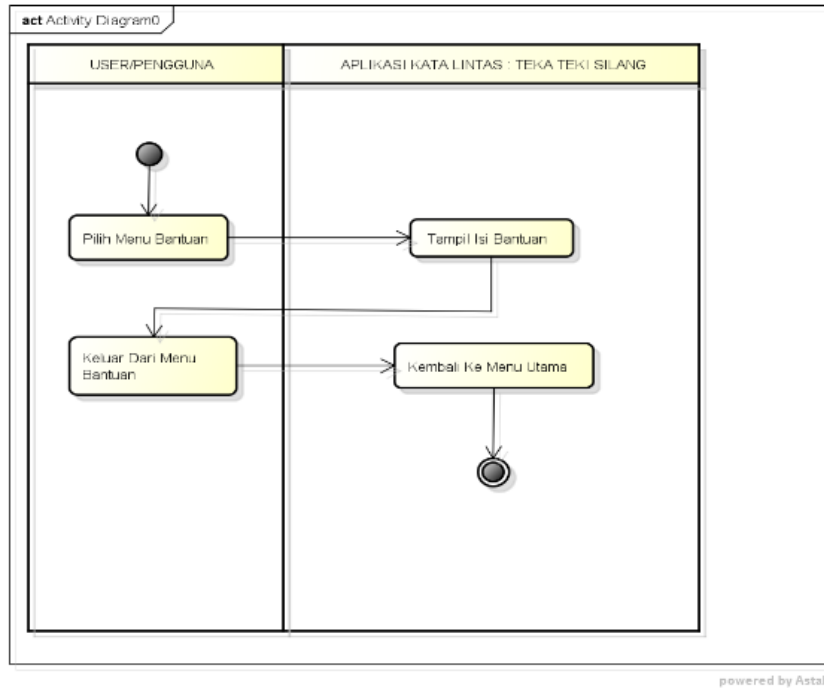
Rangkaian aktifitas form tentang dimulai pada saat pengguna memilih menu tentang kemudian tampil form untuk memberitahukan siapa perancang aplikasi kata lintas : Teka Teki Silang. selanjutnya user membaca form Tentang yang di tampilkan, setelah selesai membaca user memilih untuk keluar dari form Tentang.



Gambar 7. Activity Diagram Tentang

3. Activity Diagram Bantuan

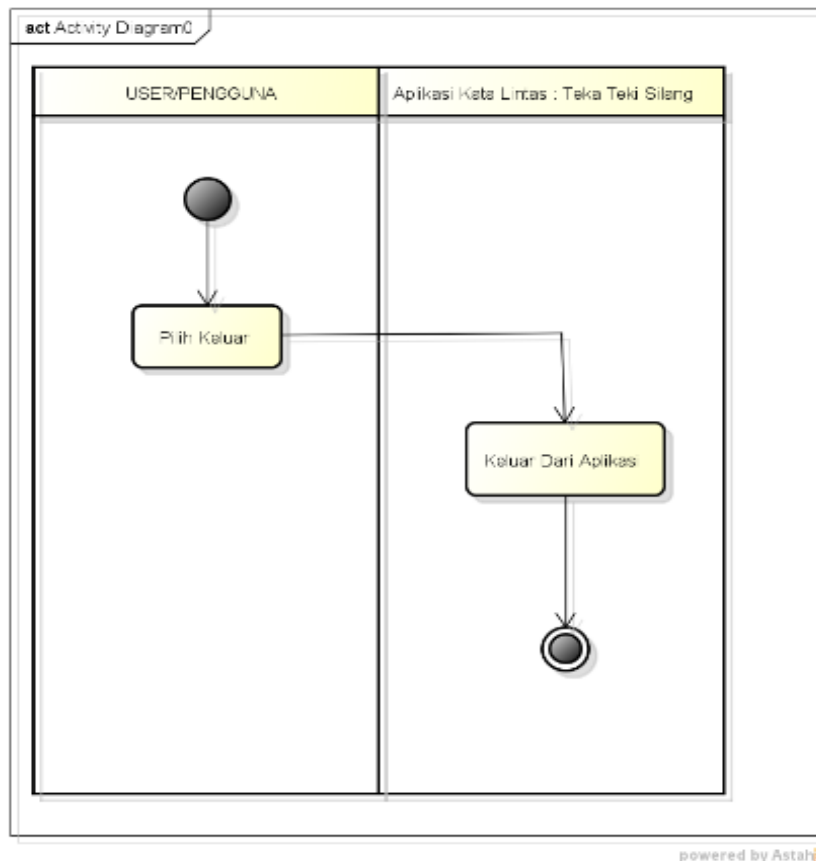
Rangkaian aktifitas form bantuan dimulai pada saat pengguna memilih menu bantuan dan kemudian tampil form untuk memberikan bantuan atau petunjuk cara permainan dari aplikasi game yang telah dirancang.



Gambar 8. Activity Diagram Bantuan

4. Activity Diagram Keluar

Aktifitas untuk form menu keluar ditunjukkan melalui dimana dalam proses rangkaian aktifitas proses Keluar dari aplikasi.

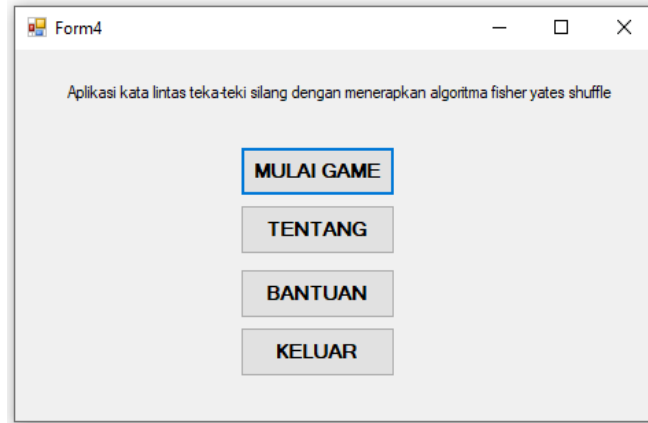


Gambar 8. Activity Diagram Keluar

3.3 Implementasi

a. Tampilan Menu Utama

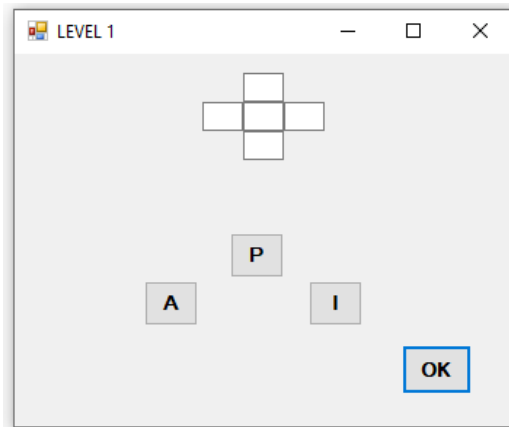
Tampilan yang muncul pertama sekali pada saat Aplikasi dijalankan. tampilan ini berisikan Rancangan menu utama menampilkan 3 (tiga) pilihan menu yaitu Mulai game, Tentang Bantuan dan Keluar. User memulai bermain dengan mengklik Mulai Bermain. Menu Bantuan berisikan keterangan cara bermain dan menu tentang menjelaskan biodata perancang, menu Exit digunakan untuk keluar dari permainan.



Gambar 9. Tampilan Menu Utama

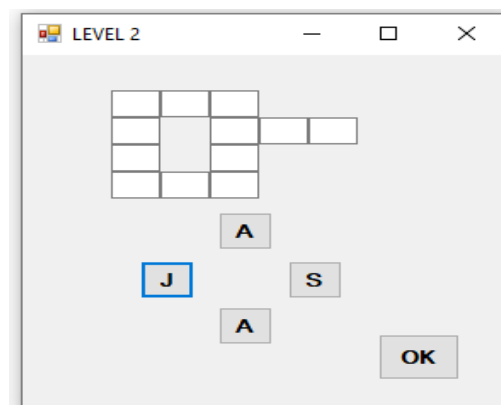
b. Tampilan Permainan

Pada aplikasi permainan yang dirancang terdiri atas beberapa level yang berbeda, dimana tingkat kesulitan serta jumlah Huruf yang diacak disesuaikan dengan tingkatan level yang ada, dimulai dari level 1 sampai level tertinggi yaitu level 3. Dimana level 1 terdiri atas 3 huruf yang dirangkai menjadi sebuah kata.



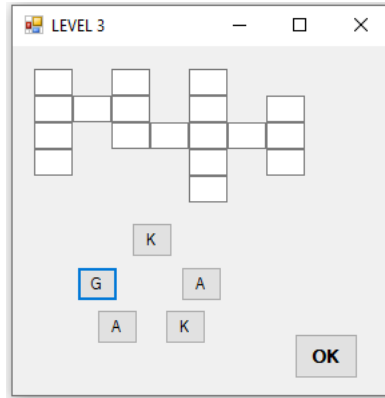
Gambar 10. Tampilan Permainan Level 1

Pada Level berikutnya yaitu level 2 terdiri atas 4 huruf yang harus dirangkai menjadi sebuah kata.



Gambar 11. Tampilan Permainan Level 2

Pada Level berikutnya yaitu level 3 terdiri atas 5 huruf yang harus dirancang menjadi sebuah kata.



Gambar 12. Tampilan Permainan Level 3

c. Tampilan Tentang

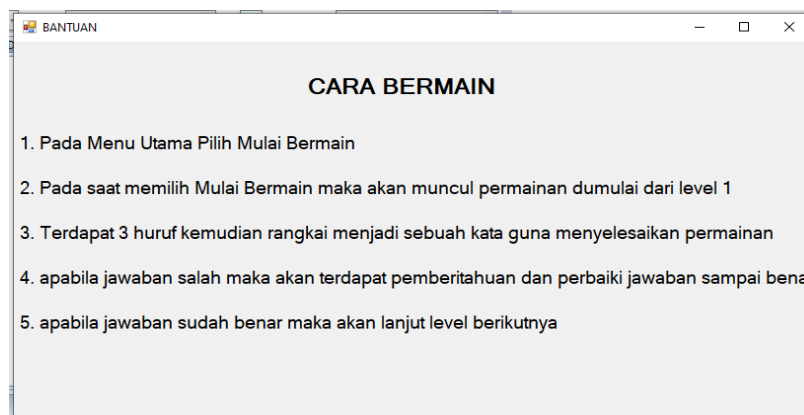
Pada Form ini berisi mengenai biodata dari penulis sekaligus perancang aplikasi Kata Lintas : Teka-Teki Silang. Dimana terdapat nama, npm, alamat, Foto, serta judul dari penelitian.



Gambar 13. Tampilan Tentang

d. Tampilan Bantuan

Pada Tampilan Form bantuan yang dirancang terdapat bantuan bagi pengguna yaitu berupa penjabaran mengenai cara bermain dari game yang dirancang.



Gambar 14. Tampilan Bantuan

4. KESIMPULAN

Saat ini banyak media pembelajaran yang digunakan dengan menerapkan teknologi, salah satunya melalui permainan (game). Meningkatkan kemampuan berfikir, meningkat kreatifitas dan menambah wawasan tentang makna suatu kata/kalimat sekarang ini bisa didapatkan hanya dengan bermain game. Salah satu permainan yang selain seru juga punya banyak manfaat bagi pemainnya adalah TTS (Teka-teki Silang) yang dimana cara bermain permainan ini adalah mengisi kotak-kotak putih dengan huruf, membentuk kata atau frasa tertentu, dengan menyelesaikan petunjuk yang mengarah ke



jawaban tertentu. Kotak yang berwarna hitam (atau warna lainnya) biasanya di gunakan untuk memisahkan kata atau frasa yang berbeda. Dalam pembuatan aplikasi ini algoritma yang digunakan adalah Fisher Yates Shuffle, dari hasil pengujian yang dilakukan pada Aplikasi maka dapat disimpulkan bahwa: prosedur pengacakan kata menggunakan Algoritma Fisher Yates Shuffle dilakukan dengan prosedur pertama yaitu penginisialisasian huruf menjadi angka guna mempermudah prosedur perhitungan menggunakan algoritma, setelah itu pengacakan beserta jumlah huruf diklasifikasikan berdasarkan level, penerapan algoritma Fisher Yates Shuffle dapat diterapkan dengan baik dimana pada aplikasi yang dirancang menghasilkan 3 level permainan..

REFERENCES

- [1] A. Ibijola and A. Olu, "A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures," *Int J Comput Appl*, vol. 54, no. 11, pp. 24-28, 2012.
- [2] M. A. Hasan, "Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online," vol. 7, no. 7, 2017.
- [3] E. R. Cahyaningrum and K. S. Batubulan, "IMPLEMENTASI METODE FISHER-YATES SHUFFLE DAN FUZZY TSUKAMOTO PADA GAME 2D GOPOH BERBASIS ANDROID," *Jurnal Informatika Polinema*, vol. 4, no. 3, p. 175, Mei 2018.
- [1] A. Ade-Ibijola, "A Simulated Enhancement Of Fisher-Yates Algorithm For Shuffling In Virtual Card Games Using Domain-Specific Data Structure," *Int. J. Comput. Appl.*, vol. 54, no. 11, hal. 24–28, 2012.
- [2] M. A. Hasan, "implementasi algoritma fisher yates untuk mengacak soal ujian online," vol. 7, no. 7, 2017.
- [3] S. Widiyanti, "perancangan aplikasi," *Peranc. Apl.*, vol. 53, no. 9, hal. 1689–1699, 2016, doi: 10.1017/CBO9781107415324.004.
- [4] IlmuKomputer.Org, "Aplikasi," *Defin. aplikasi*, vol. 3, hal. 8–31, 1385.
- [5] "Rancang Bangun Aplikasi Game Augmented Reality Permainan Tradisional Sulawesi Utara Dodorobe," *J. Tek. Inform. Univ. SAM RATULANGI*, vol. 12, no. 1, 2017, doi: 10.35793/jti.12.1.2017.17787.
- [6] E. F. Cahyadi, "Pemodelan dan Simulasi," in *Pemodelan dan Simulasi*, no. 3, 2006, hal. 2.
- [7] M. dan S. Fathoni, "Pengantar Algoritma dan Pemrograman," *Pemrograman*, vol. 4, no. 0644, hal. 10, 2015.
- [8] F. P. Juniawan dan H. Hengki, "Pengacakan Soal Ujian Penerimaan POLRI Menggunakan Algoritme Fisher Yates Shuffle," *Telematika*, vol. 12, no. 1, hal. 1, 2019, doi: 10.35671/telematika.v12i1.714.
- [9] J. Rumbaugh, "Unified Modeling Language (UML)," in *Encyclopedia of Software Engineering*, 2010, hal. 1249–1257.
- [10] C. Rupp, S. Queins, die SOPHISTen, C. Rupp, S. Queins, dan die SOPHISTen, "Use-Case-Diagramm," in *UML 2 glasklar*, 2012, hal. 239–262.