**LAPORAN TUGAS BESAR**

**16TIN1034 – Dasar-Dasar Pemrograman**

**PERMAINAN TIC TAC TOE**

****

**Disusun oleh :**

**Hafiz Maulana Azhar - 201524009**

**Muhammad Aziz Taufiqurrahman – 201524014**

**Program Studi D-IV Teknik Informatika**

**Departemen Teknik Komputer dan Informatika**

**Politeknik Negeri Bandung**

**2020/2021**

**1 Deskripsi Aplikasi**

* 1. **Permainan Tic-Tac-Toe secara umum**

Tic-tac-toe adalah sebuah permainan di mana seorang pemain menggambarkan X dan pemain lainnya menggambarkan O ke dalam suatu bidang dengan tujuan memenuhi sebuah baris dengan X atau O. Bidang tersebut pada umumnya adalah sebuah persegi yang tersusun dari 9 buah persegi lebih kecil (panjang 3 persegi dan lebar 3 persegi). Tentunya ukuran bidang ini bisa diperluas menjadi 5x5 atau 7x7. Namun, permainan dengan ukuran bidang lebih dari 3x3 tersebut kerap memiliki hasil seri. Baris yang dibentuk dalam hal ini bisa berupa sebuah baris (horisontal), sebuah kolom (vertikal), ataupun menyilang (diagonal). Jika seorang pemain berhasil mengisi sebuah baris dengan 3 bidang (3x3), 4 bidang (5x5), dan 5 bidang (7x7) secara horizontal, vertical, atau diagonal, maka permainan Tic-tac-toe ini dinyatakan selesai dengan kemenangan pemain tersebut. Tic-tac-toe sendiri termasuk ke dalam jenis *zero sum game*. *Zero sum game* adalah jenis permainan di mana jika seorang pemain menang maka pemain lainnya dinyatakan kalah.

|  |  |  |
| --- | --- | --- |
| X |  |  |
| O | X |  |
| O |  | X |

Ilustrasi aturan permainan dapat dilihat pada gambar berikut:

*Gambar 1(a) Pemain X menang secara diagonal*

|  |  |  |
| --- | --- | --- |
| X | X | X |
|  | O |  |
| O |  |  |

*Gambar 1(b) Pemain X menang secara horizontal*

|  |  |  |
| --- | --- | --- |
| X |  |  |
| X | O | O |
| X |  |  |

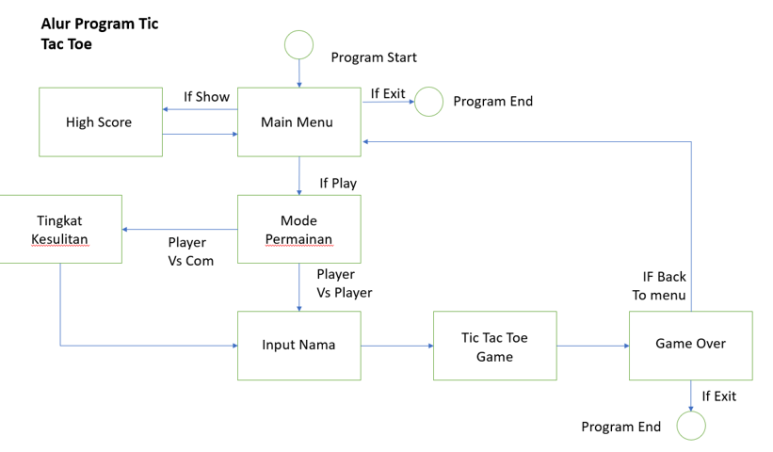
*Gambar 1(c). Pemain X menang secara vertikal*

|  |  |  |
| --- | --- | --- |
| X | X | O |
| O | O | X |
| X | O | X |

*Gambar 1(d). Permainan berakhir seri*

* 1. **Skenario Game**

**Alur Permainan**



*Gambar 1(e). Alur Program Tic Tac Toe*

Adapun tampilan menu utama pada aplikasi ini, yaitu mulai permainan, high score, cara bermain, keluar dari permainan. Pada menu utama, user diminta untuk menginputkan pilihannya berdasarkan kriteria yang tersedia.

|  |
| --- |
| **Menu Utama**  **Selamat Datang!**  **Mau mulai dari mana nih?**  **1. Mulai Permainan**  **2. High Score**  **3. Cara Bermain**  **4. Keluar Permainan**  **Masukkan pilihan Anda:** |

*Gambar 1(f). Tampilan Awal Menu Utama*

Berikut ini merupakan penjelasan dari tiap bagian yang terdapat pada pilihan menu utama:

1. **Mulai Permainan**

|  |
| --- |
| **Mulai Permainan**  **Pilihlah Mode Permainan:**   1. **Komputer** 2. **Multipemain**   **Masukan pilihan Anda:** |

*Gambar 1(g). Tampilan Mode Permainan*

Apabila user memilih mode permainan “multipemain”, maka nama pemain pertama dengan nama pemain kedua harus diinputkan. Berikut merupakan ilustrasi dari proses input nama pemain *(user)*.

|  |
| --- |
| **Multipemain**  **Masukkan nama pemain pertama.**  **Nama pemain pertama:** |

*Gambar 1(h). Tampilan Input Nama Pemain Pertama*

|  |
| --- |
| **Multipemain**  **Masukkan nama pemain kedua.**  **Nama pemain kedua:** |

*Gambar 1(i). Tampilan Input Nama Pemain Kedua*

|  |
| --- |
| **Multipemain**  **Nama pemain pertama : Player1**  **Nama pemain kedua : Player2** |

*Gambar 1(j). Tampilan Nama Para Pemain*

Selanjutnya setelah memilih mode permainan, maka user akan diarahkan untuk memilih ukuran papan permainan tic-tac-toe, terdapat ukuran 3x3, 5x5, serta 7x7.

|  |
| --- |
| **Permainan Baru Nih!**  **Pilihlah Ukuran Papan Tic-Tac-Toe**  **1. 3 x 3**  **2. 5 x 5**  **3. 7 x 7**  **Masukkan pilihan anda:** |

*Gambar 1(k). Tampilan Pilihan Ukuran Papan Permainan*

Ilustrasi tampilan papan permainan dengan ukuran 3x3, 5x5, dan 7x7 :

|  |
| --- |
| **Permainan 3 x 3**  **\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_**  **| | | |**  **| 1 | 2 | 3 |**  **|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|**  **| | | | Simbol Anda : X**  **| 4 | 5 | 6 | Simbol Lawan : O**  **|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|**  **| | | |**  **| 7 | 8 | 9 |**  **|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|**  **Masukkan posisi petak:** |

*Gambar a. Ilustrasi Permainan 3x3 (9 petak bidang)*

|  |
| --- |
| **Permainan 5 x 5**  **\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_**  **| | | | | |**  **| 1 | 2 | 3 | 4 | 5 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | Simbol Anda : X**  **| 6 | 7 | 8 | 9 | 10 | Simbol Komputer : O**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | |**  **| 11 | 12 | 13 | 14 | 15 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | |**  **| 16 | 17 | 18 | 19 | 20 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | |**  **| 21 | 22 | 23 | 24 | 25 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **Masukkan posisi petak:** |

*Gambar b. Ilustrasi Permainan 5x5 (25 petak bidang)*

|  |
| --- |
| **Permainan 7 x 7**  **\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_**  **| | | | | | | |**  **| 1 | 2 | 3 | 4 | 5 | 6 | 7 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_| Simbol Anda : X**  **| | | | | | | | Simbol Komputer : O**  **| 8 | 9 | 10 | 11 | 12 | 13 | 14 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | | |**  **| 15 | 16 | 17 | 18 | 19 | 20 | 21 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | | |**  **| 22 | 23 | 24 | 25 | 26 | 27 | 28 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | | |**  **| 29 | 30 | 31 | 32 | 33 | 34 | 35 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | | |**  **| 36 | 37 | 38 | 39 | 40 | 41 | 42 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **| | | | | | | |**  **| 43 | 44 | 45 | 46 | 47 | 48 | 49 |**  **|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|\_\_\_\_\_\_|**  **Masukkan posisi petak:** |

*Gambar c. Ilustrasi Permainan 7x7 (49 petak bidang)*

Jika setiap langkah pada menu “Mulai Permainan” telah terisi, maka permainan dimulai. Pada program aplikasi yang kami buat, batas waktu user dalam memilih pilihannya diberikan durasi selama 10 detik, jika lebih dari batas waktu maka giliran permainan akan diberikan kepada pihak lawan.

Pada awalnya semua bidang tidak terisi dengan X atau O (kosong). Kemudian pemain & komputer (lawan) mendapatkan giliran bermain sesuai dengan aturan yang sudah ditentukan. Ketika giliran seorang pemain tiba, maka pemain tersebut harus meletakan posisinya pada sebuah bidang yang masih kosong (belum terisi). Permainan terus berlanjut sehingga terdapat pemain yang mampu menyelesaikan game dengan cara diagonal (lihat Gambar 1(a)), cara horizontal (lihat Gambar 1(b)), cara vertical (lihat Gambar 1(c)), ataupun tidak ada pemenang sama sekali (seri) (lihat Gambar 1 (d)).

**Strategi/Langkah Bermain Tic-Tac-Toe**

Untuk Tic-tac-toe tanpa modifikasi, strategi langkah yang bisa diambil untuk memenangkan sebuah permainan adalah sebagai berikut :

1. Jika memungkinkan untuk memanangkan permaian maka menangkan permainan tersebut.
2. Jika memungkinan melakukan *block* maka lakukan *block*. *Block* adalah mencegah lawan memenangkan permainan.
3. Jika memungkinan melakukan *fork* maka lakukan *fork*. *Fork* adalah langkah yang memungkinkan dua cara memenangkan permainan pada langkah berikutnya. Langkah ini akan membuat lawan tidak dapat melakukan *block*.
4. Jika memungkinan melakukan *block fork* maka lakukan *block fork*. *Block fork* adalah langkah melakukan untuk mencegah lawan melakukan *fork* pada langkah berikutnya.
5. Jika memungkinkan mengisi bagian tengah maka isi pada bagian tengah.
6. Jika memungkinkan mengisi pojok berlawanan maka isi pada pojok berlawanan. Jika suatu pojok diisi oleh lawan, maka isikan pojok yang berlawanan dengan sisi tersebut.
7. Jika memungkinkan mengisi pojok maka isi pada pojok.
8. Jika memungkinkan mengisi bagian sisi maka isi pada bagian sisi.

**Aturan Permainan Tic-Tac-Toe**

Aturan sangat diperlukan dalam sebuah kegiatan, salah satunya ialah permainan. Dengan adanya aturan, maka permainan bisa berjalan dengan baik, tertib serta memiliki tujuan yang jelas. Tic-tac-toe memiliki beberapa aturan, ialah :

* Pemain pertama menempatkan posisinya pada sebuah bidang (bidang yang belum terisi)
* Pemain kedua tidak diperbolehkan menyimpan posisinya pada bidang yang sudah terisi
* Pemain diberikan selang waktu tertentu dalam menentukan posisinya
* Jika pemain melebihi batas waktu yang ditentukan untuk menentukan posisi, maka giliran bermain akan diberikan kepada lawannya
* Jika sudah ada pemain yang dinyatakan menang, maka pemain yang lain dinyatakan kalah

Di dalam sebuah permainan harus terdapat *start* (permainan dimulai) dan *game over* (permainan berakhir), pada game Tic-tac-toe ini permainan akan dinyatakan *game over* (permainan berakhir) apabila memenuhi syarat berikut :

* Permainan dinyatakan menang apabila 3 bidang terisi penuh dengan bidaknya (ukuran 3x3), 4 bidang terisi penuh dengan bidaknya (ukuran 5x5) dan 5 bidang terisi penuh dengan bidaknya (7x7), baik secara horizontal, vertical, maupun diagonal

1. **High Score**

Data ini bertujuan untuk menampung tipe data highscore, seperti winner, duration, dan boardsize. Data ini memiliki tipe struct yang didalamnya memiliki tiga field dimana 2 di antaranya merupakan variabel tunggal bertipe integer dan satu lagi merupakan array bertipe karakter.

1. **Cara Bermain Program Aplikasi Tic-Tac-Toe**

Pada game Tic-tac-toe ini ada dua pemain yang bermain, satu user dan satu lagi komputer. Urutan giliran akan dimulai dengan user memasukan pilihan posisinya pada bidang, selanjutnya akan diikuti oleh komputer, hal tersebut terus dilakukan sampai salah satu dari keduanya bisa menyusun pada bidangnya baik secara horizontal, vertical, ataupun diagonal. Tapi sebelum memulai permainannya untuk menentukan yang pertama atau yang kedua hanya pada awal sebelum game ini dimulai terdapat permainan kertas gunting dan batu untuk menentukan yang pertama jalan terlebih dahulu.

Adapun cara bermainnya pada program ini adalah :

* User tidak usah memasukan nama, akan tetapi sudah tercantum tulisan “Player 1” dengan pilihan bidaknya ialah *X* dan “Player 2” adalah komputer dengan pilihan bidaknya ialah *O*.
* Setelah itu untuk menentukan siapa yang akan mengisi bidak terlebih dahulu akan dilakukan permainan gunting kertas dan batu untuk menentukan siapa yang akan memulai terlebih dahulu untuk mengisi bidak.
* Setelah itu pemain dan AI computer bergantian untuk mengisi bidak yang sudah tersedia.
* Setiap pemain tidak diperbolehkan menempatkan posisi bidaknya pada bidang yang sudah terisi.
* Permainan dikatakan menang jika berhasil menyusun bidaknya, bisa secara horizontal, vertical, ataupun diagonal.
* Kemudian terdapat pilihan jika ingin memulai kembali permainan setelah permainan dinyatakan selesai (sudah ada pemenang permainan).

**Penentuan Giliran Bermain (Gunting-Batu-Kertas)**

Sebelum memulai putaran pertama, dilakukan permainan gunting-kertas-batu untuk menentukan urutan pemain. Jika user berhasil memenangkan permainan gunting-kertas-batu, maka user akan mendapat giliran main pertama. Jika user kalah dalam permainan gunting-kertas-batu, maka user akan mendapat giliran main kedua.

Cara bermain gunting-kertas-batu dalam program ini adalah dengan menginput angka ‘1’, angka ‘2’, atau angka ‘3’. Angka ‘1’ adalah pilihan untuk memilih ‘gunting’. Angka ‘2’ adalah pilihan untuk memilih ‘batu’. Angka ‘3’ adalah pilihan untuk memilih ‘kertas’. Setelah menginput angka yang diinginkan, user menekan tombol “enter”. Kemudian akan muncul hasil permainan gunting-kertas-batu. Jika terjadi kondisi seri, maka permainan gunting-kertas-batu akan diulangi hingga terdapat kondisi menang atau kalah.

**Penentuan Pemenang Permainan**

Ketika permainan sedang berjalan, perlu diperiksa apakah sudah ada pemain yang berhasil menyusun bidaknya pada bidang secara horizontal, vertical, ataupun diagonal (3 bidang karena rasio papannya 3x3, 4 bidang karena rasio papan 5x5, dan 5 bidang dengan rasio 7x7). Jika sudah ada, maka permainan berakhir dan pemain yang bersangkutan dinyatakan sebagai pemenang permainan. Pada program ini yang akan ditampilkan hanya pemenang permainan saja.

1. **Keluar Permainan**

Jika user ingin mengakhiri program, maka user bisa memilih pilihan “keluar permainan” pada menu utama dengan catatan permainan harus sudah mendapatkan pemenang. Jika belum mendapatkan pemenang, maka user tidak bisa memilih “keluar permainan”, karena pada program yang kami buat pemain harus menyelesaikan sampai permainan mendapatkan juara. Adapun ilustrasi tampilannya sebagai berikut:

|  |
| --- |
| **Apakah anda yakin ingin keluar dari permainan?**  **Y: Ya | N: Tidak**  **Masukan pilihan anda:** |

*Gambar 1(l). Tampilan Pilihan Keluar Program*

**Tingkat Kesulitan**

Karakter komputer menggambarkan tingkah laku dari komputer sebagai pemain yang ditentukan. Dalam menentukan isi dari petak yang dilakukan oleh komputer selaku pemain, komputer akan memperhitungkan pilihannya berdasarkan pada tingkat kesulitan dari permainan yang dijelaskan sebagaimana berikut:

1. Mode Mudah (*Easy*)

Komputer menempatkan simbol secara acak pada papan permainan. Pada tingkat kesulitan mudah, komputer memiliki kemampuan untuk memilih langkah secara acak. Tingkat kemenangan user melawan komputer dengan tingkat kesulitan mudah adalah 80% menang, 15% seri, dan 5% kalah.

1. Mode Normal (*Normal*)

Komputer menempatkan simbol secara acak pada papan, namun komputer memiliki peluang untuk memilih petak terbaik untuk menempatkan simbolnya. Pada tingkat kesulitan menengah, komputer memiliki kemampuan untuk memilih langkah secara acak serta menghalangi pergerakan lawan. Namun, pergerakan komputer diprioritaskan untuk memenangkan dirinya sendiri. Tingkat kemenangan user melawan komputer dengan tingkat kesulitan sedang adalah 40% menang, 30% seri, dan 30% kalah.

1. Mode Sulit (*Hard*)

Komputer menempatkan simbol pada papan berdasarkan dengan perhitungan untuk memilih petak terbaik sehingga tidak memungkinkan untuk menempatkan simbol secara acak. Pada tingkat kesulitan sulit, komputer memiliki kemampuan untuk memenangkan komputer itu sendiri serta menghalangi pergerakan lawan. Namun, pergerakan komputer diprioritaskan untuk menghalangi pergerakan lawan. Tingkat kemenangan user melawan komputer dengan tingkat kesulitan sulit adalah 5% menang, 50% seri, dan 45% kalah.

1. **Design dan Perancangan** 
   1. **Teori *Minimax* (Teori Pemain Komputer)**

Algoritma Minimax adalah sebuah algoritma yang mendasari pola pikir langkah penyelesaian masalah dalam beberapa jenis permainan papan yang berbasis kecerdasan buatan dalam komputer, seperti tic-tac-toe. Algoritma minimax merupakan algoritma pencarian yang akan melakukan penelusuran pada setiap node hingga diperoleh nilai maksimum. Kecerdasan buatan merupakan suatu sistem dengan menggunakan algoritma tertentu sehingga sistem tersebut seolah-olah dapat berfikir seperti manusia. Tic-tac-toe menerapkan algoritma minimax. Game ini menerapkan kecerdasan buatan dengan menggunakan metode minimax yang akan memberikan peluang bagi komputer untuk mengambil langkah terbaik. Game tic-tac-toe yang dirancang terdiri dari 3 level, level mudah, sedang dan sulit dimana setiap levelnya memiliki 3 bidak pilihan ukuran papan yaitu 3 x 3, 5 x 5, dan 7 x 7. Setiap akhir permainan diperoleh skor akhir dan waktu permaian yang digunakan selama permainan game tic-tac-toe ini.

Algoritma Minimax digunakan untuk menentukan pilihan agar dapat memperkecil kemungkinan kehilangan nilai maksimal, yang akan meminimalkan kemungkinan kekalahan dan memaksimalkan kemungkinan kemenangan. Algoritma ini diterapkan dalam game yang dimainkan oleh dua pemain dan berbasis zero-sum seperti game tic-tac-toe yaitu suatu kondisi apabila terdapat pemain yang mengalami keuntungan, pemain lain akan mengalami kerugian senilai dengan keuntungan yang diperoleh lawan dan sebaliknya.

Algoritma minimax akan melakukan pengecekan pada seluruh kemungkinan yang ada, sehingga akan menghasilkan pohon game yang berisi semua kemungkinan game tersbut. Algoritma Minimax dapat menghasilkan pilihan langkah yang baik dengan mengasumsikan bahwa pemain lawan akan selalu memilih langkah terbaik untuk dirinya dan terburuk bagi komputer. Prinsip dasar pada algoritma Minimax ini adalah jalur yang akan dipilih oleh komputer merupakan jalur maksimal (max node) yang akan menghasilkan nilai maksimum di jalur tersebut, dan saat lawan akan bermain akan meminimalkan (min node) nilai komputer. Jadi, komputer bertujuan untuk memaksimalkan kemungkinan nilai rendah yang akan diperoleh komputer.

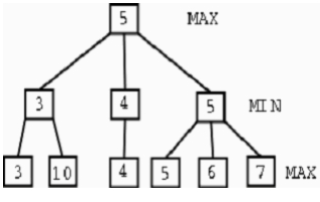
1. **Implementasi Algoritma *Minimax***

Algoritma Minimax digunakan untuk menentukan pilihan agar dapat memperkecil kemungkinan kehilangan nilai maksimal, yang akan meminimalkan kemungkinan kekalahan dan memaksimalkan kemungkinan kemenangan.

Pada permainantic-tac-toe ini mempunyai lebih sedikit rule dan knowledge base yang cenderung lebih sedikit. Dari rule yang ada mampu menghasilkan solusi-solusi yang dapat digunakan untuk memenangkan permainan. Dari rule yang sedikit, computer mampu melakukan komputasi untuk memainkan setiap kombinasi langkah darisetiap posisi dan kondisi. Namun hal ini dapatdihindari dengan membatasi sejauh mana komputer akan menganalisis hasil dari langkah langkah yang mungkin. Tetapi dengan hal ini, kita harus menambah kedalaman setiap langkahnya agar kedalaman pada state tersebut sama dengan state sebelumnya Algorit maminimax ini bekerja secara rekursif dengan mencari langkah yang akan membuatlawan mengalami keuntungan minimum. Semua strategi lawan akan di hitung dengan algoritma yang sama dan seterusnya. Ini berarti, pada langkah pertama komputer akan menganalisis seluruh kemunngkinan pada permainan. Dan untuk setiap langkahnya, komputer akan memilih langkah yang membuat lawan mendapatkan keuntungan minimum, dan membuat komputer itu sendiri mendapatkan keuntungan maksimum.

* 1. **Sistem Pemilihan Langkah Teori *Minimax***

Pemilihan langkah AI dapat diimplementasikan dengan algoritma *minimax*. Setiap langkah yang dapat dilakukan oleh AI akan diberi nilai. Semakin tinggi nilainya berarti langkah tersebut akan lebih dekat ke kemenangan.

******

Pada kasus seperti pada gambar tersebut, Sebagai contoh, ada 1 orang pemain yang mencoba bermain melawan komputer (pemain = A dankomputer = B).Apabila pemain A dapat memenangkan permainan dalam 1 langkah, maka pemain A akan menang. Pemain B yang mengetahui bahwa langkah tersebut akan membuat pemain A memenangkan permainan, dan di sisi lain ada langkah lain yang akan membuat permainan menjadi seri, maka pilihan paling logis untuk pemain B adalah langkah yang akan membuat permainan menjadi seri.Pada tahapan algoritma ini kita mengasumsikan bahwa pemain A mencoba untuk memaksimalisasi peluang permainan agar ia dapat menang. Disisi lain, pada giliran berikutnya pemain B juga berusaha mencoba untuk meminimalisir peluang kemungkinan menang pemain A. Oleh sebab itu,pemainA dapat disebut sebagai maximizing player(MAX) dan B dapat disebut sebagai minimizing player(MIN).Sehingga pembentukan pohon pencarian solusi yang digunakan akan memakai konsep depth-first,yang akan dimulai dari awal permainan hingga akhir permainan. Setelah itu, posisiterakhir permainan akan dievaluasi melalui sudut pandang MAX.

* 1. **Kebutuhan Data**

Kebutuhan (Requirements) adalah proses mengidentifikasi & mendokumentasikan data yang dibutuhkan oleh user dalam sebuah database untuk memenuhi kebutuhan informasi saat ini dan masa yang akan datang.

Dua Jenis informasi yang harus diperhatikan:

1. Informasi yang menjelaskan struktur data, seperti entitas, atribut, dan relasi. Informasi ini biasanya dinyatakan dalam bentuk grafik seperti entity-relationship diagrams (E-RD).

2. Informasi yang menggambarkan aturan atau batasan yang dapat menjaga integritas data. Biasanya disebut aturan (rules), batasan-batasan ini harus di tuangkan dalam data dictionary (atau repository) suatu organisasi.

Pada game tic-tac-toe itu kebutuhan data yang dibutuhkan diantaranya:

1. Data Pemain

Data permainan di sini bertujuan untuk menyimpan data pemain, seperti:

* Nama pemain 1 dan pemain 2 (jika memilih mode multipemain), menyimpan nama pemain 1 (jika memilih mode komputer). Tipe data untuk data pemain ini ialah tipe data char.
* Simbol pemain, pada permainan ini pemain pertama disimbolkan dengan X dan pemain kedua disimbolkan dengan O.
* Score permainan yang akan digunankan untuk menghitung score para pemain, bertipe integer.
* Input, berfungsi untuk meletakkan posisi pada petak bertipe integer

1. Data Papan Permainan

Data ini bertujuan untuk menampung tipe data papan bermain, seperti simbol dan ukuran papan permainan. Data ini berbentuk pointer of pointer yang akan digunakan sebagai representasi array 2d yang ukurannya dapat dialokasikan secara manual.

* Ukuran papan berfungsi untuk menginputkan pilihan papan permainan, papan permainan bertipe integer, terdapat 3 macam ukuran papan permainan yaitu 3x3, 5x5, dan 7x7
* Tingkat kesulitan, jika memilih mode komputer. Terdapat tiga pilihan, yaitu mudah, sedang, dan sulit. Tingkat kesulitan ini bertipe integer.

1. Data Mode Permainan

Data  ini bertujuan  untuk menampung mode permainan. Apakah bermain dengan komputer atau dengan manusia. Data ini merupakan sebuah variabel tunggal dengan tipe integer dengan nilai yang diharapkan 1 berarti bermain dengan manusia, 2 berarti bermain dengan komputer.

1. Data Highscore

Data ini bertujuan untuk menampung tipe data highscore, seperti winner, duration, dan boardsize.

1. Data Waktu

Waktu yang masih tersedia bagi pemain untuk mengisi posisi petak permainan, bertipe integer.

* 1. **Modul Program**

Modul merupakan kumpulan dari proses yang bertujuan untuk melakukan satu aktivitas tertentu. Terdapat dua tipe modul, yaitu fungsi yang memiliki tugas untuk memberikan nilai dan prosedur yang memiliki tugas untuk mengubah kondisi suatu entitas. Berikut ini merupakan daftar modul program yang ada pada aplikasi Tic-Tac-Toe.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Modul** | **Jenis Modul** | **Deskripsi Modul** |
| 1 | fullscreen | Procedure | Modul untuk membuat console windows menjadi layar penuh |
| 2 | gotoxy | Procedure | Modul untuk memfungsikan fungsi gotoxy |
| 3 | loading | Procedure | Modul untuk menampilkan tampilan loading |
| 4 | timer | Procedure | Modul untuk mengatur kecepatan tampilnya output pada layar |
| 5 | Judul | Procedure | Modul untuk menampilkan judul permainan |
| 6 | MenuUtama | Procedure | Modul untuk menampilkan menu utama dalam permainan. |
| 7 | CaraBermain | Procedure | Modul untuk menampilkan petunjuk/cara memainkan permainan. |
| 8 | showScore | Procedure | Modul untuk menampilkan peringkat skor dalam permainan. |
| 9 | AboutUs | Procedure | Modul untuk menampilkan tampilan tentang permainan |
| 10 | PilihLevel | Procedure | Modul untuk menampilkan pilihan tingkat kesulitan yang tersedia. |
| 11 | Mudah3 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan mudah pada papan 3x3. |
| 12 | Mudah5 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan mudah pada papan 5x5. |
| 13 | Mudah7 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan mudah pada papan 7x7. |
| 14 | Menengah3 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan menengah pada papan 3x3. |
| 15 | Menengah5 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan menengah pada papan 5x5. |
| 16 | Menengah7 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan menengah pada papan 7x7. |
| 17 | Sulit3 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan sulit pada papan 3x3. |
| 18 | Sulit5 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan sulit pada papan 5x5. |
| 19 | Sulit7 | Procedure | Modul untuk menampilkan permainan dengan tingkat kesulitan sulit pada papan 7x7. |
| 20 | PilihPapan | Procedure | Modul untuk menampilkan pilihan papan yang tersedia |
| 21 | board3 | Procedure | Modul untuk menampilkan papan permainan berukuran 3x3. |
| 22 | board5 | Procedure | Modul untuk menampilkan papan permainan berukuran 5x5. |
| 23 | Board7 | Procedure | Modul untuk menampilkan papan permainan berukuran 7x7. |
| 24 | GuntingKertasBatu | Procedure | Modul untuk memainkan kertas-gunting-batu. |
| 25 | gridChar | Function | Modul untuk menampilkan simbol di  dalam papan |
| 26 | CekMenang | Function | Modul untuk memeriksa apakah dalam satu permainan pada papan 3x3 sudah terdapat pemenang atau belum. |
| 27 | CekMenang5 | Function | Modul untuk memeriksa apakah dalam satu permainan pada papan 5x5 sudah terdapat pemenang atau belum. |
| 28 | CekMenang7 | Function | Modul untuk memeriksa apakah dalam satu permainan pada papan 7x7 sudah terdapat pemenang atau belum. |
| 29 | SortScore | Procedure | Modul untuk mengurutkan skor-skor dalam papan peringkat |
| 30 | LangkahPemain3 | Procedure | Modul untuk menampilkan langkah pemain di layar pada papan 3x3. |
| 31 | LangkahPemain5 | Procedure | Modul untuk menampilkan langkah pemain di layar pada papan 5x5. |
| 32 | LangkahPemain7 | Procedure | Modul untuk menampilkan langkah pemain di layar pada papan 7x7. |
| 33 | LangkahKomputer Mudah3 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 3x3 |
| 34 | LangkahKomputer Mudah5 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 5x5 |
| 35 | LangkahKomputer Mudah7 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 7x7 |
| 36 | LangkahKomputer Sulit3 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 3x3 |
| 37 | LangkahKomputer Sulit5 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 5x5 |
| 38 | LangkahKomputer Sulit7 | Procedure | Modul untuk menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 7x7 |
| 39 | Minimax | Function | Modul untuk mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 3x3 |
| 40 | Minimax5 | Function | Modul untuk mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 5x5 |
| 41 | Minimax7 | Function | Modul untuk mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 7x7 |
| 42 | HitungSkor | Procedure | Modul untuk menghitung skor setelah permainan berakhir |
| 43 | DisplayPemain3 | Procedure | Modul untuk menampilkan tampilan permainan board 3x3 |
| 44 | DisplayPemain5 | Procedure | Modul untuk menampilkan tampilan permainan board 5x5 |
| 45 | DisplayPemain7 | Procedure | Modul untuk menampilkan tampilan permainan board 7x7 |
| 46 | GetLevel | Procedure | Modul untuk memanggil modul permainan setelah bermain gunting-kertas-batu |
| 47 | SimpanSkor | Procedure | Modul untuk menginput nama pemain dan menyimpan nama pemain beserta skor ke dalam file KumpulanSkor.txt |

**Spesifikasi Modul**

|  |  |
| --- | --- |
| **Nama Modul** | Fullscreen |
| **Jenis Modul** | Procedure |
| **I.S** | Tampilan layar output windows belum penuh |
| **F.S** | Tampilan layar output windows sudah penuh |
| **Algoritma** | |
| void fullscreen(){  keybd\_event(VK\_MENU,0x38,0,0);  keybd\_event(VK\_RETURN,0x1c,0,0);  keybd\_event(VK\_RETURN,0x1c,KEYEVENTF\_KEYUP,0);  keybd\_event(VK\_MENU,0x38,KEYEVENTF\_KEYUP,0);  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Gotoxy |
| **Jenis Modul** | Procedure |
| **I.S** | Tampilan belum bisa diatur |
| **F.S** | Tampilan sudah bisa diatur |
| **Algoritma** | |
| void gotoxy(int x, int y){    HANDLE hConsoleOutput;  COORD dwCursorPosition;  dwCursorPosition.X = x;  dwCursorPosition.Y = y;  hConsoleOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);  SetConsoleCursorPosition(hConsoleOutput,dwCursorPosition);  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Loading |
| **Jenis Modul** | Procedure |
| **I.S** | Ditampilkan tampilan modul GuntingKertasBatu/level yang tersedia |
| **F.S** | Ditampilkan tampilan modul GuntingKertasBatu/level yang tersedia |
| **Algoritma** | |
| void loading() {    /\*BEGIN PROCEDURE LOADING\*/  int x;  gotoxy(75,20); printf("Harap Tunggu..");  for(x=65;x<=67;x++){  gotoxy(x,22);printf("Û\n");  timer(0.5);  }    for(x=68;x<=75;x++){  gotoxy(x,22);printf("Û\n");  timer(0.1);  }    for(x=76;x<=86;x++){  gotoxy(x,22);printf("Û\n");  timer(0.02);  }    for(x=87;x<=97;x++){  gotoxy(x,22);printf("Û\n");  timer(0.1);  }    /\*END PROCEDURE LOADING\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Timer |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Mengatur kecepatan tampilnya output pada layar |
| **Algoritma** | |
| void timer(float persentase) {  clock\_t endwait;  endwait=clock()+persentase\*CLOCKS\_PER\_SEC;  while(clock()<endwait){};  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Judul |
| **Jenis Modul** | Procedure |
| **I.S** | Layar menampilkan blankpage |
| **F.S** | Layar menampilkan judul permainan |
| **Algoritma** | |
| void Judul(){  /\*BEGIN PROCEDURE\_JUDUL\*/    /\*baris 1\*/  gotoxy(58,5); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");  gotoxy(76,5); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");  gotoxy(94,5); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");    /\*baris 2\*/  gotoxy(58,6); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");  gotoxy(76,6); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");  gotoxy(94,6); printf("ÛÛÛÛÛÛÛÛÛÛÛÛÛ\n");    /\*baris 3\*/  gotoxy(62,7); printf("ÛÛÛÛÛ\n");  gotoxy(80,7); printf("ÛÛÛÛÛ\n");  gotoxy(98,7); printf("ÛÛÛÛÛ\n");    /\*baris 4\*/  gotoxy(62,8); printf("ÛÛÛÛÛ\n");  gotoxy(80,8); printf("ÛÛÛÛÛ\n");  gotoxy(98,8); printf("ÛÛÛÛÛ\n");    /\*baris 5\*/  gotoxy(62,9); printf("ÛÛÛÛÛ\n");  gotoxy(80,9); printf("ÛÛÛÛÛ\n");  gotoxy(98,9); printf("ÛÛÛÛÛ\n");    /\*baris 6\*/  gotoxy(62,10); printf("ÛÛÛÛÛ\n");  gotoxy(80,10); printf("ÛÛÛÛÛ\n");  gotoxy(98,10); printf("ÛÛÛÛÛ\n");    /\*baris 7\*/  gotoxy(62,11); printf("ÛÛÛÛÛ\n");  gotoxy(80,11); printf("ÛÛÛÛÛ\n");  gotoxy(98,11); printf("ÛÛÛÛÛ\n");    /\*baris 8\*/  gotoxy(62,12); printf("ÛÛÛÛÛ\n");  gotoxy(80,12); printf("ÛÛÛÛÛ\n");  gotoxy(98,12); printf("ÛÛÛÛÛ\n");  gotoxy(76,15);printf("TIC-TAC-TOE");  gotoxy(67,16);printf("MULAI PERMAINAN DARI MANA NIH?");    /\*END PROCEDURE\_JUDUL\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Menu Utama |
| **Jenis Modul** | Procedure |
| **I.S** | Ditampilkan tampilan layar sebelum modul MenuUtama |
| **F.S** | Ditampilkan tampilan menu utama yang menampilkan pilihan untuk bermain, petunjuk permainan, highscore, dan exit |
| **Algoritma** | |
| void MenuUtama() {    int pilihmenu; //PilihMenu adalah variabel lokal bertipe int yang berfungsi sebagai pilihan dari case menu yang tersedia      system("CLS");  Judul();    gotoxy(70,20);printf("[1] PERMAINAN BARU\n");  gotoxy(70,21);printf("[2] CARA & ATURAN BERMAIN\n");  gotoxy(70,22);printf("[3] PAPAN PERINGKAT\n");  gotoxy(70,23);printf("[4] TENTANG PERMAINAN\n");  gotoxy(70,24);printf("[0] KELUAR\n");  gotoxy(70,26);printf("Masukkan Pilihan : \n");  gotoxy(90,26);scanf("%d",&pilihmenu);  switch(pilihmenu){  case 1:  PilihLevel();  break;    case 2:  system("CLS");  CaraBermain();  break;    case 3:  system("CLS");  showScore();  break;    case 4:  system("CLS");  AboutUs();  break;    case 0:  exit(1);  break;    default:  MenuUtama();  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Cara Bermain |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan CaraBermain di layar |
| **Algoritma** | |
| void CaraBermain() {  FILE\*file;  char length[6000];//variabel length adalah variabel lokal bertipe array of char untuk memberi batas char dari file yang dibuka    file=fopen("CaraBermain.txt","r");    while(fgets(length,6000,file)!=NULL){  printf("%s", length);  }    fclose(file);  gotoxy(65,47);printf("Tekan Tombol Untuk Kembali...\n");getch();  //KEMBALI KE MENU UTAMA  system("CLS");  MenuUtama();    /\*END PROCEDURE CARA\_BERMAIN\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | ShowScore |
| **Jenis Modul** | Procedure |
| **I.S** | Ditampilkan tampilan MenuUtama |
| **F.S** | Ditampilkan tampilan berupa nama dan skor yang disimpan |
| **Algoritma** | |
| void showScore(){  SortScore();    FILE \*tampil;  struct SaveScore data;  int y = 5;    tampil = fopen("KumpulanSkor.txt", "rb");    while (!feof(tampil)) {  fscanf(tampil, "%s - %d\n", &data.NamaPemain, &data.skor);  gotoxy(70,2);printf("Papan Peringkat\n");  gotoxy(70,y);printf("%s - %d\n\n", data.NamaPemain, data.skor);  y++;  }    gotoxy(65,y+3);system("pause");  MenuUtama();  } | |

|  |  |
| --- | --- |
| **Nama Modul** | AboutUs |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan about us tentang tujuan dari program permainan ini |
| **Algoritma** | |
| void AboutUs() {  /\*BEGIN PROCEDUR ABOUTUS\*/  FILE \*file;  char length[6000];//variabel length adalah variabel lokal bertipe array of char untuk memberi batas char dari file yang dibuka    file=fopen("TentangPermainan.txt","r");    while (fgets(length,6000,file)!=NULL){  printf("%s", length);  }    fclose(file);    gotoxy(65,30);printf("Tekan Tombol Untuk Melanjutkan...\n");getch();  //KEMBALI KE MENU UTAMA  system("CLS");  MenuUtama();  /\*END PROCEDURE ABOUTUS\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | PilihLevel |
| **Jenis Modul** | Procedure |
| **I.S** | Ditampilkan tampilan layar modul MenuUtama |
| **F.S** | Ditampilkan tampilan menu yang menampilkan level yang tersedia |
| **Algoritma** | |
| void PilihLevel(){  int pilihlevel; //variabel lokal bertipe int yang berfungsi sebagai pilihan dari case menu yang tersedia    system("CLS");    gotoxy(70,18);printf("PILIH LEVEL");  gotoxy(70,20);printf("[1] MUDAH\n");  gotoxy(70,21);printf("[2] MENENGAH\n");  gotoxy(70,22);printf("[3] SULIT\n");  gotoxy(70,23);printf("[0] Menu Utama\n");  gotoxy(70,25);printf("Masukkan Pilihan :\n");  gotoxy(90,25);scanf("%d",&pilihlevel);  menang = 0;  seri = 0;  kalah = 0;  Generate = 1;    switch(pilihlevel){  case 1:  system("CLS");  Generate=Generate\*1;  PilihPapan();  break;    case 2:  system("CLS");  Generate=Generate\*2;  PilihPapan();  break;    case 3:  system("CLS");  Generate=Generate\*3;  PilihPapan();  break;    case 0:  system("CLS");  MenuUtama();  break;    default:  PilihLevel();  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Mudah3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan mudah pada papan 3x3 |
| **Algoritma** | |
| void Mudah3(int GiliranMain){  /\*Deklarasi\*/  int board[10] = {0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 9 && CekMenang(board) == 0; ++turn){    if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerMudah3(board);  }  else {  LangkahPemain3(board);  }  }    switch(CekMenang(board)) {  case 0:  ++seri;  DisplayPemain3(board);  gotoxy(55,27);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain3(board);  gotoxy(55,27);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain3(board);  gotoxy(55,27);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Mudah5 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan mudah pada papan 5x5 |
| **Algoritma** | |
| void Mudah5(int GiliranMain){  /\*Deklarasi\*/  int board[26] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 25 && CekMenang5(board) == 0; ++turn){    if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerMudah5(board);  }  else {  LangkahPemain5(board);  }  }    switch(CekMenang5(board)) {  case 0:  ++seri;  DisplayPemain5(board);  gotoxy(55,32);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Mudah7 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan mudah pada papan 7x7 |
| **Algoritma** | |
| void Mudah7(int GiliranMain){  /\*Deklarasi\*/  int board[50] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};  unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 50 && CekMenang7(board) == 0; ++turn){    if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerMudah7(board);  }  else {  LangkahPemain7(board);  }  }    switch(CekMenang7(board)) {  case 0:  ++seri;  DisplayPemain7(board);  gotoxy(55,37);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain7(board);  gotoxy(55,37);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain7(board);  gotoxy(55,37);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Menengah3 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan menengah pada papan 3x3 |
| **Algoritma** | |
| void Menengah3(int GiliranMain){  /\*Deklarasi\*/  int board[10] = {0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 9 && CekMenang(board) == 0; ++turn){  if((turn + GiliranMain) % 2 == 0) {  if(turn < 6) {  LangkahKomputerSulit3(board);  }  else{  LangkahKomputerMudah3(board);  }  }  else {  LangkahPemain3(board);  }  }    switch(CekMenang(board)) {  case 0:  ++seri;  DisplayPemain3(board);  gotoxy(55,27);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain3(board);  gotoxy(55,27);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain3(board);  gotoxy(55,27);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Menengah5 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan menengah pada papan 5x5 |
| **Algoritma** | |
| void Menengah5(int GiliranMain){  /\*Deklarasi\*/  int board[26] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 25 && CekMenang5(board) == 0; ++turn){  if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerMudah5(board);  }  else {  LangkahPemain5(board);  }  }    switch(CekMenang5(board)) {  case 0:  ++seri;  DisplayPemain5(board);  gotoxy(55,32);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Menengah7 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan menengah pada papan 7x7 |
| **Algoritma** | |
| void Menengah7(int GiliranMain){  int board[50] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 49 && CekMenang7(board) == 0; ++turn){  if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerMudah7(board);  }  else {  LangkahPemain7(board);  }  }    switch(CekMenang7(board)) {  case 0:  ++seri;  DisplayPemain7(board);  gotoxy(55,37);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain5(board);  gotoxy(55,37);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain5(board);  gotoxy(55,37);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Sulit3 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan sulit pada papan 3x3 |
| **Algoritma** | |
| void Sulit3(int GiliranMain){  /\*Deklarasi\*/  int board[10] = {0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 9 && CekMenang(board) == 0; ++turn){    if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerSulit3(board);  }  else {  LangkahPemain3(board);  }  }    switch(CekMenang(board)) {  case 0:  ++seri;  DisplayPemain3(board);  gotoxy(45,27);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();    }  break;    case 1:  DisplayPemain3(board);  gotoxy(45,27);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain3(board);  gotoxy(45,27);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,29);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Sulit5 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan sulit pada papan 5x5 |
| **Algoritma** | |
| void Sulit5(int GiliranMain){  /\*Deklarasi\*/  int board[26] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 25 && CekMenang5(board) == 0; ++turn){  if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerSulit5(board);  }  else {  LangkahPemain5(board);  }  }    switch(CekMenang5(board)) {  case 0:  ++seri;  DisplayPemain5(board);  gotoxy(55,32);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case 1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain5(board);  gotoxy(55,32);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,34);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Sulit7 |
| **Jenis Modul** | Procedure |
| **I.S** | - |
| **F.S** | Ditampilkan tampilan permainan dengan tingkat kesulitan sulit pada papan 7x7 |
| **Algoritma** | |
| void Sulit7(int GiliranMain){  int board[50] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};    unsigned turn;    int pilihan;    /\*Proses\*/  for(turn = 0; turn < 49 && CekMenang(board) == 0; ++turn){    if((turn + GiliranMain) % 2 == 0) {  LangkahKomputerSulit7(board);  }  else {  LangkahPemain7(board);  }  }    switch(CekMenang(board)) {  case 0:  ++seri;  DisplayPemain7(board);  gotoxy(65,37);printf("Permainan Imbang!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();    }  break;    case 1:  DisplayPemain7(board);  gotoxy(65,37);printf("Kamu Kalah :(\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;    case -1:  DisplayPemain7(board);  gotoxy(65,37);printf("Kamu Menang!!\n");  if (game == 4){  gotoxy(45,39);system("pause");  system("CLS");  HitungSkor();  MenuUtama();  }  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | PilihPapan |
| **Jenis Modul** | Procedure |
| **I.S** | Ditampilkan tampilan modul PilihLevel |
| **F.S** | Ditampilkan tampilan pilihan papan yang tersedia yaitu papan 3x3, papan 5x5, serta papan 7x7, pilihan kembali ke menu sebelumnya dan menu utama |
| **Algoritma** | |
| void PilihPapan(){  int pilihpapan;  system("CLS");    gotoxy(70,18);printf("PILIH PAPAN");  gotoxy(70,20);printf("[1] PAPAN 3x3\n");  gotoxy(70,21);printf("[2] PAPAN 5x5\n");  gotoxy(70,22);printf("[3] PAPAN 7x7\n");  gotoxy(70,23);printf("[9] Kembali\n");  gotoxy(70,24);printf("[0] Menu Utama\n");  gotoxy(70,26);printf("Masukkan Pilihan :\n");  gotoxy(90,26);scanf("%d",&pilihpapan);    switch(pilihpapan){  case 1:  system("CLS");  Generate=Generate\*3;  loading();  GuntingKertasBatu();  break;    case 2:  system("CLS");  Generate=Generate\*5;  loading();  GuntingKertasBatu();  break;    case 3:  system("CLS");  Generate=Generate\*7;  loading();  GuntingKertasBatu();  break;    case 9:  system("CLS");  PilihLevel();  break;    case 0:  system("CLS");  MenuUtama();  break;    default:  PilihPapan();  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Board3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan board 3 x 3 |
| **Algoritma** | |
| void board3(int b[10]) {  int x = 50;  system("CLS");  /\*TAMPILAN PAPAN YANG AKAN DIISI INPUT PLAYER\*/  gotoxy(x,15);printf(" | |\n");  gotoxy(x,16);printf(" %c | %c | %c\n", gridChar(b[1]), gridChar(b[2]), gridChar(b[3]));  gotoxy(x,17);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(x,18);printf(" | |\n");  gotoxy(x,19);printf(" %c | %c | %c\n",gridChar(b[4]), gridChar(b[5]), gridChar(b[6]));  gotoxy(x,20);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(x,21);printf(" | |\n");  gotoxy(x,22);printf(" %c | %c | %c\n",gridChar(b[7]), gridChar(b[8]), gridChar(b[9]));  gotoxy(x,23);printf(" | |\n");  /\*END TAMPILAN PAPAN YANG AKAN DIISI INPUT PLAYER\*/  /\*TAMPILAN PAPAN MAP\*/  gotoxy(143,12);printf("| |\n");  gotoxy(138,13);printf(" 1 | 2 | 3\n");  gotoxy(138,14);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(143,15);printf("| |\n");  gotoxy(138,16);printf(" 4 | 5 | 6\n");  gotoxy(138,17);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(143,18);printf("| |\n");  gotoxy(138,19);printf(" 7 | 8 | 9\n");  gotoxy(138,20);printf(" | |\n");  /\*END TAMPILAN PAPAN MAP\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Board5 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan board 5 x 5 |
| **Algoritma** | |
| void board5(int b[26]) {  system("CLS");  /\*TAMPILAN PAPAN\*//\*TAMPILAN PAPAN YANG AKAN DIISI INPUT PLAYER\*/  gotoxy(50,15);printf("| | | |\n");  gotoxy(45,16);printf(" %c | %c | %c | %c | %c \n", gridChar(b[1]), gridChar(b[2]), gridChar(b[3]), gridChar(b[4]), gridChar(b[5])); //tempat penyimpanan bidak 1-5  gotoxy(45,17);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(50,18);printf("| | | |\n");  gotoxy(45,19);printf(" %c | %c | %c | %c | %c \n", gridChar(b[6]), gridChar(b[7]), gridChar(b[8]), gridChar(b[9]), gridChar(b[10]));//tempat penyimpanan bidak 6-10  gotoxy(45,20);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(50,21);printf("| | | |\n");  gotoxy(45,22);printf(" %c | %c | %c | %c | %c \n", gridChar(b[11]), gridChar(b[12]), gridChar(b[13]), gridChar(b[14]), gridChar(b[15])); //tempat penyimpanan bidak 11-15  gotoxy(45,23);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(50,24);printf("| | | |\n");  gotoxy(45,25);printf(" %c | %c | %c | %c | %c \n", gridChar(b[16]), gridChar(b[17]), gridChar(b[18]), gridChar(b[19]), gridChar(b[20])); //tempat penyimpanan bidak 16-20  gotoxy(45,26);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(50,27);printf("| | | |\n");  gotoxy(45,28);printf(" %c | %c | %c | %c | %c \n", gridChar(b[21]), gridChar(b[22]), gridChar(b[23]), gridChar(b[24]), gridChar(b[25])); //tempat penyimpanan bidak 21-25  gotoxy(50,29);printf("| | | |\n");  /\*END TAMPILAN PAPAN YANG AKAN DIISI INPUT PLAYER\*/  /\*TAMPILAN PAPAN MAP\*/  gotoxy(137,12);printf("| | | |\n");  gotoxy(132,13);printf(" 1 | 2 | 3 | 4 | 5 \n"); //tempat penyimpanan bidak 1-5  gotoxy(132,14);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(137,15);printf("| | | |\n");  gotoxy(132,16);printf(" 6 | 7 | 8 | 9 | 10 \n");//tempat penyimpanan bidak 6-10  gotoxy(132,17);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(137,18);printf("| | | |\n");  gotoxy(132,19);printf(" 11 | 12 | 13 | 14 | 15 \n"); //tempat penyimpanan bidak 11-15  gotoxy(132,20);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(137,21);printf("| | | |\n");  gotoxy(132,22);printf(" 16 | 17 | 18 | 19 | 20 \n"); //tempat penyimpanan bidak 16-20  gotoxy(132,23);printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy(137,24);printf("| | | |\n");  gotoxy(132,25);printf(" 21 | 22 | 23 | 24 | 25 \n"); //tempat penyimpanan bidak 21-25  gotoxy(137,26);printf("| | | |\n");  /\*END TAMPILAN PAPAN MAP\*/  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Board7 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan board 7 x 7 |
| **Algoritma** | |
| void board7(int b[50]) {  system("CLS");  /\*TAMPILAN PAPAN YANG AKAN DIISI INPUT PLAYER\*/  gotoxy (50,15); printf("| | | | | |\n");  gotoxy (45,16); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[1]), gridChar(b[2]), gridChar(b[3]), gridChar(b[4]), gridChar(b[5]),gridChar(b[6]), gridChar(b[7]));  gotoxy (45,17); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,18); printf("| | | | | |\n");  gotoxy (45,19); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[8]), gridChar(b[9]), gridChar(b[10]), gridChar(b[11]), gridChar(b[12]),gridChar(b[13]), gridChar(b[14]));  gotoxy (45,20); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,21); printf("| | | | | |\n");  gotoxy (45,22); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[15]), gridChar(b[16]), gridChar(b[17]), gridChar(b[18]), gridChar(b[19]),gridChar(b[20]), gridChar(b[21]));  gotoxy (45,23); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,24); printf("| | | | | |\n");  gotoxy (45,25); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[22]), gridChar(b[23]), gridChar(b[24]), gridChar(b[25]), gridChar(b[26]),gridChar(b[27]), gridChar(b[28]));  gotoxy (45,26); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,27); printf("| | | | | |\n");  gotoxy (45,28); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[29]), gridChar(b[30]), gridChar(b[31]), gridChar(b[32]), gridChar(b[33]),gridChar(b[34]), gridChar(b[35]));  gotoxy (45,29); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,30); printf("| | | | | |\n");  gotoxy (45,31); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[36]), gridChar(b[37]), gridChar(b[38]), gridChar(b[39]), gridChar(b[40]),gridChar(b[41]), gridChar(b[42]));  gotoxy (45,32); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (50,33); printf("| | | | | |\n");  gotoxy (45,34); printf(" %c | %c | %c | %c | %c | %c | %c \n",gridChar(b[43]), gridChar(b[44]), gridChar(b[45]), gridChar(b[46]), gridChar(b[47]),gridChar(b[48]), gridChar(b[49]));  gotoxy (50,35); printf("| | | | | |\n");  gotoxy (131,12); printf("| | | | | |\n");  gotoxy (126,13); printf(" 1 | 2 | 3 | 4 | 5 | 6 | 7 \n");  gotoxy (126,14); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,15);printf("| | | | | |\n");  gotoxy (126,16); printf(" 8 | 9 | 10 | 11 | 12 | 13 | 14 \n");  gotoxy (126,17); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,18); printf("| | | | | |\n");  gotoxy (126,19); printf(" 15 | 16 | 17 | 18 | 19 | 20 | 21 \n");  gotoxy (126,20); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,21); printf("| | | | | |\n");  gotoxy (126,22); printf(" 22 | 23 | 24 | 25 | 26 | 27 | 28 \n");  gotoxy (126,23); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,24); printf("| | | | | |\n");  gotoxy (126,25); printf(" 29 | 30 | 31 | 32 | 33 | 34 | 35 \n");  gotoxy (126,26); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,27); printf("| | | | | |\n");  gotoxy (126,28); printf(" 36 | 37 | 38 | 39 | 40 | 41 | 42 \n");  gotoxy (126,29); printf("\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n");  gotoxy (131,30); printf("| | | | | |\n");  gotoxy (126,31); printf(" 43 | 44 | 45 | 46 | 47 | 48 | 49 \n");  gotoxy (131,32); printf("| | | | | |\n");  } | |

|  |  |
| --- | --- |
| **Nama Modul** | GuntingKertasBatu |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan Permainan GuntingKertasBatu |
| **Algoritma** | |
| void GuntingKertasBatu(){  srand(time(NULL));  int pemain;  int musuh;  int GiliranMain;  system("CLS"); //berfungsi untuk clear screen atau membersihkan layar(mengosongkan layar)  musuh = rand()%3; //inisialisasi dari pilihan musuh yang acak dari 0 hingga 2  gotoxy(70,18);printf("PILIHLAH TANGANMU : \n");  gotoxy(70,20);printf("[1] Gunting\n");  gotoxy(70,21);printf("[2] Batu\n");  gotoxy(70,22);printf("[3] Kertas\n");  gotoxy(70,23);printf("Masukkan Pilihan : ");  gotoxy(90,23);scanf("%d",&pemain);  if(pemain>0 && pemain<4){  if(musuh==0) //jika hasil acak = 0 berinisialiasi bahwa pilihan musuh adalah gunting  {  if(pemain==1) //jika pemain menginput 1 maka inisialiasinya pemain gunting  {  gotoxy(70,25);printf("Gunting Lawan Gunting (Seri)\n\n");  gotoxy(70,29);system("pause");  GuntingKertasBatu();  }  else if(pemain==2) //jika pemain menginput 2 maka inisialiasinya pemain batu  {  gotoxy(70,25);printf("Gunting Lawan Batu (Menang)\n\n");  gotoxy(70,27);printf("Kamu Main Pertama\n");  gotoxy(70,29);system("pause");  GetLevel(1);  }  else if(pemain==3) //jika pemain menginput 3 maka inisialiasinya pemain kertas  {  gotoxy(70,25);printf("Gunting Lawan Kertas (Kalah)\n\n");  gotoxy(70,27);printf("Kamu Main Kedua\n");  gotoxy(70,29);system("pause");  GetLevel(2);  }  }  else if(musuh==1)  {  if(pemain==1) //jika pemain menginput 1 maka inisialiasinya pemain gunting  {  gotoxy(70,25);printf("Batu Lawan Gunting (Kalah)\n\n");  gotoxy(70,27);printf("Kamu Main Kedua\n");  gotoxy(70,29);system("pause");  GetLevel(2);  }  else if(pemain==2) //jika pemain menginput 1 maka inisialiasinya pemain batu  {  gotoxy(70,25);printf("Batu Lawan Batu (Seri)\n\n");  gotoxy(70,29);system("pause");  GuntingKertasBatu();  }  else if(pemain==3) //jika pemain menginput 1 maka inisialiasinya pemain kertas  {  gotoxy(70,25);printf("Batu Lawan Kertas (Menang)\n\n");  gotoxy(70,27);printf("Kamu Main Pertama\n");  gotoxy(70,29);system("pause");  GetLevel(1);  }  }  if(musuh==2)  {  if(pemain==1) //jika pemain menginput 1 maka inisialiasinya pemain gunting  {  gotoxy(70,25);printf("Kertas Lawan Gunting (Menang)\n\n");  gotoxy(70,27);printf("Kamu Main Pertama\n");  gotoxy(70,29);system("pause");  GetLevel(1);  }  else if(pemain==2) //jika pemain menginput 1 maka inisialiasinya pemain batu  {  gotoxy(70,25);printf("Kertas Lawan Batu (Kalah)\n\n");  gotoxy(70,27);printf("Kamu Main Kedua\n");  gotoxy(70,29);system("pause");  GetLevel(2);  }  else if(pemain==3) //jika pemain menginput 1 maka inisialiasinya pemain kertas  {  gotoxy(70,25);printf("Kertas Lawan Kertas (Seri)\n\n");  gotoxy(70,29);system("pause");  GuntingKertasBatu();  }  }  }else{  gotoxy(65,24);printf("Angka yang Anda Masukkan Tidak Valid\n");  getch();  GuntingKertasBatu();  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | GridChar |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Menampilkan simbol pada papan permainan |
| **Algoritma** | |
| char gridChar(int i){  switch(i){  case -1:  return 'O';  case 0:  return ' ';  case 1:  return 'X';  } | |

|  |  |
| --- | --- |
| **Nama Modul** | CekMenang |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Memeriksa permainan pada papan 3x3 sudah terdapat pemenang atau belum. |
| **Algoritma** | |
| int CekMenang(const int board[10]){  /\*Deklarasi\*/  unsigned wins[8][3]={{1,2,3},{4,5,6},{7,8,9},{1,4,7},{2,5,8},{3,6,9},{1,5,9},{3,5,7}};  int i;  /\*Proses\*/  for(i = 0; i < 8; ++i){  if (board[wins[i][0]] != 0 &&  board[wins[i][0]] == board[wins[i][1]] &&  board[wins[i][0]] == board[wins[i][2]])  return board[wins[i][2]];  }  return 0;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | CekMenang5 |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Memeriksa permainan pada papan 5 x 5 sudah terdapat pemenang atau belum. |
| **Algoritma** | |
| int CekMenang5(const int board[26]){  /\*Deklarasi\*/  unsigned wins[12][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15},{16,17,18,19,20},{21,22,23,24,25},{1,6,11,16,21},{2,7,12,17,22},  {3,8,13,18,23},{4,9,14,19,24},{5,10,15,20,25},{1,7,13,19,25},{5,9,13,17,21}};  int i;  /\*Proses\*/  for(i = 0; i < 12; ++i){  if (board[wins[i][0]] != 0 &&  board[wins[i][0]] == board[wins[i][1]] &&  board[wins[i][0]] == board[wins[i][2]] &&  board[wins[i][0]] == board[wins[i][3]] &&  board[wins[i][0]] == board[wins[i][4]])  return board[wins[i][4]];  }  return 0;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | CekMenang7 |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Memeriksa permainan pada papan 7x7 sudah terdapat pemenang atau belum. |
| **Algoritma** | |
| int CekMenang7 (const int board [50]) {  /\*Deklarasi\*/  unsigned wins [16][7] = {{1,2,3,4,5,6,7},{8,9,10,11,12,13,14},{15,16,17,18,19,20,21},{22,23,24,25,26,27,28},{29,30,31,32,33,34,35},  {36,37,38,39,40,41,42},{43,44,45,46,47,48,49},{1,9,17,25,33,41,49},{7,13,19,25,31,37,43},{1,8,15,22,29,36,43},{2,9,16,23,30,37,44},  {3,10,17,24,31,38,45},{4,11,18,25,32,39,46},{5,12,19,26,33,40,47},{6,13,20,27,34,41,48},{7,14,21,28,35,42,49}};  int i;  /\*Proses\*/  for (i=0;i<16;++i){  if (board[wins[i][0]] != 0 &&  board [wins[i][0]] == board [wins[i][1]] &&  board [wins[i][0]] == board [wins[i][2]] &&  board [wins[i][0]] == board [wins[i][3]] &&  board [wins[i][0]] == board [wins[i][4]] &&  board [wins[i][0]] == board [wins[i][5]] &&  board [wins[i][0]] == board [wins[i][6]])  return board[wins[i][6]];  }  return 0;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | SortScore |
| **Jenis Modul** | Procedure |
| **I.S** | Score belum terurut pada papan peringkat |
| **F.S** | Score sudah terurut pada papan peringkat |
| **Algoritma** | |
| void SortScore(){  FILE \*arrange;  struct SaveScore leaderboard[100];  struct SaveScore temp;  int n, i, j;  arrange = fopen("KumpulanSkor.txt", "rb");  n = 0;  while(!feof(arrange)){  fscanf(arrange, "%s - %d\n", &leaderboard[n].NamaPemain, &leaderboard[n].skor);  n++;  }  fclose(arrange);  for(i=0; i<n-1; i++){  for(j=0; j<n-1; j++){  if(leaderboard[j].skor < leaderboard[j+1].skor){  temp = leaderboard[j];  leaderboard[j] = leaderboard[j+1];  leaderboard[j+1] = temp;  }  }  }  arrange = fopen("KumpulanSkor.txt", "wb");  for(i=0; i<n; i++){  fprintf(arrange, "%s - %d\n", leaderboard[i].NamaPemain, leaderboard[i].skor);  };  fclose(arrange);  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahPemain3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan langkah pemain di layar pada papan 3x3. |
| **Algoritma** | |
| void LangkahPemain3(int board[10]) {  /\*Deklarsi\*/  int move = 0;  /\*Proses\*/  do{  begin:  DisplayPemain3(board);  gotoxy(45,28);printf("Masukkan Pilihan (1 - 9):");  gotoxy(72,28);scanf("%d", &move);  if (board[move] != 0){  gotoxy(45, 32);printf("Langkah Tidak Valid!\n");  gotoxy(65, 32);getch();  goto begin;  printf("\n");  }  }while( move >= 10 || move < 1 && board[move] == 0);  board[move] = -1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahPemain5 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan Langkah pemain pada papan 5 x 5 |
| **Algoritma** | |
| void LangkahPemain5(int board[26]) {  /\*Deklarsi\*/  int move = 0;  /\*Proses\*/  do{  begin:  DisplayPemain5(board);  gotoxy(45,33);printf("Masukkan Pilihan (1 - 25):");  gotoxy(72,33);scanf("%d", &move);  if (board[move] != 0){  gotoxy(45, 37);printf("Langkah Tidak Valid!\n");  gotoxy(65, 37);getch();  goto begin;  printf("\n");  }  }while( move >= 26 || move < 1 && board[move] == 0);  board[move] = -1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahPemain7 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan Langkah Penamin Pada Papan 7 x 7 |
| **Algoritma** | |
| void LangkahPemain7(int board[50]) {  int move = 0;  /\*Proses\*/  do{  begin:  DisplayPemain7(board);  gotoxy(45,38);printf("Masukkan Pilihan (1 - 49):");  gotoxy(72,38);scanf("%d", &move);  if (board[move] != 0){  gotoxy(45, 42);printf("Langkah Tidak Valid!\n");  gotoxy(65, 42);getch();  goto begin;  printf("\n");  }  }while( move >= 50 || move < 1 && board[move] == 0);  board[move] = -1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerMudah3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 3x3 |
| **Algoritma** | |
| void LangkahKomputerMudah3(int board[10]) {  /\*Deklarasi\*/  int i;  int move;  /\*Proses\*/  reset:  move = 1 + rand()% 9; // 1 adalah range angka terkecil yg akan di random, dan 9 adalah angka terbesar yang akan dirandom  if(board[move] != 0){  goto reset;  }  else{  board[move] = 1;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerMudah5 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 5x5 |
| **Algoritma** | |
| void LangkahKomputerMudah5(int board[26]){  /\*Deklarasi\*/  int i;  int move;  /\*Proses\*/  reset:  move = 1 + rand()% 25; // 1 adalah range angka terkecil yg akan di random, dan 25 adalah angka terbesar yang akan dirandom  if(board[move] != 0){  goto reset;  }  else{  board[move] = 1;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerMudah7 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan mudah pada papan 7x7 |
| **Algoritma** | |
| void LangkahKomputerMudah7(int board[50]){  int i;  int move;  reset:  move = 1 + rand()% 49; // 1 adalah range angka terkecil yg akan di random, dan 49 adalah angka terbesar yang akan dirandom  if(board[move] != 0){  goto reset;  }  else{  board[move] = 1;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerSulit3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 3x3 |
| **Algoritma** | |
| void LangkahKomputerSulit3(int board[10]){  /\*Deklarasi\*/  int move = -1;  int score = -2;  int i;  /\*Proses\*/  for(i=1; i < 10; ++i) {  if(board[i] == 0){  board[i] = 1;  int tempScore = -minimax(board, -1);  board[i] = 0;  if (tempScore > score) {  score = tempScore;  move = i;  }  }  }  board[move] = 1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerSulit5 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 5x5 |
| **Algoritma** | |
| void LangkahKomputerSulit5(int board[26]){  /\*Deklarasi\*/  int move = -1;  int score = -2;  int i;  /\*Proses\*/  for(i=1; i < 26; ++i) {  if(board[i] == 0){  board[i] = 1;  int tempScore = -minimax5(board, -1);  board[i] = 0;  if (tempScore > score) {  score = tempScore;  move = i;  }  }  }  board[move] = 1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | LangkahKomputerSulit7 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menentukan langkah komputer dalam permainan dengan tingkat kesulitan sulit pada papan 7x7 |
| **Algoritma** | |
| void LangkahKomputerSulit7(int board[50]){  int move = -1;  int score = -2;  int i;  /\*Proses\*/  for(i=1; i < 50; ++i) {  if(board[i] == 0){  board[i] = 1;  int tempScore = -minimax7(board, -1);  board[i] = 0;  if (tempScore > score) {  score = tempScore;  move = i;  }  }  }  board[move] = 1;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Minimax |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 3x3 |
| **Algoritma** | |
| int minimax(int board[10], int player) {  //How is the position like for player (their turn) on board?  int winner = CekMenang(board);  if(winner != 0) return winner\*player;  int move = -1;  int score = -2;//Losing moves are preferred to no move  int i;  for(i = 1; i < 10; ++i) {//For all moves,  if(board[i] == 0) {//If legal,  board[i] = player;//Try the move  int thisScore = -minimax(board, player\*-1);  if(thisScore > score) {  score = thisScore;  move = i;  }//Pick the one that's worst for the opponent  board[i] = 0;//Reset board after try  }  }  if(move == -1) return 0;  return score;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Minimax5 |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 5x5 |
| **Algoritma** | |
| int minimax5(int board[26], int player) {  int winner = CekMenang5(board);  if(winner != 0) return winner\*player;  int move = -1;  int score = -2;  int i;  for(i = 1; i < 26; ++i) {  if(board[i] == 0) {  board[i] = player;  int thisScore = -minimax5(board, player\*-1);  if(thisScore > score) {  score = thisScore;  move = i;  }//Pick the one that's worst for the opponent  board[i] = 0;//Reset board after try  }  }  if(move == -1) return 0;  return score;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | Minimax7 |
| **Jenis Modul** | Function |
| **I.S** |  |
| **F.S** | Mencari kemungkinan pergerakan komputer yang memungkinkan bagi komputer ukuran 7x7 |
| **Algoritma** | |
| int minimax7(int board[50], int player) {  int winner = CekMenang7(board);  if(winner != 0) return winner\*player;  int move = -1;  int score = -2;  int i;  for(i = 1; i < 50; ++i) {  if(board[i] == 0) {  board[i] = player;  int thisScore = -minimax7(board, player\*-1);  if(thisScore > score) {  score = thisScore;  move = i;  }  board[i] = 0;  }  }  if(move == -1) return 0;  return score;  } | |

|  |  |
| --- | --- |
| **Nama Modul** | HitungScore |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menghitung skor setelah permainan berakhir |
| **Algoritma** | |
| void HitungSkor(){  int SkorTotal, SkorMenang, SkorSeri;  switch(Generate){  case 3:  SkorMenang = menang\*1000;  SkorSeri = seri\*500;  SkorTotal = SkorMenang + SkorSeri;  break;  case 5:  SkorMenang = menang\*2000;  SkorSeri = seri\*1000;  SkorTotal = SkorMenang + SkorSeri;  break;  case 6:  SkorMenang = menang\*3000;  SkorSeri = seri\*1500;  SkorTotal = SkorMenang + SkorSeri;  break;  case 10:  SkorMenang = menang\*4000;  SkorSeri = seri\*2000;  SkorTotal = SkorMenang + SkorSeri;  break;  case 9:  SkorMenang = menang\*7000;  SkorSeri = seri\*3500;  SkorTotal = SkorMenang + SkorSeri;  break;  case 15:  SkorMenang = menang\*8000;  SkorSeri = seri\*4000;  SkorTotal = SkorMenang + SkorSeri;  break;  }  SimpanSkor(SkorTotal);  } | |

|  |  |
| --- | --- |
| **Nama Modul** | DisplayPemain3 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan tampilan permainan board 3x3 |
| **Algoritma** | |
| void DisplayPemain3(int board[10]){  /\*Deklarasi\*/  int y; //variabel untuk menentukan koordinat y  /\*Process\*/  switch(CekMenang(board)){  case -1:  ++menang;  break;  case 1:  ++kalah;  break;  }  /\*TAMPILAN SAAT PERMAINAN MULAI\*/  board3(board);  gotoxy(15,2);printf("ЩЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЛ");  gotoxy(39,3);printf("К");  gotoxy(15,3);printf("К Ronde Ke :\t %d",game);  gotoxy(15,4);printf("ШЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭМ");  gotoxy(43,10);printf("Pemain ( O ) Komputer ( X )\n");  gotoxy(145,10);printf("Map");  gotoxy(143,25);printf("Status");  gotoxy(135,27);printf("Menang : %d", menang);  gotoxy(135,29);printf("Seri : %d", seri);  gotoxy(135,31);printf("Kalah : %d", kalah);  for(y = 0; y < 48; y++){  gotoxy(125,y);printf("К");  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | DisplayPemain5 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan tampilan permainan board 5x5 |
| **Algoritma** | |
| void DisplayPemain5(int board[26]){  /\*Deklarasi\*/  int y; //variabel untuk menentukan koordinat y  /\*Process\*/  switch(CekMenang5(board)){  case -1:  ++menang;  break;  case 1:  ++kalah;  break;  }  /\*TAMPILAN SAAT PERMAINANA MULAI\*/  board5(board);  gotoxy(15,2);printf("ЩЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЛ");  gotoxy(39,3);printf("К");  gotoxy(15,3);printf("К Ronde Ke :\t %d",game);  gotoxy(15,4);printf("ШЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭМ");  gotoxy(43,10);printf("Pemain ( O ) Komputer ( X )\n");  gotoxy(145,10);printf("Map");  gotoxy(143,31);printf("Status");  gotoxy(135,33);printf("Menang : %d", menang);  gotoxy(135,35);printf("Seri : %d", seri);  gotoxy(135,37);printf("Kalah : %d", kalah);  for(y = 0; y < 48; y++){  gotoxy(125,y);printf("К");  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | DisplayPemain7 |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menampilkan tampilan permainan board 7x7 |
| **Algoritma** | |
| void DisplayPemain7(int board[50]){  int y;  /\*Process\*/  switch(CekMenang7(board)){  case -1:  ++menang;  break;  case 1:  ++kalah;  break;  }  /\*TAMPILAN SAAT PERMAINANA MULAI\*/  board7(board);  gotoxy(15,2);printf("ЩЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЛ");  gotoxy(39,3);printf("К");  gotoxy(15,3);printf("К Ronde Ke :\t %d",game);  gotoxy(15,4);printf("ШЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭЭМ");  gotoxy(43,10);printf("Pemain ( O ) Komputer ( X )\n");  gotoxy(145,10);printf("Map");  gotoxy(143,37);printf("Status");  gotoxy(135,39);printf("Menang : %d", menang);  gotoxy(135,41);printf("Seri : %d", seri);  gotoxy(135,43);printf("Kalah : %d", kalah);  for(y = 0; y < 48; y++){  gotoxy(125,y);printf("К");  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | GetLevel |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Memanggil modul permainan setelah bermain gunting-kertas-batu |
| **Algoritma** | |
| void GetLevel(int GiliranMain){  switch(Generate){  case 3:  game = 1;  do{  Mudah3(GiliranMain);  gotoxy(45,29);system("pause");  game++;  }while (game < 5);  break;  case 5:  game = 1;  do{  Mudah5(GiliranMain);  gotoxy(45,34);system("pause");  game++;  }while (game < 5);  break;  case 7:  game = 1;  do{  Mudah7(GiliranMain);  gotoxy(45,39);system("pause");  game++;  }while (game<5);  break;  case 6:  game = 1;  do{  Menengah3(GiliranMain);  gotoxy(45,29);system("pause");  game++;  }while (game < 5);  break;  case 10:  game = 1;  do{  Menengah5(GiliranMain);  gotoxy(45,34);system("pause");  game++;  }while (game < 5);  break;  case 14:  game = 1;  do{  Menengah7(GiliranMain);  gotoxy(45,39);system("pause");  game++;  }while (game<5);  break;  case 9:  game = 1;  do{  Sulit3(GiliranMain);  gotoxy(45,29);system("pause");  game++;  }while (game < 5);  break;  case 15:  game = 1;  do{  Sulit5(GiliranMain);  gotoxy(45,34);system("pause");  game++;  }while (game < 5);  break;  case 21:  game = 1;  do{  Sulit7(GiliranMain);  gotoxy(45,39);system("pause");  game++;  }while (game < 5);  break;  }  } | |

|  |  |
| --- | --- |
| **Nama Modul** | SimpanScore |
| **Jenis Modul** | Procedure |
| **I.S** |  |
| **F.S** | Menginput nama pemain dan menyimpan nama pemain beserta skor ke dalam file KumpulanSkor.txt |
| **Algoritma** | |
| void SimpanSkor(int skor){  struct SaveScore data;  FILE \*DataSkor;  int pilihan;  DataSkor = fopen("KumpulanSkor.txt", "ab");  gotoxy(70,22);printf("Skor kamu : %d", skor);  data.skor = skor;  gotoxy(65,24);printf("Masukkan nama kamu : ");  gotoxy(86,24);scanf("%s", &data.NamaPemain);  fprintf(DataSkor, "%s - %d\n", data.NamaPemain, data.skor);  fclose(DataSkor);  } | |

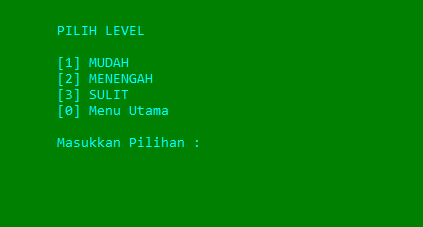
* 1. **Structure Chart**

****

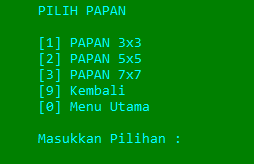
* 1. **Tampilan**
     1. **Tampilan Utama**

****

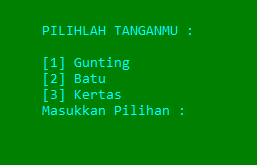
* + 1. **Tampilan Pemilihan Tingkat Kesulitan**

****

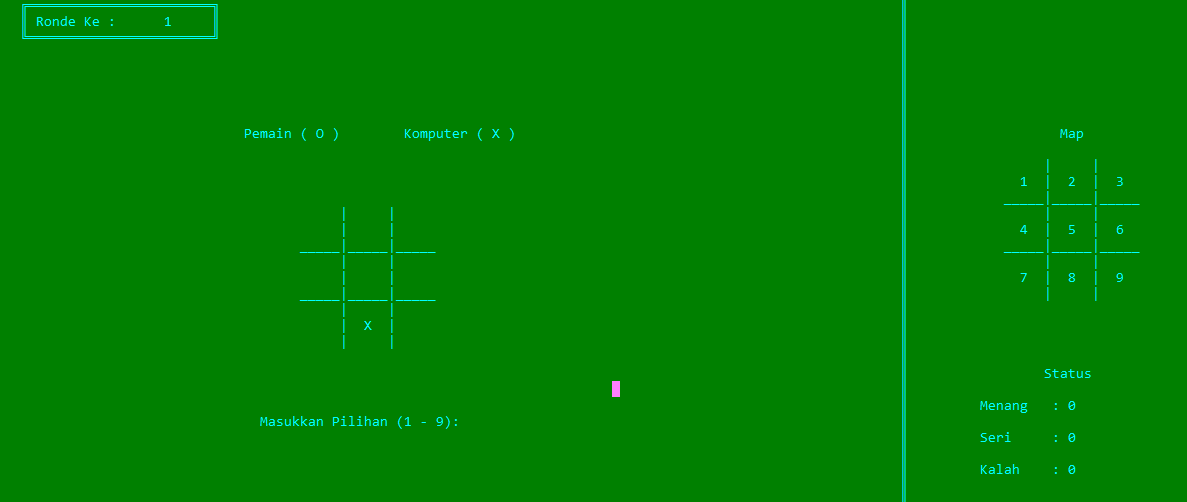
* + 1. **Tampilan Pemilihan Papan**

****

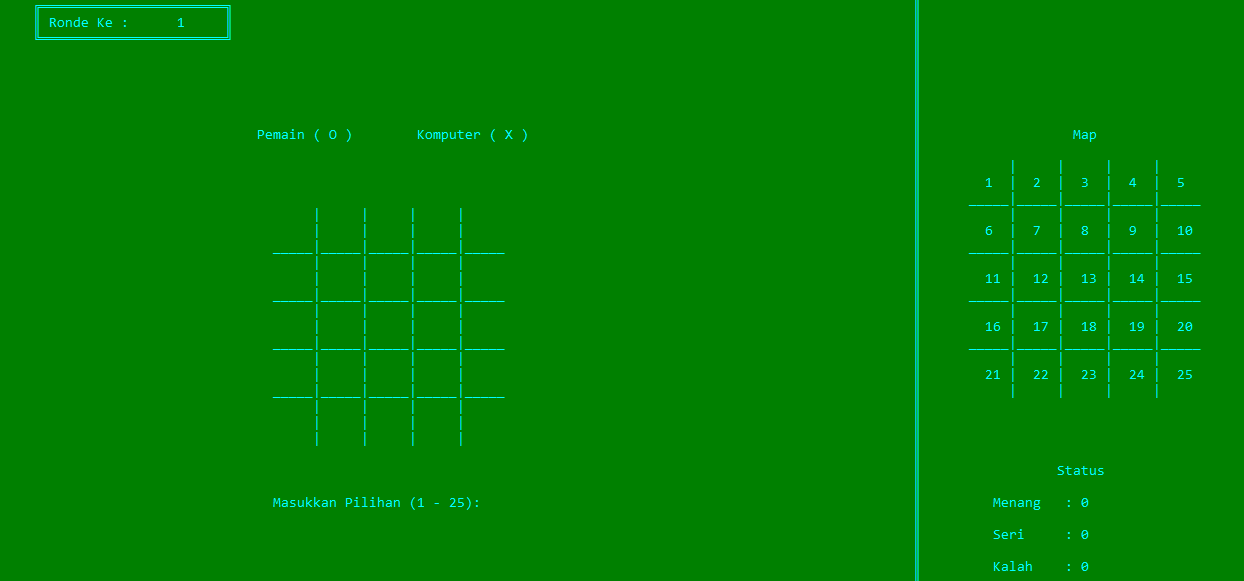
* + 1. **Tampilan Pemilihan Urutan Yang Petama menjalankan Permainan**

****

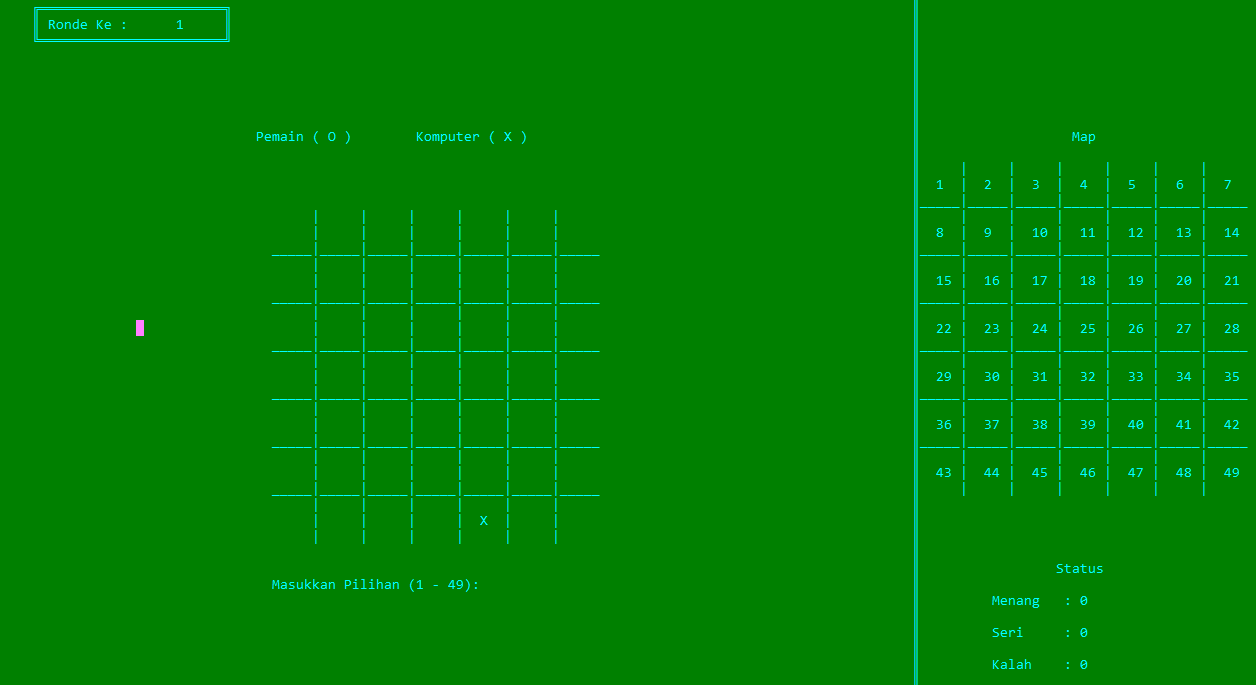
* + 1. **Tampilan Bidak Permainan 3 X 3**

****

* + 1. **Tampilan Bidak Permainan 5 X 5**

****

**2.7.7 Tampilan Bidak Permainan 7 X 7**

****

1. **Hasil Akhir Program**

**3.1 Link Github**

Berikut merupakan link github dari projrct tentang tugas besar tic-tac-toe ini:

<https://github.com/aziztaufiqurrahman/Project-Development.git>

* 1. **Petunjuk Penggunaan**

Berikut ini merupakan petunjuk penggunaan dari program yang kami buat, yaitu:

1. User diminta untuk memilih pilihan pada Menu Utama
2. Mulai Permainan, jika ingin memulai permainan baru
3. Cara Aturan Bermain, jika ingin melihat cara bermain dalam program ini
4. High Score, jika ingin melihat hasil score permainan yang sudah tersimpan
5. Tentang Permainan, jika ingin melihat keterangan pembuatan program ini
6. Keluar, jika ingin keluar dari program permainan tic-tac-toe
7. User akan diarahkan untuk memilih tingkatan permainan
8. Mudah
9. Menengah
10. Sulit
11. User akan diminta untuk memilih papan permainan
12. Papan permainan berukuran 3x3
13. Papan permainan berukuran 5x5
14. Papan permainan berukuran 7x7
15. Untuk menentukan giliran permainan, maka user akan diminta memilih pada pilihan yang telah disediakan
16. Gunting
17. Batu
18. Kertas

Jika pemain menang maka akan mendapatkan giliran pertama, jika computer menang maka computer akan mendapatkan giliran pertama dalam permainan, serta jika hasil dari program giliran permainan seri maka program akan diulang Kembali sampai terdapat menang & kalah.

1. Permainan sudah bisa dimulai
2. Pada program ini permainan akan berlangsung sebanyak 4 kali, kemudian pada akhir permainan user akan diminta untuk memasukkan nama dengan tujuan disimpan dalam high score.
3. **PENUTUP**

**4.1 Waktu Pengerjaan**

Waktu pengerjaan dari project ini selama 3 minggu, dimulai dari menyusun laporan program, konsultasi laporan, pembuatan masing-masing program, kemudian menggabungkan program menjadi sebuah program yang utuh.

**4.2 Ringkasan Kontribusi Setiap Anggota**

Pada permulaan pengerjaan, kami mendiskusikan mengenai aturan-aturan permainan dari game yang akan dibuat. Setelah aturan dasar, maka pembahasan mulai masuk ke dalam kebutuhan data yang diperlukan pada game yang akan kami buat, setelah itu masing masing dari kami membuat program tic tac toe masing masing untuk disesuaikan dengan kebutuhan, lalu ke perincian modul dari proses permainan yang berlangsung. Setelah modul-modul yang ada disepakati, kami mulai membuat program serta menggabungkan modul tersebut menjadi program yang utuh.

**4.3 Lesson Learned**

* Kami merasakan bahwa memanajemen waktu di tengah deadline berbagai tugas yang ada pada tugas besar ini.
* Kami lebih mengetahui bahwa perancangan atau langkah awal membuat suatu aplikasi atau game harus dengan teliti dan lebih memikirkan rancangan awal terlebih dahulu.
* Tahap perancangan sangat berpengaruh terhadap jalannya membuat program, sehingga tahap perancangan tersebut menjadi sebuah acuan alur pembuatan program.
* Program yang besar harus disusun menjadi berbagai program kecil dan memiliki tujuan spesifik (modul program) sehingga memudahkan untuk mencari kesalahan dalam sebuah program.
* Kerja sama, komunikasi serta pembagian tugas sangat diperlukan dalam pembuatan sebuah project.
* Efektivitas dari sebuah modul sangat terasa sehingga program menjadi lebih ringkas dan efisien.
* Banyak ilmu yang kami dapatkan dari pembuatan tugas besar ini, dimulai dari cara membuat modul yang spesifik, membuat alur program sehingga pemanggilan modul tepat sasaran, berlatih tracing sebuah program apabila terdapat kesalahan.
* Melatih jalannya logika berpikir.