**LAPORAN TUGAS BESAR**

**KKBTI4019 – Dasar-dasar Pemrograman**

**PERMAINAN LUDO**



**Disusun oleh**

**Laras Rasdiyani – 191524016**

**Mochamad Nauval Dwisatya - 191524023**

**Program Studi DIV Teknik Informatika**

**Departemen Teknik Komputer dan Informatika**

**Politeknik Negeri Bandung**

**2019**

**GoRas Ludo Games**

1. **Deskripsi GoRas Ludo Games**

GoRas Ludo Games merupakan permainan papan yang sama seperti bermain ular tangga. GoRas Ludo Games ini bisa dilakukan bersama teman atau pun dengan keluarga. Sejarah permainan Ludo bisa dilacak dari abad ke-6 di India. Ludo sendiri berasal dari permainan Pachisi, dan juga sangat mirip dengan Parchis, permainan papan dari Spanyol yang sering di mainkan oleh pendeta kristen. Permainan Ludo amat populer dari zaman dahulu hingga saat ini, hanya struktur permainannya saja yang sedikit bervariasi dan lebih modern. Ludo semakin berkembang di era teknologi digital seperti sekarang, versi digital dari permainan ini dikembangkan dalam bentuk aplikasi atau program. Jenis permainan Ludo berbentuk *board game* kini banyak dijumpai di *smartphone*. Selain itu, ada juga *developer* kreatif yang mengembangkan game ini menjadi lebih bertema dan menarik untuk dimainkan secara digital.

Dalam permainan ludo sangat terkait dengan probabilitas karena setiap pemain memiliki strategi sendiri dengan peluang yang berbeda-beda dan acak. Hal ini membuat ludo menjadi objek yang menarik untuk diteliti, apalagi dengan perkembangan teknologi yang memungkinkan untuk membuat simulasi acak yang bertujuan untuk mengamati probabilitas kemenangan dari permainan ludo.

1.1. Menu Permainan

GoRas Ludo Games terdiri dari 4 menu utama, yaitu :

* New Games, adalah menu untuk memulai permainan
* Instruction, adalah menu untuk menampilkan petunjuk atau cara memainkan GoRas Ludo Games
* Credit, adalah menu untuk menampilkan developer dari GoRas Ludo Games
* Exit, adalah menu untuk keluar dari program GoRas Ludo Games

Adapaun spesifikasi Input dan Output di bagian menu, yaitu :

* New Games

Untuk mulai bermain user harus menginputkan angka 1, kemudian setelah itu akan muncul pilihan berupa berupa jumlah pemain yang akan bermain. Untuk memilih berpa player yang akan bermain user harus menginputkan angka diantaranya :

* Angka 1 untuk bermain dengan 2 pemain ( 1 user dan 1 komputer)
* Angka 2 untuk bermain dengan 3 pemain ( 1 user dan 2 komputer)
* Angka 3 untuk bermain dengan 4 pemain ( 1 user dan 3 komputer)
* Angka 4 untuk kembali ke menu utama
* Instruction

Untuk mengetahui cara bermain user harus menginputkan angka 2, kemudian dilayar akan muncul penjelasan berupa cara bermain

* Credits

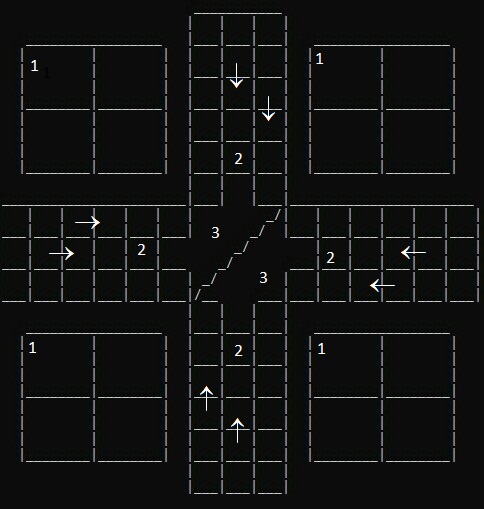
Untuk mengetahui siapa developer dari Goras Ludo Games user harus menginputkan angka 3, kemudian dilayar akan muncul penjelasan mengenai developer Goras Ludo Games

* Exit

Untuk keluar dari program GoRas Ludo, user harus menginputkan angka 4 di tampilan Menu Utama

1.2. Papan permaiann GoRas Ludo Games

Bagian-bagian pada papan permainan GoRas Ludo Games dapat dilihat pada gambar 2.1 dan akan dijelaskan per-poin.



**Gambar 2.1 Papan Permainan GoRas Ludo Games**

Papan permainan GoRas Ludo Games berbentuk persegi empat, di setiap sudutnya terdapat *home base,* pada gambar 2.1 diberi tanda dengan angka 1. Setiap *home base* memiliki empat buah bidak dengan lambang masing masing yaitu C,K,F, atau X.

Bidak berjalan memutar sesuai arah jarum jam seperti pada gambar 2.1, arah laju permainan GoRas Ludo Games ditandai dengan tanda panah. Apabila bidak telah berputar sebanyak satu purtaran ( 52 petak ) maka bidak tersebut masuk ke dalam lintasan finish yang berada di bagian tengah seperti pada gambar 2.1 yang ditandai dengan angka 2 (dua). Pada gambar 2.1 yang ditandai dengan angka 3 adalah Finish.

1.3. Pemain

Permainan GoRas Ludo Games dapat dimainkan oleh minimal terdiri dari 2 pemain (1 user dan 1 komputer) dan maksimal 4 pemain ( 1 user dan 3 komputer).

1.4. Spesifikasi Pemain (Komputer)

|  |  |  |
| --- | --- | --- |
| No. | Nama | Perilaku |
| 1. | Quick | Komputer akan berusaha untuk menggerakkan pion secepat mungkin menuju tujuan. Pemilihan pion yang akan digerakkan berdasarkan prioritas. Prioritas tertinggi adalah pion yang dapat bergerak dan berada paling dekat dengan tujuan. Dengan menggerakkan pion yang paling dekat dengan kotak tujuan, kita akan lebih cepat mencapai tujuan. Selama pion yang paling dekat dengan kotak tujuan tidak terhalangi, maka seluruh langkah yang dapat diambil akan digunakan untuk memajukan pion tersebut. Jika tidak ada yang bisa bergerak maka algoritma akan menggerakkan pion lainnya sejumlah angka yang tertera di dau atau sisa langkah yang dapat diambil. |
| 2. | Killer | Komputer akan berusaha untuk menggerakkan pion agar dapat memukul pion pemain lain. Pemilihan pion yang digerakkan adalah berdasarkan prioritas tergantung kondisi pion saat ini. Prioritas paling tinggi adalah pion yang paling dekat pion lain dan dapat memukul pion pemain lain. Prioritas kedua adalah jika tidak bisa memukul pion lain, pion tersebut akan maju mendekati pion lawan yang terdekat. Jika tidak ada pion yang memungkinkan, maka algoritma akan mencari pion lain yang dapat digerakkan. |
| 3. | Flow | Komputer akan mengerakkan semua pion menuju titik  tujuan bersama-sama. Setiap langkah yang dapat diambil digunakan untuk menggerakkan semua pion agar semakin dekat dengan tujuan. Sebelum melangkah dipilih pion dengan prioritas tertinggi. Pion dengan prioritas tertinggi adalah pion yang paling belakang dan  dapat melangkah. Dengan memilih pion paling belakang kita melangkah dengan tujuan seluruh pion bergerak bersama menuju tujuan. |

1.5. Lambang Pemain

* + Quick, memiliki bidak yang mempunyai lambang ( Q )
  + Killer, memiliki bidak yang mempunyai lambang ( K )
  + Flow, memiliki bidak yang mempunyai lambang ( F )
  + User, memiliki bidak yang mempunyai lambang ( X )

1.6. Fitur

* Times

Pemain memiliki waktu maksimal 30 detik untuk setiap giliran ( dimulai dari melempar dadu hingga bidak yang di pilih pemain berpindah )

* Pause

User dapat menjeda permainan kapanpun selagi program tidak di close dengan menekan “Esc”, dalam artian apabila user menekan tombol “Esc” maka program akan berhenti, dan untuk melanjutkan permaiann yang sebelumnya di mainkan maka user harus menekan tombol “Esc” lagi. Namun apabila user menekan tombol “Esc” dan keluar dari program dengan mengklik tanda silang(x) yang ada pada pojok kanan atas program, maka permainan yang sebelumnya terjeda tidak akan dapat dimainkan apabila user membuka ulang program.

1.7. Memasukan Nama

Sebelum memilih jumlah pemain, user harus menginputkan nama terlebih dahulu.

1.8. Kocok Dadu

Untuk mengocok dadu user harus menekan enter agar angka dari hasil random dadu muncul di layar

1.9. Syarat keluar dari home base

Untuk dapat mengeluarkan bidak ke jalur permainan, pemain harus mendapatkan angka enam dari dadu yang dilemparkan.

1.10. Pergerakan bidak

Pemain menggerakkan bidak yang sudah berada di jalur permainan sesuai dengan angka yang dihasilkan dadu. Pemain melemparkan dadu secara bergantian sesuai urutan yang telah ditentukan ( diawali dari user kemudian memutar searah jarum jam )

1.11. Mendapat angka dadu 6 (≤dua kali)

* Apabila pemain mendapat angka dadu 6 untuk pertama kali maka bidak akan keluar dari home basenya secara otomatis
* Apabila pemain mendapat angka dadu 6 dua kali dan salah satu bidaknya telah berada di lintasan maka pemain tersebut dapat mengocok dadu dua kali. Untuk user akan mendapat pilihan yang akan ditampilkan dilayar berupa “Apakah kamu ingin mengeluarkan bidakmu lagi dari home base?” inputkan “y” untuk mengeluarkan bidak atau inputkan “t” untuk melanjutkan bidak yang ada di lintasan

1.12. Mendapat angka dadu 6 (tiga kali)

Apabila pemain mendapat angka dadu 6 untuk ketiga kalinya, maka bidak yang paling dekat dengan homebasenya, tak terkecuali berapa jumlah bdaknya akan masuk kembali ke home base(dihitung berdasarkan jumlah petak dari home base ke bidak)

1.13. Syarat menuju lintasan finish

Bidak yang sudah melewati satu putaran (52 petak) masuk ke dalam lintasan finish

1.14. Syarat menuju kotak finish

Bidak tidak akan bergerak apabila angka yang dihasilkan dadu belum sesuai untuk masuk ke dalam finish. Contoh: bidak hanya perlu tiga langkah lagi menuju finish, namun angka yang dihasilkan dadu adalah empat, maka bidak tidak akan bergerak menuju finish

1.15. Game Over

Apabila salah satu pemain telah menang maka, pemain lain yang bidaknya paling jauh dari lintasan finish akan Game Over (tidak dapat bermain lagi) . Contoh: Player 1 semua bidaknya telah memasuki finish (menang), kemudian apabila player 2 salah satu bidaknya ada di petak ke-2 setelah home basenya dan player 3 salah satu bidaknya ada di petak ke-5 setelah home basenya, maka player 2 yang Game Over. Jadi di GoRas Ludo Games ini maksimal hanya ada dua pemain yang memasuki finish ( 4 player bermain, maka 2 player masuk finish dan 2 player lain Game Over )

1.16.. Winner

Pemain dikatakan menang apabila semua bidak yang dimilikinya memasuki finish dengan urutan yang petama

1.17.. Permainan selesai

Setelah semua pemain telah memasuki finish maka permainan GoRas Ludo Games telah selesai dan akan muncul tampilan akhir berupa nama pemenang ( nama user / player ke berapa ).

1.18. Membunuh bidak lawan

Jika bidak pemain berhenti di tempat yang telah terisi bidak lawan, maka bidak milik lawan tersebut masuk kembali ke home basenya (tak terkecuali berapa jumlahnya)

**2. Design / Perancangan**

2.1. Struktur Data

* Data Pemain
* File Map

Map.txt adalah file berjenis teks yang berisi tampilan papan permainan dari GoRAs Ludo Games

* File Instruction

Instruction.txt adalah file berjenis teks yang berisi panduan untuk bermain GoRas Ludo Games

* File Credits

Credits.txt adalah file berjenis teks yang berisi credits mengenai

pembuatan program GoRas Ludo Games

2.2. Daftar Modul

Bagian ini menguraikan semua modul yang digunakan dalam program GoRas Ludo Games . Modul - modul tersebut dapat berupa function atau prosedur. Tabel 2.1. adalah uraian mengenai modul-modul yang terlibat.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama Modul** | **Jenis Modul** | **Keterangan** |
| 1. | fullScreen | Procedure | Modul untuk membuat tampilan layar pada program game menjadi penuh ( Full Screen ) |
| 2. | gotoxy | Procedure | Modul yang berperan sebagai koordinat untuk memposisikan objek di layar |
| 3. | header | Procedure | Modul untuk menapilkan tulisan GORAS pada bagian atas program |
| 4. | intro | Procedure | Modul untuk menampilkan tulisan SELAMAT DATANG DO GORAS LUDO GAMES!!!! saat pertama kali membuka program Goras Ludo Games |
| 5. | mainMenu | Procedure | Modul untuk memanggil modul lain yang terdapat di bagian Menu diantaranya Modul NewGames, Instruction, Credits, dan Exit |
| 6. | newGames | Procedure | Modul ini berfungsi untuk memulai permainan dan akan memanggil modul lain yang berperan dalam permainan ludo dari awal hingga akhir |
| 7. | instruction | Procedure | Modul ini berfungsi untuk menampilkan file instruksi yang berisi cara bermain GoRas Ludo Games |
| 10. | credits | Procedure | Modul ini berfungsi untuk menampilkan file credits yang berisi penjelasan pembuat GoRas Ludo Games |
| 11. | selectPlayer | Procedure | Modul untuk menampilkan tampilan berupa pilihan berapa player yang akan bermain. Modul ini akan dipanggil ketika user menginput angka 1 di main menu untuk pilihan New Games |
| 12. | setColor | Procedure | Modul ini akan dipanggil ketika sesuatu yang di tampilkan (print) mengharuskan untuk diberi warna |
| 13. | urutanMain | Procedure | Modul untuk menentukan urutan bermain secara random |
| 14. | timer | Procedure | Modul yang akan dipanggil ketika pemain mendapat giliran untuk menggerakan bidak. Fungsi dari modul ini adalah untuk batas waktu yang diberikan pada pemain dimulai dari random dadu hingga menggerakan bidak |
| 15. | randomDadu | Function | Modul untuk mengeluarkan jumlah angka dadu secara random |
| 16. | pilihLangkah | Function | Modul ini dipanggil ketika salah satu bidak pemain berada di lintasan dan mendapat angka dadu 6. |
| 17. | outFromHomeBase | Procedure | Modul yang akan dipanggil ketika pemain memenuhi kondisi untuk keluar dari home base |
| 18. | movingPion | Procedure | Modul yang akan dipanggil pada saat bidak akan melakukan perpindahan di papan permainan GoRas Ludo Games |
| 19. | membunuhLawan | Procedure | Modul ini akan dipanggil apabila pemain memenuhi kondisi untuk dapat membunuh lawan (pemain menempati petak yang sebelumnya sudah ditempati lawan) setelah modul ini dipanggil maka modul KeHomeBase juga akan dipanggil |
| 20. | backToHomeBase | Procedure | Modul yang akan dipanggil ketika pemain melakukan sesuatu yang memungkinkan dirinya untuk kembali ke home base |
| 21. | beradaDiFinish | Function | Modul yang akan dipanggil ketika bidak pemain akan memasuki lintasan finish (petak ke-52) |
| 22. | isChekWinner | Function | Modul untuk melakukan pemeriksaan pemain ketika memenuhi kondisi yang memungkinkan dirinya menjadi pemenang |
| 23. | isGameOver | Function | Modul ini akan dipanggil oleh Modul IsChekWinner untuk melakukan pemeriksaan. Apabila pemenang telah ditentukan oleh Modul IsChekWinner maka modul ini akan memeriksa posisi petak dari pemain lain, bidak yang terdekat dengan home basenya akan Game Over dan akan muncul tampilan Game Over |
| 24. | user | Function | Modul ini akan memanggil modul Timer() yang bertujuan memulai pergerakan bidak berdasarkan waktu yang telah ditentukan |
| 25. | quick | Function | Modul ini akan berisi logika-logika dari player quick untuk mengambil keputusan cara bermain bidak yang memiliki kemampuan quick (seperti yang dijelaskan di spesifikasi pemain) |
| 26. | killer | Function | Modul ini akan berisi logika-logika dari player Killer untuk mengambil keputusan cara bermain bidak yang memiliki kemampuan Kiler (seperti yang dijelaskan di spesifikasi pemain) |
| 27. | flow | Function | Modul ini akan berisi logika-logika dari player flow untuk mengambil keputusan cara bermain bidak yang memiliki kemampuan flow (seperti yang dijelaskan di spesifikasi pemain) |
| 28. | printMap | Procedure | Modul ini berguna untuk menampilkan papan permaianan pada saat permainan berlangsung |

**Tabel 2.1. Daftar Modul**

2.3. Spesifikasi Modul

Pada bagian ini akan dijelaskan secara lebih spesifik mengenai modul yang sebelumnya diuraikan di Tabel 2.1.

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure fullScreen() |
| I.S | Tampilan program standar (tidak fullscreen) |
| F.S | Tampilan program (console) fullscreen (memenuhi layar) |
| Calling Modul | Function main() |
| Called Modul | - |
| Author | Raigne (cboard.cprogramming.com) |
| Algoritma | void fullScreen() { //make console 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);  } |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure header() |
| I.S | Tampilan atas program kosong |
| F.S | Tampilan atas program terisi dengan tampilan header |
| Calling Modul | Function main() , Procedure selectPlayer() |
| Called Modul | - |
| Author | Mochamad Nauval Dwisatya |
| Algoritma | Temp : Array[1..200] of Char  Read(file\\header.txt)Temp  While Not EOF (header.txt) do  write(layar) Temp  read(header.txt) Temp  end while |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure intro() |
| I.S | Layar pada program kosong |
| F.S | Halaman intro tertampilkan |
| Calling Modul | Function main() |
| Called Modul | - |
| Author |  |
| Algoritma | procedure intro  Deklarasi :  int i,j;  int len;  char wlc[50]  char line[50]  Algoritma  wlc[50] (“SELAMAT DATANG DI GORAS LUDO  GAMES!!!)  line[50] (“============================”)  header();  gotoxy(68,24);  len = strlen(line);  for(i=0; i<len; i++) do  putchar(line[i]);  Sleep(30);  endfor    gotoxy(68,25);  len = strlen(wlc);  for(i=0;i<len;i++) do  putchar(wlc[i]);  Sleep(70);  endfor    gotoxy(68,26);  len = strlen(line);  for(i=0; i<len; i++)do  putchar(line[i]);  Sleep(30);  endfor  write("enter")  gotoxy(68,28);system("PAUSE");  endprocedure intro |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure goToxy() |
| I.S | Text tersusun beraturan dari kiri atas |
| F.S | Text berada pada kordinat yang sudah ditentukan |
| Calling Modul | Procedure printMap(), Procedure header() , Procedure mainMenu() , Procedure selectPlayer() , Procedure intro() |
| Called Modul | - |
| Author | Duthomhas (cplusplus.com). |
| Algoritma | procedure gotoxy(int x, int y) //module for change position object  HANDLE hConsoleOutput;  COORD dwCursorPosition;  dwCursorPosition.X = x;  dwCursorPosition.Y = y;  hConsoleOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);  SetConsoleCursorPosition(hConsoleOutput,dwCursorPosition);  endprocedure gotoxy |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure setColor() |
| I.S | Tampilan pada program tidak berwarna ( warna default) |
| F.S | Program berwarna sesuai warna yang telah diatur dalam  algoritma Procedure setColor() |
| Calling Modul | Procedure header() |
| Called Modul | - |
| Author | https://49012032sehatsempurna.wordpress.com/2012/04/06/merubah-warna-teks-pada-c/ |
| Algoritma | procedure setcolor(unsigned short color)  HANDLE hCon = GetStdHandle(STD\_OUTPUT\_HANDLE);  SetConsoleTextAttribute(hCon,color);  endprocedure setcolor |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure mainMenu() |
| I.S | Layar pada program kosong |
| F.S | Mengarahkan user ke pilihan menu yang diinput |
| Calling Modul | Function main() |
| Called Modul | Procedure newGames() , Procedure instruction() , Procedure credits() |
| Author |  |
| Algoritma | procedure mainMenu  Deklarasi : input : integer  Algoritma :  header();  gotoxy(100,19);write(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  gotoxy(100,20); write ("| |");  gotoxy(100,21); write ("| |");  gotoxy(100,22); write ("| O O |");  gotoxy(100,23); write ("| |");  gotoxy(100,24); write ("| O O |");  gotoxy(100,25); write ("| |");  gotoxy(100,26);write ("| O O |");  gotoxy(100,27); write ("| |");  gotoxy(100,28); write ("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|");  gotoxy(40,19); write (" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  gotoxy(40,20); write ("| |");  gotoxy(40,21); write ("| |");  gotoxy(40,22); write ("| O O |");  gotoxy(40,23); write ("| |");  gotoxy(40,24); write ("| O O |");  gotoxy(40,25); write ("| |");  gotoxy(40,26); write ("| O O |");  gotoxy(40,27); write ("| |");  gotoxy(40,28); write ("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|");  gotoxy(73,20); write ("====================");  gotoxy(73,21); write ("Main Menu");  gotoxy(73,22); write ("====================");  gotoxy(73,24); write ("[1] Mulai Bermain");  gotoxy(73,25); write ("[2] Cara Bermain");  gotoxy(73,26); write ("[3] Credits");  gotoxy(73,27); write ("[4] Keluar Permainan");  gotoxy(65,34); write ("Masukkan angka yang akan  anda tuju = ");  read(input);  switch(input)  case 1 : system("CLS");selectPlayer();break;  case 2 : system("CLS");instruction();break;  case 3 : system("CLS");credits();break;  case 4 : exit(1);break;  default:gotoxy(65,34);write("Maaf angka yang anda inputkan tidak sesuai"); gotoxy(65,36);system("PAUSE");mainMenu();break;  endswitch  endprocedure mainMenu |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function newGames() |
| Input | 1 |
| Output | Modul selectPlayer |
| Calling Modul | Procedure mainMenu() |
| Called Modul | Procedure selectPlayer(), Procedure timer() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure instruction() |
| IS | User menginputkan angka 2 pada program |
| FS | Instruksi games tertampilkan di layar |
| Calling Modul | Procedure mainMenu() |
| Called Modul | - |
| Author |  |
| Algoritma | Temp : Array[1..200] of Char  Read(file\\instruction.txt)Temp  While Not EOF (instruction.txt) do  write(layar) Temp  read(instruction.txt) Temp  end while |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure credits() |
| IS | User menginputkan angka 3 pada program |
| FS | Credits games tertampilkan di layar |
| Calling Modul | Procedure mainMenu() |
| Called Modul | - |
| Author |  |
| Algoritma | Temp : Array[1..200] of Char  Read(file\\credits.txt)Temp  While Not EOF (credits.txt) do  write(layar) Temp  read(credits.txt) Temp  end while |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure selectPlayer() |
| I.S | Modul mainMenu |
| F.S | Menuju ke Modul Timer |
| Calling Modul | Procedure mainMenu() |
| Called Modul | Procedure header() |
| Author |  |
| Algoritma | procedure selectPlayer  Deklarasi :  Input : integer  Algoritma :  system("CLS");  header();  gotoxy(100,19);write(" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  gotoxy(100,20); write ("| |");  gotoxy(100,21); write ("| |");  gotoxy(100,22); write ("| O O |");  gotoxy(100,23); write ("| |");  gotoxy(100,24); write ("| O O |");  gotoxy(100,25); write ("| |");  gotoxy(100,26);write ("| O O |");  gotoxy(100,27); write ("| |");  gotoxy(100,28); write ("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|");  gotoxy(40,19); write (" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  gotoxy(40,20); write ("| |");  gotoxy(40,21); write ("| |");  gotoxy(40,22); write ("| O O |");  gotoxy(40,23); write ("| |");  gotoxy(40,24); write ("| O O |");  gotoxy(40,25); write ("| |");  gotoxy(40,26); write ("| O O |");  gotoxy(40,27); write ("| |");  gotoxy(40,28); write ("|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|");  gotoxy(73,20);write("============================");  gotoxy(73,22); write (" [1] 2 Pemain ");  gotoxy(73,23); write (" [2] 3 Pemain ");  gotoxy(73,24); write (" [3] 4 Pemain ");  gotoxy(73,25); write (" [4] Kembali ke Menu");  gotoxy(73,27);write("============================");  gotoxy(70,33);write("Masukkan angka yang akan anda  tuju = ")  read(input);  switch(input)  case 1 : system("CLS");gamePlay();  break;  case 2 : system("CLS");gamePlay();  break;  case 3 : system("CLS");gamePlay();  break;  case 4 : system("CLS");mainMenu();  break;  default: gotoxy(70,33);printf("Maaf angka yang  anda inputkan tidak sesuai");  gotoxy(70,35);system("PAUSE");  selectPlayer();  break;  endswitch  endprocedure selectPlayer |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure timer() |
| I.S | Waktu dimulai |
| F.S | Waktu berakhir |
| Calling Modul | Function user() , Function collector() , Function killer() , dan Function quick() |
| Called Modul | Function randomDadu() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function user() |
| Input |  |
| Output |  |
| Calling Modul | Procedure newGames() |
| Called Modul | Procedure timer() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function quick() |
| Input |  |
| Output |  |
| Calling Modul | Procedure newGames() |
| Called Modul | Procedure timer() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function killer() |
| Input |  |
| Output |  |
| Calling Modul | Procedure newGames() |
| Called Modul | Procedure Timer() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function flow() |
| Input |  |
| Output |  |
| Calling Modul | Procedure newGames() |
| Called Modul | Procedure Timer() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function randomDadu() |
| Input | Enter |
| Output | Mata dadu tertampilkan ( secara acak ) |
| Calling Modul | Procedure timer() |
| Called Modul | Function pilihLangkah() , Function movingPion() , Function beradaDiFinish() |
| Author |  |
| Algoritma | Function randomDadu  Deklarasi :  angkaDadu,I : integer  Algoritma :  srand(time(NULL));  gotoxy(127,25);printf("TEKAN ENTER UNTUK  MERANDOM");  gotoxy(127,26);printf("-----------------  -----------");  gotoxy(140,25);getch();  for(i=0;i<10;i++) do  gotoxy(127,25);printf("DADU KAMU    SEDANG DI RANDOM");  gotoxy(127,26);printf("-----------  ------------");  angkaDadu = (rand()%6)+1;  showDadu(angkaDadu);Sleep(200);  endfor  for(i=0; i<10; i++) do  setcolor(15);gotoxy(128,38);  write("KAMU MENDAPAT ANGKA DADU %d",angkaDadu);Sleep(100);  setcolor(9);gotoxy(128,38);  write("KAMU MENDAPAT ANGKA DADU %d",angkaDadu);Sleep(100);  endfor  return (angkaDadu);  endfunction ranDomdadu |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function pilihLangkah() |
| Input | y / n |
| Output | Memindahkan bidak / Mengeluarkan bidak |
| Calling Modul | Function randomDadu() |
| Called Modul | Procedure outFromHomeBase() atau Procedure movingPion() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure movingPion() |
| I.S | Bidak berada di petak[j] |
| F.S | Bidak berada di petak[j+i] |
| Calling Modul | Function randomDadu() atau Function pilihLangkah() |
| Called Modul | Procedure outFromHomeBase() atau Procedure movingPion() |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function beradaDiFinish() |
| I.S | False // Pemain belum berada di finish |
| F.S | True // Pemain memenuhi syarat untuk berada di lintasan  finish |
| Calling Modul | Function randomDadu() |
| Called Modul | Function isCheckwinner() , Function isGameOver |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure outFromHomeBase() |
| I.S | Bidak berada di Homebase |
| F.S | Bidak berada di lintasan ( petak pertama) |
| Calling Modul | Function pilihLangkah() |
| Called Modul | - |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure membunuhLawan() |
| I.S | Bidak berada pada petak yang sama dengan bidak lawan |
| F.S | Bidak yang terlebih dahulu ada pada petak kembali ke homebase |
| Calling Modul | Procedure movingPion() |
| Called Modul | Procedure backToHomeBase() |
| Author |  |
| Algoritma |  |

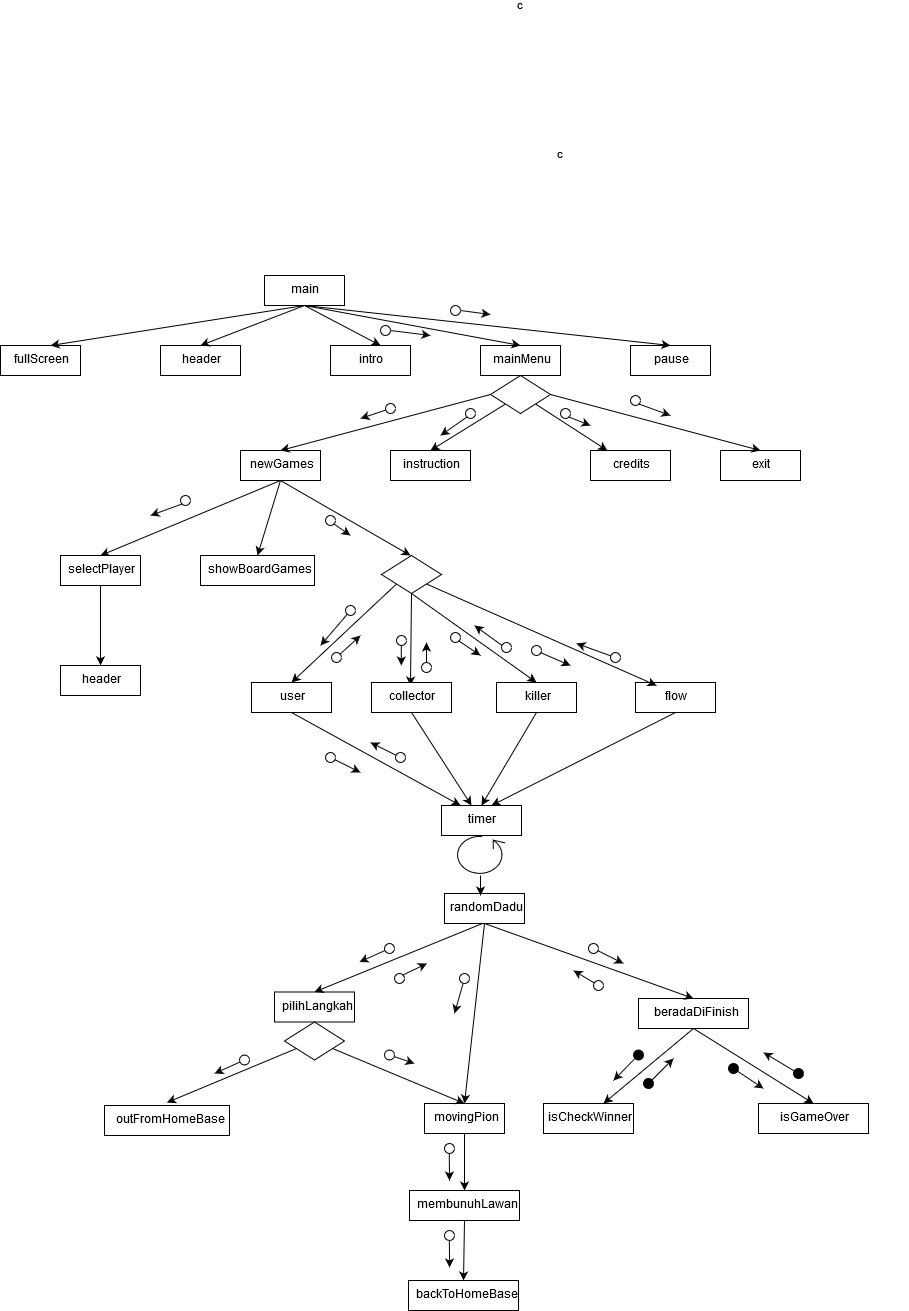
|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure backToHomeBase() |
| I.S | Bidak berada di lintasan (luar homebase) |
| F.S | Bidak berada di homebase |
| Calling Modul | Procedure membunuhLawan() |
| Called Modul | - |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function isCheckWinner() |
| Input | Data koordinat bidak |
| Output | Permainan berlanjut dan muncul tulisan Congratulations //Selama masih ada dua atau lebih pemain, maka permainan akan tetap berlanjut sampai memenuhi jumlah maksimal pemenang |
| Calling Modul | Function beradaDiFinish() |
| Called Modul | - |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Function isGameOver() |
| Input | Semua bidak pada salah satu player telah berada di kotak finish |
| Output | Tampilan Game Over tertampilkan di layar program |
| Calling Modul | Function beradaDiFinish() |
| Called Modul | - |
| Author |  |
| Algoritma |  |

|  |  |
| --- | --- |
| Tipe dan Nama Modul | Procedure printMap() |
| I.S | Layar pada program permainan belum terisi papan  Permainan |
| F.S | Papan permainan tertampilkan di layar |
| Calling Modul | Procedure newGames() |
| Called Modul | - |
| Author |  |
| Algoritma | Temp : Array[1..200] of Char  Int : 8  Read(file\\Map.txt)Temp  While Not EOF (credits.txt) do  write(layar) Temp  i++  end while |

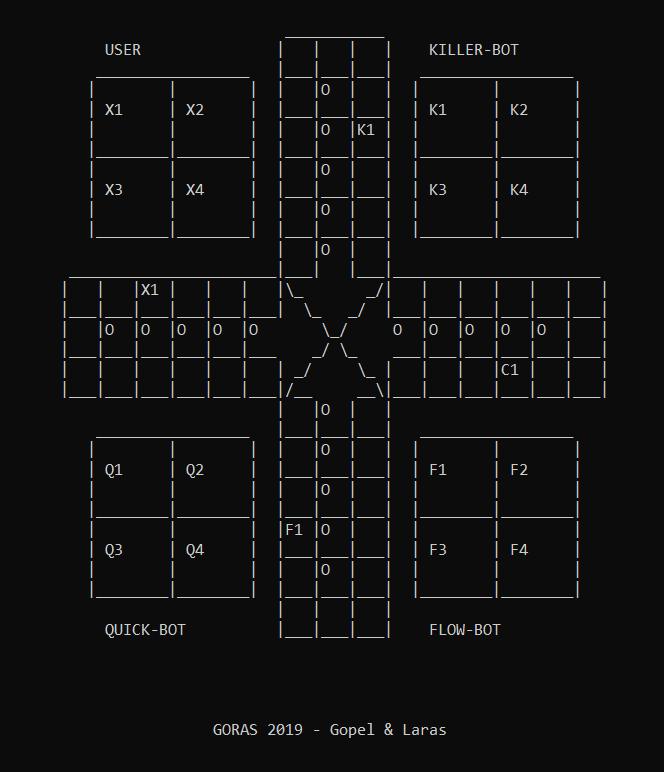
2.4. Structure Chart



**3. Tampilan**

Pada bagian ini diuraikan mengenai rancangan tampilan dalam GoRas Ludo Games, diantaranya :

3.1. Papan Permainan



3.2. Header



3.3 Intro



3.4 Main Menu



3.5 Select Player

