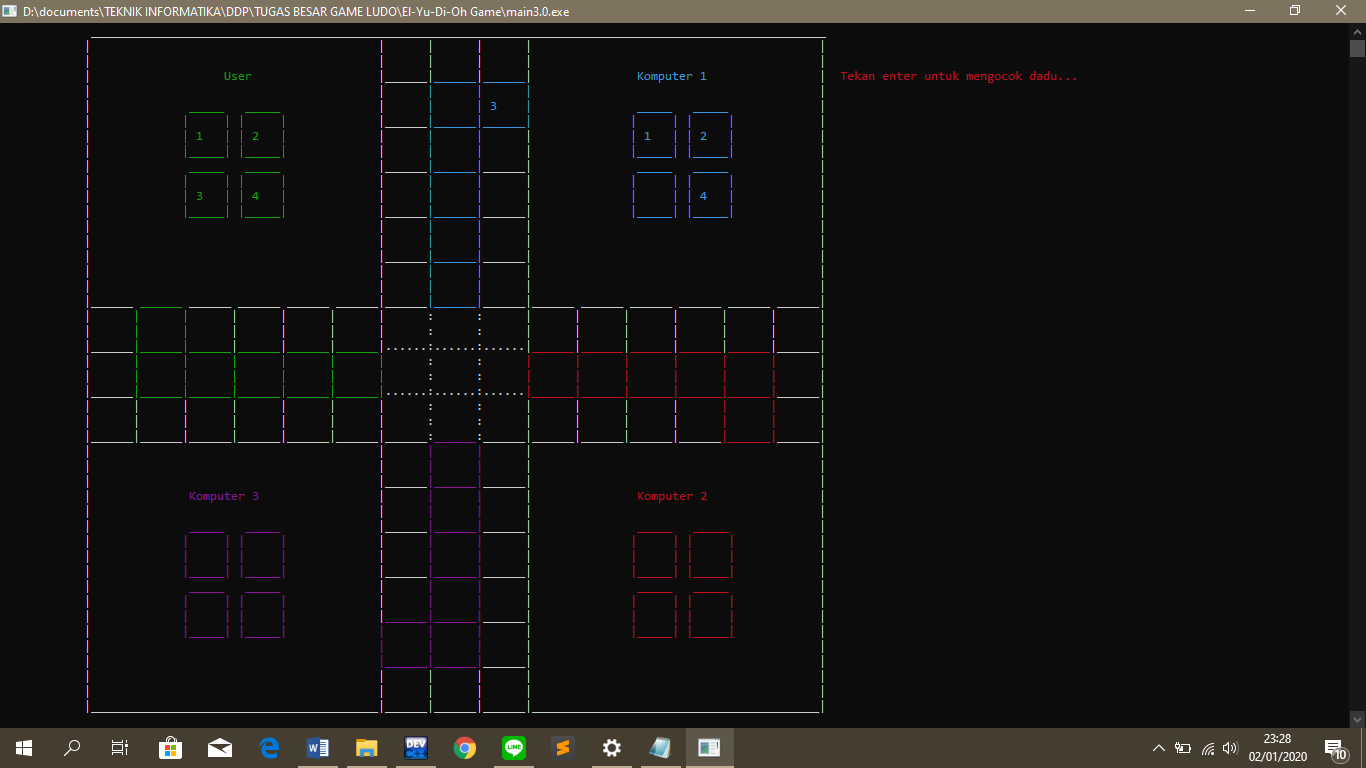
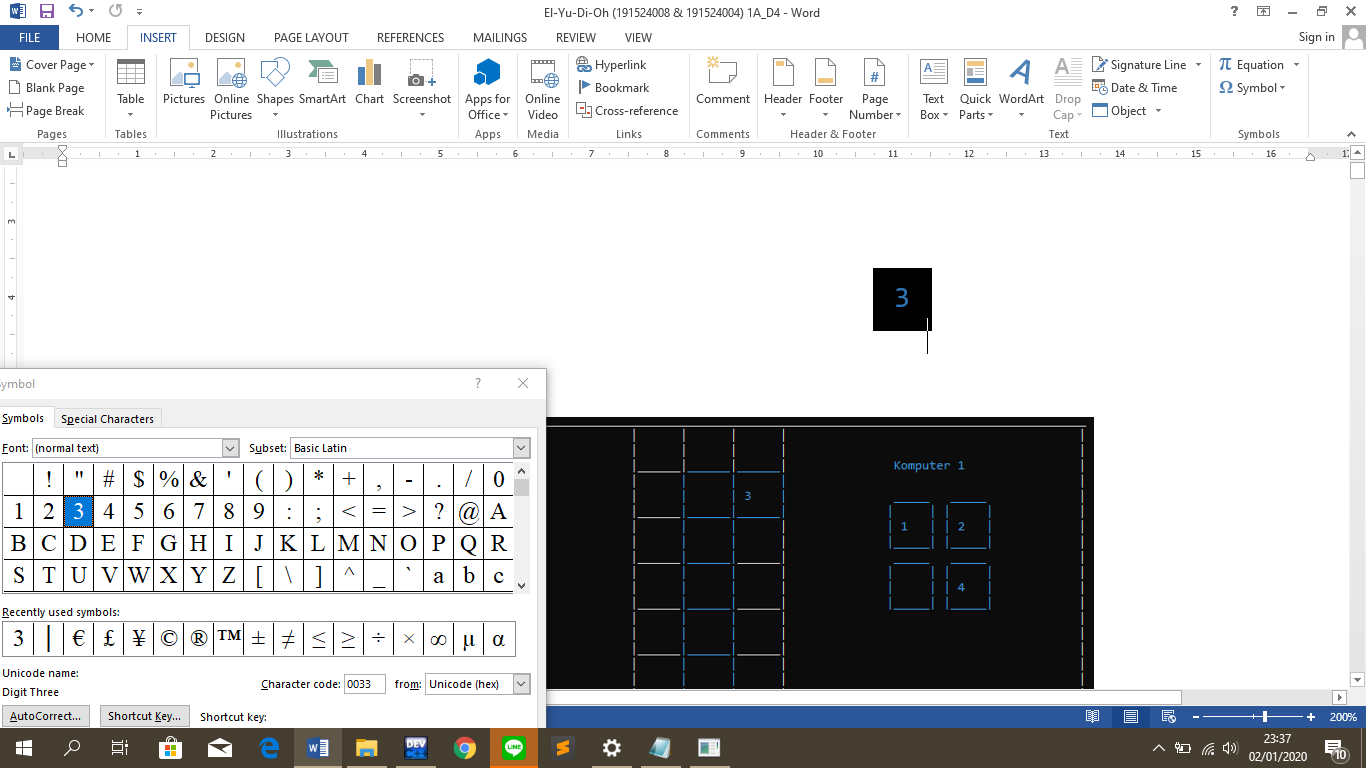
**El-Yu-Di-Oh Game Skenario**

****

**I. Deskripsi Aplikasi**

Deskripsi aplikasi adalah uraian mengenai aplikasi secara utuh. Bagian ini juga akan menjelaskan detail elemen-elemen dari permainan El-Yu-Di-Oh dan cara kerja aplikasi nya..

a). Pemain, permainan ini dimainkan oleh 2 jenis pemain, yaitu :

1. User(manusia), permainan El-Yu-Di-Oh ini hanya bisa dimainkan oleh satu user/ pemain manusia yang akan melawan pemain komputer.

2. Pemain Komputer, yaitu pemain yang berasal dan dimainkan oleh komputer. Permainan El-Yu-Di-Oh ini dapat dimainkan oleh 1-3 komputer sesuai dengan permintaan dari pemain manusia.

b). Bidak, home base dan papan permainan.

- Setiap pemain memiliki 4 bidak yang bisa dijalankan. Dan 1 home base sebagai tempat untuk bidak yang dimiliki nya.

- User dan pemain dari pihak komputer masing-masing mendapat warna yang telah ditentukan secara random untuk empat bidak dan home base nya yang akan ditampilkan pada papan permainan.

- Warna yang menjadi identitas masing-masing pemain tersebut digunakann selama permainan berlangsung.

- Ketika permainan berlangsung, bidak dipindahkan searah jarum jam mengikuti lintasan permainan yang terdapat pada papan permainan .

- Papan permainan El-Yu-Di-Oh terdiri dari 52 petak sebagai lintasan, 4 home base dan 4 lintasan menuju finish dengan warna berbeda, dan 4 petak finish.

**1.1** **Alur Permainan**

Pada game El-Yu-Di-Oh ini, ketika program dibuka akan tampil pada layar berupa sambutan selamat datang di permainan dan informasi mengenai kreator.

Setelah itu, akan ada tampilan main menu yang di dalamnya terdapat beberapa pilihan yang dijelaskan di bawah ini :

● Start Game, adalah menu untuk memulai permainan.

Untuk mulai bermain user harus menginputkan huruf S, kemudian setelah itu akan muncul pilihan berupa jumlah pemain yang akan bermain. Untuk memilih berapa pemain 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)

- Huruf E untuk kembali ke menu utama

● Instruction, adalah menu untuk menampilkan petunjuk atau cara memainkan El-Yu-Di-Oh Game.

Untuk mengetahui cara bermain, user harus menginputkan huruf I, kemudian di layar akan muncul penjelasan teks yang berisi aturan bermain.

● Exit, adalah menu untuk keluar dari program El-Yu-Di-Oh Game.

Untuk keluar dari program El-Yu-Di-Oh, user harus menginputkan huruf E di tampilan Menu Utama

Jika user memilih Start Game, maka user dapat menentukan jumlah pemain komputer yang akan dijadikan lawan, setelah itu akan ditampilkan urutan bermain yang ditentukan secara random. Tahap selanjutnya adalah tampilnya papan permainan, yang menandakan permainan dimulai.

**1.2 Aturan Permainan**

Dalam permainan El-Yu-Di-Oh terdapat beberapa aturan untuk bermain selama game dimainkan. Aturan permainan yang digunakan adalah sebagai berikut:

1. Ketika pemain sudah mendapat giliran bermain, pemain harus melempar dadu dan menggerakan bidak nya atau mengeluarkan bidak nya dari home base.

2. Pemain harus mendapatkan jumlah enam angka dadu dari hasil pelemparan untuk mengeluarkan bidak dari home base.

3. Pemain menentukan bidak mana yang akan dikeluarkan sesuai dengan nomor bidak yang tertera pada bidak miliknya. Jika tidak mendapat jumlah enam angka dadu, pemain tidak dapat mengeluarkan bidaknya.

4. Bidak yang dipilih oleh pemain, akan berpindah petak dimulai dari petak setelah base sesuai dengan jumlah angka dadu yang didapat. Kemudian dilanjutkan dengan pemain lainnya.

5. Ketika user sudah mengeluarkan lebih dari 1 bidak, dan user telah melempar dadu. User dapat memilih bidak mana yang harus berpindah petak.

7. Jika bidak berhenti di petak yang telah terisi oleh bidak lawan, maka bidak lawan diinyatakan “mati” dan akan kembali ke home base milik lawan.

8. Jika user berhenti pada petak yang telah diisi dengan bidak milik sendiri lainnya, bidak tersebut kemudian akan menempati petak yang sama tanpa harus ada salah satu bidak yang kembali ke home base.

9. Jika bidak sudah hampir melewati satu putaran sesuai lintasan dengan selisih 1 petak, maka bidak akan berbelok dan akan mengikuti lintasan yang sesuai dengan warna bidak dan home base nya.

10. Ketika bidak sudah berada di petak pada lintasan menuju finish, bidak tidak akan berpindah petak jika jumlah angka dadu yang dilempar lebih besar dari jumlah sisa petak menuju finish.

Contoh: bidak hanya perlu tiga langkah lagi menuju finish, tetapi jumlah yang dihasilkan adalah empat dadu, maka bidak akan diam di petak yang sedang dipijak. Dan seterusnya sampai bidak selesai.

**1.3 Spesifikasi Input:**

1. Setelah tampilan opening pada layar, user menekan tombola pa saja untuk memasuki tahap selanjutnya, yaitu tampilan main menu.

2. Pada main menu, user dapat menginputkan :

- Huruf S jika memilih Start Game(untuk memulai permainan);

- Huruf I jika memilih Instruction(untuk melihat aturan dan cara bermain);

- Huruf E jika memilih Exit(untuk keluar dari permainan).

3. Untuk memilih berapa pemain 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)

4. User menekan enter untuk melempar dadu.

5. Jika bidak user yang berada di lintasan lebih dari 1 bidak, user menginput bidak mana yang akan digerakkan.

**1.4 Spesifikasi Output:**

1. Jika semua bidak dari salah satu komputer lebih dulu masuk finish maka di layar akan tampil: “Kamu Kalah“ .

2. Jika semua bidak dari user lebih dulu masuk finish maka di layar akan tampil: “Selamat kamu menang“.

**1.5 Fitur tambahan:**

1. User dapat menjeda permainan dengan ketentuan sebagai berikut:

* User dapat menjeda permainan tetapi program (permainan) tidak boleh ditutup.
* Jika program ditutup, maka permainan harus diulang dari awal.

**II. Desain**

**2.1 Kebutuhan Data**

**2.1.1 Data Pemain**

Data pemain digunakan untuk menampung informasi untuk pemain. Terdiri atas 4 pemain, yaitu:

* User
* Komputer1
* Komputer2
* Komputer3

**2.2 Struktur Data :**

       Struktur data akan berisi penjelasan dari kegunaan data tersebut, isi, berikut pseudocodenya.

2.2.1 Pemain

Data pemain berisi data tentang pemain yang bermain. Data pemain bertipe data record.

Pemain akan berisi :

* Warna dari pemain tersebut. Warna akan menentukan bidak dan home base yang dimiliki pemain tersebut.
* Urutan bermain dari pemain tersebut. Urutan tersebut akan memudahkan pemain mengetahui ia berada di urutan ke berapa dan giliran para pemain selama permainan berlangsung.
* Posisi bidak dari pemain tersebut.  Posisi bidak akan menginformasikan letak bidak pemain terrsebut baik ketika dalam lintasan maupun dalam home base.

Berikut tipe data :

Player = record

int giliran

int warna

char bidak[4]

COORD posisibidak[4]

bool bidakdihomebase[4]

int arahjalanbidak[4] // 0 untuk diam, 1 untuk ke kanan, 2 untuk ke atas, 3 untuk ke kiri, dan 4 untuk ke bawah.

bool bidakdifinish[4] //true untuk ya dan false untuk tidak.

Player pemain[] : array

2.2.2 File Instruction

instruksi.txt adalah file berjenis teks yang berisi panduan untuk bermain El-Yu-Di-Oh Game.

**2.3 Modul Program:**

**2.3.1 Daftar Modul**

Bagian ini menjelaskan semua modul yang digunakan dalam program El-Yu-Di-Oh. Modul-modul tersebut dapat berupa function atau prosedur. Tabel 1 adalah uraian mengenai modul-modul yang terlibat.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Modul | Jenis | Keterangan |
| 1. | Main | Function | Modul yang berfungsi sebagai wadah atau tempat dari modul lainnya. |
| 2. | Opening | Prosedur | Menampilkan berupa sambutan selamat datang dan informasi mengenai kreator kepada user pada layar |
| 3. | MainMenu | Function | Program awal yang akan menampilkan pilihan untuk Start Game, Instruction, dan Exit.  User menginput huruf S untuk pilihan Start Game(akan memanggil modul jumlahpemain), huruf I untuk pilihan Instruction(akan memanggil modul instruction), dan huruf E untuk exit(akan memanggil modul exit). |
| 4. | StartGame | Function | Sebagai modul utama (wadah untuk bermain) dalam permainan Ludo. Modul ini akan memanggil semua modul yang berperan dalam permainan. |
| 5. | Exit | Function | Merupakan modul yang sudah tersedia dalam library di bahasa C. Modul ini untuk keluar dari menu yang ada pada menu Start Game. |
| 6. | JumlahPemain | Function | Program utama yang akan menampilkan pilihan untuk bermain dengan berapa pemain (1, 2, atau 3 komputer). |
| 7. | Initial | Prosedur | Menentukan perbedaan pemain (digunakan untuk membedakan bidak dan home base setiap pemain). |
| 8. | PenentuUrutan | Function | Menentukan urutan para pemain untuk bermain yang ditentukan secara random. |
| 9. | RandomDadu | Function | Mengeluarkan jumlah angka mata dadu yang ditentukan secara random dan akan dipakai untuk menggerakkan bidak. |
| 10. | PilihBidak | Prosedur | Untuk mengeluarkan langkah selanjutnya setelah pemain mendapatkan angka dadu 6 yang kedua kalinya atau ketika pemain mendapatkan 6 angka dadu dan sudah ada bidak pemain di lintasan. |
| 11. | KeluarHomeBase | Function | Modul untuk mengeluarkan bidak pemain dari home base. |
| 12. | PenggerakBidak | Function | Modul untuk menggerakan bidak pemain. |
| 13. | PembunuhBidak | Prosedur | Modul dieksekusi pada saat bidak pemain yang mendapat giliran bermain menempati petak yang sudah ditempati bidak lawan. (bidak lawan akan dikembalikan ke home base-nya). |
| 14. | PenentuPemenang | Prosedur | Ketika ada pemain yang semua bidaknya sudah menempati petak finish maka permainan selesai. |
| 15. | Instruction | Prosedur | Modul untuk menampilkan aturan bermain |
| 16. | TampilPapanPermainan | Prosedur | Modul untuk menampilkan papan permainan |
| 17. | PrintBidak | Prosedur | Modul untuk menampilkan bidak pada papan permainan |
| 18. | GameOver | Prosedur | Modul untuk menampilkan tampilan “Selamat Kamu Menang” di layar user yang berarti bahwa gameover untuk pemain komputer. |

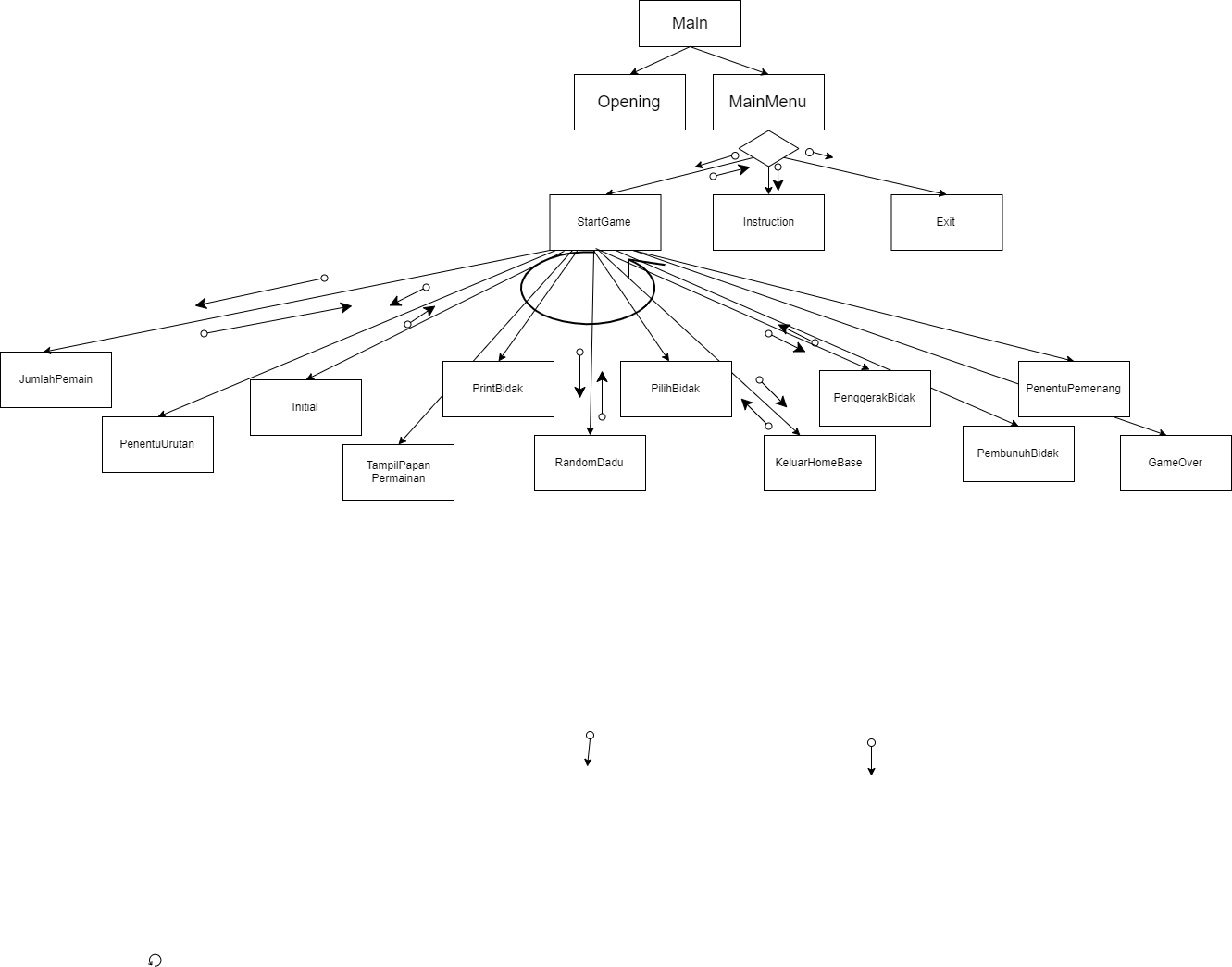
**Tabel 1.Daftar Modul**

**2.3.2 Spesifikasi Modul**

Setiap modul diuraikan atas beberapa bagian, yakni Nama modul, Deskripsi, Initial State, Final State, Modul Pemanggil, Modul yang dipanggil, Author, Version, dan Algoritma. Tabel 2 adalah uraian dari spesifikasi modul-modul yang terlibat.

|  |  |  |
| --- | --- | --- |
| **No** | **Spesifikasi Modul** | **Keterangan** |
| 1. | Function\_ Main() |  |
|  | Deskripsi | Modul yang berfungsi sebagai wadah atau tempat  dari modul lainnya. |
|  | Input | - |
|  | Output | - |
|  | Modul Pemanggil | - |
|  | Modul yang dipanggil | * Prosedur\_Opening() * Function\_MainMenu() |
|  | Author | Farra |
|  | Algoritma  (Dalam Bahasa C) | int main(){  PlaySound(TEXT("D:\\documents\\TEKNIK INFORMATIKA\\DDP\\TUGAS BESAR GAME LUDO\\El-Yu-Di-Oh Game\\soundtrack.wav"), NULL, SND\_FILENAME|SND\_LOOP|SND\_ASYNC);  Opening();  MainMenu();  } |
| 2. | Prosedur\_Opening() |  |
|  | Deskripsi | Modul untuk menampilkan berupa sambutan selamat datang dan informasi mengenai kreator kepada user pada layar |
|  | Initial State | Layar kosong |
|  | Final State | Menampilkan tampilan pembuka sebagai sambutan selamat datang pada user dan informasi mengenai kreator pada layar |
|  | Modul Pemanggil | Function\_Main() |
|  | Modul yang dipanggil | - |
|  | Author | Farra |
|  | Algoritma  (Dalam bahasa C) | void Opening(){  int i,j,k;  int d=10;  GoToXY(75, 6);  char a[]=" Welcome to\n";  for (k=0;k<24;k++){  PenentuWarna(4);  int q;  for(q=0;q<10000000;q++){  }  printf("%c", a[k]);  PenentuWarna(7);  }  char teks1[114]={"------------------------------------------------------------------------------------------------------------------"};  char teks2[114]= {"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"};  char teks3[114]={"------------------------------------------------------------------------------------------------------------------"};  char teksLudo1[111]=  {" \_|\_|\_|\_| \_| \_| \_| \_|\_|\_| \_| \_|\_| \_| "};  char teksLudo2[111]=  {" \_| \_| \_| \_| \_| \_| \_| \_| \_| \_| \_|\_|\_| "};  char teksLudo3[111]=  {" \_|\_|\_| \_| \_|\_|\_|\_|\_| \_| \_| \_| \_|\_|\_|\_|\_| \_| \_| \_| \_|\_|\_|\_|\_| \_| \_| \_| \_| "};  char teksLudo4[111]=  {" \_| \_| \_| \_| \_| \_| \_| \_| \_| \_| \_| \_| "};  char teksLudo5[111]=  {" \_|\_|\_|\_| \_|\_|\_|\_| \_| \_|\_|\_| \_|\_|\_| \_| \_|\_| \_| \_| "};  char teksLudo6[111]=  {" G A M E "};  char teks4[114]=  {"------------------------------------------------------------------------------------------------------------------"};  char teks5[114]=  {"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"};  char teks6[114]=  {"------------------------------------------------------------------------------------------------------------------"};  for(i=0;i<114;i++){  PenentuWarna(1);  GoToXY(30+i,8); printf("%c",teks1[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<114;i++){  PenentuWarna(3);  GoToXY(30+i,10); printf("%c",teks2[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<114;i++){  PenentuWarna(1);  GoToXY(30+i,12); printf("%c",teks3[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<114;i++){  PenentuWarna(1);  GoToXY(30+i,30); printf("%c",teks4[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<114;i++){  PenentuWarna(3);  GoToXY(30+i,32); printf("%c",teks5[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<114;i++){  PenentuWarna(1);  GoToXY(30+i, 34); printf("%c",teks6[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,16); printf("%c",teksLudo1[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,18); printf("%c",teksLudo2[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,20); printf("%c",teksLudo3[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,22); printf("%c",teksLudo4[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,24); printf("%c",teksLudo5[i]);  for(j=0;j<900000;j++){  }  }  for(i=0;i<111;i++){  PenentuWarna(6);  GoToXY(30+i,26); printf("%c",teksLudo6[i]);  for(j=0;j<900000;j++){  }  }  char b[]="created by Ali Nurdin (191524004)\n";  char c[]="and Farra Jessica (191524008)\n";  GoToXY(69.5, 35);  for (k=0;k<35;k++){  PenentuWarna(6);  int q;  for(q=0;q<10000000;q++){  }  printf("%c", b[k]);  }  GoToXY(71.5, 36);  for (k=0;k<31;k++){  PenentuWarna(6);  int q;  for(q=0;q<10000000;q++){  }  printf("%c", c[k]);  }  PenentuWarna(4);  GoToXY(67.5, 37);  printf("Tekan enter untuk masuk ke main menu");  getch();  system("cls");  PenentuWarna(7);  } |
| 3. | Function\_ MainMenu () |  |
|  | Deskripsi | Program awal yang akan menampilkan pilihan untuk Start Game, Instruction, dan Exit.  User menginput huruf S untuk pilihan Start Game(akan memanggil modul jumlahpemain), huruf I untuk pilihan Instruction(akan memanggil modul instruction), dan huruf E untuk exit(akan memanggil modul exit). |
|  | Input | User menekan tombol enter |
|  | Output | Tampilnya pilihan “Start Game”, “Instruction”, dan “Exit” |
|  | Modul Pemanggil | Function\_Main() |
|  | Modul yang dipanggil | * - Function\_StartGame() * - Prosedur\_Instruction() * - Function\_Exit() |
|  | Author | Farra |
|  | Algoritma  (Dalam bahasa C) | MainMenu(){  int i;  GoToXY(80.5, 10);  printf("S. Start Game\n");  GoToXY(80.5,15);  printf("I. Instruction\n");  GoToXY(80.5,20);  printf("E. Exit\n");  GoToXY(70.5, 25);  printf("Masukkan huruf yang tertera diatas ");  scanf("%c", &i);  if (i == ‘S’ || i == ‘s’){  StartGame();  } else if (i == ‘I’ || i== ‘I’){  Instruction();  } else if(i == ‘E’ || i == ‘e’){  exit(1);  }  } |
| 4. | Function\_ StartGame() |  |
|  | Deskripsi | Modul utama (sebagai wadah untuk bermain) dalam permainan Ludo. Modul ini akan memanggil semua modul yang berperan dalam permainan. |
|  | Input | Kondisi layar masih di Main Menu |
|  | Output | Modul Prosedur\_JumlahPemain() terpanggil |
|  | Modul Pemanggil | Function\_MainMenu() |
|  | Modul yang dipanggil | * Function\_JumlahPemain() * Prosedur\_Initial() * Function\_PenentuUrutan() * Prosedur\_TampilPapanPermainan() * Prosedur\_PrintBidak |
|  | Author | Farra & Ali |
|  | Algoritma  (Dalam bahasa C) | StartGame(){  int jumlahpemain, penentuurutan, i, n;  system("cls");  jumlahpemain = JumlahPemain();  int urutanpemain[jumlahpemain+1];  PenentuUrutan(urutanpemain,jumlahpemain+1);  for(i=0; i<jumlahpemain+1; i++){  int j;  pemain[i].giliran = urutanpemain[i];  pemain[i].warna = i+2;  for (j=0; j<4; j++){  pemain[i].bidak[j] = j;  }  }  initial(jumlahpemain+1);  system("cls");  bool menang = false;  int pemenang = NULL;  int playerke = 0;  while (menang != true){  int giliran, m;  for (giliran=1; giliran <= jumlahpemain+1; giliran++){  TampilPapanPermainan();  printbidak(jumlahpemain+1);  int nomorbidak;  bool semuabidakdibase = true, sudahmilih = false;  int dadu = rand()%6 + 1;  if(pemain[playerke].giliran == giliran){  for (m=0; m<4; m++){  if (pemain[playerke].bidakdihomebase[m] == false){  semuabidakdibase = false;  break;  }  }  PenentuWarna(4);  GoToXY(120, 4); printf("Tekan enter untuk mengocok dadu...");  PenentuWarna(7);  getch();  GoToXY(120, 4); printf("Kamu mendapat angka dadu %d ", dadu);  if(semuabidakdibase && dadu<6){  GoToXY(120, 5); printf("Maaf Kamu tidak bisa mengeluarkan bidak...\n");  GoToXY(120, 6); system("pause");  }else if(semuabidakdibase && dadu==6){  PenentuWarna(4);  GoToXY(120, 5); printf("Pilih angka bidak yang mau dikeluarkan (1-4) ");  PenentuWarna(7);  PilihBidak(&nomorbidak);  KeluarHomeBase(playerke, nomorbidak-1);  }else if(!semuabidakdibase && dadu==6){  PenentuWarna(4);  GoToXY(120, 5); printf("Pilih angka bidak yang mau dikeluarkan/digerakkan (1-4) ");  PenentuWarna(7);  PilihBidak(&nomorbidak);//scanf("%d", &nomorbidak);  if(pemain[playerke].bidakdihomebase[nomor bidak-1] == true ){  KeluarHomeBase(playerke, nomorbidak-1);  }else{  for(n=0; n<dadu;n++){  PenggerakBidak(playerke, nomorbidak-1, dadu);  }  }  }else if(!semuabidakdibase && dadu<6){  PenentuWarna(4);  GoToXY(120, 5); printf("Pilih angka bidak yang mau dijalankan...");  PenentuWarna(7);  PilihBidak(&nomorbidak);  bool boleh = false;  while (!boleh){  if(pemain[playerke].bidakdihomebase[nomor bidak-1] == false){  boleh = true;  }else{  PenentuWarna(4);  GoToXY(120, 5); printf("Pilih angka bidak yang mau dijalankan... ");  PenentuWarna(7);  PilihBidak(&nomorbidak);  }  }  if(pemain[playerke].bidakdihomebase[nomor bidak-1] == false ){  for(n=0; n<dadu;n++){  PenggerakBidak(playerke, nomorbidak-1, dadu);  }  }  }  }else if (pemain[playerke+1].giliran == giliran){  for (m=0; m<4; m++){  if (pemain[playerke+1].bidakdihomebase[m] == false){  semuabidakdibase = false;  break;  }  }  GoToXY(120, 4); printf("Komputer 1 silahkan mengocok dadu");  sleep(2);  GoToXY(120, 4); printf("Komputer 1 mendapat angka dadu %d ", dadu);  sleep(2);  if(semuabidakdibase && dadu<6){  }else if(semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  GoToXY(120, 5); printf("Komputer 1 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+1, nomorbidak-1);  }else if(!semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  if(pemain[playerke+1].bidakdihomebase[nomorbidak-1] == true ){  GoToXY(120, 5); printf("Komputer 1 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+1, nomorbidak-1);  }else{  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 1 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+1, nomorbidak-1, dadu);  }  }  }else if(!semuabidakdibase && dadu<6){  bool terpilih = false;  while (!terpilih){  int terpilih01 = rand()%4;  if(pemain[playerke+1].bidakdihomebase[ter pilih01] == false){  nomorbidak = terpilih01+1;  terpilih = true;  }  }  if(pemain[playerke+1].bidakdihomebase[nomorbidak-1] == false ){  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 1 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+1, nomorbidak-1, dadu);  }  }  }  sleep(2);  }else if (pemain[playerke+2].giliran == giliran){  for (m=0; m<4; m++){  if (pemain[playerke+2].bidakdihomebase[m] == false){  semuabidakdibase = false;  break;  }  }  GoToXY(120, 4); printf("Komputer 2 silahkan mengocok dadu");  sleep(2);  GoToXY(120, 4); printf("Komputer 2 mendapat angka dadu %d ", dadu);  sleep(2);  if(semuabidakdibase && dadu<6){  }else if(semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  GoToXY(120, 5); printf("Komputer 2 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+2, nomorbidak-1);  }else if(!semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  if(pemain[playerke+2].bidakdihomebase[nomorbidak-1] == true ){  GoToXY(120, 5); printf("Komputer 2 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+2, nomorbidak-1);  }else{  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 2 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+2, nomorbidak-1, dadu);  }  }  }else if(!semuabidakdibase && dadu<6){  bool terpilih = false;  while (!terpilih){  int terpilih01 = rand()%4;  if(pemain[playerke+2].bidakdihomebase[ter pilih01] == false){  nomorbidak = terpilih01+1;  terpilih = true;  }  }  if(pemain[playerke+2].bidakdihomebase[nomorbidak-1] == false ){  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 2 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+2, nomorbidak-1, dadu);  }  }  }  sleep(2);  }else if (pemain[playerke+3].giliran == giliran){  for (m=0; m<4; m++){  if (pemain[playerke+3].bidakdihomebase[m] == false){  semuabidakdibase = false;  break;  }  }  GoToXY(120, 4); printf("Komputer 3 silahkan mengocok dadu");  sleep(2);  GoToXY(120, 4); printf("Komputer 3 mendapat angka dadu %d ", dadu);  sleep(2);  if(semuabidakdibase && dadu<6){  }else if(semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  GoToXY(120, 5); printf("Komputer 3 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+3, nomorbidak-1);  }else if(!semuabidakdibase && dadu==6){  nomorbidak = rand()%4+1;  if(pemain[playerke+3].bidakdihomebase[nomorbidak-1] == true ){  GoToXY(120, 5); printf("Komputer 3 mengeluarkan bidak %d", nomorbidak);  KeluarHomeBase(playerke+3, nomorbidak-1);  }else{  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 3 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+3, nomorbidak-1, dadu);  }  }  }else if(!semuabidakdibase && dadu<6){  bool terpilih = false;  while (!terpilih){  int terpilih01 = rand()%4;  if(pemain[playerke+3].bidakdihomebase[ter pilih01] == false){  nomorbidak = terpilih01+1;  terpilih = true;  }  }  if (pemain[playerke+3].bidakdihomebase[nomor bidak-1] == false ){  for(n=0; n<dadu;n++){  GoToXY(120, 5); printf("Komputer 3 menggerakkan bidak %d", nomorbidak);  PenggerakBidak(playerke+3, nomorbidak-1, dadu);  }  }  }  sleep(2);  }  PenentuPemenang(&menang, &pemenang, giliran);//cek menang atau engga...  }  }  } |
| 5. | Function\_ Exit() | Merupakan modul yang sudah tersedia dalam library di bahasa C |
|  | Deskripsi | Modul ini untuk keluar dari menu yang ada pada menu Start Game. |
|  | Input | Program yang masih berjalan |
|  | Output | Program sudah berakhir |
|  | Modul Pemanggil | Function\_MainMenu() |
|  | Modul yang dipanggil | - |
|  | Author | - |
|  | Algoritma  (Dalam Bahasa C) | {  exit(1);  } |
| 6. | Function\_ JumlahPemain() |  |
|  | Deskripsi | Modul yang berperan sebagai program utama yang akan menampilkan pilihan untuk bermain dengan berapa pemain (1, 2, atau 3 komputer). |
|  | Input | Layar kosong |
|  | Output | Tampilnya pilihan jumlah pemain komputer untuk lawan dari user, yaitu 1 pemain, 2 pemain, dan 3 pemain. |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Farra |
|  | Algoritma  (Dalam bahasa C) | int JumlahPemain(){  int j=60;  GoToXY(j, 20);  printf("Masukkan jumlah lawan (komputer) (1-3):\n");  int jumlahpemain, i;  GoToXY(j+40, 20); scanf("%d", &jumlahpemain);  if (jumlahpemain<1 || jumlahpemain>3){  GoToXY(62, 20); printf("Masukan jumlah pemain >= 1 atau <=3.\n");  return JumlahPemain();  }  return jumlahpemain;  } |
| 7. | Prosedur\_Initial() |  |
|  | Deskripsi | Modul untuk menentukan perbedaan pemain (digunakan untuk membedakan bidak dan home base setiap pemain). |
|  | Inisial State | Pemain belum mendapatkan warna |
|  | Final State | Tampilan warna untuk masing-masing pemain |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Ali |
|  | Algoritma  (Dalam bahasa C) | void initial(int jumlahpemain){  int j, i;  for (j=0; j<jumlahpemain; j++){  for(i=0; i<4; i++){  pemain[j].bidakdihomebase[i] = true;  pemain[j].bidakdifinish[i] = false;  }  }  for(i=0; i<4; i++){  pemain[0].arahjalanbidak[i] = 1;  pemain[1].arahjalanbidak[i] = 4;  pemain[2].arahjalanbidak[i] = 3;  pemain[3].arahjalanbidak[i] = 2;  }  if(jumlahpemain >= 2){  pemain[0].posisibidak[0].X = 28;  pemain[0].posisibidak[0].Y = 8;  pemain[0].posisibidak[1].X = 36;  pemain[0].posisibidak[1].Y = 8;  pemain[0].posisibidak[2].X = 28;  pemain[0].posisibidak[2].Y = 12;  pemain[0].posisibidak[3].X = 36;  pemain[0].posisibidak[3].Y = 12;  pemain[1].posisibidak[0].X = 92;  pemain[1].posisibidak[0].Y = 8;  pemain[1].posisibidak[1].X = 100;  pemain[1].posisibidak[1].Y = 8;  pemain[1].posisibidak[2].X = 92;  pemain[1].posisibidak[2].Y = 12;  pemain[1].posisibidak[3].X = 100;  pemain[1].posisibidak[3].Y = 12;  }  if (jumlahpemain >= 3){  pemain[2].posisibidak[0].X = 92;  pemain[2].posisibidak[0].Y = 36;  pemain[2].posisibidak[1].X = 100;  pemain[2].posisibidak[1].Y = 36;  pemain[2].posisibidak[2].X = 92;  pemain[2].posisibidak[2].Y = 40;  pemain[2].posisibidak[3].X = 100;  pemain[2].posisibidak[3].Y = 40;  }  if (jumlahpemain == 4){  pemain[3].posisibidak[0].X = 28;  pemain[3].posisibidak[0].Y = 36;  pemain[3].posisibidak[1].X = 36;  pemain[3].posisibidak[1].Y = 36;  pemain[3].posisibidak[2].X = 28;  pemain[3].posisibidak[2].Y = 40;  pemain[3].posisibidak[3].X = 36;  pemain[3].posisibidak[3].Y = 40;  }  } |
| 8. | Function\_ PenentuUrutan() |  |
|  | Deskripsi | Modul untuk menentukan urutan para pemain untuk bermain secara random. |
|  | Input | Jumlah pemain |
|  | Output | Tampilan urutan pemain yang telah ditentukan secara random sesuai dengan jumlah pemain |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Ali |
|  | Algoritma  (Dalam bahasa C) | int \* PenentuUrutan(int urutanpemain[], int jumlahpemain){  int a[jumlahpemain], i, urutan;  int urutanakhir = 0;  urutan = (rand()%jumlahpemain)+1;  a[0] = urutan;  for (i=1; i<jumlahpemain; i++){  a[i] = a[i-1]-1;  }  for(i=1; i<jumlahpemain; i++){  if (a[i]<1 || a[i]>jumlahpemain){  a[i] =jumlahpemain;  urutanakhir=i;  i=jumlahpemain+1;  }  }  if (urutanakhir != 0){  for (i=urutanakhir+1; i<jumlahpemain;i++){  a[i]=a[i-1]-1;  }  }  for(i=0; i<jumlahpemain;i++){  urutanpemain[i] = a[i];  }  for(i=0; i<jumlahpemain; i++){  if(i>=1){  GoToXY(60, 22+i);  printf("- Urutan komputer%d : %d", i, urutanpemain[i]);  }else{  GoToXY(60, 22+i);  printf("- Urutan pemain: %d", urutanpemain[i]);  }  }  getch();  return a;  } |
| 9. | Function\_ RandomDadu() |  |
|  | Deskripsi | Modul untuk mengeluarkan jumlah angka dadu yang ditentukan secara random dan akan dipakai untuk menggerakkan bidak. |
|  | Input | - |
|  | Output | Tampilnya jumlah angka mata dadu yang didapat oleh pemain yang mendapat giliran. |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | Prosedur\_PenentuWarna |
|  | Author | Farra |
|  | Algoritma  (Dalam Bahasa C) | void RandomDadu(int angkadadu){  switch(angkadadu) {  case 0 :  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| |");  GoToXY(135,13);printf("| |");  GoToXY(135,14);printf("| |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  break;  PenentuWarna(7);  case 1 :  PenentuWarna(1); //warna : biru tua  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| |");  GoToXY(135,13);printf("| \* |");  GoToXY(135,14);printf("| |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  case 2 :  PenentuWarna(2); //warna : hijau  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| \* |");  GoToXY(135,13);printf("| |");  GoToXY(135,14);printf("| \* |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  case 3 :  PenentuWarna(3); //warna : biru muda  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| \* |");  GoToXY(135,13);printf("| \* |");  GoToXY(135,14);printf("| \* |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  case 4 :  PenentuWarna(4); //warna : merah  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| \* \* |");  GoToXY(135,13);printf("| |");  GoToXY(135,14);printf("| \* \* |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  case 5 :  PenentuWarna(5); //warna : ungu  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| \* \* |");  GoToXY(135,13);printf("| \* |");  GoToXY(135,14);printf("| \* \* |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  case 6 :  PenentuWarna(6); //warna : orange  GoToXY(135,10);printf(" \_\_\_\_\_\_\_\_\_\_\_");  GoToXY(135,11);printf("| |");  GoToXY(135,12);printf("| \* \* |");  GoToXY(135,13);printf("| \* \* |");  GoToXY(135,14);printf("| \* \* |");  GoToXY(135,15);printf("|\_\_\_\_\_\_\_\_\_\_\_|");  PenentuWarna(7);  break;  }  } |
| 10. | Prosedur\_ PilihBidak() |  |
|  | Deskripsi | Modul untuk mengeluarkan langkah selanjutnya setelah pemain mendapatkan angka dadu 6 yang kedua kalinya atau ketika pemain mendapatkan 6 angka dadu dan sudah ada bidak pemain di lintasan. |
|  | Inisial State | Pemain mendapat angka dadu dan bidak pemain masih berada di petak semula |
|  | Final State | Terpilih langkah selanjutnya, diantara pengeluaran bidak atau penggerakan bidak |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Ali |
|  | Algoritma | void PilihBidak(int \*nomorbidak){  bool sudahmemilih = false;  while(!sudahmemilih){  int nomor;  scanf("%d", &nomor);  if (nomor<1 || nomor>4){  GoToXY(100, 5); printf("Nomor bidak tidak ada...");  }else{  \*nomorbidak = nomor;  sudahmemilih = true;  }  }  } |
| 11. | Function\_ KeluarHomeBase() |  |
|  | Deskripsi | Modul untuk mengeluarkan bidak pemain dari home base. |
|  | Input | Angka 6 dari mata dadu yang telah dirandom |
|  | Output | Bidak pemain berada di lintasan pada papan permainan |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Farra & Ali |
|  | Algoritma  (Dalam Bahasa C) | int KeluarHomeBase(int playerke, int nomorbidak){  if (playerke == 0){  pemain[playerke].posisibidak[ nomorbidak].X = 21;  pemain[playerke].posisibidak[ nomorbidak].Y = 21;  pemain[playerke].homebase[nomorbidak] = false;  }  if (playerke == 1){  pemain[playerke].posisibidak[ nomorbidak].X = 70;  pemain[playerke].posisibidak[ nomorbidak].Y = 6;  pemain[playerke].homebase[nomorbidak] = false;  }  if (playerke == 2){  pemain[playerke].posisibidak[ nomorbidak].X = 105;  pemain[playerke].posisibidak[  nomorbidak].Y = 27;  pemain[playerke].homebase[nomor bidak] = false;  }  if (playerke == 3){  pemain[playerke].posisibidak[ nomorbidak].X = 56;  pemain[playerke].posisibidak[ nomorbidak].Y = 42;  pemain[playerke].homebase[nomor bidak] = false;  }  } |
| 12. | Function\_ Penggerak Bidak() |  |
|  | Deskripsi | Modul untuk menggerakan bidak pemain. |
|  | Input | Bidak berada dalam petak |
|  | Output | Bidak berpindah petak |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - |
|  | Author | Ali |
|  | Algoritma | int PenggerakBidak(int playerke, int nomorbidak, int dadu){  if((pemain[playerke].posisibidak [nomorbidak].X-(dadu\*7) < 70) && (70<pemain[playerke].posisibidak [nomorbidak].X) && (pemain[playerke].posisibidak [nomorbidak].X<=105) && playerke==2 && pemain[playerke].posisibidak [nomorbidak].Y == 24){  }else if(pemain[playerke].posisibidak [nomorbidak].Y-(dadu\*3) < 27 && (27<pemain[playerke].posisibidak [nomorbidak].Y) && (pemain[playerke].posisibidak[nomorbidak].Y<=42) && playerke==3 && pemain[playerke].posisibidak[nomor bidak].X == 63){  }else if(playerke == 0 && pemain[playerke].posisibidak[nomor bidak].X == 49 && pemain[playerke].posisibidak[nomor bidak].Y == 24){ pemain[playerke].posisibidak [nomorbidak].X = 56;  pemain[playerke].posisibidak [nomorbidak].Y = 24;  pemain[playerke].arahjalanbidak [nomorbidak] = 0;  pemain[playerke].bidakdifinish [nomorbidak] = true; //finish hijau (player 1);  }else if(playerke == 1 && pemain[playerke].posisibidak [nomorbidak].X == 63 && pemain[playerke].posisibidak [nomorbidak].Y == 18){  pemain[playerke].posisibidak[nomorbidak].X = 63;  pemain[playerke].posisibidak[nomorbidak].Y = 21;  pemain[playerke].arahjalan bidak[nomorbidak] = 0;  pemain[playerke].bidakdi finish[nomorbidak] = true; //finish biru (player 2);  }else if(playerke == 2 && pemain[playerke].posisibidak[nomorbidak].X == 77 && pemain[playerke].posisibidak[nomorbidak].Y == 24){  pemain[playerke].posisibidak[nomorbidak].X = 70;  pemain[playerke].posisibidak[nomorbidak].Y = 24;  pemain[playerke].arahjalan bidak[nomorbidak] = 0;  pemain[playerke].bidakdi finish[nomorbidak] = true; //finish merah (player 3);  }else if(playerke == 3 && pemain[playerke].posisibidak[nomorbidak].X == 63 && pemain[playerke].posisibidak[nomorbidak].Y == 30){  pemain[playerke].posisibidak[nomorbidak].X = 63;  pemain[playerke].posisibidak[nomorbidak].Y = 27;  pemain[playerke].arahjalan bidak[nomorbidak] = 0;  pemain[playerke].bidakdi finish[nomorbidak] = true; //finish ungu (player 1);  }else if(playerke == 0 && pemain[playerke].posisibidak[nomorbidak].X == 14 && pemain[playerke].posisibidak[nomorbidak].Y == 24){  pemain[playerke].posisibidak[nomorbidak].X = 21;  pemain[playerke].posisibidak[nomorbidak].Y = 24;  pemain[playerke].arahjalan bidak[nomorbidak] = 1; //menuju garis finish hijau (player 1);  }else if(playerke == 1 && pemain[playerke].posisibidak[nomorbidak].X == 63 && pemain[playerke].posisibidak[nomorbidak].Y == 3){  pemain[playerke].posisibidak[nomorbidak].X = 63;  pemain[playerke].posisibidak[nomorbidak].Y = 3;  pemain[playerke].arahjalan bidak[nomorbidak] = 4; //menuju garis finish biru (player 2);  }else if(playerke == 2 && pemain[playerke].posisibidak[nomorbidak].X == 112 && pemain[playerke].posisibidak[nomorbidak].Y == 24){  pemain[playerke].posisibidak[nomorbidak].X = 105;  pemain[playerke].posisibidak[nomorbidak].Y = 24;  pemain[playerke].arahjalan bidak[nomorbidak] = 3; //menuju garis finish merah (player 3);  }else if(playerke == 3 && pemain[playerke].posisibidak[nomorbidak].X == 63 && pemain[playerke].posisibidak[nomorbidak].Y == 45){  pemain[playerke].posisibidak[nomorbidak].X = 63;  pemain[playerke].posisibidak[nomorbidak].Y = 42;  pemain[playerke].arahjalan bidak[nomorbidak] = 2; //menuju garis finish ungu (player 3);  }else if(pemain[playerke].posisibidak[nomorbidak].X == 49 && pemain[playerke].posisibidak[nomorbidak].Y == 21){  pemain[playerke].posisibidak[nomorbidak].X = 56;  pemain[playerke].posisibidak[nomorbidak].Y = 18;  pemain[playerke].arahjalan bidak[nomorbidak] = 2; //tengah kiri atas  }else if(pemain[playerke].posisibidak [nomorbidak].X == 56 && pemain[playerke].posisibidak [nomorbidak].Y == 3){  pemain[playerke].posisibidak [nomorbidak].X = 63;  pemain[playerke].posisibidak [nomorbidak].Y = 3;  pemain[playerke].arahjalanbidak [nomorbidak] = 1; //atas kiri  }else if(pemain[playerke].posisibidak [nomorbidak].X == 70 && pemain[playerke].posisibidak [nomorbidak].Y == 3){  pemain[playerke].posisibidak [nomorbidak].X = 70;  pemain[playerke].posisibidak [nomorbidak].Y = 6;  pemain[playerke].arahjalanbidak [nomorbidak] = 4; //atas kanan  }else if(pemain[playerke].posisibidak [nomorbidak].X == 70 && pemain[playerke].posisibidak [nomorbidak].Y == 18){  pemain[playerke].posisibidak [nomorbidak].X = 77;  pemain[playerke].posisibidak[nomorbidak].Y = 21;  pemain[playerke].arahjalan bidak[nomorbidak] =1; //tengah kanan atas  }else if(pemain[playerke].posisibidak [nomorbidak].X == 112 && pemain[playerke].posisibidak [nomorbidak].Y == 21){  pemain[playerke].posisibidak [nomorbidak].X = 112;  pemain[playerke].posisibidak [nomorbidak].Y = 24;  pemain[playerke].arahjalan bidak[nomorbidak] = 4; //kanan atas  }else if(pemain[playerke].posisibidak [nomorbidak].X == 112 && pemain[playerke].posisibidak [nomorbidak].Y == 27){  pemain[playerke].posisibidak [nomorbidak].X = 105;  pemain[playerke].posisibidak [nomorbidak].Y = 27;  pemain[playerke].arahjalan bidak[nomorbidak] = 3; //kanan bawah  }else if(pemain[playerke].posisibidak [nomorbidak].X == 77 && pemain[playerke].posisibidak [nomorbidak].Y == 27){  pemain[playerke].posisibidak [nomorbidak].X = 70;  pemain[playerke].posisibidak[nomorbidak].Y = 30;  pemain[playerke].arahjalan bidak[nomorbidak] = 4; //tengah kanan bawah  }else if(pemain[playerke].posisibidak [nomorbidak].X == 70 && pemain[playerke].posisibidak [nomorbidak].Y == 45){  pemain[playerke].posisibidak [nomorbidak].X = 63;  pemain[playerke].posisibidak [nomorbidak].Y = 45;  pemain[playerke].arahjalanbidak [nomorbidak] = 3;//bawah kanan  }else if(pemain[playerke].posisibidak [nomorbidak].X == 56 && pemain[playerke].posisibidak [nomorbidak].Y == 45){  pemain[playerke].posisibidak [nomorbidak].X = 56;  pemain[playerke].posisibidak [nomorbidak].Y = 42;  pemain[playerke].arahjalan bidak[nomorbidak] = 2; //bawah kiri  }else if(pemain[playerke].posisibidak [nomorbidak].X == 56 && pemain[playerke].posisibidak [nomorbidak].Y == 30){  pemain[playerke].posisibidak [nomorbidak].X = 49;  pemain[playerke].posisibidak [nomorbidak].Y = 27;  pemain[playerke].arahjalan bidak[nomorbidak] = 3; //tengah kiri bawah  }else if(pemain[playerke].posisibidak [nomorbidak].X == 14 && pemain[playerke].posisibidak [nomorbidak].Y == 27){  pemain[playerke].posisibidak[nomorbidak].X = 14;  pemain[playerke].posisibidak[nomorbidak].Y = 24;  pemain[playerke].arahjalan bidak[nomorbidak] = 2; //kiri bawah  }else if(pemain[playerke].posisibidak [nomorbidak].X == 14 && pemain[playerke].posisibidak [nomorbidak].Y == 21){  pemain[playerke].posisibidak[nomorbidak].X = 21;  pemain[playerke].posisibidak[nomorbidak].Y = 21;  pemain[playerke].arahjalan bidak[nomorbidak] = 1; //kiri atas  }else if(pemain[playerke].arahjalanbidak[nomorbidak] == 1){  pemain[playerke].posisibidak[nomorbidak].X += 7;  }else if(pemain[playerke].arahjalanbidak[nomorbidak] == 2){  pemain[playerke].posisibidak [nomorbidak].Y -= 3;  }else if(pemain[playerke].arahjalanbidak[nomorbidak] == 3){  pemain[playerke].posisibidak[nomorbidak].X -= 7;  }else if(pemain[playerke].arahjalanbidak[nomorbidak] == 4){  pemain[playerke].posisibidak[nomorbidak].Y += 3;  } |
| 13. | Prosedur\_ PembunuhBidak() |  |
|  | Deskripsi | Modul dieksekusi pada saat bidak pemain yang mendapat giliran bermain menempati petak yang sudah ditempati bidak lawan. (bidak lawan akan dikembalikan ke home base-nya). |
|  | Initial State | Bidak pemain yang sedang mendapat giliran bermain menempati petak yang sudah terisi oleh bidak lawan |
|  | Final State | Bidak lawan kembali ke homebase nya |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | - - |
|  | Author | Ali |
|  | Algoritma  (Dalam Bahasa C) | void PembunuhBidak(int playerke, int nomorbidak){  int i, j;  for (i=0; i<4; i++){  for (j = 0; j<4; j++){  if ((i != playerke) && (pemain[i].posisibidak[j].X == pem ain[playerke].posisibidak[nomorbidak].X) && (pemain[i].posisibidak[j].Y == pemain[playerke].posisibidak[nomorbidak].Y)){  if (i==0 && j==0){  pemain[0].posisibidak[0].X = 28;  pemain[0].posisibidak[0].Y = 8;  pemain[0].bidakdihomebase[0] = true;  pemain[0].arahjalanbidak[0] = 1;  }  if (i==0 && j==1){  pemain[0].posisibidak[1].X = 36;  pemain[0].posisibidak[1].Y = 8;  pemain[0].bidakdihomebase[1] = true;  pemain[0].arahjalanbidak[1] = 1;  }  if (i==0 && j==2){  pemain[0].posisibidak[2].X = 28;  pemain[0].posisibidak[2].Y = 12;  pemain[0].bidakdihomebase[2] = true;  pemain[0].arahjalanbidak[2] = 1;  }  if (i==0 && j==3){  pemain[0].posisibidak[3].X = 36;  pemain[0].posisibidak[3].Y = 12;  pemain[0].bidakdihomebase[3] = true;  pemain[0].arahjalanbidak[3] = 1;  }  if (i==1 && j==0){  pemain[1].posisibidak[0].X = 92;  pemain[1].posisibidak[0].Y = 8;  pemain[1].bidakdihomebase[0] = true;  pemain[1].arahjalanbidak[0] = 4;  }  if (i==1 && j==1){  pemain[1].posisibidak[1].X = 100;  pemain[1].posisibidak[1].Y = 8;  pemain[1].bidakdihomebase[1] = true;  pemain[1].arahjalanbidak[1] = 4;  }  if (i==1 && j==2){  pemain[1].posisibidak[2].X = 92;  pemain[1].posisibidak[2].Y = 12;  pemain[1].bidakdihomebase[2] = true;  pemain[1].arahjalanbidak[2] = 4;  }  if (i==1 && j==3){  pemain[1].posisibidak[3].X = 100;  pemain[1].posisibidak[3].Y = 12;  pemain[1].bidakdihomebase[3] = true;  pemain[1].arahjalanbidak[3] = 4;  }  if (i==2 && j==0){  pemain[2].posisibidak[0].X = 92;  pemain[2].posisibidak[0].Y = 36;  pemain[2].bidakdihomebase[0] = true;  pemain[2].arahjalanbidak[0] = 3;  }  if (i==2 && j==1){  pemain[2].posisibidak[1].X = 100;  pemain[2].posisibidak[1].Y = 36;  pemain[2].bidakdihomebase[1] = true;  pemain[2].arahjalanbidak[1] = 3;  }  if (i==2 && j==2){  pemain[2].posisibidak[2].X = 92;  pemain[2].posisibidak[2].Y = 40;  pemain[2].bidakdihomebase[2] = true;  pemain[2].arahjalanbidak[2] = 3;  }  if (i==2 && j==3){  pemain[2].posisibidak[3].X = 100;  pemain[2].posisibidak[3].Y = 40;  pemain[2].bidakdihomebase[3] = true;  pemain[2].arahjalanbidak[3] = 3;  }  if (i==3 && j==0){  pemain[3].posisibidak[0].X = 28;  pemain[3].posisibidak[0].Y = 36;  pemain[3].bidakdihomebase[0] = true;  pemain[3].arahjalanbidak[0] = 2;  }  if (i==3 && j==1){  pemain[3].posisibidak[1].X = 36;  pemain[3].posisibidak[1].Y = 36;  pemain[3].bidakdihomebase[1] = true;  pemain[3].arahjalanbidak[1] = 2;  }  if (i==3 && j==2){  pemain[3].posisibidak[2].X = 28;  pemain[3].posisibidak[2].Y = 40;  pemain[3].bidakdihomebase[2] = true;  pemain[3].arahjalanbidak[1] = 2;  }  if (i==3 && j==3){  pemain[3].posisibidak[2].X = 36;  pemain[3].posisibidak[2].Y = 40;  pemain[3].bidakdihomebase[2] = true;  pemain[3].arahjalanbidak[1] = 2;  }  }  }  }  } |
| 14. | Prosedur\_ PenentuPemenang() |  |
|  | Deskripsi | Ketika ada pemain yang semua bidaknya sudah menempati petak finish maka permainan selesai. |
|  | Initial State | Belum ada pemain yang menjadi pemenang |
|  | Final State | Tampilan “Selamat kamu menang“ atau “Game Over” |
|  | Modul Pemanggil | Function\_MemasukiLintasanFinish() |
|  | Modul yang dipanggil | - |
|  | Author | Ali |
|  | Algoritma  (Dalam Bahasa C) | void PenentuPemenang(bool \*menang, int \*pemenang, int playerke){  int i;  for (i=0; i<4; i++){  if(pemain[playerke].bidakdi finish [i] == false){  \*menang = false;  \*pemenang = NULL;  return;  }else {  \*menang = true;  \*pemenang = playerke;  }  }  } |
| 15. | Prosedur\_Instruction() |  |
|  | Deskripsi | Modul untuk menampilkan aturan bermain |
|  | Initial State | Layar kosong |
|  | Final State | Tampil nya instruksi/aturan bermain |
|  | Modul Pemanggil | Prosedur\_MainMenu() |
|  | Modul yang dipanggil | **-** |
|  | Author | Farra |
|  | Algoritma  (Dalam bahasa C) | void Instruction(){  system("cls");  FILE \*instruksi;  char instruksi01[2395];  instruksi=fopen("instruksi.txt","r");  if(!instruksi){  printf("File Instruksi tidak  ditemukan!.");  }  GoToXY(78,1);  printf("INSTRUKSI\n--------------------  ----------------------------------\n");  While (fgets(instruksi01, 2395, instruksi) !=NULL){  printf("%s", instruksi01);  }  fclose(instruksi);  printf("------------------------------------------------------------------------------------------------------------------------------------------------------------------------\n");  PenentuWarna(4);  printf("Tekan huruf S untuk memulai permainan");  PenentuWarna(7);  getch();  system("cls");  MainMenu();  } |
| 16. | Prosedur\_TampilPapanPermainan() |  |
|  | Deskripsi | Modul untuk menampilkan papan permainan |
|  | Initial State | Layar Kosong |
|  | Final State | Tampilnya papan permainan |
|  | Modul Pemanggil | Function\_StartGame() |
|  | Modul yang dipanggil | Prosedur\_PenentuWarna() |
|  | Author | Farra & Ali |
|  | Algoritma  (Dalam bahasa C) | void TampilPapanPermainan(){  int i;  system("cls");  for (i=0; i<=8; i+=8){  PenentuWarna(2);  GoToXY(26+i, 6); printf(" \_\_\_\_\_ ");  GoToXY(26+i, 7); printf("| |");  GoToXY(26+i, 8); printf("| |");  GoToXY(26+i, 9); printf("|\_\_\_\_\_|");  GoToXY(26+i, 10); printf(" \_\_\_\_\_ ");  GoToXY(26+i, 11); printf("| |");  GoToXY(26+i, 12); printf("| |");  GoToXY(26+i, 13); printf("|\_\_\_\_\_|");  PenentuWarna(7);  }  GoToXY(13, 46);  for(i=0; i<105; i++){  printf("\_");  }  for (i=14; i<52; i=i+8){  int y = 19, a = 62;  if (i!=12){  i=i-1;  }  //timur  GoToXY(i, y); printf("\_\_\_\_\_\_\n");  GoToXY(i, y+1); printf(" |\n");  GoToXY(i, y+2); printf(" |\n");  GoToXY(i, y+3); printf("\_\_\_\_\_\_|\n");  if(i>=14 && i<=21){  PenentuWarna(2);  GoToXY(i-1, y); printf(" \_\_\_\_\_\_\n");  GoToXY(i-1, y+1); printf("| |\n");  GoToXY(i-1, y+2); printf("| |\n");  GoToXY(i-1, y+3); printf("|\_\_\_\_\_\_|\n");  GoToXY(i-1, y+4);printf("| |\n");  GoToXY(i-1, y+5); printf("| |\n");  GoToXY(i-1, y+6); printf("|\_\_\_\_\_\_|\n");  }  if(i>=14){  PenentuWarna(2);  }  GoToXY(i, y+3); printf("\_\_\_\_\_\_\n");  GoToXY(i, y+4); printf(" |\n");  GoToXY(i, y+5); printf(" |\n");  GoToXY(i, y+6); printf("\_\_\_\_\_\_|\n");  PenentuWarna(7);  GoToXY(i, y+7); printf(" |\n");  GoToXY(i, y+8); printf(" |\n");  GoToXY(i, y+9); printf("\_\_\_\_\_\_|\n");  //barat  GoToXY(i+a, y); printf(" \_\_\_\_\_\_\n");  GoToXY(i+a, y+1); printf("| \n");  GoToXY(i+a, y+2); printf("| \n");  GoToXY(i+a, y+3); printf("|\_\_\_\_\_\_\n");  if(i>=13 && i<44){  PenentuWarna(4);  }  GoToXY(i+a, y+4); printf("| \n");  GoToXY(i+a, y+5); printf("| \n");  GoToXY(i+a, y+6); printf("|\_\_\_\_\_\_\n");  int barat;  PenentuWarna(7);  GoToXY(i+a, y+7); printf("| \n");  GoToXY(i+a, y+8); printf("| \n");  GoToXY(i+a, y+9); printf("|\_\_\_\_\_\_\n");  PenentuWarna(4);  if(i>=35 &&i<54){  for(barat=0; barat<=4; barat+=4){  GoToXY(41+a, y+4); printf("| |\n");  GoToXY(41+a, y+5); printf("| |\n");  GoToXY(41+a, y+6); printf("|\_\_\_\_\_\_|\n");  GoToXY(41+a, y+7); printf("| |\n");  GoToXY(41+a, y+8); printf("| |\n");  GoToXY(41+a, y+9); printf("|\_\_\_\_\_\_|\n");  }  }  if(i<+42){  GoToXY(i+a+1, y+3); printf("\_\_\_\_\_\_\n");  }  PenentuWarna(7);  }  for (i=3; i<20; i=i+3){  int x = 54;  //utara  GoToXY(x, i-1); printf("| |\n");  GoToXY(x, i); printf("| |\n");  GoToXY(x, i+1);printf("|\_\_\_\_\_\_|\n");  GoToXY(x+14, i-1); printf("| |\n");  GoToXY(x+14, i); printf("| |\n");  GoToXY(x+14, i+1); printf("|\_\_\_\_\_\_|\n");  int utara;  if (i==6){  PenentuWarna(3);  GoToXY(x+15, i-2); printf("\_\_\_\_\_\_\n");  GoToXY(x+8, i-2); printf("\_\_\_\_\_\_\n");  GoToXY(x+14, i-1);printf("| |\n");  GoToXY(x+14, i); printf("| |\n");  GoToXY(x+14, i+1);printf("|\_\_\_\_\_\_|\n");  } PenentuWarna(7);  if(i>=6 && i<20){  PenentuWarna(3);  }  GoToXY(x+7, i-1); printf("| |\n");  GoToXY(x+7, i); printf("| |\n");  GoToXY(x+7, i+1); printf("|\_\_\_\_\_\_|\n");  PenentuWarna(7);  //selatan  GoToXY(x, i+26); printf("| |\n");  GoToXY(x, i+27); printf("| |\n");  GoToXY(x, i+28); printf("|\_\_\_\_\_\_|\n");  GoToXY(x+14, i+26); printf("| |\n");  GoToXY(x+14, i+27); printf("| |\n");  GoToXY(x+14, i+28); printf("|\_\_\_\_\_\_|\n");  if (i>=0 && i<13){  PenentuWarna(5);  }  GoToXY(x+7, i+26); printf("| |\n");  GoToXY(x+7, i+27); printf("| |\n");  GoToXY(x+7, i+28); printf("|\_\_\_\_\_\_|\n");  PenentuWarna(7);  }  GoToXY(13, 1);  for(i=1; i<=105; i++){  printf("\_");  }  for (i=2; i<=46; i++){  GoToXY(12, i);  printf("|");  }  for (i=2; i<=46; i++){  GoToXY(117, i);  printf("|");  }  i= 15;  if(i>13){  int x=54;  PenentuWarna(5);  GoToXY(x+1, i+25); printf("\_\_\_\_\_\_\n");  GoToXY(x, i+26); printf("| |\n");  GoToXY(x, i+27); printf("| |\n");  GoToXY(x, i+28); printf("|\_\_\_\_\_\_|\n");  GoToXY(x+8, i+25); printf("\_\_\_\_\_\_\n");  GoToXY(x+7, i+26); printf("| |\n");  GoToXY(x+7, i+27); printf("| |\n");  GoToXY(x+7, i+28); printf("|\_\_\_\_\_\_|\n");  PenentuWarna(7);  }  for (i=0; i<=8; i+=8){  int j=34;  PenentuWarna(5);  GoToXY(26+i, j); printf(" \_\_\_\_\_ ");  GoToXY(26+i, j+1);printf("| |");  GoToXY(26+i, j+2);printf("| |");  GoToXY(26+i, j+3);printf("|\_\_\_\_\_|");  GoToXY(26+i, j+4);printf(" \_\_\_\_\_ ");  GoToXY(26+i, j+5);printf("| |");  GoToXY(26+i, j+6);printf("| |");  GoToXY(26+i, j+7);printf("|\_\_\_\_\_|");  PenentuWarna(7);  }  for (i=0; i<=8; i+=8){  PenentuWarna(3);  int j=90;  GoToXY(j+i, 6); printf(" \_\_\_\_\_ ");  GoToXY(j+i, 7); printf("| |");  GoToXY(j+i, 8); printf("| |");  GoToXY(j+i, 9); printf("|\_\_\_\_\_|");  GoToXY(j+i, 10);printf(" \_\_\_\_\_ ");  GoToXY(j+i, 11);printf("| |");  GoToXY(j+i, 12); printf("| |");  GoToXY(j+i, 13); printf("|\_\_\_\_\_|");  PenentuWarna(7);  }  for (i=0; i<=8; i+=8){  PenentuWarna(4);  int j=90, n=34;  GoToXY(j+i, n); printf(" \_\_\_\_\_ ");  GoToXY(j+i, n+1); printf("| |");  GoToXY(j+i, n+2); printf("| |");  GoToXY(j+i, n+3); printf("|\_\_\_\_\_|");  GoToXY(j+i, n+4); printf(" \_\_\_\_\_ ");  GoToXY(j+i, n+5); printf("| |");  GoToXY(j+i, n+6); printf("| |");  GoToXY(j+i, n+7); printf("|\_\_\_\_\_|");  PenentuWarna(7);  }  GoToXY(55, 28); printf("\_\_\_\_\_\_ ");  PenentuWarna(5); GoToXY(62, 28); printf("\_\_\_\_\_\_ "); PenentuWarna(7);  GoToXY(69, 28); printf("\_\_\_\_\_\_ ");  GoToXY(27, 4); PenentuWarna(2); printf("User");  GoToXY(91, 4); PenentuWarna(3); printf("Komputer 1");  GoToXY(91, 32); PenentuWarna(4); printf("Komputer 2");  GoToXY(27, 32); PenentuWarna(5); printf("Komputer 3");  PenentuWarna(7);  GoToXY(75, 28); printf("|");  GoToXY(55, 20); printf(" : : ");  GoToXY(55, 21); printf(" : : ");  GoToXY(55, 22); printf("......:......:......");  GoToXY(55, 23); printf(" : : ");  GoToXY(55, 24); printf(" : : ");  GoToXY(55, 25); printf("......:......:......");  GoToXY(55, 26); printf(" : : ");  GoToXY(55, 27); printf(" : : ");  GoToXY(61, 28); printf(":");GoToXY(68, 28); printf(":");  } |
| 17. | Prosedur\_PrintBidak() |  |
|  | Deskripsi | Modul untuk menampilkan bidak pada papan permainan |
|  | Initial State | Bidak belum ditampilkan pada papan permainan |
|  | Final State | Bidak sudah ditampilkan pada papan permainan |
|  | Modul Pemanggil | Prosedur\_StartGame() |
|  | Modul yang dipanggil | * Function\_JumlahPemain() * Function\_PenentuWarna() |
|  | Author | Ali |
|  | Algoritma  (Dalam bahasa C) | void printbidak(int jumlahpemain){  int j, i;  for(j=0; j<jumlahpemain+1; j++){  for(i=0; i<4; i++){  PenentuWarna(pemain[j].warna);  GoToXY(pemain[j].posisibidak[i].X,  pemain[j].posisibidak[i].Y);  printf("%d", i+1);  PenentuWarna(7);  }  }  } |
| 18. | Prosedur\_GameOver() |  |
|  | Deskripsi | Modul untuk menampilkan tampilan “Selamat Kamu Menang” di layar user yang berarti bahwa gameover untuk pemain komputer. |
|  | Initial State | Layar Kosong |
|  | Final State |  |
|  | Modul Pemanggil | Function\_StartGame |
|  | Modul yang dipanggil | - |
|  | Author | Farra |
|  | Algoritma  (Dalam Bahasa C) | void GameOver(int pemenang){  system("cls");  if (pemenang == 0){  sleep(1);  GoToXY(27, 4); printf(" \_\_\_ \_\_\_ \_ \_ \_\_ \_\_ \_ \_\_\_\_\_ \_ \_\_ \_ \_\_ \_\_ \_ \_ \_\_ \_\_ \_\_\_ \_ \_ \_ \_ \_ \_\_\_ \_ \_ \_ \n");  GoToXY(27, 5); printf(" / \_\_| \_\_| | /\_\\ | \\/ | /\_\\\_ \_| | |/ / /\_\\ | \\/ | | | | | \\/ | \_\_| \\| | /\_\\ | \\| |/ \_\_| | | |\n");  GoToXY(27, 6); printf(" \\\_\_ \\ \_|| |\_\_ / \_ \\| |\\/| |/ \_ \\| | | ' < / \_ \\| |\\/| | |\_| | | |\\/| | \_|| .` |/ \_ \\| .` | (\_ |\_|\_|\_|\n");  GoToXY(27, 7); printf(" |\_\_\_/\_\_\_|\_\_\_\_/\_/ \\\_\\\_| |\_/\_/ \\\_\\\_| |\_|\\\_\\/\_/ \\\_\\\_| |\_|\\\_\_\_/ |\_| |\_|\_\_\_|\_|\\\_/\_/ \\\_\\\_|\\\_|\\\_\_\_(\_|\_|\_)\n");  }else{  GoToXY(27, 8); printf(" \_ \_\_ \_ \_\_ \_\_ \_ \_ \_ \_\_ \_ \_ \_ \_ \_");  GoToXY(27, 9); printf("| |/ / /\_\\ | \\/ | | | | | |/ / /\_\\ | | /\_\\ | || |");  GoToXY(27, 10); printf("| ' < / \_ \\| |\\/| | |\_| | | ' < / \_ \\| |\_\_ / \_ \\| \_\_ |");  GoToXY(27, 11); printf("|\_|\\\_\\/\_/ \\\_\\\_| |\_|\\\_\_\_/ |\_|\\\_\\/\_/ \\\_\\\_\_\_\_/\_/ \\\_\\\_||\_|");  }  getch();  MainMenu();  } |

**Tabel 2.Spesifikasi Modul**

**Structure Chart**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| **2.3 Interface/tampilan**  Pada bagian ini diuraikan rancangan tampilan dalam game EL-Yu-Di-Oh.  2.3.1 Tampilan Opening    Gambar 1.Tampilan Opening pada Layar User  2.3.2 Tampilan Main Menu    Gambar 2.1 Tampilan Main Menu  2.3.3 Tampilan Menu    Gambar 2.2 Tampilan jumlah pemain pada menu start game    Gambar 2.2.1 Tampilan urutan Gambar 2.2.2 Tampilan urutan Gambar 2.2.3 Tampilan urutan  pemain (1 user vs 1 komputer) pemain (1 user vs 2 komputer) pemain (1 user vs 3 komputer)        Gambar 2.3 Tampilan menu Instruksi  2.3.4 Papan Permainan  Gambar 2.4 Tampilan Papan Permainan  2.3.5 Tampilan Kocok Dadu  Gambar 2.5 Tampilan Kocok Dadu(untuk angka dadu 3)  2.3.6 Tampilan Giliran Bermain    Gambar 2.6 Tampilan Giliran Bermain(Untuk pemain dari Komputer)  2.3.7 Tampilan Menang    Gambar 2.7 Tampilan ketika user sebagai pemenang | | | | | | | | |
|  |  |  |  |  |  |  |  |  |
|  | | | | |  | | |  |
|  |  | | |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | |  | | | | |  |  |
|  |  |  |  |  |  |  |  |  |
|  | | |  | | |  |  |  |