**1.5 PENJELASAN CODE**

1. Pendeklarasian Variabel

|  |
| --- |
| const int MAX\_PASIEN = 5;  const int MAX\_HP = 100;  struct Entity {  bool pasien;  int day;  bool virus;  int\* hp;  int num\_pasien;  bool obat\_used;  string names[MAX\_PASIEN];  };  Entity earth; |

*Script* diatas merupakan pendeklarasian beberapa variabel dan juga struktur yang penting dalam penggunaan program yang akan dipanggil di *line* *code* selanjutnya seperti struktur “Entity” memiliki fungsi yang spesifik dalam permainan. Variabel global “const int MAX\_PASIEN” dan “const int MAX\_HP” menentukan jumlah maksimal pasien dan HP maksimum masing-masing, sementara “Entity earth” adalah struktur yang menyimpan semua data dan status terkait bumi dan pasien.

Variabel dalam Entity seperti “bool pasien”, “int day”, “bool virus”, “int\* hp”, ”int num\_pasien”, “bool obat\_used”, dan “string names[MAX\_PASIEN]” digunakan untuk melacak status umum pasien, jumlah hari, keberadaan virus, kesehatan setiap pasien, jumlah pasien yang hidup, penggunaan obat, dan nama-nama pasien.

1. Main Menu

|  |
| --- |
| int main() {  int choose; earth.day = 0;  earth.pasien = true; earth.virus = false;  day\_one();  system("cls");  display();  cin.get();  system("cls");  narasi();  cin.get();  system("cls");  while (earth.day <= 10) {  cout << "===============" << endl;  cout << " Hari ke-" << earth.day << endl;  cout << "===============" << endl;  cout << "1. Lihat kondisi pasien" << endl;  cout << "2. Hari selanjutnya" << endl;  cout << "3. Berikan obat" << endl;  cout << "Masukkan Pilihan: ";  cin >> choose;  system("cls");  switch (choose) {  case 1:  tampilkanKondisiBumi();  pause();  system("cls");  break;  case 2:  day();  break;  case 3:  berikanObat();  pause();  system("cls");  break;  default:  cout << "Pilihan tidak valid. Silakan pilih lagi." << endl;  break;  }  if (earth.day >= 10) {  if (checkWinCondition()) {  cout << "Selamat! anda menang karena telah menyelamatkan minimal 2 pasien" << endl;  }else{  cout << "Semua Pasien mati anda kalah!" << endl;  }  break;  }  if (checklosecondition()){  cout<<"Anda Kalah! Karena anda tidak bisa menyelamatkan minimal 2 pasien"<<endl;  pause();  exit(0);  }  if (earth.num\_pasien == 0) {  cout << "Semua pasien telah mati. Permainan selesai." << endl;  break;  }  }  deallocateMemory(); return 0;} |

*Script* di atas merupakan sebuah pengimplementasian dari menu utama *game* yang dibuat. Pada awalnya terdapat pendeklarasian berupa ”int choose”, ”earth.day = 0”, ” earth.pasien = true” dan ” earth.virus = false”. Lalu terdapat beberapa *input* *function* yang dimasukkan ke dalam *code* yang akan digunakan sebagai jalannya program.

Dalam fungsi “main()”, variabel “choose” digunakan untuk menyimpan pilihan input dari pemain dalam menu utama permainan. Program dimulai dengan inisialisasi hari pertama menggunakan “day\_one()”, diikuti oleh tampilan layar pembuka dan narasi.

Loop utama permainan berjalan hingga hari ke-10, menampilkan menu pilihan kepada pemain, dan mengolah input pemain untuk melihat kondisi pasien, melanjutkan ke hari berikutnya, atau memberikan obat. Program juga mengecek kondisi kemenangan setelah 10 hari menggunakan kondisi *if* “earth.day >= 10” dan kondisi kekalahan setiap hari, serta menghentikan permainan jika semua pasien mati. Terakhir, memori yang dialokasikan dibebaskan menggunakan “deallocateMemory()”

1. Game

|  |
| --- |
| void display() {  cout << " \_ \_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_ \n";  cout << "| | / /| \_\_\_ \\|\_ \_|/ \_\_\_||\_ \_|/ \_\_\_| \n";  cout << "| |/ / | |\_/ / | | \\ `--. | | \\ `--. \n";  cout << "| \\ | / | | `--. \\ | | `--. \\ \n";  cout << "| |\\ \\| |\\ \\ \_| |\_ /\\\_\_/ / \_| |\_ /\\\_\_/ / \n";  cout << "\\\_| \\\_/\\\_| \\\_| \\\_\_\_/ \\\_\_\_\_/ \\\_\_\_/ \\\_\_\_\_/ \n";  cout << "\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_\_\_ \_\_\_ \_\_\_\_\_\n";  cout << "| \_ \\|\_ \_| | \_\_\_ \\| | | || \\/ ||\_ \_|\n";  cout << "| | | | | | | |\_/ /| | | || . . | | |\n";  cout << "| | | | | | | \_\_\_ \\| | | || |\\/| | | |\n";  cout << "| |/ / \_| |\_ | |\_/ /| |\_| || | | | \_| |\_\n";  cout << "|\_\_\_/ \\\_\_\_/ \\\_\_\_\_/ \\\_\_\_/ \\\_| |\_/ \\\_\_\_/\n";  cout << endl;  cout << "Press any key to continue........................." << endl;  } |

*Script* di atas merupakan tampilan untuk awal program dimulai sebagai bentuk judul yang akan diangkat berdasarkan program yang telah dibuat dimana dalam ”void display()” ini terdapat suatu *output* yang berbentuk sebuah seni kata yang disusun menggunakan ASCII untuk menampilkan sebuah judul yang tidak membosankan dengan *style font* judul dari sebuah video game DOOM.

|  |
| --- |
| void narasi() {  cout << "=====================================\n";  cout << "- - - - - - - -\_\_\_\_\_- - - - - - - \n"  "- - - - - ,-:` \\;',`'-, - - - - - - \n"  "- - - - -.'-;\_,; ':-;\_,'.- - - - - \n"  "- - - - /; '/ , \_`.-\\- - - - - \n"  "- - - -| '`. (` /` ` \\`|- - - - - \n"  "- - - -|:. `\\`-. \\\_ / |- - - - - \n"  "- - - -| ( `, .`\\ ;'|- - - - - \n"  "- - - - \\ | .' `-'/- - - - - \n"  "- - - - -`. ;/ .'- - - - - \n"  "- - - - - -`'-.\_\_\_\_\_.'- - - - - - -\n";  cout << "=====================================\n";  cout << "Bumi yang dulu damai dan harmonis kini terguncang oleh serangan\n"  "virus mematikan. Dalam kepanikan global ini, Anda, seorang\n"  "perawat berbakat, diberi tugas krusial: merawat lima pasien\n"  "penting yang memegang kunci untuk menyelamatkan dunia.\n"  "\n"  "1. Dr. Majdi - ilmuwan yang sedang mengembangkan vaksin.\n"  " Tanpa perawatannya, riset vaksin bisa terhenti.\n"  "2. Jenderal Tegar - pemimpin militer yang mengoordinasikan\n"  " upaya pengendalian virus. Kehadirannya penting untuk\n"  " menjaga ketertiban dan keamanan.\n"  "3. Presiden Aqsha - pemimpin negara yang memberikan arahan\n"  " dan harapan kepada rakyat. Stabilitas pemerintahan\n"  " bergantung padanya.\n"  "4. Mentri Faza - seorang mentri yang berpegang teguh arahan\n"  " Presiden jika jabatan tersebut diserahkan ke dirinya.\n"  "5. Prof. Ayu - penemu yang membantu Dr. Majdi dalam mencari\n"  " obat vaksin yang berguna sebagai penyelamat bumi ini.\n"  "\n"  "Tugas Anda adalah memastikan 5 pasien tersebut tetap sehat\n"  "dan mampu melanjutkan misi mereka.";} |

*Script* diatas merupakan *function* yang berguna untuk memberi latar belakang terhadap narasi program yang akan disampaikan guna mendapat pandangan yang dapat mempermudah *user* dalam mengerti alur cerita yangn akan disampaikan oleh program.

|  |
| --- |
| void pause() {  cin.get();  cin.fail();  cin.ignore();  } |

Bagian ini merupakan bentuk *function* yang akan mengantisipasi jika terdapat kesalahan dalam peng-*input*an yang dilakukan oleh *user* yang mana jika terdapat kesalahan maka program tidak mengeluarkan *output* *error* melainkan akan terus mengulang hingga *user* memasukkan *input* yang tepat.

|  |
| --- |
| void allocateMemory() {  earth.hp = new int[MAX\_PASIEN];  }  void deallocateMemory() {  delete[] earth.hp;} |

Bagian ini digunakan sebagai pengalokasian terhadap memori *integer* dari sebuah array ”hp” yang terdapat pada struktur ”Entity”. *Pointer* dari ”hp” pertama kali akan di *set* sebagai bentuk NULL. *Function* dari ”allocateMemory” berguna untuk mengalokasi ukuran memori dari “MAX\_PASIEN” dan akan mengantarkannya terhadap variabel ”hp” hal ini berguna untuk mengatasi jumlah pasien dan poin darah yang mereka milikk secara efisien. Lalu untuk *function* “deallocateMemory” digunakan untuk menghapus memori yang telah di inisialisasikan menggunakan operator “delete[]” yang akan dipanggil di akhir *game*.

|  |
| --- |
| void randomizeHP() {  int base\_hp[MAX\_PASIEN] = {70, 50, 30, 30, 35};  for (int i = 0; i < MAX\_PASIEN; i++) {  int rand\_index = (i + 1) % MAX\_PASIEN;  swap(base\_hp[i], base\_hp[rand\_index]);  }  for (int i = 0; i < MAX\_PASIEN; i++) {  earth.hp[i] = base\_hp[i];  }  } |

*Script* di atas merupakan sebuah *function* yang akan memberi suatu nilai acak terhadap *array* dari HP yang ditetapkan dengan syarat sebagai “70, 50, 30, 30, 35”. Terdapat sebuah perulangan untuk *array* dari “base\_hp” dengan cara menggunakan rumus “(i + 1) % MAX\_PASIEN” ini digunakan agar indeks tidak dipilih dua kali sebagai nilai HP yang akan diterima oleh pasien. Lalu untuk fungsi dari “swap” yaitu untuk menukar elemen “base\_hp” saat ini dengan elemen acak dari indeks yang akan memungkinkan agar nilai yang diterima pasien akan acak dari syarat yang telah di set di dalam *array*. Terdapat perulangan lagi yang mana nilai dari “base\_hp” diberikan ke variabel “earth.hp”.

|  |
| --- |
| void tampilkanPasien() {  if (earth.num\_pasien == 0) {  cout << "Semua pasien telah mati." << endl;  return;  }  cout<<"========================"<<endl;  cout << "Status pasien hari ke-" << earth.day << ":" << endl;  cout<<"========================"<<endl;  for (int i = 0; i < earth.num\_pasien; i++) {  for (int i = 0; i < earth.num\_pasien - 1; i++) {  for (int j = 0; j < earth.num\_pasien - i - 1; j++) {  if (earth.hp[j] > earth.hp[j + 1]) {  swap(earth.hp[j], earth.hp[j + 1]);  swap(earth.names[j], earth.names[j + 1]);  }  }  }  cout << "Pasien " << earth.names[i] << ": HP = " << earth.hp[i] << endl;  }  } |

Script “void tampilkanPasien()” merupakan sebuah perintah untuk menampilkan kondisi HP dari setiap pasien yang masih hidup pada hari tertentu dalam simulasi. Jika semua pasien telah mati, fungsi akan menampilkan pesan yang sesuai dan keluar. Jika ada pasien yang masih hidup, fungsi akan mencetak status HP dari setiap pasien tersebut. Terdapat dua iterasi perulangan yang digunakan sebagai proses mengurutkan nama pasien yang memiliki darah yang lebih sedikit dibandingkan pasien lainnya.

|  |
| --- |
| void tampilkanKondisiBumi() {  if (earth.num\_pasien == 0) {  cout << "Semua pasien telah mati." << endl;  return;  }  tampilkanPasien();  if (earth.day == 1) {  cout<<"========================"<<endl;  cout<< " Bumi Baik Baik Saja " << endl;  cout<<"========================"<<endl;  } else if (earth.day == 2 || earth.day == 3) {  cout<<"========================"<<endl;  cout << " Bumi Terkena Virus " << endl;  cout<<"========================"<<endl;  } else if (earth.day >= 4) {  cout<<"========================"<<endl;  cout << "Cepat Sembuhkan Bumi" << endl;  cout<<"========================"<<endl;  }  } |

Dalam *function* “tampilkanKondisiBumi()” digunakan sebagai pemberi tanda bahwa keadaan bumi yang telah terkena oleh virus. “earth.num\_pasien == 0” yang berarti jika pasien tidak ada maka terdapat indikasi pesan bahwa semua pasien mati, lalu untuk indikasi “earth.day” yaitu jika terdapat syarat lebih dari beberapa hari maka terdapat *output* yang memberi tahu tentang kondisi bumi saat ini yang akan memberi gambaran untuk *user* agar segera membantu pasien yang terkena virus agar dapat sembuh dan melawan virus yang mneyerang.

|  |
| --- |
| void kondisi() {  if (earth.day == 2) {  earth.virus = true;  randomizeHP();  } else if (earth.day >= 4) {  earth.pasien = false;  }  } |

*Script* ini digunakan sebagai cara untuk memberi aktivasi terhadap virus yang akan dimulai pada hari kedua dengan menggunakan penggunaan *if* “earth.day == 2” maka variabel “earth.virus” di-*set* menjadi *true* lalu untuk “earth.day >= 4” untuk mengindikasikan jika tidak ada pasien yang hidup untuk menentukan menang atau kalahnya dari game ini.

|  |
| --- |
| void berikanObat() {  if (earth.obat\_used) {  cout << "Obat sudah digunakan hari ini. Tunggu hari berikutnya." << endl;  return;  }  if (earth.num\_pasien == 0) {  cout << "Tidak ada pasien yang bisa diberi obat." << endl;  return;  }  string pasien\_name;  cout<<"===========================\n";  cout << "Pilih pasien untuk diberi obat\n";  cout<<"===========================\n";  cout<<"- Dokter\n";  cout<<"- Jendral\n";  cout<<"- Presiden\n";  cout<<"- Mentri\n";  cout<<"- Professor\n";  cout<<"Masukkan nama pasien yang akan diobati : ";  cin >> pasien\_name;  bool valid\_choice = false;  for (int i = 0; i < earth.num\_pasien; i++) {  if (earth.names[i] == pasien\_name) {  int& hp = earth.hp[i];  if (hp < MAX\_HP) {  hp = min(hp + 35, MAX\_HP);  cout << "Pasien " << pasien\_name << " telah diberikan obat. HP sekarang = " << hp << endl;  } else {  cout << "Pasien " << pasien\_name << " sudah memiliki HP maksimum." << endl;  }  earth.obat\_used = true;  valid\_choice = true;  break;  }  }  if (!valid\_choice) {  cout << "Pasien tidak valid." << endl;  }  } |

*Script* “void berikanObat()” merupakan sebuah perintah untuk memberikan obat kepada salah satu pasien yang masih hidup, meningkat HP pasien sebanyak 35, dan memastikan bahwa obat hanya bisa diberikan sekali perhari. “if (earth.obat\_used)” fungsi untuk mengecek apakah obat sudah digunakan hari ini. “if (earth.num\_pasien == 0)” fungsi untuk mengecek apakah tidak ada pasien yang hidup “earth.num\_pasien == 0” jika semua pasien telah mati, maka fungsi akan menampilkan pesan dan keluar dari fungsi dengan “return;”.

Untuk *code* “string pasien\_name” digunakan sebagai pendeklarasian untuk memanggil nama pasien seperti Dokter, Jendral, Presiden dan lain lain yang akan dimasukkan saat memberikan obat yang dipakai dalam *code* “cin >> pasien\_name”. “valid\_choice” di-*set* menjadi “false” agar memeriksa jika nama pasien yang dimasukkan benar lalu hal itu berguna sebagai iterasi *array* “earth.names” lalu jika hal itu sesuai dengan elemen nama pasien yang berada dalam *array* maka hal itu merespon nilai HP yang terdapat di dalam *array* lalu menambahkan poin HP tersebut sesuai dengan nama pasien yang telah dipilih namun sebagai syarat yang mana pasien tidak dapat menerima obat jika HP pasien telah dinilai maksimum. Lalu “earth.obat\_use” di-*set* menjadi “true” yang berarti penggunaan obat telah dipakai untuk hari ini dan “valid\_choice” juga di-*set* menjadi “true” yang berarti menandakan untuk keluar dari iterasi pengulangan dan jika masih di-*set* sebagai *false* maka nama pasien yang dimasukkan salah.

|  |
| --- |
| void hapusPasien(int index) {  for (int i = index; i < earth.num\_pasien - 1; i++) {  earth.hp[i] = earth.hp[i + 1];  earth.names[i] = earth.names[i + 1];  }  earth.num\_pasien--;  } |

*Script* ini digunakan untuk mengetahui pasien mana yang poin HP nya menyentuh angka 0 atau dapat dibilang pasien tersebut mati. *Script* ini menggunakan perulangan dengan cara mengambil parameter “index” yang merepresentasikan posisi dari “earth.names” dan “earth.hp” lalu mulai dengan perulangan “earth.num\_pasien - 1” yang berguna menggeser elemen ke kanan sebanyak 1 posisi hal ini digunakan untuk memberi nilai ulang pada indeks selanjutnya. Setelah menggeser nilai, fungsi dari “earth.num\_pasien--” sebagai tanda untuk mengurangi nilai dari jumlah pasien yang mengindikasikan bahwa pasien berkurang 1

|  |
| --- |
| void day()  {  earth.day++;  earth.obat\_used = false;  for (int i = 0; i < earth.num\_pasien; i++)  {  if (earth.hp[i] > 0)  {  earth.hp[i] -= 20;  if (earth.hp[i] <= 0) {  cout << "Pasien " << earth.names[i] << " telah mati." << endl;  hapusPasien(i);  i--;  }  }  }  kondisi();  } |

*Script* “void day()” merupakan sebuah fungsi untuk mensimulasikan berjalannya satu hari dalam program. “earth.day++;” *code* yang berfungsi untuk meningkatkan nilai sebesar satu yang menandakan satu hari telah berlalu. “earth.obat\_used = false;” *code*  yang berfungsi untuk mengatur nilai menjadi *false* yang menandakan bahwa obat belum digunakan pada hari yang baru dimulai. “for(int i = 0; i < earth.num\_pasien; i++)” *code* yang berfungsi untuk *looping* ke iterasi melalui semua pasien yang masih hidup. “if (earth.hp[i] > 0)” *code*  yang berfungsi untuk mengecek apakah HP pasien di indeks “i” lebih besar dari 0. Jika iya, maka akan melanjutkan ke langkah berikutnya. “earth.hp[i] -= 20;” *code* yang berfungsi untuk mengurangi nilai HP pasien sebesar 20 untuk menandakan bahwa pasien tersebut kehilangan HP hari ini. “if (earth.hp[i] <= 0)” *code* yang mengecek apakah setelah pengurangan HP, niali HP pasien menjadi kurang dari atau sama dengan 0. Jika iya, maka lanjutkan ke langkah berikutnya. “cout << "Pasien " << earth.names[i] << " telah mati." << endl;” *code* yang berfungsi untuk menampilkan pesan bahwa pasien yang bersangkutan telah mati. “hapusPasien(i);” *code* yang berfungsi untuk memanggil fungsi untuk menghapus pasien yang telah mati dari daftar pasien yang hidup. “i--;” *code* yang berfungsi untuk mengurangi nilai “i” sebesar satu untuk menyesuaikan indeks. “kondisi();” *code* yang berfungsi untuk memanggil fungsi untuk memperbarui kondisi bumi.

|  |
| --- |
| void day\_one() \  {  earth.day = 1;  earth.pasien = true;  earth.virus = false;  earth.obat\_used = false;  earth.num\_pasien = MAX\_PASIEN;  allocateMemory();  string default\_names[MAX\_PASIEN] = {"Dokter", "Jendral", "Presiden", "Mentri", "Professor"};  for (int i = 0; i < MAX\_PASIEN; i++)  {  earth.hp[i] = MAX\_HP;  earth.names[i] = default\_names[i];  }  } |

Bagian ini merupakan pendeklarasian dari semua struktur awal dari game yang dibuat yang mana bagian ini mengubah seluruh nilai variabel menjadi *default* agar proses game dapat berjalan sesuai yang diinginkan. “earth.day = 1” mendeklarasikan hari yang dimulai yaitu hari pertama. “earth.pasien = true” mengindikasikan bahwa ada pasien dalam simulasi ini, “earth.virus = false” dan “earth.obat\_used = false” diatur agar virus dan obat tidak dapat digunakan terlebih dahulu. “earth.num\_pasien = MAX\_PASIEN” yaitu men-*set* banyak pasien yang akan diuji yaitu sebanyak 5 pasien. “allocateMemory();” *code* ini digunakan untuk mengalokasikan jumlah HP kepada *array* “hp”. “string default\_names[MAX\_PASIEN] = {"Dokter", "Jendral", "Presiden", "Mentri", "Professor"}” yaitu nama pasien yang harus disembuhkan dan iterasi pengulangan di sini yaitu memeriksa pasien. “earth.hp[i] = MAX\_HP” yaitu mengatur darah dari masing masing pasien yaitu 100 “earth.names[i] = default\_names[i];” yaitu memberi nama *default* pada setiap pasien.

|  |
| --- |
| bool checkWinCondition() {  return earth.num\_pasien >= 2;  }  bool checklosecondition() {  return earth.num\_pasien < 2;  } |

Guna *script* diatas adalah untuk memeriksa bahwa *user* telah menamatkan simulasi game ini dengan syarat bahwa pasien yang diselamatkan lebih dari sama dengan 2 pasien yang mana apabila *user* hanya mampu menyelamatkan kurang dari 2 maka game dinyatakan gagal dan *user* kalah dalam menyelamatkan bumi