

IDENTITAS :

Nama : Muhammad Irfan

NIM : L0124063

Kelas : B

Judul Program : C-Responsi1KP_L0124063_MuhammadIrfan_01.c

Deskripsi Program :Game Simulasi menjadi Bupati menggunakan Bahasa C

Dokumentasi Program
<div>Analisis Kode</div> <div><pre>1 #include <stdio.h> 2 #include <stdlib.h> 3 #include <time.h> 4 #include <ctype.h></pre></div> <div><ul style="list-style-type: none">o #include <stdio.h> untuk menggunakan standard input output seperti, scanf, printf, dsb.o #include <stdlib.h> untuk menggunakan library yang tidak tersedia di standard input output Seperti NULL, dsb.o #include <time.h> untuk menggunakan library yang tidak tersedia di standard input output dan standard library, seperti rand(), dsb.o #include <ctype.h> untuk menggunakan library yang tidak tersedia di standard input output, standard library, dan time. Seperti toupper(), tolower(), dsb</div> <div><pre>6 typedef struct { 7 int anggaran; 8 int populasi; 9 int kebahagiaan; 10 int infrastruktur; 11 } Kota;</pre></div> <div><ul style="list-style-type: none">o Typdef struct {} Kota; untuk menggabungkan beberapa variabel kedalam satu variabel bernama Kota. Beberapa variabel tsb berupa int, karena akan memiliki nilai berupa bilangan bulat</div> <div><pre>13 typedef struct { 14 char jenisFasilitas[50]; 15 int biayaPembangunan; 16 int dampakMasyarakat; 17 int dampakInfrastruktur; 18 int dampakPopulasi; 19 } Fasilitas;</pre></div> <div><ul style="list-style-type: none">o Typedef struct {} Fasilitas; gunanya sama dengan variabel Kota, bedanya hanya isinya saja. Di Fasilitas terdapat variabel berupa char untuk menyimpan karakter, tapi disini saya menggunakan char yang berbentuk array agar bisa menerima nilai berupa kata(bukan hanya karakter).</div>

```

21 void clear(){
22     system("cls");
23 };

```

- Fungsi ini berguna untuk menghapus kata/huruf/apapun itu yang ada di terminal pada saat menjalankan program.

```

25 Kota kota = {500, 1000, 35, 20};
26 int persentasePajak = 0;
27 int aksiCounter = 0;

```

- Variabel “Kota kota”, berfungsi sebagai inisialisasi dari variabel berbentuk typedef struct {} Kota, kedalam variabel kota dan di beri nilai yang sesuai berdasarkan barisan/urutan yang ada pada variabel Kota. Misalkan pada urutan pertama yakni 500, maka pada variabel int anggaran akan terisi nilai sebesar 500.
- “int persentasePajak = 0” berfungsi untuk menginisialisasi variabel yang berbentuk integer yang diberi nama persentasePajak dan diberi nilai 0 untuk mencegah adanya logic error.
- “int aksiCounter = 0” juga berfungsi sama seperti variabel persentasePajak, aksiCounter digunakan sebagai tempat penyimpanan nilai setiap kali player melakukan sebuah aksi pada saat bermain game tsb. Bagian ini akan lebih masuk akal di penjelasan yang lebih lanjut.

```

29 void pengelolaanAnggaran() {
30     int pendapatanDariPajak = ((persentasePajak * kota.populasi)/100);
31     kota.anggaran += pendapatanDariPajak;
32
33     if (kota.anggaran < 0) {
34         kota.anggaran = 0;
35     }
36 }

```

- “void pengelolaanAnggaran(){}” merupakan function yang digunakan untuk menyimpan logika sederhana untuk mendapatkan uang/anggaran dari pajak. Di dalam function tersebut juga terdapat logika, jika anggaran kota kurang dari 0 atau bernilai negative, maka anggaran akan di kembalikan menjadi 0, agar menghindari anggaran bernilai negative.

```

38 int pengaturanPajak(){
39     int besarPajak;
40     char masukan[10];
41
42     while (1) {
43         clear();
44         printf("Berapa besar pajak yang kamu inginkan?\n");
45         printf("(Ingat semakin besar pajak yang kamu buat, semakin sedih wargamu.)(1-100):");
46         fgets(masukan, sizeof(masukan), stdin);
47
48         if (sscanf(masukan, "%d", &besarPajak) != 1 || besarPajak < 0 || besarPajak > 100) {
49             printf("Input tidak valid. Harap masukkan angka antara 0 dan 100.\n");
50             continue;
51         }
52
53         break;
54     }
55
56     return besarPajak;
57 };

```

- Function “int pengaturanPajak(){}” digunakan sebagai sarana bagi player untuk menginput/memasukkan berapa besar pajak yang player inginkan di kotanya (berkisar dari 1-100). Di dalam fungsi ini memuat beberapa variabel berupa int dan char yang digunakan untuk menyimpan input dari nilai yang akan player masukkan.
- “While(1)” adalah perulangan/loop yang bernilai 1 (true) yang digunakan untuk mengulang fungsi tersebut hingga player memasukkan nilai/input yang valid, setelah itu variabel break akan dijalankan dan program akan mengembalikan nilai besarPajak.
- Saya menggunakan fgets dan sscanf untuk memvalidasi input, yang akan membuat player tidak akan bisa melanjutkan game jika dia tidak memasukkan input yang sesuai dengan keterangan yang sudah diberikan yakni (1-100).

```

59 void pajak(){
60     while (1){
61         persentasePajak = pengaturanPajak();
62
63         if (persentasePajak == -1 || persentasePajak < 1 || persentasePajak > 100) {
64             printf("Nilai pajak tidak valid. Harus antara 1-100.\n");
65         }
66
67         int kebahagiaan = (persentasePajak * kota.populasi);
68         int infrastruktur = (persentasePajak * kota.populasi);
69
70         if (persentasePajak >= 80 && persentasePajak <= 100) {
71             kota.kebahagiaan -= kebahagiaan/2000;
72             kota.infrastruktur += infrastruktur/2000;
73             kurangiPopulasi(&kota, persentasePajak * kota.populasi / 1000, 10);
74         } else if (persentasePajak >= 60 && persentasePajak < 80) {
75             kota.kebahagiaan -= kebahagiaan/3000;
76             kota.infrastruktur += infrastruktur/3000;
77             kurangiPopulasi(&kota, persentasePajak * kota.populasi / 1200, 7);
78         } else if (persentasePajak >= 40 && persentasePajak < 60) {
79             kota.kebahagiaan -= kebahagiaan/4000;
80             kota.infrastruktur += infrastruktur/4000;
81             kurangiPopulasi(&kota, persentasePajak * kota.populasi / 1400, 5);
82         } else if (persentasePajak >= 20 && persentasePajak < 40) {
83             kota.kebahagiaan -= kebahagiaan/5000;
84             kota.infrastruktur += infrastruktur/5000;
85         } else if (persentasePajak >= 10 && persentasePajak < 20) {
86             kota.kebahagiaan -= kebahagiaan/5500;
87             kota.infrastruktur += infrastruktur/5500;
88         } else if (persentasePajak >= 1 && persentasePajak < 10) {
89             kota.kebahagiaan -= kebahagiaan/6000;
90             kota.infrastruktur += infrastruktur/6000;
91         }
92
93         break;
94     }
95     printf("Kamu menetapkan pajak sebesar %d%%\n", persentasePajak);
96     printf("kebahagiaan masyarakat sekarang adalah %d dan infrastruktur %d\n", kota.kebahagiaan, kota.infrastruktur);
97     pengelolaanAnggaran();
98 }

```

Pada fungsi pajak() terdapat beberapa komponen/variabel yang akan dijelaskan lebih rinci:

- “while(1)” perulangan/loop tak terhingga, digunakan untuk memvalidasi input player. Proses pengaturan pajak akan terus mengulang hingga player memasukkan nilai yang ditentukan yakni 1 sampai dengan 100. Setelah valid maka dengan menggunakan “break;” loop tersebut akan berhenti dan lanjut ke fungsi selanjutnya.
- Fungsi pengaturanPajak() dipanggil untuk mendapatkan input persentase pajak dari player.
- Input yang diterima dari pengaturanPajak() kemudian disimpan dalam variabel persentasePajak.
- Nilai persentasePajak divalidasi untuk memastikan berada dalam rentang 1 hingga 100. Jika nilainya tidak valid (misalnya, -1 atau di luar rentang), maka program akan menampilkan pesan error: "Nilai pajak tidak valid. Harus antara 1-100."
- Nilai kebahagiaan (kebahagiaan) dan infrastruktur (infrastruktur) dihitung berdasarkan persentase pajak dan jumlah populasi kota (kota.populasi).
- Nilai awal kebahagiaan dan infrastruktur disesuaikan dengan formula (persentasePajak * kota.populasi).
- Berdasarkan nilai persentasePajak, kebahagiaan, infrastruktur akan terpengaruh. Semakin tinggi pajak maka akan semakin rendah kebahagiaan warga, namun infrastrukturnya akan semakin tinggi.
- Setelah penyesuaian kebahagiaan dan infrastruktur, program menampilkan pesan kepada player mengenai persentase pajak yang telah ditetapkan serta nilai kebahagiaan dan infrastruktur yang berlaku setelah penetapan pajak tersebut.
- Fungsi pengelolaanAnggaran() dipanggil di akhir untuk melanjutkan ke proses manajemen anggaran yang akan menambah jumlah anggaran kota dari pajak yang telah di dapat.

```

100  Fasilitas fasilitas[5] = {
101      {"Sekolah", 100, 40, 45, 35},
102      {"Rumah Sakit", 90, 37, 44, 74},
103      {"Pasar", 24, 12, 23, 12},
104      {"GrandMall", 78, 30, 40, 26},
105      {"City Club", 120, 55, 54, 92}
106  };

```

- Fungsi ini digunakan untuk memberikan nilai kepada variabel bertipe typedef struct{} Fasilitas yang telah di jelaskan di nomor 2, dengan memberikan nama fasilitas[5] yang diisi dengan 5 elemen. Setiap elemen dalam array berisi data untuk satu fasilitas, terdiri dari beberapa parameter:
 - Nama Fasilitas (String), misalnya "Sekolah", "Rumah Sakit", dll.
 - Biaya Pembangunan (Integer), yaitu biaya dalam satuan miliar yang dibutuhkan untuk membangun fasilitas tersebut.
 - Dampak Kebahagiaan (Integer), yaitu nilai peningkatan kebahagiaan warga kota setelah fasilitas dibangun.
 - Dampak Infrastruktur (Integer), yaitu nilai peningkatan infrastruktur kota.
 - Dampak Populasi (Integer), yaitu jumlah tambahan populasi setelah fasilitas dibangun.

```

108  int pilihFasilitas(){
109      int pilihan;
110      printf("Apa yang ingin kamu bangun(1-5):");
111      if (scanf("%d", &pilihan) != 1){
112          while (getchar() != '\n');
113      }
114      };
115
116      return pilihan - 1;
117  }

```

Fungsi int pilihFasilitas() mendapatkan input dari player tentang fasilitas yang ingin dibangun di kota, dengan pilihan yang berkisar antara 1 hingga 5.

- Variabel int pilihan dideklarasikan untuk menyimpan input pilihan player.
- Fungsi printf digunakan untuk menampilkan pesan kepada player: "Apa yang ingin kamu bangun(1-5):". Ini memberi tahu player bahwa mereka harus memasukkan angka antara 1 hingga 5 sesuai pilihan fasilitas yang diinginkan.
- Fungsi scanf("%d", &pilihan) digunakan untuk membaca input dari player dan menyimpannya di variabel pilihan. Jika scanf gagal membaca sebuah angka (artinya input bukan angka), ia akan mengembalikan nilai selain 1.
- Jika input tidak valid, maka kondisi if (scanf(...) != 1) akan terpenuhi, dan program menjalankan loop while (getchar() != '\n');. Loop ini berfungsi untuk membersihkan karakter yang tidak diinginkan dari input buffer, sehingga input tidak valid tersebut diabaikan dan player bisa memasukkan pilihan yang benar pada kesempatan berikutnya.
- Fungsi return pilihan - 1; mengembalikan nilai input player, tetapi dikurangi 1. Dengan demikian, jika player memasukkan 1, maka nilai yang dikembalikan adalah 0. Hal ini sering dilakukan untuk mengonversi input dari player (biasanya dalam skala 1-5) ke indeks array yang dimulai dari 0.

```

119 int pembangunanInfrastruktur() {
120     while (1) {
121         clear();
122         printf("====Apa yang ingin anda bangun====\n");
123         for (int i = 0; i < 5; i++) {
124             printf("%d. %s\t%d miliar\t+ %d kebahagiaan\t+ %d kemajuan\t+ %d populasi\n",
125                 i+1,
126                 fasilitas[i].jenisFasilitas,
127                 fasilitas[i].biayaPembangunan,
128                 fasilitas[i].dampakMasyarakat,
129                 fasilitas[i].dampakInfrastruktur,
130                 fasilitas[i].dampakPopulasi);
131         }
132
133         int pilihan = pilihFasilitas();
134         if (pilihan >= 0 && pilihan < 5) {
135             if (kota.anggaran >= fasilitas[pilihan].biayaPembangunan) {
136
137                 kota.anggaran -= fasilitas[pilihan].biayaPembangunan;
138                 kota.kebahagiaan += fasilitas[pilihan].dampakMasyarakat;
139                 kota.infrastruktur += fasilitas[pilihan].dampakInfrastruktur;
140                 kota.populasi += fasilitas[pilihan].dampakPopulasi;
141
142                 printf("Anda Membangun %s dengan biaya %d miliar, anda memperoleh %d kebahagiaan, %d kemajuan serta %d penambahan populasi\n",
143                     fasilitas[pilihan].jenisFasilitas,
144                     fasilitas[pilihan].biayaPembangunan,
145                     fasilitas[pilihan].dampakMasyarakat,
146                     fasilitas[pilihan].dampakInfrastruktur,
147                     fasilitas[pilihan].dampakPopulasi);
148             } else {
149                 printf("Anggaran tidak cukup untuk membangun %s\n", fasilitas[pilihan].jenisFasilitas);
150                 continue;
151             }
152         } else {
153             printf("Pilihan tidak valid\n");
154             continue;
155         }
156         break;
157     }
158 }
159

```

Fungsi int pembangunanInfrastruktur() memungkinkan player memilih fasilitas yang ingin dibangun di kota, kemudian memperbarui kondisi kota berdasarkan fasilitas yang dibangun, termasuk efek pada anggaran, kebahagiaan, kemajuan infrastruktur, dan populasi. Berikut adalah penjelasan lebih rinci:

- Fungsi ini menggunakan loop while (1) untuk memastikan bahwa player terus diberikan opsi untuk memilih fasilitas hingga pilihan yang valid dan anggaran yang cukup terpenuhi. Jika ada kesalahan input atau anggaran tidak mencukupi, player akan diberi kesempatan untuk mencoba lagi.
- Fungsi clear() (kemungkinan membersihkan layar) dipanggil untuk mengatur tampilan yang bersih sebelum daftar fasilitas ditampilkan.
- Program menampilkan header "====Apa yang ingin anda bangun====" sebagai judul untuk daftar fasilitas.
- Loop for (int i = 0; i < 5; i++) digunakan untuk mencetak informasi setiap fasilitas yang tersedia dalam array fasilitas, dengan data sebagai berikut:
 - Nomor urut fasilitas.
 - Nama atau jenis fasilitas (fasilitas[i].jenisFasilitas).
 - Biaya pembangunan dalam satuan miliar (fasilitas[i].biayaPembangunan).
 - Dampak pada kebahagiaan masyarakat (fasilitas[i].dampakMasyarakat).
 - Dampak pada kemajuan infrastruktur (fasilitas[i].dampakInfrastruktur).
 - Dampak pada penambahan populasi (fasilitas[i].dampakPopulasi).
- Fungsi pilihFasilitas() dipanggil untuk menerima input dari player dan mengembalikan pilihan fasilitas dalam bentuk indeks (nilai 0 hingga 4, yang sesuai dengan array fasilitas).
- Fungsi memeriksa apakah pilihan berada dalam rentang yang valid (0 hingga 4). Jika tidak valid, pesan "Pilihan tidak valid" akan ditampilkan, dan loop dimulai kembali.
- Jika pilihan valid, maka program memeriksa apakah anggaran kota (kota.anggaran) mencukupi untuk biaya pembangunan fasilitas yang dipilih (fasilitas[pilihan].biayaPembangunan).
- Jika anggaran tidak cukup, pesan error "Anggaran tidak cukup untuk membangun" akan ditampilkan, dan loop dimulai kembali.
- Jika anggaran mencukupi, maka biaya pembangunan akan dikurangi dari kota.anggaran.
- Dampak fasilitas pada kota diterapkan dengan memperbarui beberapa atribut kota:
 - kota.kebahagiaan ditambah dengan nilai fasilitas[pilihan].dampakMasyarakat.
 - kota.infrastruktur ditambah dengan nilai fasilitas[pilihan].dampakInfrastruktur.
 - kota.populasi ditambah dengan nilai fasilitas[pilihan].dampakPopulasi.
- Setelah fasilitas berhasil dibangun, fungsi menampilkan ringkasan berupa nama fasilitas yang dibangun, biaya, serta peningkatan kebahagiaan, kemajuan infrastruktur, dan populasi sebagai hasil dari pembangunan tersebut.
- Setelah pembangunan berhasil, pernyataan break; digunakan untuk keluar dari loop while (1), sehingga proses pembangunan selesai dan fungsi berakhir.

```

161 void penambahanPopulasi(Kota *kota) {
162     int penambahan;
163     printf("Berapa jumlah populasi yang ingin ditambah?:");
164     scanf(" %d", &penambahan);
165     kota->populasi += penambahan;
166     printf("Populasi sekarang: %d\n", kota->populasi);
167 }

```

Fungsi void penambahanPopulasi(Kota *kota) berfungsi untuk menambah populasi kota sesuai jumlah yang diinginkan oleh player. Berikut adalah penjelasan proses dalam fungsi ini:

- Variabel int penambahan dideklarasikan untuk menyimpan jumlah populasi yang ingin ditambahkan.
- printf("Berapa jumlah populasi yang ingin ditambah?:") digunakan untuk meminta player memasukkan jumlah populasi yang ingin ditambahkan.
- scanf(" %d", &penambahan); membaca input jumlah populasi yang dimasukkan oleh player dan menyimpannya di variabel penambahan.
- kota->populasi += penambahan; menambahkan nilai penambahan ke atribut populasi dalam struktur kota. Tanda -> digunakan karena kota adalah pointer ke struktur Kota.
- printf("Populasi sekarang: %d\n", kota->populasi); menampilkan jumlah populasi terbaru setelah penambahan.

```

169 void penambahanPopulasiAcak(Kota *kota) {
170     int penambahan = rand() % 100 + 1;
171     printf("Populasi bertambah %d\n", penambahan);
172     kota->populasi += penambahan;
173     printf("Populasi sekarang %d\n", kota->populasi);
174 }

```

Fungsi void penambahanPopulasiAcak(Kota *kota) digunakan untuk menambah populasi kota secara acak dengan nilai antara 1 hingga 100. Berikut adalah proses yang terjadi di dalam fungsi ini:

- int penambahan = rand() % 100 + 1; menentukan jumlah penambahan populasi secara acak.
- Fungsi rand() menghasilkan bilangan acak, dan rand() % 100 + 1 memastikan nilai tersebut berada dalam rentang 1 hingga 100.
- printf("Populasi bertambah %d\n", penambahan); menampilkan jumlah populasi yang bertambah, sehingga player mengetahui penambahan populasi yang diterima secara acak.
- kota->populasi += penambahan; menambahkan nilai penambahan ke atribut populasi dalam struktur kota. Karena kota adalah pointer ke struktur Kota, operator -> digunakan untuk mengakses anggota populasi.
- printf("Populasi sekarang %d\n", kota->populasi); menampilkan total populasi terbaru setelah penambahan acak dilakukan.

```

176 void kurangiPopulasi(Kota *kota, int jumlah, int langkah) {
177     if (langkah == 0 || kota->populasi <= 0) {
178         if (kota->populasi < 0) kota->populasi = 0;
179         return;
180     }
181     kota->populasi -= jumlah;
182     if (kota->populasi < 0) kota->populasi = 0;
183     kurangiPopulasi(kota, jumlah, langkah - 1);
184 }

```

Fungsi void kurangiPopulasi(Kota *kota, int jumlah, int langkah) digunakan untuk mengurangi populasi kota secara bertahap dalam beberapa langkah yang ditentukan oleh parameter langkah. Berikut adalah penjelasan prosesnya:

1. Mengecek Kondisi Penghentian (Base Case)

- if (langkah == 0 || kota->populasi <= 0) merupakan kondisi penghentian fungsi rekursif:
 - Jika langkah mencapai 0 atau populasi kota sudah mencapai 0, proses pengurangan dihentikan.

- Jika kota->populasi < 0, nilai populasi diatur menjadi 0 untuk mencegah populasi negatif.
2. Mengurangi Populasi
 - kota->populasi -= jumlah; mengurangi populasi kota sebesar nilai jumlah yang diberikan sebagai parameter.
 - Setelah pengurangan, fungsi memeriksa kembali apakah kota->populasi < 0; jika ya, maka nilai populasi diatur menjadi 0.
 3. Memanggil Fungsi Secara Rekursif
 - kurangiPopulasi(kota, jumlah, langkah - 1); adalah panggilan rekursif yang memanggil fungsi dengan nilai langkah yang berkurang satu. Ini memastikan bahwa fungsi akan terus mengurangi populasi sebanyak jumlah hingga langkah mencapai 0 atau populasi menjadi 0.

```

186 void pilihanPenambahan() {
187     while (1){
188         clear();
189         int pilihan;
190         printf("Pilih cara penambahan populasi:\n");
191         printf("1. Tambah sesuai keinginan\n");
192         printf("2. Tambah secara acak\n(1-2):");
193         if (scanf(" %d", &pilihan) != 1){
194             while (getchar() != '\n');
195             continue;
196         };
197
198         if (pilihan == 1) {
199             penambahanPopulasi(&kota);
200         } else if (pilihan == 2) {
201             penambahanPopulasiAcak(&kota);
202         } else {
203             continue;
204         }
205         break;
206     }
207 }

```

Fungsi void pilihanPenambahan() memberikan opsi kepada player untuk memilih cara menambah populasi kota, baik secara manual sesuai keinginan atau secara acak. Berikut adalah rincian proses yang terjadi di dalam fungsi ini:

- Fungsi ini menggunakan loop while (1) untuk memberikan kesempatan kepada player untuk membuat pilihan hingga pilihan yang valid diberikan. Jika terjadi kesalahan, player akan diminta untuk memilih kembali.
- clear(); digunakan untuk membersihkan layar, sehingga tampilan menu pilihan lebih rapi.
- Menu pilihan dicetak dengan printf, menawarkan dua cara untuk menambah populasi:
 - Opsi 1: "Tambah sesuai keinginan"
 - Opsi 2: "Tambah secara acak"
- Player diminta untuk memasukkan pilihan antara 1 dan 2 dengan pesan (1-2):.
- if (scanf(" %d", &pilihan) != 1) digunakan untuk membaca input dari player. Jika input tidak valid (bukan angka), maka loop while (getchar() != '\n'); membersihkan input buffer untuk mengabaikan karakter yang tidak valid dan melanjutkan kembali ke permintaan input.
- Jika pilihan == 1, fungsi penambahanPopulasi(&kota); dipanggil untuk menambah populasi sesuai keinginan player.
- Jika pilihan == 2, fungsi penambahanPopulasiAcak(&kota); dipanggil untuk menambah populasi secara acak.

- Jika player memberikan pilihan yang tidak valid (bukan 1 atau 2), maka loop akan terus berlangsung, dan player akan diminta untuk memilih kembali.
- break; digunakan untuk keluar dari loop while (1) setelah player memberikan pilihan yang valid dan salah satu fungsi penambahan populasi dipanggil.

```
289 void gempaBumi(){
290     int yangKamuPilih;
291     do {
292         clear();
293         printf("Kota Anda diguncang oleh gempa bumi yang sangat dahsyat, seolah-olah bumi mengering dalam kesakitan. Suara gemuruh menggema, merobek kesunyian malam, saat bangunan-bangun
294         printf("Kini saatnya kamu menentukan arah nasib kota ini. Apa yang akan kamu pilih?\n");
295         printf("1. Membangun kembali kota dengan menggunakan teknologi canggih dan material tahan gempa.\n");
296         printf("2. Mengalihkan dana bantuan ke program hiburan dan rekreasi untuk meningkatkan kebahagiaan warga.\n");
297         printf("3. Mendirikan kamp pengungsi sementara dan menunda pembangunan infrastruktur.\n");
298         printf("4. Memfokuskan semua sumber daya untuk membangun infrastruktur dasar dengan cepat, tanpa mempertimbangkan kebutuhan lainnya.\n");
299         printf("(1-4):");
300
301         if (scanf("%d", &yangKamuPilih) != 1) {
302             while (getchar() != '\n');
303             printf("Input tidak valid, silakan masukkan angka 1-4.\n");
304             continue;
305         }
306
307         if (yangKamuPilih < 1 || yangKamuPilih > 4) {
308             printf("Silakan masukkan pilihan yang valid.\n");
309             continue;
310         }
311
312         switch (yangKamuPilih) {
313             case 1:
314                 kota.anggaran -= 289;
315                 kota.infrastruktur += 80;
316                 kota.kebahagiaan += 78;
317                 printf("Kamu menghabiskan 289 miliar untuk membangun kembali kota.\n");
318                 printf("Kebahagiaan dan infrastruktur meningkat +78 dan +80.\n");
319                 break;
320             case 2:
321                 kota.anggaran -= 120;
322                 kota.infrastruktur -= 120;
323                 kota.kebahagiaan += 120;
324                 kota.populasi -= kota.populasi / 30;
325                 if (kota.populasi < 0) kota.populasi = 0; // Prevent negative population
326                 printf("Kamu menghabiskan 120 miliar untuk menghibur wargamu.\n");
327                 printf("Kebahagiaan meningkat +120, tetapi infrastruktur menurun -120.\n");
328                 printf("Populasi berkurang 30%.\n");
329                 break;
330             case 3:
331                 kota.kebahagiaan -= 20;
332                 kota.infrastruktur -= kota.infrastruktur / 20;
333                 kota.anggaran -= 98;
334                 printf("Kamu menghabiskan 98 miliar.\n");
335                 printf("Kota mengalami kerusakan -20% dan kebahagiaan berkurang -20.\n");
336                 break;
337             case 4:
338                 kota.anggaran -= 321;
339                 kota.infrastruktur += 150;
340                 kota.kebahagiaan -= 70;
341                 printf("Kamu menghabiskan 321 miliar.\n");
342                 printf("Infrastruktur meningkat +150, tetapi kebahagiaan menurun -70.\n");
343                 break;
344             }
345         } while (1);
346     }
```

Fungsi void gempaBumi() digunakan untuk menangani kejadian gempa bumi dalam permainan, di mana player diberikan pilihan untuk menentukan tindakan pascagempa yang akan mempengaruhi anggaran, infrastruktur, kebahagiaan, dan populasi kota. Berikut adalah proses yang terjadi di dalam fungsi ini:

- Fungsi dimulai dengan menampilkan narasi tentang gempa bumi besar yang melanda kota. Narasi ini menggambarkan dampak gempa secara dramatis, membangun suasana dan memberikan konteks bagi player untuk membuat keputusan penting.
- Setelah narasi, player diberikan empat opsi tindakan dengan dampak yang berbeda pada kota:
 - Opsi 1: Membangun kembali kota dengan teknologi dan material tahan gempa.
 - Opsi 2: Mengalihkan dana bantuan untuk meningkatkan kebahagiaan melalui program hiburan.
 - Opsi 3: Mendirikan kamp pengungsi sementara dan menunda pembangunan.
 - Opsi 4: Memfokuskan semua sumber daya untuk membangun infrastruktur dasar.
- do...while(1) digunakan untuk memastikan player memberikan input yang valid antara 1 hingga 4. Jika input tidak valid, player diminta untuk mengulangi input.
- scanf("%d", &yangKamuPilih); digunakan untuk membaca pilihan player.
- Jika input tidak berupa angka atau berada di luar rentang 1 hingga 4, player diminta menginput ulang hingga pilihan yang valid diberikan.
- switch (yangKamuPilih) digunakan untuk menentukan dampak dari pilihan player berdasarkan input yang valid:
 - Pilihan 1: Mengurangi anggaran sebesar 289, meningkatkan infrastruktur sebesar 80, dan kebahagiaan sebesar 78. Narasi menunjukkan bahwa player telah memutuskan membangun ulang kota dengan material tahan gempa.
 - Pilihan 2: Mengurangi anggaran dan infrastruktur sebesar 120, meningkatkan kebahagiaan sebesar 120, dan mengurangi populasi sebesar 30%. Pilihan ini fokus pada peningkatan kebahagiaan melalui hiburan, tetapi berdampak negatif pada infrastruktur.
 - Pilihan 3: Mengurangi anggaran sebesar 98, mengurangi infrastruktur sebesar 20%, dan menurunkan kebahagiaan sebesar 20. Pilihan ini berfokus pada pendirian kamp pengungsi dengan penundaan pembangunan.

- Pilihan 4: Mengurangi anggaran sebesar 321, meningkatkan infrastruktur sebesar 150, tetapi menurunkan kebahagiaan sebesar 70. Pilihan ini memprioritaskan pembangunan infrastruktur dasar dengan mengorbankan
 - Setelah pilihan yang valid diproses, loop do...while(1) dihentikan dengan break;.

Ada beberapa fungsi yang sama seperti gempaBumi(), seperti kebakaranHutan(), Banjir(), dsb. Ada juga yang mirip, namun alih-alih mendatangkan bencana, fungsi tsb akan mendatangkan keuntungan bagi game ini. Contoh fungsinya adalah investasiMasuk(), dan kerjaSamaAntarKota().

```

466 void randomEvent() {
467     int random = rand() % 100 + 1;
468     if (random >= 90 && random <= 100) {
469         gempaBumi();
470     } else if (random >= 75 && random < 90) {
471         banjir();
472     } else if (random >= 60 && random < 75) {
473         kebakaranHutan();
474     } else if (random >= 50 && random < 60) {
475         investasiMasuk();
476     } else if (random >= 40 && random < 50) {
477         kerjasamaAntarKota();
478     }
479 }
```

Fungsi randomEvent() menentukan kejadian acak di kota berdasarkan nilai acak dari rand(). Nilai ini menentukan apakah bencana seperti gempa, banjir, atau kebakaran hutan akan terjadi, atau jika peluang seperti investasi atau kerja sama akan muncul. Setiap kejadian acak memicu fungsi tertentu yang menjalankan skenario spesifik, memberikan variasi dinamika kota dalam permainan.

- Random = rand() % 100 + 1; merupakan variabel berbentuk integer yang memiliki nilai acak antara 1 sampai dengan 100.
- Jika nilainya berkisar 90-100 maka yang dipanggil adalah fungsi gempaBumi().
- Nilai 75-89 yang dipanggil adalah fungsi banjir().
- Nilai 60-74 yang dipanggil adalah fungsi kebakaranHutan().
- Nilai 50-9 yang dipanggil adalah fungsi investasiMasuk().
- Nilai 40-49 yang dipanggil adalah fungsi kerjasamaAntarKota().
- Selain daripada itu yakni < 40, tidak akan terjadi apa apa/program akan berjalan seperti biasa(normal).

```

481 void cekPanggilRandomEvent(){
482     if (aksiCounter >= 3){
483         randomEvent();
484         aksiCounter = 0;
485     }
486 }
```

Fungsi cekPanggilRandomEvent() digunakan untuk menghitung/melihat apakah player telah melakukan 3 kali perubahan, jika player telah melakukan perubahan/aksi sebanyak 3 kali maka fungsi ini akan memanggil fungsi randomEvent() yang akan memanggil fungsi lainnya. Setelah itu aksiCounter akan dikembalikan menjadi 0.

```

488 void statusKota(Kota *kota){
489     clear();
490     printf("Anggaran: %d miliar\n", kota->anggaran);
491     printf("Populasi: %d manusia\n", kota->populasi);
492     printf("Kebahagiaan: %d\n", kota->kebahagiaan);
493     printf("Infrastruktur: %d\n", kota->infrastruktur);
494 }

```

Fungsi void statusKota(Kota *kota) digunakan untuk menampilkan status terkini dari kota, termasuk informasi penting seperti anggaran, populasi, kebahagiaan, dan infrastruktur. Tujuannya adalah untuk memberikan gambaran kondisi kota yang dapat digunakan oleh player dalam pengambilan keputusan selanjutnya.

- o Fungsi clear() dipanggil pertama kali untuk membersihkan layar sehingga tampilan status kota menjadi lebih rapi dan mudah dibaca oleh player.
- o printf("Anggaran: %d miliar\n", kota->anggaran); mencetak nilai anggaran kota dalam satuan miliar, memberi tahu player jumlah dana yang tersedia untuk berbagai keperluan, seperti pembangunan dan perbaikan.
- o printf("Populasi: %d manusia\n", kota->populasi); mencetak jumlah penduduk kota. Informasi ini penting karena jumlah penduduk dapat dipengaruhi oleh kejadian seperti bencana atau kebijakan yang player buat.
- o printf("Kebahagiaan: %d\n", kota->kebahagiaan); menampilkan tingkat kebahagiaan penduduk. Nilai kebahagiaan ini dipengaruhi oleh keputusan-keputusan yang dibuat player dan bisa berdampak pada pertumbuhan atau penurunan populasi di masa depan.
- o printf("Infrastruktur: %d\n", kota->infrastruktur); mencetak nilai infrastruktur yang mencerminkan kualitas dan jumlah fasilitas publik. Infrastruktur bisa meningkat atau menurun berdasarkan tindakan player dan kejadian acak yang memengaruhi kondisi kota.

```

496 void lihatSkorKepemimpinan(){
497     clear();
498     int skor = (kota.kebahagiaan * 2 + kota.infrastruktur * 2 + kota.anggaran / 10 + kota.populasi / 100) / 6;
499
500     printf("\n===== Skor Kepemimpinan =====\n");
501     printf("Skor Kepemimpinan Anda: %d\n", skor);
502
503     if (skor >= 80) {
504         printf("Luar biasa! Anda adalah pemimpin yang dicintai dan dihormati. Kota Anda berkembang dengan pesat,\n"
505             "dan warganya bahagia di bawah kepemimpinan Anda. Kota Anda mendekati era keemasan!\n");
506     } else if (skor >= 60) {
507         printf("Anda adalah pemimpin yang kuat dan kompeten. Meski ada tantangan, kota ini terus maju dan\n"
508             "warganya merasa sejahtera. Dengan sedikit usaha lebih, kota ini bisa lebih makmur.\n");
509     } else if (skor >= 40) {
510         printf("Kepemimpinan Anda mendapat kritik, tetapi kota tetap berjalan. Warga memiliki harapan, \n"
511             "meskipun terkadang merasa ragu. Perbaikan di beberapa area dapat meningkatkan kepercayaan masyarakat pada Anda.\n");
512     } else if (skor >= 20) {
513         printf("Kota Anda menghadapi banyak masalah dan warganya tidak puas dengan kepemimpinan Anda.\n"
514             "Jika keadaan tidak segera diperbaiki, kota ini bisa jatuh ke dalam kehancuran.\n");
515     } else {
516         printf("Kota berada di ambang kehancuran. Warga mulai kehilangan harapan, dan beberapa bahkan menyerukan\n"
517             "pergantian pemimpin. Saatnya untuk mengambil tindakan drastis, atau Anda bisa kehilangan segalanya.\n");
518     }
519
520     printf("=====\n\n");
521 }

```

Fungsi void lihatSkorKepemimpinan() menampilkan skor kepemimpinan dari player berdasarkan kondisi kota. Skor ini dihitung dari berbagai aspek kota seperti kebahagiaan, infrastruktur, anggaran, dan populasi. Berdasarkan skor tersebut, fungsi ini memberikan evaluasi terhadap kepemimpinan player dalam bentuk teks deskriptif yang menunjukkan tingkat keberhasilan atau tantangan dalam memimpin kota.

- o Fungsi dimulai dengan clear() untuk membersihkan layar, memastikan tampilan skor kepemimpinan mudah dibaca tanpa gangguan dari tampilan sebelumnya.
- o $\text{int skor} = (\text{kota.kebahagiaan} * 2 + \text{kota.infrastruktur} * 2 + \text{kota.anggaran} / 10 + \text{kota.populasi} / 100) / 6;$
 - Skor dihitung dengan mempertimbangkan berbagai faktor:
 - kota.kebahagiaan dan kota.infrastruktur masing-masing diberi bobot lebih tinggi dengan perkalian 2, menunjukkan betapa pentingnya faktor-faktor ini dalam menciptakan kesejahteraan kota.
 - kota.anggaran dibagi 10, dan kota.populasi dibagi 100 untuk menyesuaikan skala kontribusi mereka terhadap skor keseluruhan.
 - Hasil perhitungan ini dibagi 6 untuk mendapatkan nilai akhir dari skor kepemimpinan.
- o Setelah skor dihitung, nilai ini ditampilkan di layar dengan printf("Skor
- o Berdasarkan nilai skor, fungsi memberikan evaluasi sebagai berikut:

- 80 ke atas: Player dipuji sebagai pemimpin luar biasa yang dicintai warga dan membawa kota ke puncak kejayaan.
 - 60 hingga 79: Player dianggap pemimpin yang kuat, meskipun ada beberapa tantangan, kota tetap berkembang dan warganya makmur.
 - 40 hingga 59: Player mendapat evaluasi campuran; ada kritik, tetapi kota masih berjalan dengan harapan untuk perbaikan.
 - 20 hingga 39: Warga mulai merasa tidak puas, dan kota menghadapi banyak masalah. Player diberi peringatan untuk segera mengambil tindakan.
 - Di bawah 20: Kota hampir hancur, dan warga mulai kehilangan kepercayaan pada kepemimpinan player, mendesak perubahan besar.
- `printf("=====\n\n");` digunakan sebagai penutup untuk tampilan, memberikan batas visual agar informasi lebih terstruktur.

```

522
523 void mainGameBupati(){
524     char main, lanjutMain = 'y', nama[50];
525     int kerja;
526
527     while (1){
528         clear();
529         printf("-----Selamat Datang di Kota Ketapang-----\n");
530         printf("-----Apakah anda ingin menjadi Bupati?-----\n");
531         printf("(Y' or 'N)");
532         if (scanf("%c", &main) != 1){
533             while(getchar() != '\n');
534         };
535
536         getchar();
537
538         printf("-----Masukkan Nama Anda-----\n");
539         scanf("%[^\n]", nama);
540         while(getchar() != '\n');
541
542         if (toupper(main) == 'Y') {
543             do {
544                 while (1){
545                     clear();
546                     printf("-----Selamat Datang di Kota Ketapang, Bupati %s-----\n", nama);
547
548                     printf("Apa yang ingin anda lakukan\n");
549                     printf("1. Membangun fasilitas umum\n");
550                     printf("2. Menambah populasi\n");
551                     printf("3. Mengatur Pajak\n");
552                     printf("4. Melihat status Kota\n");
553                     printf("5. Melihat skor kepemimpinan\n");
554                     printf("6. keluar\n");
555                     printf("Pilih (1-6):");
556                     if (scanf("%d", &kerja) != 1){
557                         while(getchar() != '\n');
558                     };
559                     switch (kerja)
560                     {
561                         case 1:
562                             pembangunanInfrastruktur();
563                             aksiCounter++;
564                             cekPanggilRandomEvent();
565                             break;
566                         case 2:
567                             pilihanPenambahan();
568                             aksiCounter++;
569                             cekPanggilRandomEvent();
570                             break;
571                         case 3:
572                             pajak();
573                             aksiCounter++;
574                             cekPanggilRandomEvent();
575                             break;
576                         case 4:
577                             statusKota(&kota);
578                             break;
579                         case 5:
580                             lihatSkorKepemimpinan();
581                             break;
582                         case 6:
583                             return;
584                         default:
585                             continue;;
586                     }
587
588                     printf("Apakah anda ingin lanjut main?(Y/N):");
589                     getchar();
590                     if (scanf("%c", &lanjutMain) != 1){
591                         while (getchar() != '\n');
592                     }
593                 }
594             } while (toupper(lanjutMain) == 'Y');
595
596             if (toupper(lanjutMain) == 'N'){
597                 printf("Terima kasih telah bermain!\n");
598                 break;
599             }
600         } else if (toupper(main) == 'N'){
601             clear();
602             printf("Kamu cupu!!!!\n");
603             break;
604         }
605     }
606 }

```

Fungsi void mainGameBupati() mengatur alur utama permainan simulasi kepemimpinan di Kota Ketapang. Player akan memulai dengan memilih apakah ingin menjadi bupati dan mengisi nama. Jika setuju, mereka bisa mengelola kota melalui berbagai opsi, termasuk membangun fasilitas, menambah populasi, mengatur pajak, melihat status kota, dan mengevaluasi skor kepemimpinan mereka.

- char main, lanjutMain = 'y', nama[50]; — main untuk menyimpan pilihan awal player menjadi bupati atau tidak, lanjutMain untuk memilih apakah ingin terus bermain atau tidak, dan nama untuk menyimpan nama player.
- int kerja; — variabel yang digunakan untuk memilih opsi aksi di dalam permainan.

- Fungsi dimulai dengan perulangan while(1) yang menampilkan prompt selamat datang di Kota Ketapang dan bertanya apakah player ingin menjadi bupati. Jika scanf gagal membaca input, fungsi membersihkan buffer input.
- toupper(main) == 'Y' — Jika player memilih 'Y', program melanjutkan ke permainan utama. Jika tidak, program menampilkan pesan “Kamu cupu!” dan keluar.
- Setelah persetujuan, player diminta untuk memasukkan nama mereka dengan scanf("%[^\\n]", nama); yang memungkinkan nama lengkap dengan spasi.
- Submenu Aksi Permainan: Di dalam do-while loop, terdapat opsi yang player dapat pilih untuk melakukan aksi tertentu:
 - 1: Membangun fasilitas umum melalui pembangunanInfrastruktur(), menambah jumlah aksi dengan aksiCounter++, dan memeriksa apakah event acak terjadi dengan cekPanggilRandomEvent().
 - 2: Menambah populasi dengan pilihanPenambahan(), juga diikuti dengan penambahan aksi dan cek event acak.
 - 3: Mengatur pajak dengan pajak(), dengan penambahan aksi dan cek event acak.
 - 4: Melihat status kota dengan statusKota(&kota).
 - 5: Melihat skor kepemimpinan dengan lihatSkorKepemimpinan().
 - 6: Memilih untuk keluar dari permainan dengan return.
- Setelah aksi selesai, player ditanya apakah ingin melanjutkan bermain (lanjutMain).
- Jika player memilih N, program keluar dari do-while loop dan menampilkan pesan “Terima kasih telah bermain!”
- Jika player memilih N pada prompt awal (tidak ingin menjadi bupati), fungsi keluar dengan menampilkan pesan "Kamu cupu!!!!".

```

608     int main () {
609         srand(time(NULL));
610         mainGameBupati();
611
612         return 0;
613     }

```

Fungsi main() ini menyiapkan dan memulai permainan simulasi kepemimpinan Kota Ketapang dengan langkah-langkah berikut:

- pertama, mengacak event dalam permainan melalui srand(time(NULL)) untuk memastikan setiap permainan memiliki hasil yang berbeda.
- Selanjutnya, mainGameBupati() dijalankan sebagai fungsi utama yang berisi seluruh mekanisme simulasi, mulai dari pengambilan keputusan hingga pengelolaan kota.
- Terakhir, return 0; memastikan program berhenti dengan sukses, menandakan permainan telah berakhir tanpa error.

Analisis Program (saat dijalankan)

```

=====Selamat Datang di Kota Ketapang=====
=====Apakah anda ingin menjadi Bupati?=====
('Y' or 'N')

```

Gambar di atas menunjukkan pada saat program di jalankan, program akan memintan konfirmasi apakah player ingin bermain menjadi Bupati. (pilihan jawaban 'y'/'Y' dan 'n'/'N').

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

=====Selamat Datang di Kota Ketapang=====
=====Apakah anda ingin menjadi Bupati?=====
('Y' or 'N)y
=====Masukkan Nama Anda=====
█
```

Jika player memilih y/Y maka program akan meminta nama dari player, seperti pada gambar di atas.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  Code + ▢

Kamu cupu!!!!
PS C:\Users\User\OneDrive\Documents\INFORMATIKA UNS\konsep pemrograman\responsi 1> █
```

Jika player memilih/ mengetik n/N maka akan ada kata kata “kamu cupu!!!”, lalu program dihentikan.

```
=====Selamat Datang di Kota Ketapang=====
=====Apakah anda ingin menjadi Bupati?=====
('Y' or 'N)y
=====Masukkan Nama Anda=====
irfan█
```

Setelah player memasukkan nama maka program akan masuk ke permainan.

```
=====Selamat Datang di Kota Ketapang, Bupati irfan=====
Apa yang ingin anda lakukan
1. Membangun fasilitas umum
2. Menambah populasi
3. Mengatur Pajak
4. Melihat status Kota
5. Melihat skor kepemimpinan
6. keluar
Pilih (1-6):█
```

Ini adalah menu pada game simulasi ini, terdapat banyak pilihan dari 1-6.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  Code +

=====Apa yang ingin anda bangun=====
1. Sekolah      100miliar      +40 kebahagiaan +45 kemajuan      +35 populasi
2. Rumah Sakit  90miliar        +37 kebahagiaan +44 kemajuan      +74 populasi
3. Pasar        24miliar        +12 kebahagiaan +23 kemajuan      +12 populasi
4. GrandMall    78miliar        +30 kebahagiaan +40 kemajuan      +26 populasi
5. City Club    120miliar       +55 kebahagiaan +54 kemajuan      +92 populasi
Apa yang ingin kamu bangun(1-5):█
```


Jika kamu memilih 1 pada menu utama, akan muncul pilihan untuk player ingin membangun fasilitas apa.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR
=====Apa yang ingin anda bangun=====
1. Sekolah      100miliar      +40 kebahagiaan +45 kemajuan      +35 populasi
2. Rumah Sakit  90miliar        +37 kebahagiaan +44 kemajuan      +74 populasi
3. Pasar         24miliar        +12 kebahagiaan +23 kemajuan      +12 populasi
4. GrandMall     78miliar        +30 kebahagiaan +40 kemajuan      +26 populasi
5. City Club     120miliar       +55 kebahagiaan +54 kemajuan      +92 populasi
Apa yang ingin kamu bangun(1-5):1
Anda Membangun Sekolah dengan biaya 100 miliar, anda memperoleh 40 kebahagiaan, 45 kemajuan serta 35
penambahan populasi
Apakah anda ingin lanjut main?(Y/N):
```

Setelah memilih fasilitas apa yang akan dibangun, player akan diberikan pop up message yakni player telah berhasil membangun fasilitas. Setelah itu ada konfirmasi apakah player ingin lanjut bermain atau tidak. Jika player memilih y/Y maka program akan kebalik ke menu utama, jika tidak maka player akan diberikan ucapan selamat dan program akan berhenti.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

Pilih cara penambahan populasi:
1. Tambah sesuai keinginan
2. Tambah secara acak
(1-2):
```

Jika pada menu utama player memilih 2, maka player akan di arahkan untuk menambah populasi, di bagian ini terdapat 2 pilihan, yakni menambah sesuai keinginan player atau menambah secara acak.

```
Pilih cara penambahan populasi:
1. Tambah sesuai keinginan
2. Tambah secara acak
(1-2):1
Berapa jumlah populasi yang ingin ditambah?:200
Populasi sekarang: 1235
Apakah anda ingin lanjut main?(Y/N):
```

Misalkan player memilih 1, maka player harus memasukkan berapa jumlah populasi yang ingin di tambah, lalu akan ada pesan berapa jumlah populasi sekarang. Lalu akan ada konfirmasi seperti biasa.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

Pilih cara penambahan populasi:
1. Tambah sesuai keinginan
2. Tambah secara acak
(1-2):2
Populasi bertambah 60
Populasi sekarang 1357
Apakah anda ingin lanjut main?(Y/N):
```

Ketika player memilih menambah secara acak, maka akan di acak dari 1-100, seperti gambar di atas, populasinya bertambah sebanyak 60 dan akan ada pesan mengenai berapa jumlah penduduk sekarang. Lalu akan ada konfirmasi seperti biasa.

```
Kota Anda diguncang oleh gempa bumi yang sangat dahsyat, seolah-olah bumi mengerang dalam kesakitan. Suara gemuruh menggema, merobek kesunyian malam, saat bangunan-bangunan yang sebelumnya kokoh mulai runtuh satu per satu. Jalanan bergetar dan retak, memisahkan teman dari keluarganya. Warga berlarian dengan wajah penuh ketakutan, mata mereka dipenuhi air mata dan keprihatinan. Di tengah kekacauan, teriakan minta tolong terdengar, menggaung di antara puing-puing yang berserakan. Ini adalah malam yang takkan pernah terlupakan, saat segala sesuatu yang dicintai seolah lenyap dalam sekejap mata.
Kini saatnya kamu menentukan arah nasib kota ini. Apa yang akan kamu pilih?
1. Membangun kembali kota dengan menggunakan teknologi canggih dan material tahan gempa.
2. Mengalihkan dana bantuan ke program hiburan dan rekreasi untuk meningkatkan kebahagiaan warga.
3. Mendirikan kamp pengungsi sementara dan menunda pembangunan infrastruktur.
4. Memfokuskan semua sumber daya untuk membangun infrastruktur dasar dengan cepat, tanpa mempertimbangkan kebutuhan lainnya.
(1-4):
```

Ketika player telah melakukan 3x perubahan/aksi, maka akan ada random event yang muncul, ini dapat berupa bencana atau keuntungan, seperti gambar di atas, random eventnnya adalah bencana. Setiap Keputusan akan mempengaruhi segala aspek di kota. Namun akan ada kemungkinan bahwa tidak akan ada random event (yaaa, Namanya juga random kadang ada, kadang tidak).

```
Kota Anda diguncang oleh gempa bumi yang sangat dahsyat, seolah-olah bumi mengerang dalam kesakitan. Suara gemuruh menggema, merobek kesunyian malam, saat bangunan-bangunan yang sebelumnya kokoh mulai runtuh satu per satu. Jalanan bergetar dan retak, memisahkan teman dari keluarganya. Warga berlarian dengan wajah penuh ketakutan, mata mereka dipenuhi air mata dan keprihatinan. Di tengah kekacauan, teriakan minta tolong terdengar, menggaung di antara puing-puing yang berserakan. Ini adalah malam yang takkan pernah terlupakan, saat segala sesuatu yang dicintai seolah lenyap dalam sekejap mata.
Kini saatnya kamu menentukan arah nasib kota ini. Apa yang akan kamu pilih?
1. Membangun kembali kota dengan menggunakan teknologi canggih dan material tahan gempa.
2. Mengalihkan dana bantuan ke program hiburan dan rekreasi untuk meningkatkan kebahagiaan warga.
3. Mendirikan kamp pengungsi sementara dan menunda pembangunan infrastruktur.
4. Memfokuskan semua sumber daya untuk membangun infrastruktur dasar dengan cepat, tanpa mempertimbangkan kebutuhan lainnya.
(1-4):1
Kamu menghabiskan 289 miliar untuk membangun kembali kota.
Kebahagiaan dan infrastruktur meningkat +78 dan +80.
Apakah anda ingin lanjut main?(Y/N):
```

Misalkan kita menjawab 1 maka akan ada pesan mengenai apa saja yang terpengaruh. Lalu akan ada konfirmasi seperti biasa.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

Berapa besar pajak yang kamu inginkan?
(Ingat semakin besar pajak yang kamu buat, semakin sedih wargamu.)(1-100):
```

Ketika player memilih nomor 3 pada menu utama, maka akan ada pesan mengenai berapa jumlah pajak yang ingin player ambil dari penduduk, pajak berkisar antara 1-100 dalam bentuk persen.

Semakin tinggi pajaknya maka akan semakin rendah kebahagiaan Masyarakat, tapi player akan semakin kaya dan infrastruktur nya akan semakin berkembang.

```

Berapa besar pajak yang kamu inginkan?
(Ingat semakin besar pajak yang kamu buat, semakin sedih wargamu.)(1-100):60
Kamu menetapkan pajak sebesar 60%
kebahagiaan masyarakat sekarang adalah 126 dan infrastruktur 172
Apakah anda ingin lanjut main?(Y/N):

```

Pada gambar diatas, dimisalkan player memasukkan input sebesar 60% pajak, maka akan ada pesan berapa kebahagiaan Masyarakat sekarang, dan infrastruktur sekarang.

```

Anggaran: 643 miliar
Populasi: 888 manusia
Kebahagiaan: 126
Infrastruktur: 172
Apakah anda ingin lanjut main?(Y/N):

```

Jika player memilih 4 pada meu utama, player akan diberitahu status kota, yakni anggaran, populasi, kebahagiaan, dan infrastruktur. Lalu seperti biasa akan ada konfirmasi permainan.

```

===== Skor Kepemimpinan =====
Skor Kepemimpinan Anda: 111
Luar biasa! Anda adalah pemimpin yang dicintai dan dihormati. Kota Anda berkembang dengan pesat,
dan warganya bahagia di bawah kepemimpinan Anda. Kota Anda mendekati era keemasan!
=====

Apakah anda ingin lanjut main?(Y/N):

```

Jika player memilih 5 maka akan ada pesan mengenai skor kepemimpinan selama player bermain, semakin tinggi skor maka akan semakin bagus deskripsi yang didapat oleh player.

```

=====Selamat Datang di Kota Ketapang, Bupati irfan=====
Apa yang ingin anda lakukan
1. Membangun fasilitas umum
2. Menambah populasi
3. Mengatur Pajak
4. Melihat status Kota
5. Melihat skor kepemimpinan
6. keluar
Pilih (1-6):6
PS C:\Users\User\OneDrive\Documents\INFORMATIKA UNS\konsep pemrograman\responsi 1>

```

Jika player memilih nomor 6 maka program akan berhenti dan player dapat berhenti bermain.

Tabel Pengimplemetasian materi :

No.	MATERI	POIN	PERAN DALAM PROGRAM
1.	Tipe Data	5	<pre>//INT untuk mendeklarasikan variabel anggaran, populasi, kebahagiaan, infrastruktur, biayaPembangunan, dampakMasyarakat, dampakInfrastruktur, dampakPopulasi, persentasePajak, aksiCounter, pengaturanPajak(); pilihFasilitas(); pembangunanInfrastruktur(); main(); dsb. //CHAR jenisFasilitas[50]; masukan[10]; main, lanjutMAin, nama[50]; //VOID clear(); pengelolaanAnggaran(); pajak(); penambahanPopulasi(Kota *kota); kurangiPopulasi(Kota *kota, int jumlah, int langkah); randomEvent(); dsb</pre>
2.	I/O	5	<pre>//INPUT untuk menginput apakah player ingin bermain pada fungsi mainGameBupatti(); siapa nama player, apa yang ingin player lakukan, berapa jumlah populasi yang ingin ditambah pada fungsi penambahanPopulasi(); apakah player masih ingin bermain atau berhenti. //OUTPUT untuk mencetak/print, siapa nama player, pemilihan menu tentang apa yang ingin player lakukan yang ada di fungsi mainGameBupati(); untuk mencetak status kota juga mencetak kinerja player sebagai bupati yang ada pada fungsi statusKota(); dan lihatSkorKepemimpinan(); </pre>

3	Looping	15	<pre>//while berfungsi sebagai logika untuk <u>perulangan</u> agar ketika <u>player</u> memasukkan input yang <u>tidak</u> sesuai dengan <u>ketentuan</u>, maka <u>program</u> akan terus meminta mengulang hingga jawaban valid. ada beberapa kasus pada program saya yang menggunakan logika ini contohnya di <u>fungsi</u> vodi <u>mainGameBupati()</u>; ada juga <u>penggunaan</u> <u>do{} while()</u>.</pre> <pre>//for digunakan untuk <u>mencetak pilihan</u> yang <u>tersedia</u>, contohnya <u>seperti</u> <u>pilihan</u> pada menu <u>utam</u> <u>dsb.</u> ini terdapat pada <u>fungsi</u> int <u>pembangunanInfrastrutur()</u>;</pre>
4.	Conditional	15	<pre>//if else digunakan untuk <u>menciptakan kondisi</u> <u>dimana</u> jika <u>player</u> memasukkan input karakter kedalam variabel <u>int</u> maka program tidak akan lanjut hingga <u>player</u> memasukkan bilangan bulat. terdapat bebrapa kasus yang menggunakan <u>if else</u> ini, contohnya ada di <u>int pilihFasilitas()</u>; ada juga <u>if else</u>; jika <u>uang/anggaran</u> <u>player</u> tidak cukup, maka <u>fasilitas</u> tidak akan dibangun, ini terdapat di <u>fungsi</u> int <u>pembangunanInfrastruktur()</u>;</pre> <pre>//Switch case ini digunakan untuk menentukan apa yang akan terjadi ketika <u>player</u> memilih <u>pilihan tertentu</u> contohnya ada di menu utama yakni <u>fungsi</u> void <u>mainGameBupati()</u>; <u>player</u> diberikan beberapa pilihan, setiap <u>pilihan</u> akan memberikan <u>output</u> yang berbeda beda. ada juga <u>void randomEvent()</u>; yang akan <u>menciptakan random event</u>, ini akan memberikan <u>player</u> <u>pilihan</u> yang akan <u>mempengaruhi</u> <u>permainan</u></pre>
5.	Rekursi	5	<pre>//rekursi rekursi terdapat pada <u>fungsi void kurangPopulasi()</u>; fungsi ini digunakan untuk <u>mengurangi populasi</u> secara berkala ketika <u>fungsi ini</u> di <u>panggil</u> fungsi <u>tsb</u> akan memanggil dirinya sendiri, setiap kali <u>fungsi</u> memanggil dirinya, <u>nilai</u> fungsi tersebut <u>dikutangi 1</u> hingga <u>kondisi</u> dimana <u>fungsi</u> mencapai <u>nilai 0</u>, lalu program akan berhenti. ini dapat ditemukan pada <u>fungsi void pajak()</u>; ketika <u>player</u> menginput <u>pajak terlalu besar</u>, maka <u>fungsi</u> rekursi ini akan di <u>panggil</u>.</pre>

6.	Array	15	<pre>//array untuk mendeklarasikan array nama[50]; player untuk memebrikan nilai/menginisialisasi typedef struct {} Fasiliias yang telah di deklarasikan dan di beri nilai pada fungsi Fasilitas fasilitas[5] = {}; yang berisi 5 elemen untuk di isi</pre>
7.	Pointer	10	<pre>//pointer untuk menerima alamat dari variabel kota di fungsi void penambahanPopulasi(Kota *kota) {}; yang di panggil dengan penambahanPopulasi(&kota);; void penambahanPopulasiAcak(Kota *kota) {}; void penambahanPopulasiAcak(Kota *kota) {};</pre>
8.	Function	10	<pre>//function untuk membagi suatu program menjadi beberapa bagian agar lebih mudah dimanage/diatur. contoh beberapa fungsi yakni int pembangunanInfrastruktur() {}; void pengelolaanAnggaran() {}; int main(); juga merupakan fungsi, yang adalah fungsi utama dimana akan menjalankan fungsi fungsi lainnya.</pre>
Total : 80			

NB: +10 poin kerapian dan kompleksitas, +10 poin kelengkapan laporan.