

**Makalah Proyek Akhir  
Kelompok 4 Lobi Lobi - Pemrograman Dasar 02**



**Disusun Oleh:**

Ahmad Fariz Khairi	(2306211370)
Aisya Rivelia Azzahra	(2306161864)
Aliya Rizqiningrum Salamun	(2306161813)
Maharaka Fadhilah	(2306225520)

**Dosen Pengampu:**

Dr. Prima Dewi Purnamasari, ST MSc

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER  
UNIVERSITAS INDONESIA  
DEPOK 2023**

- **Pembagian Tugas Anggota Kelompok**

Ahmaf Fariz Khairi

- Mendiskusikan tema program.
- Membuat flowchart.
- Membuat source code program.
- Membuat source code function//fitur.
- Membuat video.

Aisya Rivelia Azzahra

- Mendiskusikan tema program.
- Mendiskusikan ide fitur dan panduan program.
- Merapikan code.
- Membuat pseudocode.
- Mengetik dan menyusun makalah.
- Membuat video.

Aliya Rizqiningrum Salamun

- Mendiskusikan tema program.
- Mendiskusikan ide fitur dan panduan program.
- Merapikan code.
- Membuat list variable/array/struct dll dan kegunaannya.
- Mengetik dan menyusun makalah.
- Membuat video.

Maharaka Fadhilah

- Mendiskusikan tema program.
- Membuat list fungsi dan kegunaannya.
- Membuat source code main program.
- Membuat source code function//fitur.
- Membuat video.

- **Penjelasan Tema Program**

Kelompok kami membuat Aplikasi Mini Quiz dengan tema 7SDG: Affordable and clean energy, kami menamai aplikasi kami "Energy Smart, Energy Fun!". Game quiz ini ditujukan untuk menguji tingkat kesadaran masyarakat tentang pentingnya akses energi yang terjangkau dan bersih. Aplikasi ini diharapkan dapat membantu mengedukasi dan memahami pentingnya tujuan 7SDG, sehingga masyarakat dapat mengambil tindakan untuk menyelesaikan permasalahan 7SDG tersebut.

```
=====
| ~ ~ ~ ~ Main Menu ~ ~ ~ ~ | | ~ ~ ~ ~ Player's Menu ~ ~ ~ ~ | | ~ ~ ~ ~ Admin's Menu ~ ~ ~ ~ |
|=====|=====|=====|
| 1. Login                | | 1. Start Quiz          | | 1. Show Question      |
| 2. Register             | | 2. Show Result        | | 2. Update Question   |
| 3. HelpDesk             | | 0. Logout             | | 0. Logout            |
| 4. About Us             | |=====|=====|=====|
| 5. Exit                 | | User selection (0-2):  | | User selection (0-2): |
|=====|=====|=====|
User selection (1-5):
```

Program ini memungkinkan pengguna untuk mendaftar/register, masuk/login, melihat panduan program/help, dan keluar dari aplikasi/exit. Setelah mengisi data, pengguna akan dialihkan ke menu Admin atau Player sesuai peran pengguna.

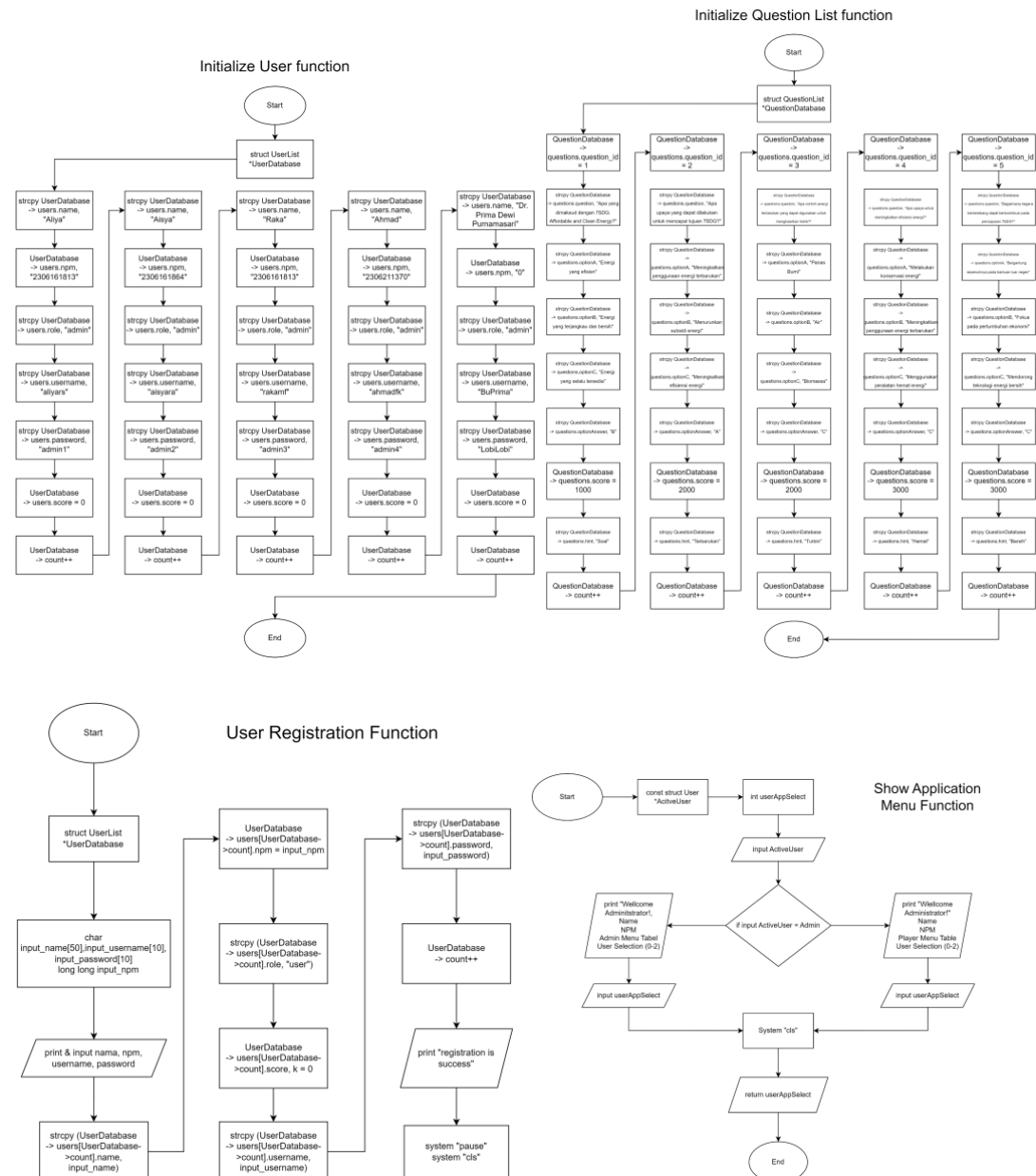
Menu Admin terdiri dari 3 pilihan:

- Show Question: untuk melihat semua pertanyaan dan database kuis.
- Update Question: untuk mengedit pertanyaan, opsi jawaban, dan petunjuk. serta memperbarui skor.
- Logout: untuk kembali ke menu utama.

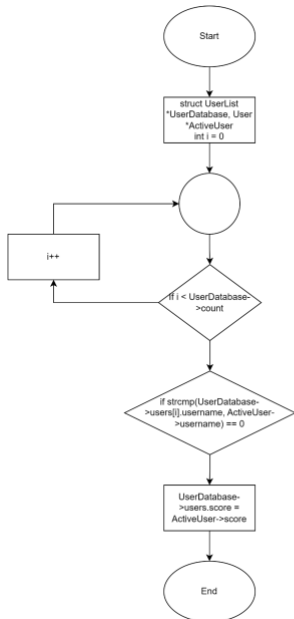
Menu Player terdiri dari 3 pilihan:

- Start Quiz: untuk memulai mini kuis tentang 7SDG.
- Show Result: untuk melihat skor kuis.
- Logout: untuk kembali ke menu utama.

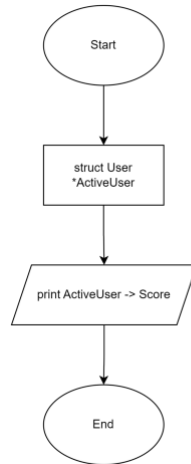
## ● Flowchart



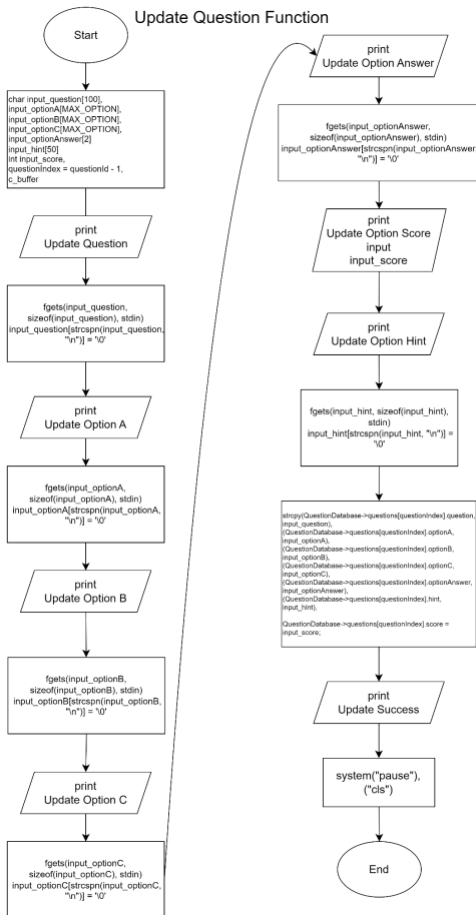
### Save Active User Data Function



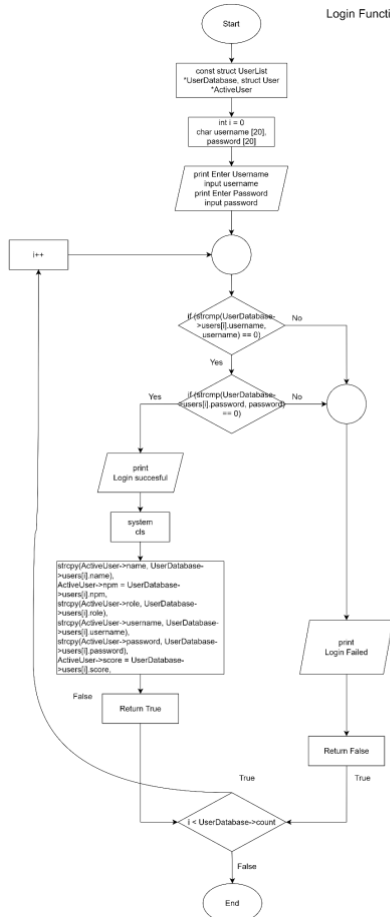
### Show User Score Function



### Update Question Function



### Login Function





---

```
12: End Struct
13:
14: Struct Question
15:     Integer question_id
16:     String question[100]
17:     String optionA[MAX_OPTION]
18:     String optionB[MAX_OPTION]
19:     String optionC[MAX_OPTION]
20:     String optionAnswer[2]
21:     Integer score
22:     String hint[50]
23: End Struct
24:
25: Struct QuestionList
26:     Array of Question questions[MAX_QUESTION]
27:     Integer count
28: End Struct
29:
30: Function Welcome()
31:     Integer i
32:     Print("! ~ Please go fullscreen and then press ENTER to start the game ~ !")

33:     Read a character
34:     Call System("CLS")
35:     Sleep(500)
36:     String banner = "Banner text here..."
37:     For i = 0 To Length of banner - 1
38:         Print banner[i]
39:         Sleep(5)
40:     End For
41:     Print("Loading Game...")
42:     Beep(659, 400)
43:     Sleep(1000)
44:     For i = 1 To 156
45:         Print Character with ASCII code 223
46:         If i = 60 Or i = 100 Then
47:             Sleep(500)
48:         End If
49:         Sleep(12)
50:     End For
51:     Sleep(1600)
52:     Call System("CLS")
53: End Function
54:
55: Function InitializeUserList(UserDatabase)
56:     Set UserDatabase.users[0].name = "Aliya"
57:     Set UserDatabase.users[0].npm = 2306161813
58:     Set UserDatabase.users[0].role = "admin"
59:     Set UserDatabase.users[0].username = "aliyars"
60:     Set UserDatabase.users[0].password = "admin1"
61:     Set UserDatabase.users[0].score = 0
62:     Increment UserDatabase.count
```

---

---

```
63: End Function
64:
65: Function UpdateQuestion(QuestionDatabase, questionId)
66:     Integer questionIndex
67:     String input_question
68:     String input_optionA
69:     String input_optionB
70:     String input_optionC
71:     String input_optionAnswer
72:     Integer input_score
73:     String input_hint
74:     Set questionIndex = questionId - 1
75:     While There is character in input buffer
76:         Read and discard a character
77:     End While
78:     Print "Update Question: "
79:     Read input_question
80:     Remove newline character from input_question
81:     Print "Update Option A: "
82:     Read input_optionA
83:     Remove newline character from input_optionA
84:     Print "Update Option B: "
85:     Read input_optionB
86:     Remove newline character from input_optionB
87:     Print "Update Option C: "
88:     Read input_optionC
89:     Remove newline character from input_optionC
90:     Print "Update Option Answer: "
91:     Read input_optionAnswer
92:     Remove newline character from input_optionAnswer
93:     Print "Update Option Score: "
94:     Read input_score
95:     While There is character in input buffer
96:         Read and discard a character
97:     End While
98:     Print "Update Option Hint: "
99:     Read input_hint
100:    Remove newline character from input_hint
101:    Set QuestionDatabase.questions[questionIndex].question = input_question
102:    Set QuestionDatabase.questions[questionIndex].optionA = input_optionA
103:    Set QuestionDatabase.questions[questionIndex].optionB = input_optionB
104:    Set QuestionDatabase.questions[questionIndex].optionC = input_optionC
105:    Set QuestionDatabase.questions[questionIndex].optionAnswer =
    input_optionAnswer
106:    Set QuestionDatabase.questions[questionIndex].score = input_score
107:    Set QuestionDatabase.questions[questionIndex].hint = input_hint
108:    Print "Update Success!"
109:    Call System("pause")
110:    Call System("cls")
```

---

---

```
111: End Function
112:
113: Function InitializeQuestionList(QuestionDatabase)
114:     Set QuestionDatabase.questions[0].question_id = 1
115:     Set QuestionDatabase.questions[0].question = "Apa yang dimaksud dengan
7SDG: Affordable and Clean Energy?"
116:     Set QuestionDatabase.questions[0].optionA = "Energi yang efisien"
117:     Set QuestionDatabase.questions[0].optionB = "Energi yang terjangkau dan
bersih"
118:     Set QuestionDatabase.questions[0].optionC = "Energi yang selalu tersedia"

119:     Set QuestionDatabase.questions[0].optionAnswer = "B"
120:     Set QuestionDatabase.questions[0].score = 20
121:     Set QuestionDatabase.questions[0].hint = "Soal"
122:     Increment QuestionDatabase.count
123: End Function
124:
125: // Main program
126: Function Main()
127:     Declare UserList UserDatabase
128:     Declare QuestionList QuestionDatabase
129:     Call InitializeUserList(UserDatabase)
130:     Call InitializeQuestionList(QuestionDatabase)
131:     Call Welcome()
132: End Function
133:
134: Call Main()
135: Function welcome():
136:     Display welcome message
137:     Run loading animation
138:
139: Function initializeUserList(UserDatabase):
140:     Input name, NPM, username, and password from the user
141:     Copy user data to the user database
142:     Increment the user count by one
143:
144: Function updateQuestion(QuestionDatabase, questionId):
145:     Input the question ID to be updated
146:     Find the question with the matching ID
147:     If found, update the question data
148:
149: Function initializeQuestionList(QuestionDatabase):
150:     Initialize the question list with default values
151:
152: Function userRegistration(UserDatabase):
153:     Input name, NPM, username, and password from the user
154:     Copy user data to the user database
155:     Increment the user count by one
156:     Display registration success message
157:
158: Function showLoginPage():
```

---



---

```
159:     Display the main menu with options for login, registration, helpdesk, about us,
      and exit
160:
161:  Function login(UserDatabase, ActiveUser):
162:      Input username and password from the user
163:      Loop to find a user based on the username
164:          If a user with a matching username is found
165:              Check the password
166:              If the password matches, save the active user's data
167:              Display login successful message
168:              Return true
169:      If no matching user is found, display login failed message
170:      Return false
171:
172:  Function showAllQuestionDetail(QuestionDatabase):
173:      Loop for each question in the question database
174:          Display details of the question
175:          Wait for user input to continue
176:
177:  Function showApplicationMenu(ActiveUser):
178:      If the user's role is admin
179:          Display the admin menu with options to show questions, update questions,
          and logout
180:      If the user's role is user
181:          Display the user menu with options to start a quiz, show results, and logout
182:
183:      Input the user's choice
184:      Return the user's choice
185:
186:  Function startQuiz(ActiveUser, QuestionDatabase):
187:      Initialize score accumulation
188:      Initialize hint count
189:      Loop for each question in the question database
190:          Display the question and answer choices
191:          If hint count is less than maximum hint
192:              Display an option to request a hint
193:              If the user requests a hint, display the hint and increment hint count
194:
195:          Wait for user answer
196:          If the answer is correct, add to the score
197:          Update the user's score
198:
199:  Function showUserScore(ActiveUser):
200:      Display the user's score
201:
202:  Function showAppreciation(ActiveUser):
203:      Switch case based on the user's score
204:      Display an appreciation message according to the score
205:
206:  Function saveActiveUserData(UserDatabase, ActiveUser):
207:      For each user in UserDatabase:
208:          If user.username matches ActiveUser.username:
209:              Update user.score with ActiveUser.score
```

---

---

```
208:
209:  Function exitGame():
210:    Clear the screen
211:    Display an exit message
212:    Wait for 1.6 seconds
213:    Clear the screen
214:
215:  Function helpdeskMenu():
216:    Display a helpdesk menu with options:
217:      1. How To Login
218:      2. How to Register
219:      3. How to View Questions (Admin)
220:      4. How to Edit Questions (Admin)
221:      5. How to Start A Quiz (Player)
222:      6. How to View Score (Player)
223:      0. Back To Main
224:
225:  Function aboutUs():
226:    Display information about the development team and the purpose of the
    application
227:    Wait for user input to continue
228:
229:  Function main():
230:    Initialize UserDatabase as an empty list
231:    Initialize ActiveUser as an empty user
232:    Initialize QuestionDatabase as an empty list
233:    Initialize loginUserSelect as 0
234:    Initialize appUserSelect as 0
235:    Initialize questionIdToUpdate as 0
236:
237:    Call welcome() to display a welcome message
238:
239:    Repeat:
240:      Call showLoginPage() to display the main menu
241:      Read loginUserSelect from the user
242:
243:      If loginUserSelect is 1:
244:        If not login(UserDatabase, ActiveUser):
245:          Exit the program
246:        Call showApplicationMenu(ActiveUser) and store the result in
    appUserSelect
247:        Repeat:
248:          If ActiveUser.role is "admin":
249:            If appUserSelect is 1:
250:              Call showAllQuestionDetail(QuestionDatabase)
251:            Else if appUserSelect is 2:
252:              Read questionIdToUpdate from the user
253:              Call updateQuestion(QuestionDatabase, questionIdToUpdate)
254:          Else if ActiveUser.role is "user":
255:            If appUserSelect is 1:
256:              Call startQuiz(ActiveUser, QuestionDatabase)
```

---

---

```

257:         Else if appUserSelect is 2:
258:             Call showUserScore(ActiveUser)
259:             Call showAppreciation(ActiveUser)
260:             Call saveActiveUserData(UserDatabase, ActiveUser)
261:             Call showApplicationMenu(ActiveUser) and update appUserSelect

262:     Until appUserSelect is 0
263:
264:     Else if loginUserSelect is 2:
265:         Call userRegistration(UserDatabase)
266:
267:     Else if loginUserSelect is 3:
268:         Call helpdeskMenu()
269:
270:     Else if loginUserSelect is 4:
271:         Call aboutUs()
272:
273:     Until loginUserSelect is 5
274:
275:     Call exitGame() to exit the program

```

---

- **List Variable/Array/Struct/Pointer dan Kegunaannya**

**Variable**

- int i : digunakan dalam perulangan untuk iterasi.
- char a : array karakter untuk menyimpan dan menampilkan function welcome.
- int c\_buffer : untuk deklarasi karakter dari input buffer.
- char input\_question : untuk menyimpan input pertanyaan dari user saat memperbarui pertanyaan quiz
- char input\_option : untuk menyimpan input opsi jawaban dari user saat memperbarui pertanyaan quiz
- int input\_score : menyimpan skor yang diinput oleh pengguna.
- int userAppSelect : menyimpan pilihan menu dari pengguna.
- int questionIdToUpdate : menyimpan ID pertanyaan yang akan diperbarui.
- char userAnswer : menyimpan jawaban dari pengguna.
- char userRequestHint : menyimpan permintaan hint dari pengguna.
- int scoreAccumulation : mengakumulasikan skor pengguna.
- int maximumHint : maksimum hint yang diperbolehkan.
- int hintCount : menghitung jumlah hint yang sudah diberikan.
- int loginUserSelect : menyimpan pilihan pengguna di halaman login.
- long long input\_npm : digunakan untuk menyimpan NPM (nomor pokok mahasiswa) yang diinput oleh pengguna saat registrasi.
- char input\_username : menyimpan username dan password yang diinput oleh pengguna.
- char input\_password : menyimpan username dan password yang diinput oleh pengguna.
- int UserSelect : digunakan dalam fungsi help untuk menangani pilihan pengguna.

**Array**

- char a[] : digunakan untuk menampilkan pesan selamat datang.

- char input\_question[100] : digunakan untuk menyimpan input pertanyaan dan pilihan.
- input\_optionA[MAX\_OPTION]: digunakan untuk menyimpan input pertanyaan dan pilihan.
- char userAnswer[2] : digunakan untuk menyimpan jawaban pengguna.
- char name[50] : menyimpan informasi pengguna yang terdapat dalam struct User.
- char role[6] : menyimpan informasi pengguna yang terdapat dalam struct User.
- char username[20] : menyimpan informasi pengguna yang terdapat dalam struct User.
- char password[20] : menyimpan informasi pengguna yang terdapat dalam struct User.
- char question[100] : menyimpan informasi tentang pertanyaan kuis yang terdapat dalam struct Question.
- char optionA[MAX\_OPTION] : menyimpan informasi tentang opsi jawaban A yang terdapat dalam struct Question.
- char optionB[MAX\_OPTION] : menyimpan informasi tentang opsi jawaban B yang terdapat dalam struct Question.
- char optionC[MAX\_OPTION] : menyimpan informasi tentang opsi jawaban C yang terdapat dalam struct Question.
- char optionAnswer[2] : menyimpan informasi tentang opsi jawaban kuis yang terdapat dalam struct Question.
- char hint[50] : menyimpan informasi tentang petunjuk dari pertanyaan kuis yang terdapat dalam struct Question.

#### **Struct**

- struct User : menyimpan informasi tentang pengguna, termasuk nama, NPM, peran, username, password, dan skor.
- struct UserList : menyimpan daftar pengguna dan jumlah pengguna.
- struct Question : menyimpan informasi tentang pertanyaan, termasuk ID, pertanyaan, pilihan, jawaban, skor, dan hint.
- struct QuestionList : menyimpan daftar pertanyaan dan jumlah pertanyaan.

#### ● **List Fungsi dan Kegunaannya**

1. welcome() : menampilkan pesan selamat datang dan banner game ketika program pertama kali dijalankan. Menggunakan beberapa fungsi dari library `stdio.h`, `stdlib.h`, dan windows.h untuk tampilan dan pengaturan layar.
2. initializeUserList() : menginisialisasi daftar pengguna (user list) pada awal program. Menambahkan beberapa pengguna dengan peran admin dan mengisi data seperti nama, NPM, peran, username, password, dan skor ke dalam struktur data `UserList`.
3. updateQuestion() : memungkinkan admin untuk memperbarui pertanyaan kuis. Admin diminta untuk memasukkan ID pertanyaan yang ingin diperbarui dan mengisi ulang data pertanyaan seperti teks pertanyaan, opsi jawaban, jawaban yang benar, skor, dan petunjuk.
4. initializeQuestionList() : menginisialisasi daftar pertanyaan pada awal

program. Menambahkan beberapa pertanyaan ke dalam struktur data QuestionList beserta dengan opsi jawaban, jawaban yang benar, skor, dan petunjuk.

- 5. userRegistration() : memungkinkan pengguna untuk mendaftar dengan menginput nama, NPM, username, dan password. Data pengguna baru akan ditambahkan ke dalam daftar pengguna.
- 6. showLoginPage() : menampilkan menu utama login, meminta pengguna untuk memilih antara login, registrasi, help desk, about us, atau keluar dari program.
- 7. login() : mencocokkan input username dan password pengguna dengan data yang ada di daftar pengguna. Jika cocok, fungsi akan mengembalikan nilai true dan menyimpan data pengguna aktif.
- 8. showAllQuestionDetail() : menampilkan detail semua pertanyaan, termasuk teks pertanyaan, opsi jawaban, jawaban yang benar, skor, dan petunjuk.
- 9. showApplicationMenu() : menampilkan menu aplikasi berdasarkan peran pengguna (admin atau user) dan meminta input pengguna untuk memilih opsi aplikasi.
- 10. startQuiz() : memulai kuis untuk pengguna dengan peran "user". Pengguna diminta untuk menjawab pertanyaan, dan skor akumulatifnya dihitung berdasarkan jawaban yang benar.
- 11. showUserScore() : menampilkan skor pengguna setelah menyelesaikan kuis.
- 12. showAppreciation() : menampilkan apresiasi sesuai dengan rentang skor yang diperoleh oleh pengguna setelah menyelesaikan kuis.
- 13. saveActiveUserData() : menyimpan skor terbaru pengguna aktif ke dalam daftar pengguna.
- 14. keluar() : menampilkan pesan keluar dan membersihkan layar sebelum program berakhir.
- 15. help() : menampilkan menu help desk yang memberikan petunjuk singkat tentang cara menggunakan program.
- 16. AboutUs() : menampilkan informasi singkat tentang anggota tim pembuat program.
- 17. main() : fungsi utama yang memanggil fungsi-fungsi di atas untuk menjalankan program kuis.

- **Copy Sourcecode**

[illegible]

```
--|\n"\t\t\t\t\t\t |----- A Quiz Game on 7th SDGs Version -----  
--|\n"\t\t\t\t\t\t\t |----- Kelompok Lobi-Lobi -----  
--|\n"\t\t\t\t\t\t\t  
=====\\n\\n"};  
for(i = 0; a[i] != '\\0'; i++) {  
    printf("%c", a[i]);  
    sleep(5);  
}  
  
// Menampilkan progress loading dan beep  
printf("\\t\t\t\t\t\t\t\t\t Loading Game...\\n\\n");  
Beep(659,400);  
  
// Menunda program selama 1 detik  
sleep(1000);  
  
// Menampilkan loading bar  
for(i = 1; i <= 156; i++) {  
    printf("%c", 223); // 223 adalah kode karakter beta di dalam ASCII2  
    if (i == 60 || i == 100)  
        sleep(500); // Menunda program selama 0.5 detik  
    sleep(12);  
}  
  
// Menunda program selama 1.6 detik  
sleep(1600);  
  
// Membersihkan layar  
system("CLS");  
}  
//fungsi untuk menyimpan database admin dan player  
void initializeUserList(struct UserList *UserDatabase) {  
    strcpy(UserDatabase->users[0].name, "Aliya");  
    UserDatabase->users[0].npm = 2306161813;  
    strcpy(UserDatabase->users[0].role, "admin");  
    strcpy(UserDatabase->users[0].username, "aliyars");  
    strcpy(UserDatabase->users[0].password, "admin1");  
    UserDatabase->users[0].score = 0;  
    UserDatabase->count++; //update count  
  
    strcpy(UserDatabase->users[1].name, "Aisyah");  
    UserDatabase->users[1].npm = 2306161864;  
    strcpy(UserDatabase->users[1].role, "admin");  
    strcpy(UserDatabase->users[1].username, "aisyara");  
    strcpy(UserDatabase->users[1].password, "admin2");  
    UserDatabase->users[1].score = 0;  
    UserDatabase->count++;  
  
    strcpy(UserDatabase->users[2].name, "Raka");  
    UserDatabase->users[2].npm = 2306225520;  
    strcpy(UserDatabase->users[2].role, "admin");  
    strcpy(UserDatabase->users[2].username, "rakamf");  
    strcpy(UserDatabase->users[2].password, "admin3");  
    UserDatabase->users[2].score = 0;  
    UserDatabase->count++;  
  
    strcpy(UserDatabase->users[3].name, "Ahmad");  
    UserDatabase->users[3].npm = 23061211370;  
    strcpy(UserDatabase->users[3].role, "admin");  
    strcpy(UserDatabase->users[3].username, "ahmadfk");  
    strcpy(UserDatabase->users[3].password, "admin4");  
    UserDatabase->users[3].score = 0;  
    UserDatabase->count++;  
  
    strcpy(UserDatabase->users[4].name, "Dr. Prima Dewi Purnamasari");
```

```

        UserDatabase->users[4].npm = 0;
        strcpy(UserDatabase->users[4].role, "admin");
        strcpy(UserDatabase->users[4].username, "BuPrima");
        strcpy(UserDatabase->users[4].password, "LobiLobi");
        UserDatabase->users[4].score = 0;
        UserDatabase->count++;
    }

    //fungsi untuk mengubah soal bagi admin
    void updateQuestion(struct QuestionList *QuestionDatabase, int
    questionId) {
        char input_question[100];
        char input_optionA[MAX_OPTION];
        char input_optionB[MAX_OPTION];
        char input_optionC[MAX_OPTION];
        char input_optionAnswer[2];
        int input_score;
        char input_hint[50];
        int questionIndex = questionId - 1;
        int c_buffer;

        // clear buffer sebelum input fgets
        while ((c_buffer = getchar()) != '\n' && c_buffer != EOF);

        // ambil input user
        printf("Update Question\t: ");
        fgets(input_question, sizeof(input_question), stdin); //dipake
        untuk membaca spasi
        input_question[strcspn(input_question, "\n")] = '\0'; // Remove
        newline if present

        printf("Update Option A\t: ");
        fgets(input_optionA, sizeof(input_optionA), stdin);
        input_optionA[strcspn(input_optionA, "\n")] = '\0'; // Remove
        newline if present

        printf("Update Option B\t: ");
        fgets(input_optionB, sizeof(input_optionB), stdin);
        input_optionB[strcspn(input_optionB, "\n")] = '\0'; // Remove
        newline if present

        printf("Update Option C\t: ");
        fgets(input_optionC, sizeof(input_optionC), stdin);
        input_optionC[strcspn(input_optionC, "\n")] = '\0'; // Remove
        newline if present

        printf("Update Option Answer\t: ");
        fgets(input_optionAnswer, sizeof(input_optionAnswer), stdin);
        input_optionAnswer[strcspn(input_optionAnswer, "\n")] = '\0'; //
        Remove newline if present

        printf("Update Option Score\t: ");
        scanf("%d", &input_score);

        // clear the input buffer
        while ((c_buffer = getchar()) != '\n' && c_buffer != EOF);

        printf("Update Option Hint: ");
        fgets(input_hint, sizeof(input_hint), stdin);
        input_hint[strcspn(input_hint, "\n")] = '\0'; // Remove newline
        if present

        // simpan perubahan
        strcpy(QuestionDatabase->questions[questionIndex].question,
        input_question);
        strcpy(QuestionDatabase->questions[questionIndex].optionA,
        input_optionA);
        strcpy(QuestionDatabase->questions[questionIndex].optionB,
        input_optionB);
    }

```



```

        strcpy(QuestionDatabase->questions[questionIndex].optionC,
input_optionC);
        strcpy(QuestionDatabase->questions[questionIndex].optionAnswer,
input_optionAnswer);
        QuestionDatabase->questions[questionIndex].score = input_score;
        strcpy(QuestionDatabase->questions[questionIndex].hint,
input_hint);

        printf("\n Update Success! \n");
        system("pause");
        system("cls");
    }
//fungsi yang menyimpan pertanyaan yang akan ditanyakan dalam quiz
void initializeQuestionList(struct QuestionList *QuestionDatabase) {
    QuestionDatabase->questions[0].question_id = 1;
    strcpy(QuestionDatabase->questions[0].question, "Apa yang
dimaksud dengan 7SDG: Affordable and Clean Energy?");
    strcpy(QuestionDatabase->questions[0].optionA, "Energi yang
efisien");
    strcpy(QuestionDatabase->questions[0].optionB, "Energi yang
terjangkau dan bersih");
    strcpy(QuestionDatabase->questions[0].optionC, "Energi yang
selalu tersedia");
    strcpy(QuestionDatabase->questions[0].optionAnswer, "B");
    QuestionDatabase->questions[0].score = 20; //score dari soal
    strcpy(QuestionDatabase->questions[0].hint, "Soal"); //hint yang
tersedia
    QuestionDatabase->count++; //update count

    QuestionDatabase->questions[1].question_id = 2;
    strcpy(QuestionDatabase->questions[1].question, "Apa upaya yang
dapat dilakukan untuk mencapai tujuan 7SDG?");
    strcpy(QuestionDatabase->questions[1].optionA, "Meningkatkan
penggunaan energi terbarukan");
    strcpy(QuestionDatabase->questions[1].optionB, "Menurunkan
subsidi energi");
    strcpy(QuestionDatabase->questions[1].optionC, "Meningkatkan
efisiensi energi");
    strcpy(QuestionDatabase->questions[1].optionAnswer, "A");
    QuestionDatabase->questions[1].score = 20;
    strcpy(QuestionDatabase->questions[1].hint, "Terbarukan");
    QuestionDatabase->count++;

    QuestionDatabase->questions[2].question_id = 3;
    strcpy(QuestionDatabase->questions[2].question, "Apa contoh
energi terbarukan yang dapat digunakan untuk menghasilkan
listrik?");
    strcpy(QuestionDatabase->questions[2].optionA, "Panas Bumi");
    strcpy(QuestionDatabase->questions[2].optionB, "Air");
    strcpy(QuestionDatabase->questions[2].optionC, "Biomassa");
    strcpy(QuestionDatabase->questions[2].optionAnswer, "B");
    QuestionDatabase->questions[2].score = 20;
    strcpy(QuestionDatabase->questions[2].hint, "Turbin");
    QuestionDatabase->count++;

    QuestionDatabase->questions[3].question_id = 4;
    strcpy(QuestionDatabase->questions[3].question, "Apa upaya untuk
meningkatkan efisiensi energi?");
    strcpy(QuestionDatabase->questions[3].optionA, "Melakukan
konservasi energi");
    strcpy(QuestionDatabase->questions[3].optionB, "Meningkatkan
penggunaan energi terbarukan");
    strcpy(QuestionDatabase->questions[3].optionC, "Menggunakan
peralatan hemat energi");
    strcpy(QuestionDatabase->questions[3].optionAnswer, "C");
    QuestionDatabase->questions[3].score = 20;
    strcpy(QuestionDatabase->questions[3].hint, "Hemat");
    QuestionDatabase->count++;
}

```

```

        QuestionDatabase->questions[4].question_id = 5;
        strcpy(QuestionDatabase->questions[4].question, "Bagaimana
negara berkembang dapat berkontribusi pada pencapaian 7SDG?");
        strcpy(QuestionDatabase->questions[4].optionA, "Bergantung
sepenuhnya pada bantuan luar negeri");
        strcpy(QuestionDatabase->questions[4].optionB, "Fokus pada
pertumbuhan ekonomi");
        strcpy(QuestionDatabase->questions[4].optionC, "Mendorong
teknologi energi bersih");
        strcpy(QuestionDatabase->questions[4].optionAnswer, "C");
        QuestionDatabase->questions[4].score = 20;
        strcpy(QuestionDatabase->questions[4].hint, "Bersih");
        QuestionDatabase->count++;
    }
    //fungsi untuk meminta data user ketika register sebelum memainkan
    quiz
    void userRegistration(struct UserList *UserDatabase){
        char input_name[50];
        long long input_npm;
        char input_username[10];
        char input_password[10];

        // ambil input user
        printf("\n\nYour Name: ");
        scanf("%50s", &input_name);
        printf("Your NPM: ");
        scanf("%d", &input_npm);
        printf("New Username: ");
        scanf("%10s", &input_username);
        printf("New Password: ");
        scanf("%10s", &input_password);

        // tambahkan data pada database user
        strcpy(UserDatabase->users[UserDatabase->count].name,
input_name);
        UserDatabase->users[UserDatabase->count].npm = input_npm;
        strcpy(UserDatabase->users[UserDatabase->count].role, "user");
        UserDatabase->users[UserDatabase->count].score = 0;
        strcpy(UserDatabase->users[UserDatabase->count].username,
input_username);
        strcpy(UserDatabase->users[UserDatabase->count].password,
input_password);
        UserDatabase->count++;

        printf("\nregistration is success \n");
        system("pause");
        system("cls");
    }

```

```

    }
    //fungsi menampilkan interface main menu
    void showLoginPage(){
        printf("\n===== \n"
            "| ~ ~ ~ ~ ~ Main Menu ~ ~ ~ ~ ~ | \n"
            "===== \n"
            "===== \n"
            "| \n"
            "| 1. Login \n"
            "| 2. Register \n"
            "| 3. HelpDesk \n"
            "| 4. About Us \n"
            "| 5. Exit | \n"
            "| \n"
            "===== \n");
    }

```

```

// Fungsi untuk simulasi proses logina
bool login(const struct UserList *UserDatabase, struct User
*ActiveUser) { //syntax pake const untuk yang list const (tidak ada
perubahannya)

```

```

char username[20];
char password[20];

// user input username dan password untuk login
printf("\n\nEnter Username\t: ");
scanf("%20s", username);
printf("Enter Password\t: ");
scanf("%20s", password);

// loop find user
int i;
for (i = 0; i < UserDatabase->count; i++) {
    // cari berdasarkan username
    // melakukan string comparison, jika sama maka bernilai 0
    if (strcmp(UserDatabase->users[i].username, username) == 0)
    {
        // jika user di temukan maka cek password
        if (strcmp(UserDatabase->users[i].password, password) ==
0){
            printf("Login successful!\n");
            system("cls");

            // simpan status user yang sedang aktif
            strcpy(ActiveUser->name, UserDatabase-
>users[i].name); //disimpan dari user database yang sudah ditemukan
tadi
            ActiveUser->npm = UserDatabase->users[i].npm;
            strcpy(ActiveUser->role, UserDatabase-
>users[i].role);
            strcpy(ActiveUser->username, UserDatabase-
>users[i].username);
            strcpy(ActiveUser->password, UserDatabase-
>users[i].password);
            ActiveUser->score = UserDatabase->users[i].score;
            return true;
        }
        else{
            printf("Login failed!\n"); //jika data tidak
ditemukan
            return false;
        }
    }
}
// data tidak di temukan
printf("Login failed!\n");
return false;
}

//fungsi untuk menampilkan pertanyaan quiz
void showAllQuestionDetail(const struct QuestionList
*QuestionDatabase){
    int i;
    for (i = 0; i < QuestionDatabase->count; i++) {
        printf("\n");
        printf("=====\\n");
        printf("|                Question ID %d                |\\n"
,QuestionDatabase->questions[i].question_id);

        printf("=====\\n\\n");
        printf("Question : %s\\n\\n",QuestionDatabase-
>questions[i].question);
        printf("Option A : %s\\n",QuestionDatabase-
>questions[i].optionA);
        printf("Option B : %s\\n",QuestionDatabase-
>questions[i].optionB);
        printf("Option C : %s\\n",QuestionDatabase-
>questions[i].optionC);
        printf("Option Answer : %s\\n",QuestionDatabase-
>questions[i].optionAnswer);
    }
}

```

```

        printf("Question Score : %i\n",QuestionDatabase-
>questions[i].score);
        printf("Option Hint      : %s\n",QuestionDatabase-
>questions[i].hint);
        system("pause");
        system("cls");
    }
}

//fungsi untuk menampilkan menu bagi admin dan player
int showApplicationMenu(const struct User *ActiveUser){
    int userAppSelect;
    if (strcmp(ActiveUser->role, "admin") == 0){ //sesuai role user
        printf("\n~ ~ Welcome Administrator! ~ ~\n");
        printf("Name : %s\n", ActiveUser->name);
        printf("NPM : %lld\n", ActiveUser->npm);
        printf("\n===== \n"
            "| ~ ~ ~ ~ Admin's Menu ~ ~ ~ ~ | \n"
            "===== \n"
            "===== \n"
            "| \n"
            "| 1. Show Question | \n"
            "| 2. Update Question | \n"
            "| 0. Logout | \n"
            "| \n"
            "===== \n");
        printf("User selection (0-2): ");
        scanf("%d", &userAppSelect);
        system("cls");
        return userAppSelect;
    }
    else if (strcmp(ActiveUser->role, "user") == 0){
        printf("\n~ ~ Welcome Player! ~ ~\n");
        printf("Name : %s\n", ActiveUser->name);
        printf("NPM : %lld\n", ActiveUser->npm);
        printf("\n===== \n"
            "| ~ ~ ~ ~ Player's Menu ~ ~ ~ ~ | \n"
            "===== \n"
            "===== \n"
            "| \n"
            "| 1. Start Quiz | \n"
            "| 2. Show Result | \n"
            "| 0. Logout | \n"
            "| \n"
            "===== \n");
        printf("User selection (0-2): ");
        scanf("%d", &userAppSelect);
        system("cls");
        return userAppSelect;
    }
}

//fungsi untuk memulai quiz
void startQuiz(struct User *ActiveUser, const struct QuestionList
*QuestionDatabase) {
    int scoreAccumulation = 0;
    int maximumHint = 3; //hanya bisa memakai maksimal 3 hint dari 5
    soal
    char userAnswer[2];
    char userRequestHint;
    int hintCount = 0; //Hitung jumlah hint yang sudah diberikan

    int i;
    for (i = 0; i < QuestionDatabase->count; i++) { //menampilkan
    pertanyaan
        printf("\n<~~~~~ Question ID %d ~~~~~>\n",
QuestionDatabase->questions[i].question_id);
        printf("Question: %s\n", QuestionDatabase-
>questions[i].question);
        printf("A) %s\n", QuestionDatabase->questions[i].optionA);

```

```

        printf("B) %s\n", QuestionDatabase->questions[i].optionB);
        printf("C) %s\n", QuestionDatabase->questions[i].optionC);

        // Tanya apakah user ingin hint jika belum melebihi maksimum
hint
        if (hintCount < maximumHint) {
            printf("Do you want a hint? (Y/N): ");
            scanf(" %c", &userRequestHint);

            if (userRequestHint == 'Y' || userRequestHint == 'y') {
                printf("Hint: %s\n", QuestionDatabase-
>questions[i].hint);
                hintCount++; // Tambah hitungan hint
                printf("User answer (A-C): "); //lanjut menjawab
                jika sudah mendapat hint
                scanf("%s", &userAnswer);
                if (strcmp(QuestionDatabase-
>questions[i].optionAnswer, userAnswer) == 0) {
                    scoreAccumulation += QuestionDatabase-
>questions[i].score;
                }
            }
            else if (userRequestHint == 'N' || userRequestHint ==
'n'){
                printf("User answer (A-C): "); //jika tidak
                ingin hint maka lanjut menjawab soal
                scanf("%s", &userAnswer);
                if (strcmp(QuestionDatabase-
>questions[i].optionAnswer, userAnswer) == 0) {
                    scoreAccumulation += QuestionDatabase-
>questions[i].score;
                }
            }
        }
        else{ //jika sudah mencapai maksimum hint maka hint tidak
        akan ditawarkan lagi
            printf("User answer (A-C): "); //langsung menjawab soal
            scanf("%s", &userAnswer);
            if (strcmp(QuestionDatabase->questions[i].optionAnswer,
userAnswer) == 0) {
                scoreAccumulation += QuestionDatabase-
>questions[i].score;
            }
        }
        }
        system("cls");

        // update data active user
        ActiveUser->score = scoreAccumulation; //update score
    }

    //fungsi untuk menampilkan score
    void showUserScore(struct User *ActiveUser){
        printf(" \n\n =====\n");
        printf(" || Your Score is: %d ||\n",ActiveUser->score);
        printf(" =====\n");
    }

    //fungsi untuk menampilkan parameter seberapa tereduasinya user
    tentang materi
    void showAppreciation(struct User *ActiveUser){
        // switch case
        switch (ActiveUser->score / 20) {
            case 0:
                // Range 0-20
                printf(" Wah, pemahaman kamu masih kurang banget nih
semangat terus yaa belajarnya!\n\n");
                system("pause");
                system("cls");
                break;

```



```

        "=====\n");
do {
    printf("User Selection(0-6): ");
    scanf("%d", &UserSelect);

    if (UserSelect < 0 || UserSelect > 6) {
        printf("Invalid selection, please try again.\n");
        continue;
    }

    // Clearing the screen, note: system("cls") is platform-
dependent
    system("cls");

    switch(UserSelect) {
        case 1:

            printf("=====\n"
                " |                                     How To Login
            |\n"

            "=====\n\n"
            " 1. Enter the number 1 on the main
menu\n"
            " 2. You will be prompted to enter your
username and password\n"
            " 3. For admin, enter the username and
password as per the program's stored data\n"
            " 4. For players, enter the username and
password registered during the registration process\n"
            " 5. The program will check if the username and
password are registered\n"
            " 6. If login successful, your Name and Student
ID (NPM) will be displayed, and you will be directed to the
Admin/User menu\n"
            " 7. If the username and password are
incorrect, it will display -Login failed- and exit the program\n\n"
            );
            system ("pause");//menjeda program sampai ada input
            break;//keluar dari pengkondisian
        case 2:

            printf("=====\n"
                " |                                     How To Register
            |\n"

            "=====\n\n"
            " 1. Enter the number 2 on the main
menu\n"
            " 2. You will be prompted to enter your
Name, Student ID (NPM), and create a new account with a username and
password\n"
            " 3. Registration is successful.\n"
            " 4. You will then return to the main menu
to log in (follow guide 1)\n\n"
            );
            system ("pause");//menjeda program sampai ada input
            break;//keluar dari pengkondisian
        case 3:

            printf("=====\n"
                " |                                     How To View Questions
            |\n"

            "=====\n\n"
            " 1. Log in with the admin account (Follow
guide 1).\n"

```

```

questions.\n"
" 2. Enter the number 1 to view all
displayed one by one.\n"
" 3. All provided questions will be
return to the admin menu.\n\n"
);
system ("pause");//menjeda program sampai ada input
break;//keluar dari pengkondisian
case 4:

printf("=====\n"
"| How to Edit a Question
|\n"

"=====\n\n"
" 1. Log in with the admin account
(Follow guide 1).\n"
" 2. Enter the number 2 on the admin
menu to edit questions.\n"
" 3. Enter the question ID you want to
edit.\n"
" 4. You can then modify the question
and its answer options.\n"
" 5. You can also change the answer and
the score for that question.\n"
" 6. Lastly, you will be prompted to
enter the updated hint.\n"
" 7. The question is successfully
updated.\n"
" 8. You will be returned to the admin
menu.\n\n"
);
system ("pause");//menjeda program sampai ada input
break;//keluar dari pengkondisian
case 5:

printf("=====\n"
"| How to Start a Quiz
|\n"

"=====\n\n"
" 1. Log in with the user account. (If you
don't have an account, follow guide 2 -How to Register-).\n"
" 2. After logging in, you will see the -Player
Menu-\n"
" 3. Enter the number 1 -Start Quiz- to
begin the quiz.\n"
" 4. You can start working on the quiz.\n"
" 5. Pay close attention to the
instructions while answering the quiz.\n\n"
);
system ("pause");//menjeda program sampai ada input
break;//keluar dari pengkondisian
case 6:

printf("=====\n"
"| How to View Quiz Results
|\n"

"=====\n\n"
" 1. Register and log in with the player
account. (Follow guides 1 and 2).\n"
" 2. Start the quiz first to earn points;
otherwise, your score will be 0.\n"
" 3. Choose the number 2 on the player
menu.\n"
" 4. The quiz result score will be
displayed.\n"

```



```

        " 5. After that, you will be immediately
returned to the player menu.\n\n"
    );
    system("pause");//menjeda program sampai ada input
    break;//keluar dari pengkondisian
    default:
        break;

    }

    getchar();
    system("cls");
} while (UserSelect > 7 || UserSelect < 0); //akan ditampilkan
terus sampai pilihan di input user
}
//menampilkan profil developer dan latar belakang program dibuat
void AboutUs(){
    printf("\n\n\n===== Greetings!
===== \n\n\nHello! we are from the group
4 <Lobi - Lobi>. Our group consists of 4 members:\n"
        "Ahmad Fariz Khairi, Aisya Rivelia Azzahra, Aliya
Rizqiningrum Salamun, and\n"
        "Maharaka Fadhilah. As game developers , we designed
a mini quiz to assess\n"
        "public awareness regarding the importance of
affordable and clean energy\n"
        "access, as well as knowledge about the 7th
Sustainable Development Goal (SDGs):\n"
        "Affordable and Clean Energy. This application aims
to help educate and promote\n"
        "understanding of the significance of the 7th SDG,
encouraging people to take\n"
        "action to address the related challenges.
Enjoy!\n\n\n");
    system("pause");
    system("cls");
}

int main() {
    //menghias terminal
    system("color D");
    system("color B5");
    // deklarasi variable user
    struct UserList UserDatabase; //ketika di logout bakal tersimpan
di user database
    struct User ActiveUser; //active user saat dia login akan diisi
dari data yang input
    UserDatabase.count = 0;

    // deklarasi variable question
    struct QuestionList QuestionDatabase;
    int loginUserSelect; // menu utama
    int appUserSelect; // player atau admin
    int questionIdToUpdate; //untuk update pertanyaan
    QuestionDatabase.count = 0;

    // Inisialisasi data user dan question
    initializeUserList(&UserDatabase); // dikirim alamat user
database
    initializeQuestionList(&QuestionDatabase); // dikirim alamat
dari question database
    welcome(); //menampilkan interface masuk ke game
    // show login page
    showLoginPage(); //panggil function menu utama
    printf("User selection (1-5): ");
    scanf("%d", &loginUserSelect); //meminta user memilih fitur yang
ingin digunakan

```

```

system("cls");

while(loginUserSelect != 5){ //selama tidak memilih 5 menu akan
terus tampil (loop)
    if (loginUserSelect == 1){ //kalau user memilih 1
        //akan diminta login
        if (!login(&UserDatabase, &ActiveUser)) { // lookup
database
            system("pause");
            system("cls");
            continue; //jika login gagal maka akan diminta lagi
username dan pass sampai benar
        }
        // tampilkan menu aplikasi
        appUserSelect = showApplicationMenu(&ActiveUser);

        while(appUserSelect != 0){ //selama tidak memilih 0 loop
akan jalan
            // process pilihan user
            if (strcmp(ActiveUser.role, "admin") == 0){
//kondisi jika role adalah admin
                if(appUserSelect == 1){
                    // tampilkan detail soal
                    showAllQuestionDetail(&QuestionDatabase);
                }
                else if (appUserSelect == 2){
                    // update soal
                    printf("Question ID to Update: ");
                    scanf("%d", &questionIdToUpdate);
                    updateQuestion(&QuestionDatabase,
questionIdToUpdate);
                }
            }
            else if (strcmp(ActiveUser.role, "user") == 0){
//kondisi jika role adalah user(player)
                if(appUserSelect == 1){
                    // mulai quiz
                    startQuiz(&ActiveUser, &QuestionDatabase);
                }
                else if(appUserSelect == 2){
                    showUserScore(&ActiveUser); //tampilkan
score
                    showAppreciation(&ActiveUser); //dan
parameter
                }
            }
            saveActiveUserData(&UserDatabase, &ActiveUser);
//simpan data player selama bermain
            appUserSelect = showApplicationMenu(&ActiveUser);
//tampilkan terus application menu
        }

    }
    else if (loginUserSelect == 2){
        userRegistration(&UserDatabase); //kalau memilih 2 maka
user akan diminta register data
    }
    else if (loginUserSelect == 3){
        help(); //jika memilih 3 user akan dibawa ke menu
panduan menggunakan program
    }
    else if (loginUserSelect == 4){
        AboutUs(); //jika memilih 4 akan muncul biodata
developer
    }
    showLoginPage(); //akan muncul terus menu selam belum input
5 (exit)
    printf("User selection: ");
    scanf("%d", &loginUserSelect);
}

```

```
        system("cls");  
    }  
    keluar(); //jika sudah input 5 akan muncul interface exit  
    return 0;  
}
```