

**MAKALAH PEMROGRAMAN DASAR
TRANS UI**



Disusun Oleh :
Brian Yudha Sandi – 2106637082
Leonardo Jeremy Pongpare Munda – 2106707914
Raden Bagus Senopati Kresna Ramdani Galih Rahayu – 2106702610
Syauqi Auliya Muhammad – 2106702610

Program Studi Teknik Komputer
Fakultas Teknik
Universitas Indonesia
Tahun Akademik 2021 / 2022

A) PENJELASAN TEMA PROGRAM SECARA SINGKAT

Program kami yang berjudul TransUI (Transport UI) ini bertujuan untuk memudahkan mahasiswa UI dalam bepergian terutama dalam ruang lingkup UI dengan menggabungkan berbagai macam metode transportasi yang bisa digunakan, antara lain: Bikun (bis kuning), Spekun (sepeda kuning), KRL, dan Ojek Online. Program juga menggunakan mekanisme login agar dapat digunakan oleh lebih dari satu orang di waktu yang bersamaan.

B) PEMBAGIAN TUGAS

a. Syauqi :

i. Functions :

1. main
2. daftar
3. mainmenu
4. topup
5. helpdesk
6. ojek
7. delay

ii. Makalah :

1. Pembagian tugas
2. Pseudocode
3. Flowchart
4. List fungsi beserta kegunaannya

iii. Video :

1. Penjelasan mengenai main function dan mainmenu serta mekanisme login/registrasi
2. Penjelasan mengenai function ojek

b. Brian :

i. Functions :

1. kereta
2. displaykereta

ii. Makalah :

1. Flowchart
2. List variabel dan library beserta kegunaannya

iii. Video :

1. Penjelasan tentang function kereta (+ jakartakota dan bogor)

c. Kresna :

i. Functions :

1. bikun
2. jumlahbikun

ii. Makalah :

1. Flowchart
2. List variabel/struct/dan array beserta kegunaannya

iii. Video :

1. Penjelasan mengenai function bikun (+ jumlahbikun)
2. Recording

d. Leonardo :

i. Functions :

1. spekun
2. displayspekun

ii. Makalah :

1. Pseudocode
2. List fungsi beserta kegunaannya
3. List struct dan array beserta kegunaannya

- iii. Video :
 1. Penjelasan mengenai function spekun (+displayspekun)
 2. Editing

C) LIST VARIABEL, STRUCT, ARRAY, FUNGSI, DAN LIBRARY

- a. Variabel :
 - i. main function :
 1. struct akun *pUser : pointer to struct akun yang akan melalui *dynamic memory allocation* untuk menyimpan dan membuat akun sebanyak yang diinginkan user dengan menghemat memori
 2. int jumlah : menyimpan jumlah dari akun yang telah teregistrasi
 3. int input : meyimpan input sebagai pilihan user, type integer
 4. int yesorno : menyimpan ekspresi Boolean mengenai apakah ID dan password yang diinput oleh user sudah terdaftar dan sesuai atau belum
 5. int currentuser : menyimpan dan menspesifikasi akun yang mana yang akan digunakan dan dipassing ke tahap selanjutnya
 6. iint : sebagai counter dalam looping
 - ii. daftar function :
 1. int jumlah : menyimpan jumlah dari akun yang telah teregistrasi
 2. Struct akun *pUser : pointer to struct akun
 - iii. mainmenu function :
 1. int Input : meyimpan input sebagai pilihan user, type integer
 2. Struct akun *pUser : pointer to struct akun
 - iv. topup :
 1. int Input : meyimpan input sebagai pilihan user, type integer
 2. int Saldo : pointer to integer yaitu saldo yang merupakan elemen dari struct akun
 - v. helpdesk :
 1. int pilihan : menyimpan input sebagai pilihan user, type integer
 2. char temp : sebagai tempat penyimpanan sementara dalam mekanisme "press any key to continue"
 - vi. ojek :
 1. int *Saldo : pointer to integer yaitu saldo yang merupakan elemen dari struct akun
 2. int Input & input2 : menyimpan input sebagai pilihan user
 3. int Pickup : menyimpan input user sebagai tempat pick-up ojek
 4. int Destination : menyimpan input user untuk tujuan ojek
 5. char temp : sebagai tempat penyimpanan sementara dalam mekanisme "press any key to continue"
 6. float harga : menyimpan harga dari ojek online
 - vii. delay :
 1. int seconds : menyimpan jumlah detik untuk delay
 2. clock_t start_time : menyimpan waktu mulai untuk looping dan mensimulasikan *loading* dengan time delay
 - viii. spekun :
 1. int input, input2 & input3 : menyimpan input sebagai pilihan user
 2. int i : counter untuk looping
 - ix. displaySpekun :
 1. int i : counter untuk looping
 2. char temp : sebagai tempat penyimpanan sementara dalam mekanisme "press any key to continue"

- x. kereta :
 - 1. int tujuan : menyimpan input user mengenai tujuan kereta yang diinginkan
 - 2. int pilihan : menyimpan input pilihan user
- xi. displaykereta :
 - 1. char temp: sebagai tempat penyimpanan sementara dalam mekanisme "press any key to continue"
 - 2. int i : counter untuk looping
 - 3. int jumlahkereta : untuk mengatur jumlah kereta yang ada
 - 4. int tujuan : untuk mengatur array mana yang akan digunakan untuk didisplay (1 = jakarta, 2 = bogor)
- xii. bikun :
 - 1. int loc : menyimpan input user mengenai *location* yang diinginkan user
 - 2. int input : menyimpan input sebagai pilihan user
- xiii. jumlahbikun :
 - 1. int i, j : counter untuk looping
 - 2. int jumlahbikun : mengatur jumlah bikun yang terdapat di suatu lokasi
 - 3. char temp : sebagai tempat penyimpanan sementara dalam mekanisme "press any key to continue"
- b. Struct :
 - i. akun, untuk menyimpan data akun, elemennya :
 - 1. character array "id" yang kapasitasnya 25
 - 2. int pass
 - 3. int keberapa
 - 4. int saldo
 - ii. Spek, untuk menyimpan data jumlah spekun di suatu titik, elemennya :
 - 1. Int jmlSpekun
 - 2. Character array "nama" yang kapasitasnya 15
- c. Array :
 - i. ojek function :
 - 1. float jarak[9] : berisi jarak-jarak ojek dari user yang nantinya akan dipilih acak
 - ii. displayKereta function :
 - 1. int hour [19] : berisi jam kedatangan kereta yang nantinya akan dipilih acak
 - 2. int stamformasi [3] : berisi jumlah gerbong pada kereta yang akan dipilih acak
 - 3. int minute [9] : berisi menit kedatangan kereta tujuan Jakarta Kota yang akan dipilih acak
 - 4. int minute2[9] : berisi menit kedatangan kereta tujuan Bogor yang akan dipilih acak
 - iii. jumlahbikun function :
 - 1. int berapalamalagi [6] : berisi berapa menit lagi waktu bikun untuk sampai ke lokasi user yang isinya akan diacak antara 0 – 19
 - iv. spekun function :
 - 1. struct Spek spot[7] : menyimpan list nama spot dan jumlah spekun yang tersedia di masing-masing spot

d. Fungsi :

- i. main : untuk run C sebagai fungsi utama dan sebagai inisialisasi

```
Selamat datang di TransUI !

===== INISIALISASI =====

Masukkan angka dari opsi yang diinginkan :
1. Login
2. Register
3. Exit program
>
```

- ii. mainmenu : untuk menampilkan program inti setelah login

```
===== MAIN MENU =====

Selamat datang, haha

Masukkan angka dari opsi yang diinginkan :
1. BiKun
2. SpeKun
3. KRL
4. Ojek Online

5. Helpdesk
6. Topup Saldo
7. Log out
>
```

- iii. helpdesk : sebagai petunjuk pengoperasian dan penjelasan singkat mengenai program untuk user

```
===== HELPDESK =====

Program TransUI ini berfungsi untuk memudahkan mahasiswa UI bepergian di dalam ruang lingkup UI

Pilih bagian yang dibingungkan:
1) Inisialisasi
2) Main menu
3) Kembali ke main menu
> 2

Main Menu:
Pada tahap ini, user diberikan 4 pilihan mode transportasi: kereta, bikun, spekun, ojol
User juga diberikan opsi untuk top-up saldo yang dapat digunakan untuk membayar ojol dan opsi kembali ke inisialisasi
- Kereta: menampilkan jadwal kereta untuk stasiun UI
- Bikun: menampilkan jadwal bikun (bis kuning) pada setiap halte
- Spekun: menampilkan ketersediaan spekun (sepeda kuning) pada setiap lokasi dan opsi untuk meminjam/mengembalikan sepeda
- Ojol: memesan ojek online dengan memilih pick-up location dan destination (berbayar)

Press any key to continue . . . |
```

- iv. delay : memberikan sensasi *loading* dengan memberi buffer dalam bentuk detik untuk program menggunakan library time.h
- v. topup : mengisi ulang saldo user pada program

```
===== TOPUP SALDO =====

Saldo awal: Rp. 7750
Jumlah yang ingin ditopup: 20000

Saldo akhir: Rp. 27750

===== MAIN MENU =====
```

- vi. spekun : informasi mengenai sepeda kuning untuk bisa dipinjam dan dikembalikan
- vii. displayspekun : mendisplay jumlah spekun yang ada di setiap lokasi

```
===== SPEKUN =====

No.  Lokasi      Jumlah
1. Stasiun UI   : 20  sepeda
2. Stasiun Pocin : 10  sepeda
3. Perpustakaan : 6   sepeda
4. FKM          : 2   sepeda
5. FIB          : 19  sepeda
6. FMIPA        : 0   sepeda
7. FTeknik      : 10  sepeda

Press any key to continue . . . 1

1. Meminjam
2. Mengembalikan
3. Exit

Pilih operasi yang ingin dilakukan : 1
Pilih spot peminjaman : 4

Saldo anda tersisa Rp 10000

No.  Lokasi      Jumlah
1. Stasiun UI   : 20  sepeda
2. Stasiun Pocin : 10  sepeda
3. Perpustakaan : 6   sepeda
4. FKM          : 1   sepeda
5. FIB          : 19  sepeda
6. FMIPA        : 0   sepeda
7. FTeknik      : 10  sepeda

Press any key to continue . . . r

1. Meminjam
2. Mengembalikan
3. Exit
```

- viii. daftar : untuk menambah akun

```
===== REGISTER =====

Masukkan ID (no space / max. 25 character): haha
Masukkan password (angka): 123

===== INISIALISASI =====
```

- ix. kereta : menampilkan menu kereta di stasiun UI
- x. displaykereta : menampilkan jadwal kereta arah Jakarta Kota dan Bogor

```
===== KERETA =====  
  
JADWAL KRL COMMUTER LINE JABODETABEK  
  
Masukkan tujuan anda:  
  
1. Jakarta Kota  
2. Bogor  
  
Atau:  
3. Main Menu  
  
> 1  
  
Jadwal KRL tujuan Jakarta Kota  
-----  
  
Jam Kedatangan      Stamformasi  
4.20                12  
5.15                8  
6.57                12  
7.33                10  
8.20                12  
9.50                8  
10.50               12  
11.25               12  
12.45               10  
13.50               10  
14.50               8  
15.50               8  
16.57               10  
17.10               8  
18.15               10  
19.10               10  
20.20               10  
21.10               12  
22.50               10  
  
Press any key to continue . . . f
```

- xi. bikun : menampilkan menu kereta dan pilihan lokasi pick-up bikun (bis kuning)
- xii. jumlahbikun : menampilkan jumlah bikun yang sedang mendekati lokasi-lokasi tertentu dan berapa menit lagi sebelum sampai

```

===== BIKUN =====

1. Jadwal Bikun
2. Main Menu

> 1

Masukkan lokasi Anda

1. Stasiun UI
2. FT
3. Fasilkom
4. FK
5. FH

> 2

Jadwal kedatangan Bikun di FT

Terdapat 2 Bikun menuju tempat Anda:
- Bikun 1 akan sampai dalam 0 menit
- Bikun 2 akan sampai dalam 9 menit

Press any key to continue . . .

```

- xiii. ojek : menampilkan menu memesan ojek online

```

===== OJEK ONLINE =====

Saldo anda adalah: Rp. 10000
Jika ingin isi ulang, silakan kembali ke main-menu dan pilih opsi topup

1) Pesan Ojol
2) Main menu

> 1

Titik-titik ojek yang sudah ditentukan:
1. Stasiun UI
2. FH
3. Masjid
4. Balairung UI
5. RS UI
6. FMIPA
7. Perpus UI
8. Stadion
9. FT/FEB

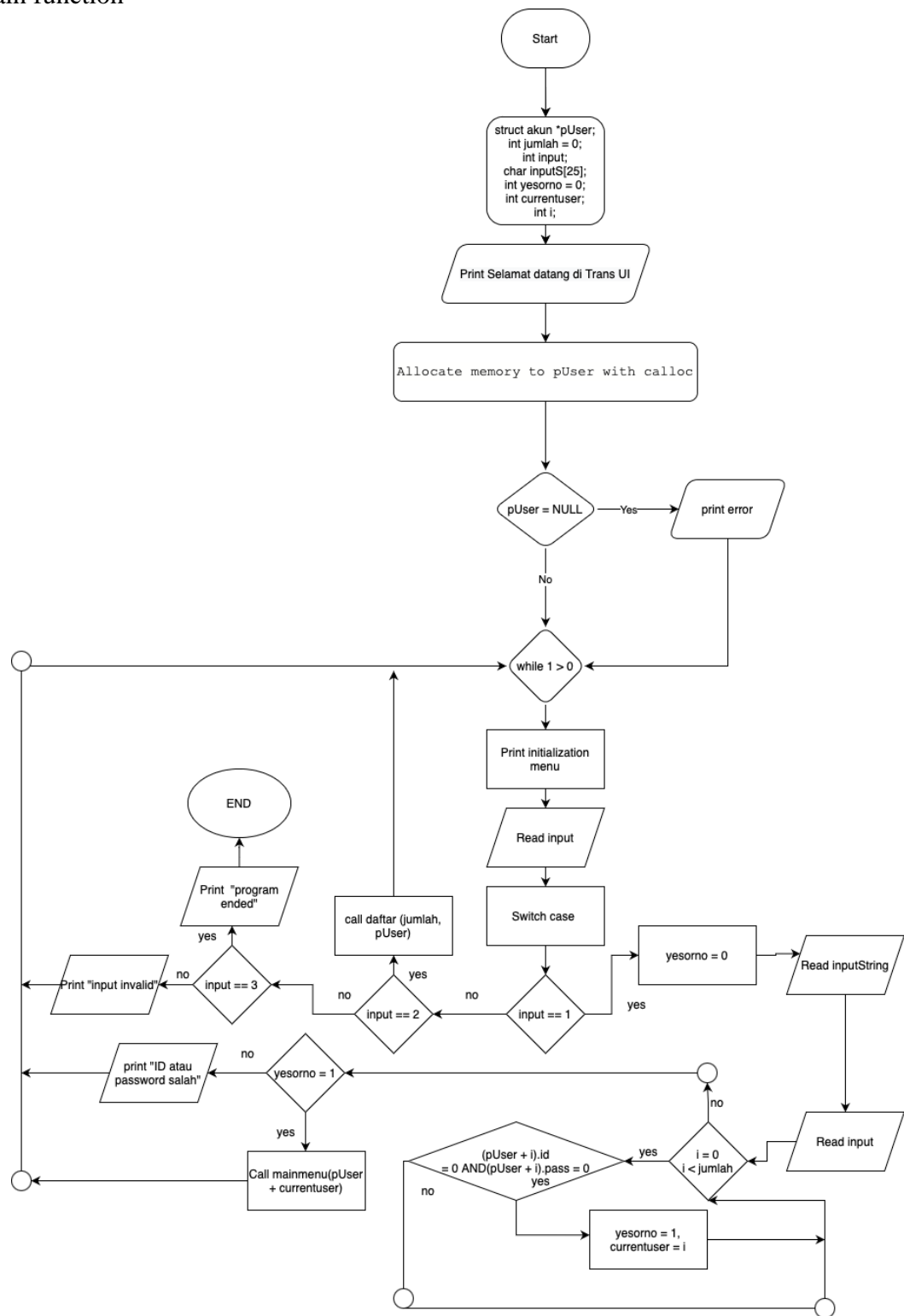
Ketik lokasi pick-up anda
> 1
Ketik destinasi anda
> 3
Harganya adalah Rp. 2250.00
Driver ditemukan!
Saldo anda sekarang adalah Rp. 7750! Silakan menunggu di lokasi pickup

Press any key to continue . . . |

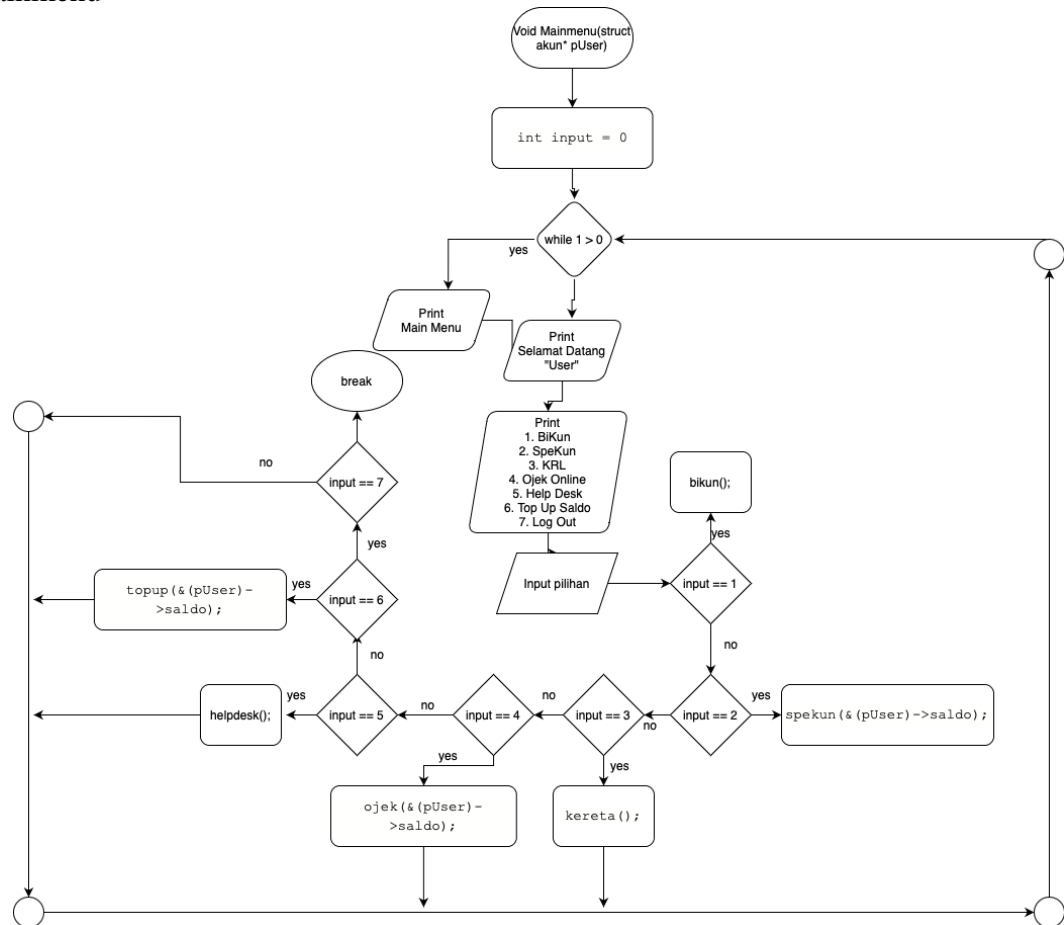
```


- e. Library :
 - i. Stdio.h : standard input output library untuk C seperti printf() dll.
 - ii. Stdlib.h : standard library yang digunakan untuk dinamic memory allocation
 - iii. String.h : memudahkan dalam menangani string seperti untuk fungsi strcpy()
 - iv. Time.h : membuat fungsi delay() untuk mensimulasikan adanya *loading* dan menggunakan fungsi srand untuk mendapatkan nilai integer random dalam rentang tertentu
 - v. Math.h : mendapatkan nilai absolut/mutlak dalam penentuan harga ojek online

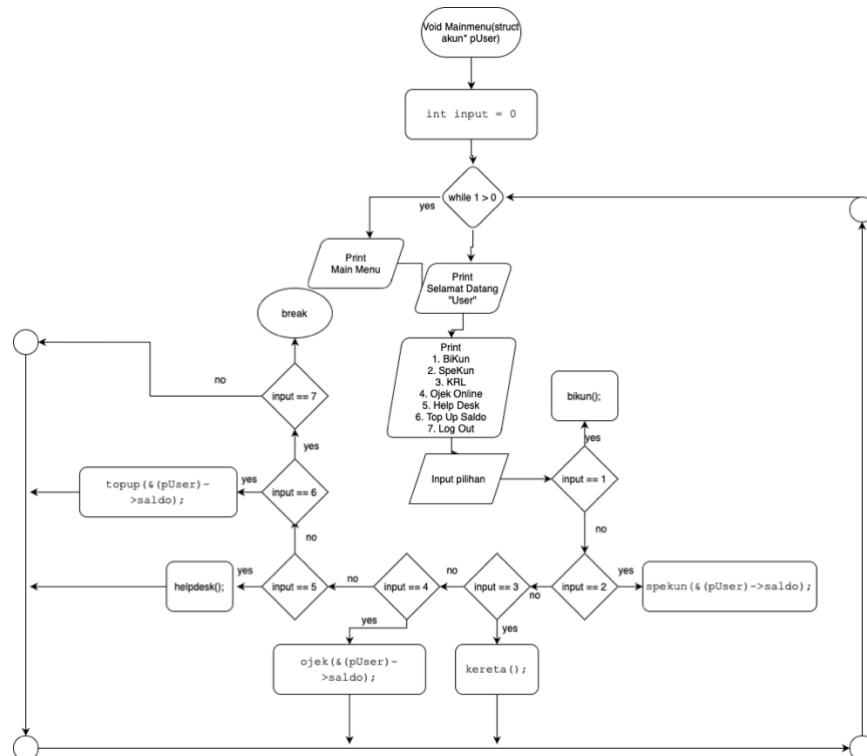
D) FLOWCHART
a. Main function



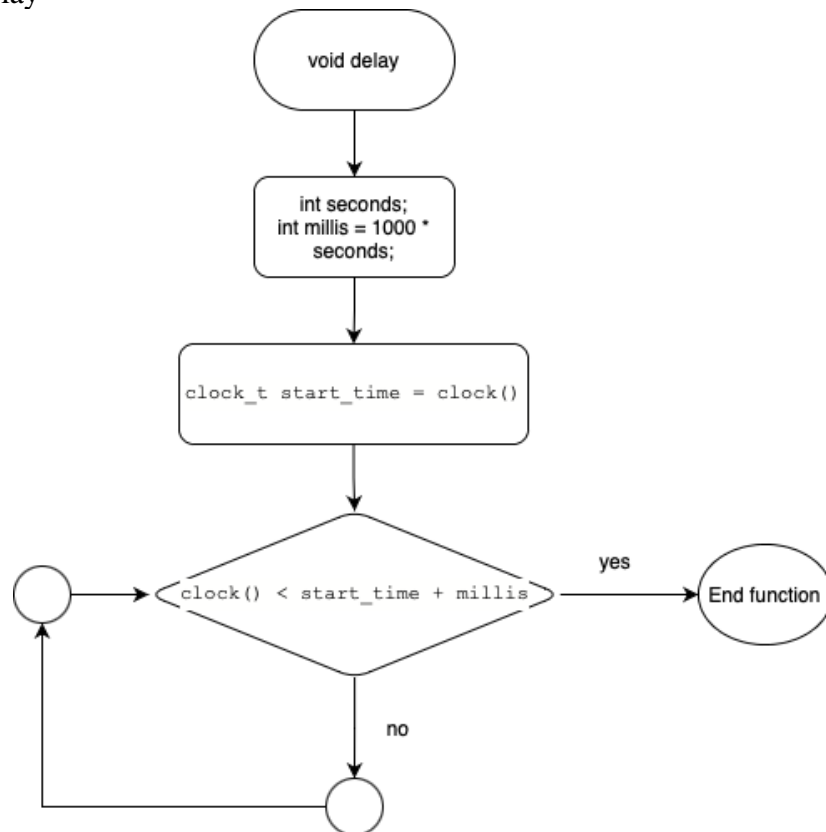
b. Mainmenu



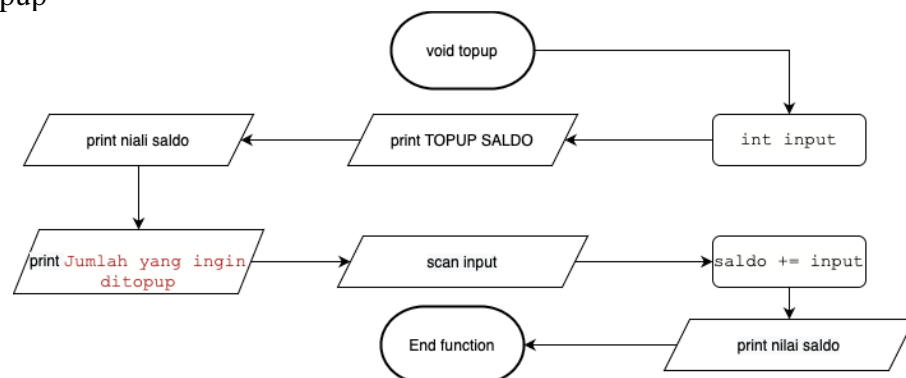
c. Helpdesk



d. Delay

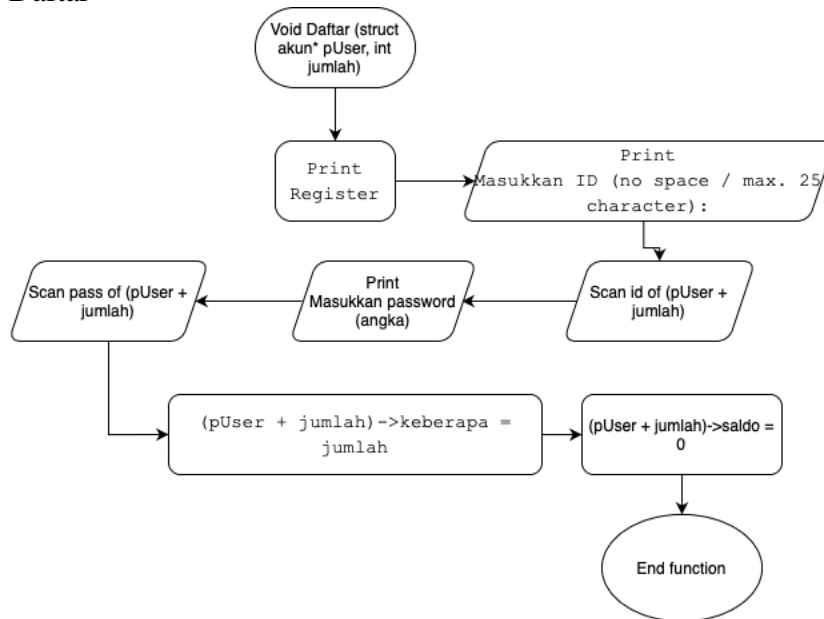


e. Topup

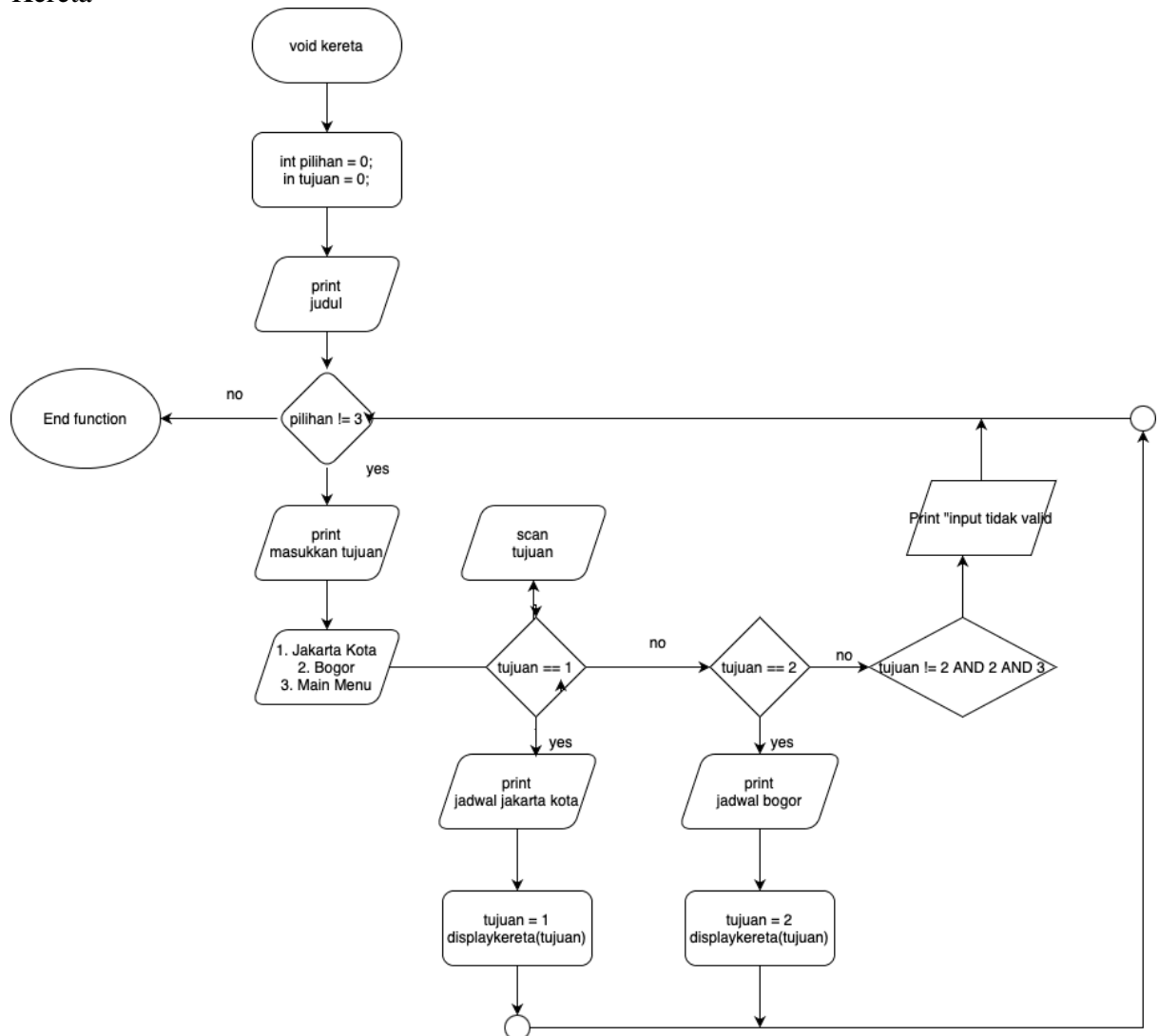


f. Spekun

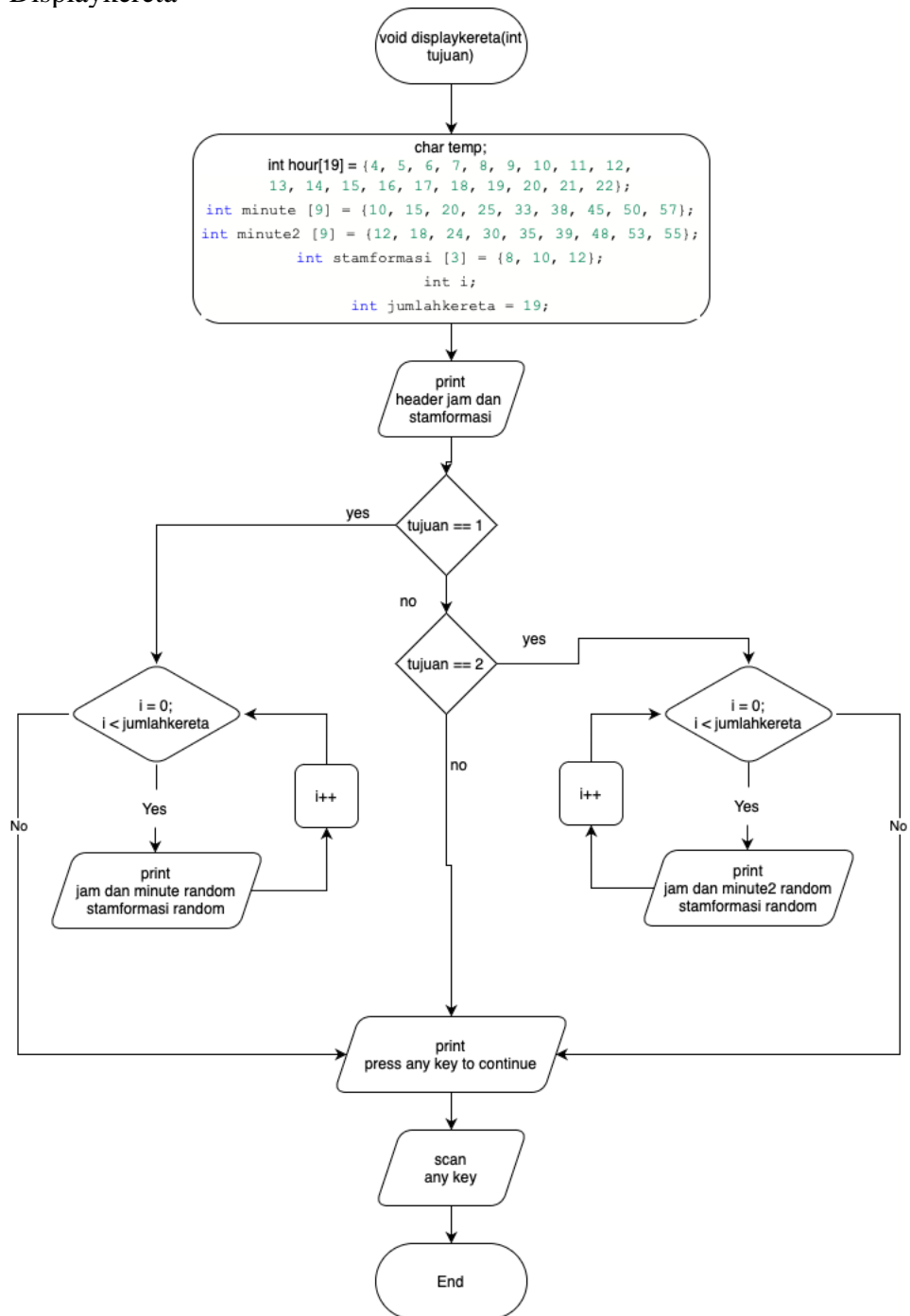
h. Daftar



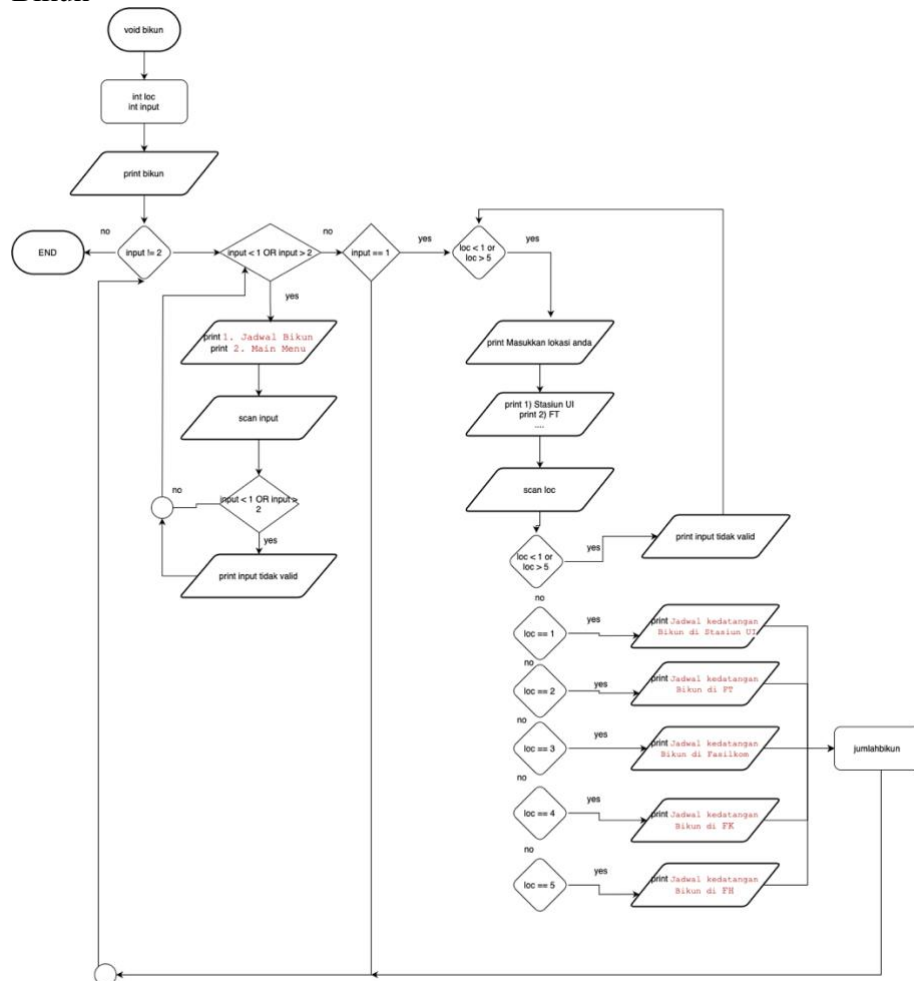
i. Kereta



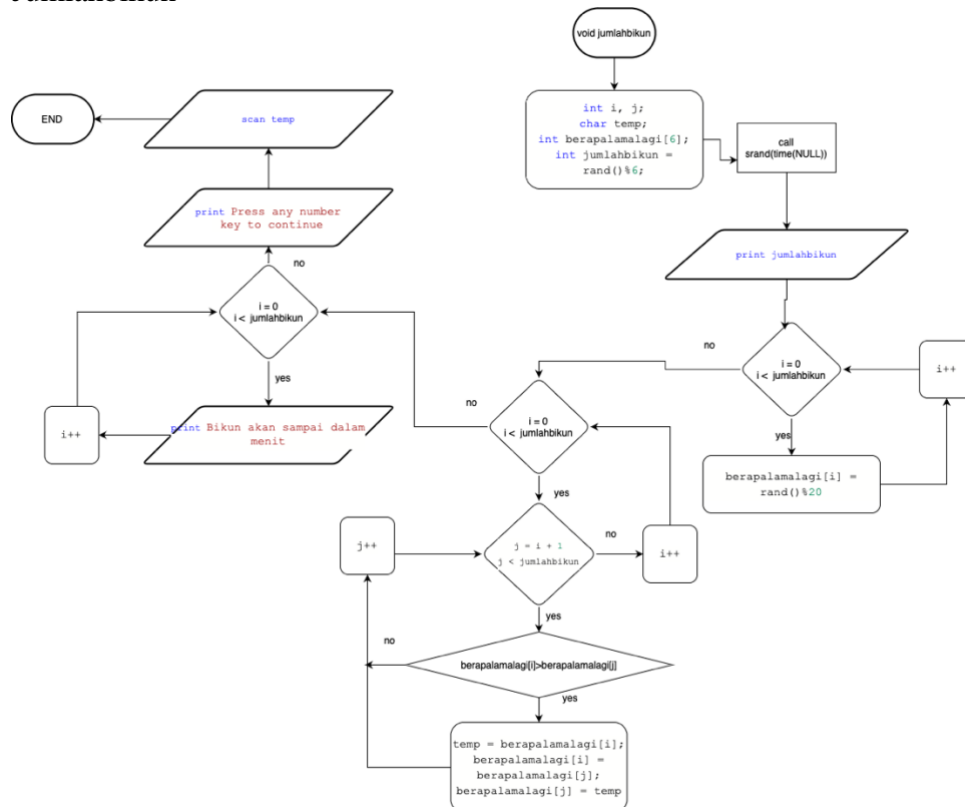
j. Displaykereta



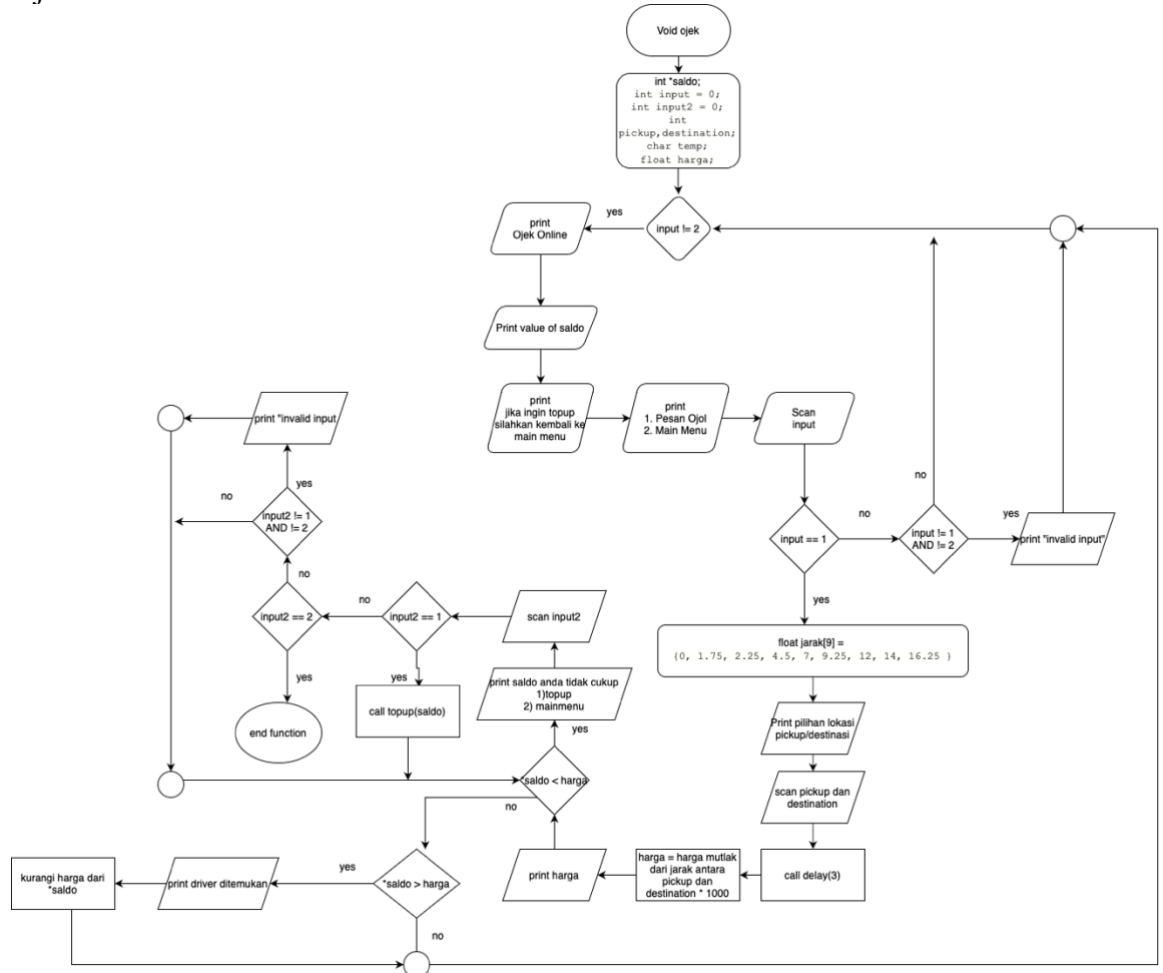
k. Bikun



l. Jumlahbikun



m. Ojek



E) PSEUDOCODE

Main function

- 1 **Declare** pointer to struct named pUser
- 2 **Initialize** jumlah to 0
- 3 **Initialize** input to 0
- 4 **Declare** character array named inputS
- 5 **Initialize** yesorno to 0
- 6 **Initialize** curretuser to 0
- 7 **Initialize** i to 0

- 8 **Print** "Selamat datang di TransUI"
- 9 **Allocate memory** to pUser with struct type akun
- 10 **If** pUser == NULL
- 11 **Print** "memory allocation failed"
- 12 **Endif**

- 13 **While** 1 is greater than 0
- 14 **Print** the initialization menu
- 15 **Read** input
- 16 **Case of** input
- 17 1 : **Set** yesorno to 0

```

18      Read inputString
19      Read input
20      For i = 0, i < jumlah, increment i
21          If id of pUser + I == 0 AND pass of pUser + I == 0
22              Set yesorno to 1
23              Set currentuser to i
24          Endif
25      Endfor
26      If yesorno == 1
27          Call mainmenu with pUser + currentuser
28      Else
29          Print "ID atau password salah"
30      2 : Increment jumlah
31          Reallocate memory to pUser
32          Call daftar with (jumlah-1) and pUser
33      3 : Print "program ended"
34      Return 0
35      Default : Print "input invalid"
36  Endcase
37 Endwhile

```

Topup function (int *saldo)

```

1  Initialize input
2  Print value of *saldo
3  Read input
4  Add input to *saldo
5  Print value of *saldo

```

Helpdesk function (void)

```

1  Set pilihan to 0
2  Initialize temp
3  Print brief explanation of helpdesk function
4  While pilihan is not equal to 3
5      Print "1. Initialization\n 2. Main Menu"
6      Read pilihan
7      If pilihan == 1
8          Print explanation about initialization
9      Else if pilihan == 2
10         Print explanation about main menu
11     Else if pilihan == 3
12         Break from loop
13     Else
14         Print "input tidak valid"
15     Endif
16 Endwhile

```

Daftar function (struct akun* pUser, int jumlah)

```
1  Read id of (pUser + jumlah)
2  Read pass of (pUser + jumlah)
3  Set beberapa of (pUser + jumlah) to jumlah
4  Set saldo of (pUser + jumlah) to 0
```

Mainmenu function (struct akun* pUser)

```
1  Initialize input to 0
2  While 1 > 0
3      Print "selamat datang" + id of pUser
4      Print "1. Bikun\n 2. Spekun\n3. KRL\n4. Ojek Online\n5. Helpdesk\n6. Topup Saldo\n7. Logout
5      Read pilihan
6      If pilihan == 1
7          Call bikun
8      Else If pilihan == 2
9          Call spekun
10     Else If pilihan == 3
11         Call kereta
12     Else If pilihan == 4
13         Call ojek with address of (saldo of pUser)
14     Else If pilihan == 5
15         Call helpdesk
16     Else If pilihan == 6
17         Call topup with address of (saldo of pUser)
18     Else If pilihan == 7
19         Break from looping
20     Else
21         Print "invalid input!"
22     Endif
23 Endwhile
```

Delay function (int seconds)

```
1  Initialize millis to seconds* 1000
2  Initialize start_time to clock() function
3  While clock() function is smaller than start_time + millis
4  Endwhile
```

Ojek function (int *saldo)

```
1  Initialize input and input2 to 0
2  Initialize pickup
3  Initialize destination
4  Initialize i
5  Initialize harga

6  While input is not equal to 2
7      Print value of *saldo
8      Print "1. Pesan Ojol\n2. Main menu"
9      Read input
10     If input == 1
11         Initialize jarak array to {0,1.75,2.25,4.5,7,9.25,12,14,16.25}
12         Print "1. Stasiun UI\n2. FH\n3. Masjid\n4. Balairung\n5. RS UI\n6. FMIPA\n7. Perpus UI
              8. Stadion\n9. FT/FEB"
```

```

13  Read pickup
14  Read destination
15  Call delay
16  Set harga to absolute value of (jarak[pickup-1] – jarak[destination-1]) * 1000
17  Print value of harga

18  While *saldo smaller than harga
19      Print “saldo anda tidak cukup”
20      Print “1. Topup\n2. Kembali”
21      Read input2
22      If input2 == 1
23          Call topup
24      Else if input2 == 2
25          Break from looping
26      Else if input2 != 1 AND != 2
27          Print “invalid input
28      Endif
29  Endwhile

30  If *saldo > harga
31      Print “Driver ditemukan”
32      Set *saldo to *saldo – harga
33  Endif
34  Else if input != 1 AND != 2
35      Print “invalid input!
36  Endif
37  Endwhile

```

Bikun function (void)

```

1  Initialize loc
2  Initialize input
3  While input is not equal 2
4      While input < 1 OR input > 2
5          Print “1. Jadwal bikun\n 2. Main menu”
6          Read input
7          If input < 1 OR input > 2
8              Print “input tidak valid”
9          Endif
10     Endwhile
11     If input == 1
12         While loc < 1 OR loc > 5
13             Print “1. Stasiun UI\n 2. FT\n 3. Fasilkom\n 4. FK\n5. FH
14             Read loc
15             If loc < 1 OR loc > 5
16                 Print “Input tidak valid”
17             Endif
18         Endwhile
19     If loc == 1
20         Print “Jadwal kedatangan bikun di stasiun UI”
21     If loc == 2
22         Print “Jadwal kedatangan bikun di FT”
23     If loc == 3
24         Print “Jadwal kedatangan bikun di Fasilkom”

```

```

25     If loc == 4
26         Print "Jadwal kedatangan bikun di stasiun FK"
27     If loc == 5
28         Print "Jadwal kedatangan bikun di stasiun FH"
29     Endif
30     Call jumlahbikun
31 Endif
32 Endwhile

```

Jumlahbikun function (void)

```

1  Initialize i
2  Initialize j
3  Initialize temp
4  Call srand with time(NULL)
5  Initialize berapalamalagi[6]
6  Initialize jumlahbikun to random value between 0 and 20
7  Print value of jumlahbikun
8  For i = 0, i < jumlahbikun, i++
9      Set berapalamalagi[i] to random value between 0 and 20
10 Endfor

11 For i = 0, i < jumlahbikun, i++
12     For j = i+1, j < jumlahbikun, j++
13         If berapalamalagi[i] > berapalamalagi[j]
14             Swap element i and element j using temp variable
15         Endif
16     Endfor
17 Endfor

18 For i=0, i < jumlahbikun, i++
19     Print "bikun i+1 sampai berapalamalagi[i] menit"
20 Endfor

```

Spekun function (int *saldo)

```

1  Declare integer named i, input, input2, input 3, and character temp
2  Declare spot[7] of type struct akun
3  Set spot[1].nama to "Stasiun UI"
4  Set spot[2].nama to "Stasiun Pocin"
5  Set spot[3].nama to "Perpustakaan"
6  Set spot[4].nama to "FKM"
7  Set spot[5].nama to "FIB"
8  Set spot[6].nama to "FMIPA"
9  Set spot[7].nama to "FTeknik"

10 For i = 0, i < 7, increment i
11     Set spot[i].jumlahspekun to random number between 0 - 20
12 Endfor
13 Print title of Spekun
14 Do While input not equal to 3
15     Call displaySpekun spot as argument
16     Print options menu "1. Meminjam, 2. Mengembalikan, 3. Exit"
17 Do While input < 1 OR input > 3

```

```

18 Print "Pilih operasi yang ingin dilakukan"
19 Read input
20 Case of input
21   1 : While input2 < 1 OR input 2 > 7 OR jmlSpekun of any spot < 0
22     For i = 0, i < 7, increment i
23       If jmlSpekun of spot number i < 0
24         increment spot[i].jmlSpekun
25       Endif
26     Endfor
27     If jmlSpekun of all spot == 0
28       Print "Tidak ada spekun yang tersedia"
29       Break fromc loop
30     Endif
31     Print "Pilih spot peminjaman :'"
32     Read input2
33     While *saldo < 5000
34       Print saldo , "saldo tidak mencukupi"
35       Print options menu "1. Topup, 2. Kembali"
36       Read input3
37       If input 3 == 1
38         Call topup with saldo as parameter
39       Else If input3 == 2
40         Break from loop
41       Endif
42     Endwhile
43     If *saldo > 5000
44       Decrement spot[input2 - 1].jmlSpekun
45       Subtract 5000 from *saldo
46     Endif
47     If input2 < 1 OR input2 > 7
48       Print "Pilihan anda tidak tersedia"
49     Endif
50   Endwhile

51   2 : While input2 < 1 OR input 2 > 7 OR jmlSpekun of any spot > 20
52     For i = 0, i < 7, increment i
53       If jmlSpekun of spot number i > 20
54         decrement spot[i].jmlSPekun
55       Print parkirane di spot ke i penuh
56     Endif
57   Endfor
58   If jmlSpekun of all spot == 20
59     Print "Parkiran spekun pada semua spot penuh"
60     Break from loop
61   Endif
62   Print "Pilih spot pengembalian :'"
63   Read input2
64   increment jmlSpekun of chosen spot
65   If input2 < 1 OR input2 > 7
66     Print "Pilihan anda tidak tersedia"
67   Endif
68   Endwhile
69   3 : Break from switch and function
70   Default: Print "Pilihan anda tidak tersedia"
71 Endwhile

```

```

72     If input = 1
73         Print saldo tersisa
74     Endif
75 Endwhile

```

displaySpekun (struct Spek spot[7])

```

1  Initialize i
2  Initialize temp
3  Print "No. Lokasi  Jumlah"
4  For i = 0, i < 7, increment i
5      Print i+1, nama of spot number i+1, jmlSpekun of spot number i+1
5  Endfor
6  Print "Press any key to continue"
7  Read temp

```

Kereta function (void)

```

1  Set pilihan and tujuan to 0
2  Print title of Kereta
3  Print "Jadwal KRL commuter line JABODETABEK"
4  While pilihan not equal to 3
5      Print "Masukkan tujuan anda :
6      Print "1. Jakarta Kota, 2. Bogor, Atau: 3. Main Menu"
7      Read tujuan
8      If tujuan = 1
9          Print "Jadwal KRL Tujuan Jakarta Kota"
10         Set tujuan to 1
11         Call displaykereta with tujuan
12     Else if tujuan = 2
13         Print "Jadwal KRL tujuan Bogor"
14         Set tujuan to 2
15         Call displaykereta with tujuan
16     Else if tujuan not equal to 1 AND 2 AND 3
17         Print "Input tidak valid!"
17     Endif
18 End While

```

displaykereta function (int tujuan)

```

1  Initialize temp
2  Initialize integer hour[19] = {4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22}
3  Initialize integer stamformasi[3] = {8,10,12}
4  Initialize integer minute[9] = {10,15,20,25,33,38,45,50,57}
5  Initialize integer minute[9] = {12,18,24,30,35,39,48,53,55}
6  Initialize i
7  Initialize jumlahKereta to 19
8  Print "Jam Kedatangan  Stamformasi"
9  If tujuan == 1
10     For i = 0, i < jumlahKereta, increment i
11         Print hour[i] and random element of minute
12         Print random element from stamformasi

```

```
13  Endfor
14  Else if tujuan == 2
15      For i = 0, i < jumlahKereta, increment i
16          Print hour[i] random element of minute2
17          Print random element from stamformasi
18      Endfor
19  Endif
20  Print "Press any number key to continue..."
21  Read temp
```

F) SOURCE CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>

struct akun {
    char id[25];
    int pass;
    int keberapa;
    int saldo;
};

struct Spek{
    int jmlSpekun;
    char nama[15];
};

void daftar(int jumlah, struct akun* );
void mainmenu(struct akun*);
void helpdesk(void);
void delay(int seconds);
void topup(int *saldo);

void ojek(int *saldo);

void spekun(int *saldo);
void displaySpekun(struct Spek spek spot[7]);

void kereta(void);
void displaykereta (int tujuan);

void bikun(void);
void jumlahbikun(void);

int main(void) {
    struct akun *pUser;
    int jumlah = 0;
    int input;
    char inputs[25];
    int yesorno = 0;
    int currentuser;
    int i;

    printf("Selamat datang di TransUI !\n");
    //allocate memory for first account
    pUser = (struct akun*) calloc (1, sizeof(struct akun));
    if (pUser == NULL)
        printf("\nMemory allocation failed");

    //while loop agar ngelooop selamanya sampai user memilih input 3 (program end)
    while (1 > 0){
        printf("\n===== INISIALISASI =====");
        printf("\n\nMasukkan angka dari opsi yang diinginkan :\n1. Login\n2. Register\n3. Exit program\n\n");
        scanf("%d", &input);
        switch (input){
            case 1:
                //meminta input ID dan password dari user untuk login
                yesorno = 0;
                printf("\n\nMasukkan ID: ");
                scanf("%s", inputs);
                printf("Masukkan password: ");
                scanf("%d", &input);
                //mengecek terdapat pasangan ID dan password yang sama dalam akun-akun yang sudah terdaftar
                for (i = 0; i < jumlah; i++){
                    if (strcmp(inputs, (pUser + i)->id) == 0 && input == (pUser + i)->pass){
```

```

        //jika iya, maka akun keberapanya akan disimpan dalam variabel currentuser
dan diteruskan ke mainmenu
        yesorno = 1;
        currentuser = i;
    }
}
if (yesorno == 1){
    mainmenu(pUser + currentuser);
} else
    printf("\nID atau password salah");
break;
case 2:
    //saat ingin menambahkan akun, jumlah akun total juga bertambah dan pUser
bertambah besar memorinya
    jumlah++;
    pUser = realloc (pUser, sizeof(struct akun) * (jumlah + 1));
    daftar(jumlah-1, pUser);
    break;
case 3:
    printf("\nProgram ended");
    //setelah program selesai, pointer dibebaskan
    free(pUser);
    return 0;
default:
    printf("\nInput invalid");
}
}
}

void daftar(int jumlah, struct akun* pUser ){
    //memasukkan informasi mengenai akun yang sedang didaftarkan
    printf("\n===== REGISTER =====");
    printf("\n\nMasukkan ID (no space / max. 25 character): ");
    scanf("%s", (pUser + jumlah)->id);
    printf("Masukkan password (angka): ");
    scanf("%d", &(pUser + jumlah)->pass);
    (pUser + jumlah)->keberapa = jumlah;
    (pUser + jumlah)->saldo = 0;
}

void mainmenu(struct akun* pUser){
    int input = 0;
    while (input != 7) {
        printf("\n\n===== MAIN MENU =====");
        //print ID dari akun yang sudah login
        printf("\n\nSelamat datang, %s", (pUser)->id);
        //print opsi main menu
        printf (" \n\nMasukkan angka dari opsi yang diinginkan :\n1. BiKun\n2. SpeKun\n3.
KRL\n4. Ojek Online\n5. Helpdesk");
        printf("\n6. Topup Saldo\n7. Log out\n\n> ");
        scanf ("%d", &input);

        if (input == 1){
            bikun();
        } else if (input == 2){
            spekun(&(pUser)->saldo);
        } else if (input == 3){
            kereta();
        } else if (input == 4){
            ojek(&(pUser)->saldo);
        } else if (input == 5){
            helpdesk();
        } else if (input == 6){
            topup(&(pUser)->saldo);
        } else if (input == 7){
            break;
        } else
            printf("\nInvalid input!");
    }
}
}

```

```

void topup(int *saldo){
    int input;
    //mengisi ulang saldo menggunakan passing by reference
    printf("\n===== TOPUP SALDO =====");
    printf("\n\nSaldo awal: Rp. %d", *saldo);
    printf("\nJumlah yang ingin ditopup: ");
    scanf("%d", &input);
    *saldo += input;
    printf("\nSaldo akhir: Rp. %d", *saldo);
}

void helpdesk(void){
    int pilihan = 0;
    char temp;
    printf("\n===== HELPDESK =====");
    printf("\n\nProgram TransUI ini berfungsi untuk memudahkan mahasiswa UI bepergian di
dalam ruang lingkup UI");
    while (pilihan != 3){
        //print pilihan menu
        printf("\n\nPilih bagian yang dibingungkan: ");
        printf("\n1) Inisialisasi ");
        printf("\n2) Main menu ");
        printf("\n3) Kembali ke main menu\n\n> ");
        scanf("%d", &pilihan);

        if (pilihan == 1){
            printf("\nInisialisasi:");
            printf("\nPada tahap ini, user diberikan 3 pilihan: login, register,          end
program ");
            printf("\n - Untuk masuk ke program inti, user harus login dengan memasukkan ID
dan password akunnya");
            printf("\n - Jika belum memiliki akun, user dapat daftar dengan memilih
register");
            printf("\n - Untuk mengakhiri program, user dapat memilih exit program");
            printf("\n\nPress any key to continue . . . ");
            scanf(" %c", &temp);
        } else if (pilihan == 2){
            printf("\nMain Menu: ");
            printf("\nPada tahap ini, user diberikan 4 pilihan mode transportasi: kereta,
bikun, spekun, ojol");
            printf("\nUser juga diberikan opsi untuk top-up saldo yang dapat digunakan untuk
membayar ojol dan opsi kembali ke inisialisasi");
            printf("\n - Kereta: menampilkan jadwal kereta untuk stasiun UI");
            printf("\n - Bikun: menampilkan jadwal bikun (bis kuning) pada setiap halte");
            printf("\n - Spekun: menampilkan ketersediaan spekun (sepeda kuning) pada setiap
lokasi ");
            printf("\ndan opsi untuk meminjam/mengembalikan sepeda");
            printf("\n - Ojol: memesan ojek online dengan memilih pick-up location dan
destination (berbayar)");
            printf("\n\nPress any key to continue . . . ");
            scanf(" %c", &temp);
        } else if (pilihan == 3){
            break;
        } else if (pilihan != 1 && pilihan != 2 && pilihan != 3)
            printf("\nInput tidak valid");
        }
    }

void ojek(int *saldo){
    int input = 0;
    int input2 = 0;
    int pickup, destination;
    char temp;
    float harga;
    while (input != 2){
        printf("\n===== OJEK ONLINE =====");
        printf("\n\nSaldo anda adalah: Rp. %d", *saldo);
        printf("\nJika ingin isi ulang, silakan kembali ke main-menu dan pilih opsi topup");
        //menampilkan menu dari function ojek online

```

```

printf("\n\n1) Pesan Ojol");
printf("\n2) Main menu\n\n> ");
scanf("%d", &input);

if (input == 1){
    //array yang berfungsi untuk perhitungan harga, anggap elemen pertama adalah
    //semacam kilometer 0 seperti dalam jalan tol
    float jarak[9] = {0, 1.75, 2.25, 4.5, 7, 9.25, 12, 14, 16.25 };

    printf("\n\nTitik-titik ojek yang sudah ditentukan:");
    printf("\n1. Stasiun UI\n2. FH \n3. Masjid\n4. Balairung UI\n5. RS UI");
    printf("\n6. FMIPA\n7. Perpus UI\n8. Stadion\n9. FT/FEB");
    printf("\n\nKetik lokasi pick-up anda\n> ");
    scanf("%d", &pickup);
    printf("Ketik destinasi anda\n> ");
    scanf("%d", &destination);
    //function delay agar mendapatkan sensasi loading
    delay(3);
    //perhitungan harga adalah dengan mencari jarak antara lokasi pickup dan destinasi
    //kemudian dikali dengan 1000
    harga = fabsf(jarak[pickup-1] - jarak[destination-1]) * 1000;
    printf("Harganya adalah Rp. %.2f", harga );

    //kalau saldonya tidak cukup, akan diminta untuk memilih antara topup atau kembali
    //ke menu
    while (*saldo < harga){
        printf("\nSaldo anda tidak cukup!");
        printf("\n1. Topup\n2. Kembali\n\n> ");
        scanf("%d", &input2);
        if (input2 == 1)
            topup(saldo);
        else if (input2 == 2)
            break;
        else if (input2 != 1 && input2 != 2)
            printf("\nInvalid input!");
    }
    //kalau saldonya cukup, akan langsung menemukan driver
    if (*saldo > harga){
        printf("\nDriver ditemukan! ");
        *saldo -= harga;
        printf("\nSaldo anda sekarang adalah Rp. %d! Silakan menunggu di lokasi pickup",
        *saldo);
        printf("\n\nPress any key to continue . . . ");
        scanf(" %c", &temp);
    }
    else if (input != 1 && input != 2)
        printf("\nInvalid input!");
    }
}

void delay(int seconds){
    //function berfungsi untuk memberikan sensasi loading
    int millis = 1000 * seconds;
    clock_t start_time = clock();
    while (clock() < start_time + millis) ;
}

void spekun(int *saldo){
    int i, input, input2, input3;

    struct Spek spot[7];
    //Pemberian nama untuk masing-masing spot
    strcpy(spot[0].nama, "Stasiun UI");
    strcpy(spot[1].nama, "Stasiun Pocin");
    strcpy(spot[2].nama, "Perpustakaan");
    strcpy(spot[3].nama, "FKM");
    strcpy(spot[4].nama, "FIB");
    strcpy(spot[5].nama, "FMIPA");
    strcpy(spot[6].nama, "FTeknik");

```

```

//randomisasi jumlah spekun pada setiap spot antara 0 - 20
for (i = 0; i < 7; i++){
    spot[i].jmlSpekun = (rand() % (21));
}
//judul spekun
printf("\n\n===== SPEKUN =====");

do{//menggunakan while loop agar program dapat berulang dengan sentinel 3 untuk exit
    displaySpekun(spot);//function yang akan menampilkan list nama-nama spot dan jumlah
    spekun pada spot
    printf("\n1. Meminjam");
    printf("\n2. Mengembalikan");
    printf("\n3. Exit");
    do{//menggunakan while untuk error handling jika user memasukkan input selain 1/2/3
        printf("\n\nPilih operasi yang ingin dilakukan : ");
        scanf("%d", &input);
        switch(input){
            case 1 :
                do{
                    //menggunakan for loop dan if untuk mengecek apakah ada spot yang kosong agar
                    user tidak bisa memilih spot tersebut
                    for (i = 0; i < 7; i++){
                        if(spot[i].jmlSpekun < 0) {
                            jumlah minus
                            spot[i].jmlSpekun++;//jumlah diincrement kembali agar tidak menampilkan
                            printf("Spekun di spot yang anda pilih sudah habis, silahkan memilih spot
                            lain\n\n");
                        }
                    }
                    //mengecek jika semua spot sudah kosong agar user sudah tidak bisa meminjam
                    lagi dan mengeluarkan dari loop
                    if (spot[0].jmlSpekun == 0 && spot[1].jmlSpekun == 0 && spot[2].jmlSpekun ==
                    0 &&
                    spot[3].jmlSpekun == 0 && spot[4].jmlSpekun == 0 && spot[5].jmlSpekun == 0
                    &&
                    spot[6].jmlSpekun == 0){
                        printf("Tidak ada spekun yang tersedia");
                        break;
                    }
                    printf("Pilih spot peminjaman : ");
                    scanf("%d", &input2);

                    while (*saldo < 5000){//menggunakan while agar jika saldo tidak mencukupi
                    maka user harus topup
                        printf("\n\nSaldo anda sebanyak Rp. %d", *saldo);
                        printf("\nSaldo anda tidak mencukupi!");
                        printf("\n1. Topup\n2. Kembali\n\n> ");
                        scanf("%d", &input3);
                        if (input3 == 1)
                            topup(saldo);//menjalankan function topup untuk menambah saldo
                        else if (input3 == 2)
                            break;//akan keluar dari loop
                    }
                    if (*saldo > 5000){//jika saldo mencukupi maka peminjaman akan berjalan
                    dikurangi
                        spot[input2-1].jmlSpekun--;//jumlah spekun pada spot yang dipilih akan
                        *saldo -= 5000;//saldo akan dipotong sebanyak 5000
                    }
                    if(input2 < 1 || input2 > 7)//Jika input yang dimasukkan bukan antara 1-7
                    maka akan menampilkan warning
                        printf("Pilihan anda tidak tersedia\n");
                    //error handling jika input tidak di antara 1-7 dan jika jika spekun spot
                    yang dipilih sudah habis
                } while ( input2 < 1 || input2 > 7 || spot[0].jmlSpekun < 0 ||
                spot[1].jmlSpekun < 0 || spot[2].jmlSpekun < 0 || spot[3].jmlSpekun < 0 | spot[4].jmlSpekun
                < 0 || spot[5].jmlSpekun < 0 ||
                spot[6].jmlSpekun < 0 );
                break;
            case 2 :

```

```

do{
    //menggunakan for loop dan if untuk mengecek apakah ada spot yang penuh agar
    user tidak bisa memilih spot tersebut
    for (i = 0; i < 7; i++){
        if(spot[i].jmlSpekun > 20){
            spot[i].jmlSpekun--;
            printf("Parkiran spekun di spot yang anda pilih sudah penuh, silahkan
mengembalikan di spot lain\n\n");
        }
        //mengecek jika semua spot sudah penuh (max 20) agar user sudah tidak bisa
        mengembalikan lagi dan mengeluarkan dari loop
        if (spot[0].jmlSpekun == 20 && spot[1].jmlSpekun == 20 && spot[2].jmlSpekun
== 20 &&
        spot[3].jmlSpekun == 20 && spot[4].jmlSpekun == 20 && spot[5].jmlSpekun
== 20 &&
        spot[6].jmlSpekun == 20){
            printf("Parkiran spekun pada semua spot sudah terisi penuh\n"); break;
        }
        printf("Pilih spot pengembalian (Maksimal spekun pada 1 spot adalah 20) :
");
        scanf("%d", &input2);
        spot[input2-1].jmlSpekun++; //jumlah spekun pada spot yang dipilih akan
        bertambah
        if(input2 < 1 || input2 > 7) //error handling jika input tidak di antara 1-7
            printf("Pilihan anda tidak tersedia\n");
        //error handling jika spot sudah penuh
        } while ( input2 < 1 || input2 > 7 || spot[0].jmlSpekun > 20 ||
spot[1].jmlSpekun > 20 || spot[2].jmlSpekun > 20 || spot[3].jmlSpekun > 20 ||
spot[4].jmlSpekun > 20 || spot[5].jmlSpekun > 20 || spot[6].jmlSpekun > 20 );
        break;
        case 3:
            break; //keluar dari menu spekun
        default : printf("Pilihan anda tidak tersedia\n");
    }
    } while (input < 1 || input > 3);
    if (input == 1) { //jika melakukan peminjaman maka akan menampilkan sisa saldo
        printf("\nSaldo anda tersisa Rp %d\n", *saldo);
    }
} while (input != 3);
}

void displaySpekun(struct Spek spot[7]){
    int i;
    char temp;
    printf("\n\nNo. Lokasi          Jumlah\n"); //menampilkan indeks tabel
    for (i = 0; i < 7; i++){
        //menampilkan nama dan jumlah spekun dari masing-masing spot
        printf("%d. %s : %d\tsepeda\n", i+1, spot[i].nama, spot[i].jmlSpekun);
    }
    printf("\nPress any key to continue . . . "); //sebagai delay agar program tidak
    langsung dilanjutkan, user perlu memasukkan input apapun untuk melanjutkan program
    scanf(" %c", &temp);
}

void kereta(void){
    int pilihan = 0;
    int tujuan;
    printf("\n\n===== KERETA =====\n\n");
    printf("JADWAL KRL COMMUTER LINE JABODETABEK\n"); //header menu kereta

    while (pilihan != 3){
        printf("\nMasukkan tujuan anda: \n\n");
        printf("1. Jakarta Kota\n2. Bogor\n\nAtau:\n3. Main Menu\n\n> ");
        scanf("%d", &pilihan); //meminta user input
        if (pilihan == 1) { //jika user input = 1
            printf("\nJadwal KRL tujuan Jakarta Kota\n-----\n\n");
            tujuan = 1; //set variabel tujuan = 1
            displaykereta(tujuan); //memanggil func displaykereta
        }
    }
}

```

```

else if (pilihan == 2) { //jika user input =2
    printf ("\nJadwal KRL tujuan Bogor\n-----\n");
    tujuan = 2; //set variabel tujuan = 2
    displaykereta(tujuan); //memanggil func displaykereta
}
else if (pilihan != 1 && pilihan != 2 && pilihan != 3) {
    printf ("Input tidak valid!"); //error message jika input tidak valid
}
}
}

void displaykereta(int tujuan) {
    char temp;
    int hour [19] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22};
    int stamformasi [3] = {8, 10, 12};
    int i;
    int minute [9] = {10, 15, 20, 25, 33, 38, 45, 50, 57};
    int minute2 [9] = {12, 18, 24, 30, 35, 39, 48, 53, 55};
    int jumlahkereta = 19;
    printf ("\nJam Kedatangan\t\tStamformasi"); //header timetable kedatangan kereta

    if (tujuan == 1) { //jika variabel tujuan = 1
        for (i = 0; i < jumlahkereta; i++) {
            printf ("\n%6d.%d", hour[i], minute[rand() % 9]); //print jam dan menit
            printf ("\t\t%6d", stamformasi [rand() % 3]); //print rangkaian / stamformasi
        }
    } else if (tujuan == 2) { //jika variabel tujuan = 2
        for (i = 0; i < jumlahkereta; i++) {
            printf ("\n%6d.%d", hour[i], minute2[rand() % 9]); //print jam dan menit
            printf ("\t\t%6d", stamformasi [rand() % 3]);
        } //print rangkaian / stamformasi
    }
    printf ("\n\nPress any key to continue . . . ");
    scanf(" %c", &temp);
}

void bikun(void) {
    int loc;
    int input;

    printf("\n===== BIKUN ====="); //judul bikun

    while(input != 2) {
        do { //main menu dari bikun
            printf("\nn1. Jadwal Bikun\nn2. Main Menu\n\n> ");
            scanf("%d", &input); //mengscan pilihan user
            if (input < 1 || input > 2)
                printf("\nInput tidak valid!"); //apabila input selain 1 dan 2
        } while (input < 1 || input > 2);

        if (input == 1) {
            do {
                printf("\nMasukkan lokasi Anda\n");
                printf("\nn1. Stasiun UI\nn2. FT\nn3. Fasilkom\nn4. FK\nn5. FH\n\n> ");
                scanf("%d", &loc); //program akan meminta user untuk memasukkan lokasi
                if (loc < 1 || loc > 5)
                    printf("\nInput tidak valid!"); //apabila user menginput diluar 1-5 maka akan
                    diprint input tidak valid dan kembali ke menu pilihan
            } while (loc < 1 || loc > 5); //program akan print lokasi yang use pilih dan
            melanjutkannya ke function jumlahbikun

            if (loc == 1)
                printf("\nJadwal kedatangan Bikun di Stasiun UI");
            else if (loc == 2)
                printf("\nJadwal kedatangan Bikun di FT");
            else if (loc == 3)
                printf("\nJadwal kedatangan Bikun di Fasilkom");
            else if (loc == 4)
                printf("\nJadwal kedatangan Bikun di FK");
            else if (loc == 5)

```

```

        printf("\nJadwal kedatangan Bikun di FH");
    }
    jumlahbikun();
}

void jumlahbikun(){
    int i, j;
    char temp;
    srand(time(NULL));
    int berapalamalagi[6];
    int jumlahbikun = rand()%6;

    printf("\n\nTerdapat %d Bikun menuju tempat Anda:", jumlahbikun); //header dari jumlah
    dan waktu bikun yang akan sampai kepada user

    for (i = 0; i < jumlahbikun; i++){
        berapalamalagi[i] = rand()%20; //akan merandom waktu bikun yang akan sampai pada user
    }
    //algoritma penyortir agar menit jumlahbikun dimulai dari menit terkecil ke menit
    terbesar
    for (i = 0; i < jumlahbikun; i++){
        for (j = i + 1; j < jumlahbikun; j++){
            if (berapalamalagi[i] > berapalamalagi[j]){
                temp = berapalamalagi[i];
                berapalamalagi[i] = berapalamalagi[j];
                berapalamalagi[j] = temp;
            }
        }
    }
    for(i = 0; i < jumlahbikun; i++){
        printf("\n -   Bikun %d akan sampai dalam %d menit", i + 1, berapalamalagi[i]);
    }
    printf("\n\nPress any key to continue . . . ");
    scanf(" %c", &temp); //delay program jadi tidak langsung pindah dan memberi jeda pada
    program
}

```