

Nama: Dimas Arifil Azizi

NIM: 120320097

Kelas: IF 02 01

Tugas 1#2

PESPEC KATEGORI PAGE

1. Penjelasan Fitur

Fitur kategori adalah bagian dari program yang memungkinkan pengguna untuk mengelola dan mengatur berbagai kategori data. Pada fitur ini biasanya digunakan untuk mengelompokkan item, informasi barang, dan suatu data tertentu yang lebih mudah diorganisir.

2. Inputs :

Pada inputan fitur pengguna bisa memilih beberapa fungsi seperti :

- Menambahkan Kategori baru
Pada fungsi ini Pengguna bisa menambahkan kategori barang sesuai yang ingin dibutuhkan.
- Mengubah Kategori yang ada
Pada fungsi ini Pengguna bisa mengubah kategori yang sudah di tambahkan menjadi kategori baru.
- Menghapus kategori
Pada fungsi ini Pengguna bisa menghapus kategori yang tidak perlu pada daftar kategori.
- Mencari Kategori
Pada fungsi ini pengguna bisa mencari kategori yang di inginkan pengguna sesuai kebutuhannya.

3. Outputs

Pada outputan fitur ini hasil dari inputan pengguna seperti :

- Menambahkan Kategori baru
Output pada fungsi ini menampilkan kategori barang yang sudah di tambahkan oleh pengguna.
- Mengubah Kategori yang ada
Outputan pada fungsi ini pengguna bisa megubah kategori barang yang sudah ditambahkan pada daftar kategori.
- Menghapus kategori
Outputan pada fungsi ini pengguna bisa menghapus suatu kategori yang sudah ditambahkan pada daftar kategori.
- Mencari Kategori
Pada fungsi ini pengguna bisa mencari kategori barang yang sudah ditambahkan pada daftar kategori.

4. Preconditions :

Pada Preconditions pengguna harus mengisi setiap fitur dengan benar agar fungsi berjalan dengan benar.

- Menambahkan Kategori baru
 - Pada menambahkan kategori baru harus sesuai dengan kapasitas barang. Jika tidak, fungsi ini tidak akan dapat menambah kategori baru.
 - Pengguna harus memasukkan nama kategori yang valid (tidak kosong).
 - Tidak bisa menambahkan nama kategori yang sudah ada pada daftar kategori.
- Mengubah Kategori yang ada
 - Setidaknya satu kategori harus ada atau lebih kategori barang agar bisa diedit.
 - Nama kategori yang dimasukkan harus sudah ada dalam daftar kategori. Jika tidak ada, tidak ada yang bisa diedit.
- Menghapus kategori
 - Setidaknya satu kategori harus ada atau lebih kategori barang agar bisa dihapus.
 - Nama kategori yang dimasukkan harus ada dalam daftar kategori. Jika tidak ada, tidak ada yang bisa dihapus.
- Mencari Kategori
 - Setidaknya satu kategori harus ada atau lebih kategori barang agar bisa dicari.
 - Nama kategori yang dicari harus tidak kosong.

5. Postconditions

Pada Postconditions ini pengguna saat menginputan suatu kategori harus benar setelah sebuah fungsi dijalankan.

- Menambahkan Kategori baru
 - Kategori baru telah ditambahkan ke dalam categories, dan categoryCount telah bertambah satu.
 - Status kategori yang ditambahkan adalah "Aktif".
 - Jika penambahan berhasil, tidak ada perubahan pada kategori yang sudah ada.
- Mengubah kategori yang ada
 - Jika kategori dengan nama yang dimasukkan ditemukan, nama kategori tersebut telah diperbarui.
 - categoryCount tetap sama.
 - Jika tidak ditemukan, tidak ada perubahan pada categories.
- Menghapus kategori
 - Jika kategori dengan nama yang dimasukkan ditemukan, kategori tersebut telah dihapus dari categories, dan semua kategori setelahnya telah digeser satu posisi ke depan. categoryCount telah berkurang satu.
 - Jika tidak ditemukan, tidak ada perubahan pada categories.
- Mencari kategori
 - Jika kategori ditemukan, informasi kategori (nama dan status) ditampilkan.
 - Jika kategori tidak ditemukan, informasi kategori (nama dan status) tidak muncul.
 - Tidak ada perubahan pada categories dan categoryCount.

6. Proses logic

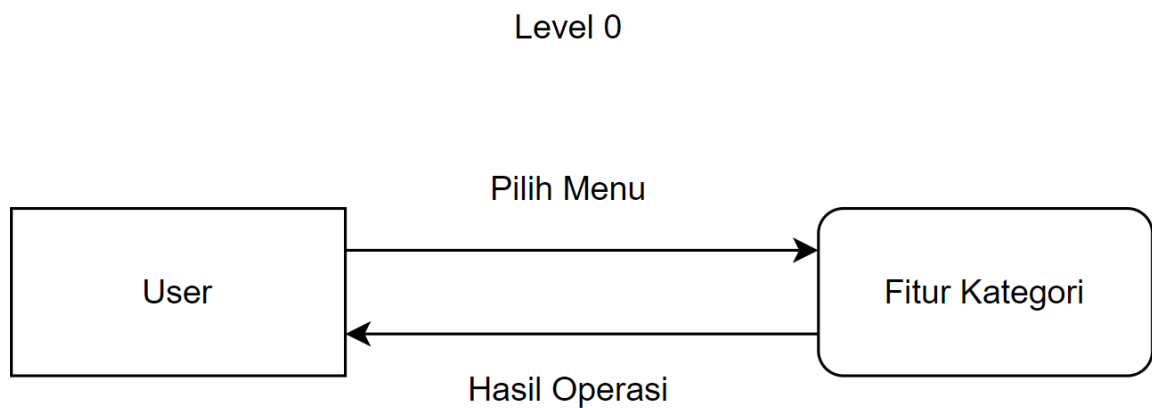
Pada proses logic ini bertujuan untuk mengelola kategori dengan menyediakan fungsi untuk menambah, mengubah, menghapus, dan mencari kategori barang.

- a. Fungsi menambahkan kategori baru
 - Receive input:

- Menerima input dari pengguna berupa nama kategori.
- Validation:
 - Memeriksa apakah categoryCount sudah mencapai batas kategori.
 - Memastikan nama kategori yang dimasukkan tidak kosong.
- Output:
 - Menampilkan pesan bahwa kategori berhasil ditambahkan atau pesan kesalahan jika tidak dapat menambah kategori.
- Data Store:
 - Menyimpan kategori baru di dalam categories dan meningkatkan categoryCount.
- Error Handling:
 - Menampilkan pesan kesalahan jika batas kategori telah tercapai.
- b. Fungsi mengubah kategori yang ada
 - Receive Input:
 - Menerima input dari pengguna berupa nama kategori yang ingin diubah dan nama kategori baru.
 - Validation:
 - Memeriksa apakah categoryCount lebih dari 0.
 - Memeriksa apakah nama kategori yang ingin diubah ada dalam daftar kategori.
 - Output:
 - Menampilkan pesan bahwa kategori berhasil diubah atau pesan kesalahan jika kategori tidak ditemukan.
 - Data Store:
 - Memperbarui nama kategori yang ada dalam categories.
 - Error Handling:
 - Menampilkan pesan kesalahan jika tidak ada kategori yang tersedia untuk diubah atau jika kategori tidak ditemukan.
- c. Fungsi hapus kategori
 - Receive Input:
 - Menerima input dari pengguna berupa nama kategori yang ingin dihapus.
 - Validation:
 - Memeriksa apakah categoryCount lebih dari 0.
 - Memeriksa apakah nama kategori yang ingin dihapus ada dalam daftar kategori.
 - Output:
 - Menampilkan pesan bahwa kategori berhasil dihapus atau pesan kesalahan jika kategori tidak ditemukan.
 - Data Store:
 - Menggeser kategori setelah kategori yang dihapus satu posisi ke kiri dan mengurangi categoryCount.
 - Error Handling:
 - Menampilkan pesan kesalahan jika tidak ada kategori yang tersedia untuk dihapus atau jika kategori tidak ditemukan.
- d. Fungsi cari kategori
 - Receive Input:

- Menerima input dari pengguna berupa nama kategori yang ingin dicari.
- Validation:
 - Memeriksa apakah categoryCount lebih dari 0.
- Output:
 - Menampilkan informasi kategori yang ditemukan atau pesan bahwa kategori tidak ditemukan.
- Data Store:
 - Tidak ada perubahan pada data, hanya melakukan pencarian dalam array categories.
- Error Handling:
 - Menampilkan pesan kesalahan jika tidak ada kategori yang tersedia untuk dicari atau jika kategori tidak ditemukan.

7. DFD (Data flow diagram)

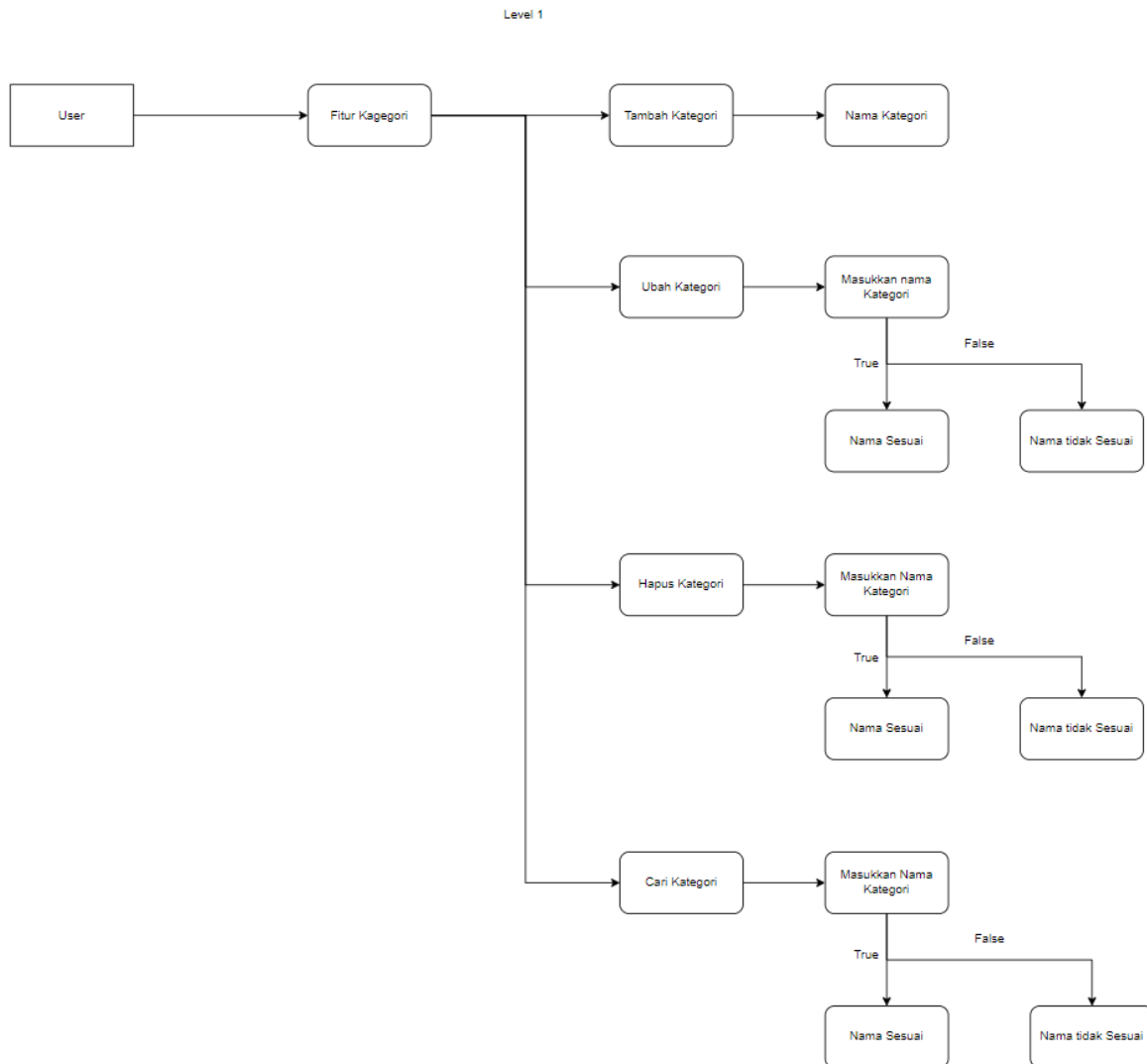


Penjelasan :

1. User:

Pengguna yang berinteraksi dengan sistem. Dalam konteks ini, pengguna akan memilih menu atau fitur dalam sistem.
2. Fitur Kategori:

Bagian dari sistem yang menyediakan berbagai kategori fitur. Sistem ini menerima input dari pengguna berupa pilihan menu, lalu melakukan operasi sesuai dengan pilihan tersebut.
3. Arus Data:
 - a. Pilih Menu: Aliran data dari pengguna ke sistem berupa pilihan fitur atau kategori menu yang ingin dipilih oleh pengguna.
 - b. Hasil Operasi: Aliran data dari sistem kembali ke pengguna berupa hasil dari proses atau operasi yang telah dilakukan berdasarkan menu yang dipilih pengguna.



Penjelasan:

1. User:
Pengguna masih berperan sebagai aktor utama yang berinteraksi dengan sistem. Pengguna memilih tindakan yang ingin dilakukan terkait kategori.
2. Fitur Kagegori:
Ini adalah proses utama di mana sistem memungkinkan pengguna untuk melakukan beberapa operasi pada kategori, seperti menambah, mengubah, menghapus, dan mencari kategori.
3. Tambah Kategori:
Setelah pengguna memilih opsi ini, pengguna akan diminta untuk memasukkan "Nama Kategori" yang baru. Sistem kemudian akan menambahkan kategori baru dengan nama yang diberikan.
4. Ubah Kategori:
Dalam proses ini, pengguna memasukkan nama kategori yang ingin diubah. Sistem memeriksa apakah nama yang dimasukkan sesuai (True) atau tidak sesuai (False). Jika sesuai, pengguna bisa melanjutkan untuk mengubah kategori. Jika tidak, sistem akan memberikan umpan balik bahwa nama tersebut tidak sesuai.
5. Hapus Kategori:

Proses ini mirip dengan "Ubah Kategori". Pengguna memasukkan nama kategori yang ingin dihapus, dan sistem memvalidasi nama tersebut. Jika nama sesuai, kategori dihapus, jika tidak, sistem memberikan notifikasi bahwa nama tidak sesuai.

6. Cari Kategori:

Pengguna juga bisa mencari kategori dengan memasukkan nama kategori. Sama seperti operasi lainnya, sistem akan memverifikasi nama yang dimasukkan, dan jika nama sesuai, hasil pencarian akan ditampilkan.

8. Structured English (pseudocode)

```
1 DO WHILE user input is not 0 (exit)
2   DISPLAY menu options:
3     - Add Category
4     - Edit Category
5     - Delete Category
6     - Search Category
7     - Display All Categories
8     - Exit
9   GET user input
10
11   IF user input is 1 THEN
12     CALL Add Category
13   ELSE IF user input is 2 THEN
14     CALL Edit Category
15   ELSE IF user input is 3 THEN
16     CALL Delete Category
17   ELSE IF user input is 4 THEN
18     CALL Search Category
19   ELSE IF user input is 5 THEN
20     CALL Display All Categories
21   ELSE IF user input is 0 THEN
22     DISPLAY exit message
23   ELSE
24     DISPLAY "Invalid choice"
25
26 ADD CATEGORY
27 IF category count is greater than or equal to MAX_CATEGORIES THEN
28   DISPLAY "Cannot add more categories. Limit reached."
29   RETURN
30
31 PROMPT user to enter new category name
32 SET new category status to "Active"
33 ADD the new category to the list of categories
34 INCREMENT category count
35 DISPLAY "Category successfully added."
36
37 EDIT CATEGORY
38 IF there are no categories THEN
39   DISPLAY "No categories available to edit."
40   RETURN
41
42 PROMPT user to enter the category name to edit
43 FOR each category in the list
44   IF category name matches user input THEN
45     PROMPT user to enter new category name
46     UPDATE the category's name
47     DISPLAY "Category successfully updated."
48     RETURN
49   DISPLAY "Category not found."
50
51 DELETE CATEGORY
52 IF there are no categories THEN
53   DISPLAY "No categories available to delete."
54   RETURN
55
56 PROMPT user to enter the category name to delete
57 FOR each category in the list
58   IF category name matches user input THEN
59     REMOVE the category from the list by shifting the rest of the categories
60     left
61     DECREMENT category count
62     DISPLAY "Category successfully deleted."
63     RETURN
64   DISPLAY "Category not found."
65
66 SEARCH CATEGORY
67 PROMPT user to enter category name to search for
68 FOR each category in the list
69   IF category name matches user input THEN
70     DISPLAY category details (name and status)
71     RETURN
72   DISPLAY "Category not found."
73
74 DISPLAY ALL CATEGORIES
75 IF there are no categories THEN
76   DISPLAY "No categories available."
77   RETURN
78
79 FOR each category in the list
80   DISPLAY category details (name and status)
```

Penjelasan :

1. Penjelasan Detail

Loop Utama:

- DO WHILE user input is not 0 (exit): Ini memulai loop yang berlanjut hingga pengguna memasukkan 0, yang menunjukkan bahwa mereka ingin keluar dari program.
- DISPLAY menu options: Menampilkan aksi yang tersedia bagi pengguna (menambah, mengedit, menghapus, mencari, dan menampilkan kategori).

2. Penanganan Input Pengguna:

- GET user input: Mengambil pilihan pengguna dari menu yang ditampilkan.
- Pernyataan kondisional berikut (IF, ELSE IF) menentukan fungsi mana yang akan dipanggil berdasarkan input pengguna. Setiap nomor sesuai dengan aksi spesifik:
 - 1 untuk menambah kategori
 - 2 untuk mengedit kategori
 - 3 untuk menghapus kategori
 - 4 untuk mencari kategori
 - 5 untuk menampilkan semua kategori
 - 0 untuk keluar dari program
- Jika pengguna memasukkan opsi yang tidak valid, program memberi tahu mereka tentang pilihan yang tidak valid.

3. Fungsi

Setiap fungsi mewakili aksi spesifik yang dapat dilakukan pada kategori:

- Menambah Kategori:
 - Memeriksa apakah batas maksimum kategori telah tercapai. Jika ya, maka memberi tahu pengguna dan keluar.
 - Meminta pengguna untuk memasukkan nama kategori baru, mengatur statusnya menjadi "Aktif," menambahkannya ke daftar, meningkatkan jumlah kategori, dan mengonfirmasi penambahan yang berhasil.
- Mengedit Kategori:
 - Memeriksa apakah ada kategori yang sudah ada. Jika tidak, memberi tahu pengguna dan keluar.
 - Meminta nama kategori yang akan diedit, mencari dalam daftar untuk mencocokkan, dan jika ditemukan, meminta nama baru untuk diperbarui. Jika diperbarui dengan sukses, mengonfirmasi perubahan; jika tidak, memberi tahu pengguna bahwa kategori tidak ditemukan.
- Menghapus Kategori:
 - Mirip dengan fungsi edit, terlebih dahulu memeriksa kategori yang ada. Jika tidak ada, keluar dari program.
 - Meminta pengguna untuk memasukkan nama kategori yang akan dihapus. Jika ditemukan, kategori dihapus dari daftar dan jumlah kategori dikurangi, lalu mengonfirmasi penghapusan. Jika tidak ditemukan, memberi tahu pengguna.
- Mencari Kategori:
 - Meminta pengguna untuk memasukkan nama kategori yang ingin dicari.
 - Mencari dalam daftar untuk mencocokkan dan menampilkan detail kategori jika ditemukan. Jika tidak ditemukan, memberi tahu pengguna bahwa kategori tidak ditemukan.
- Menampilkan Semua Kategori:
 - Memeriksa apakah ada kategori yang ada. Jika tidak, memberi tahu pengguna dan keluar.

- b. Jika kategori tersedia, iterasi melalui daftar dan menampilkan detail setiap kategori.

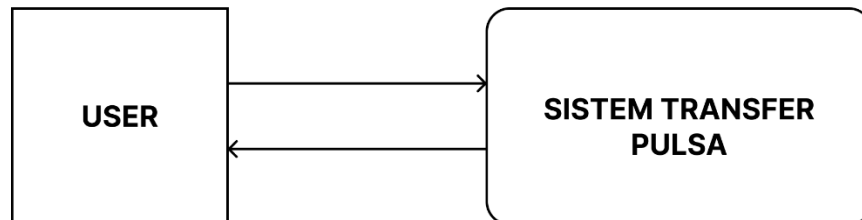
Link GitHub : <https://github.com/Azizidimas/Impal>

Tugas 2#2

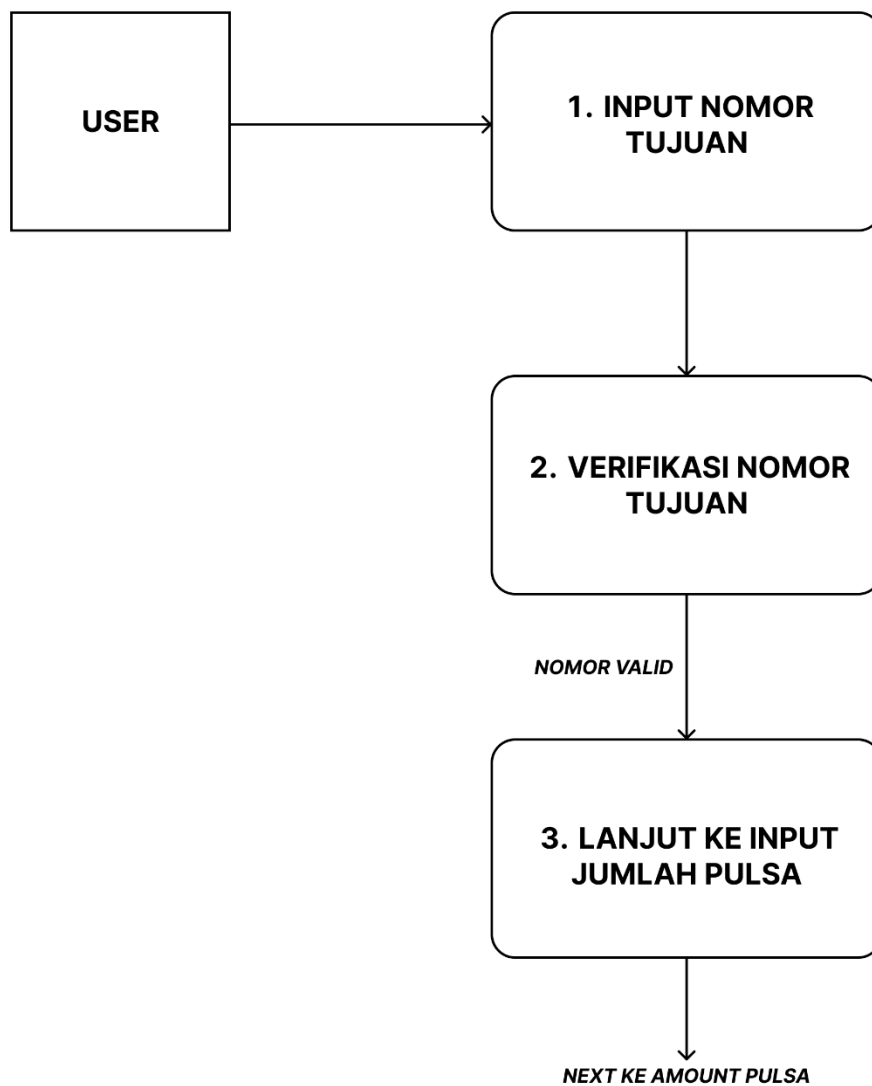
1. DFD (Data Flow Diagram)

a. Fitur Tranfer Pulsa

Lvl 0

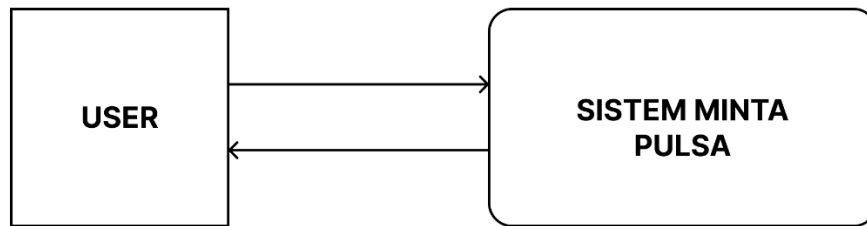


Lvl 1

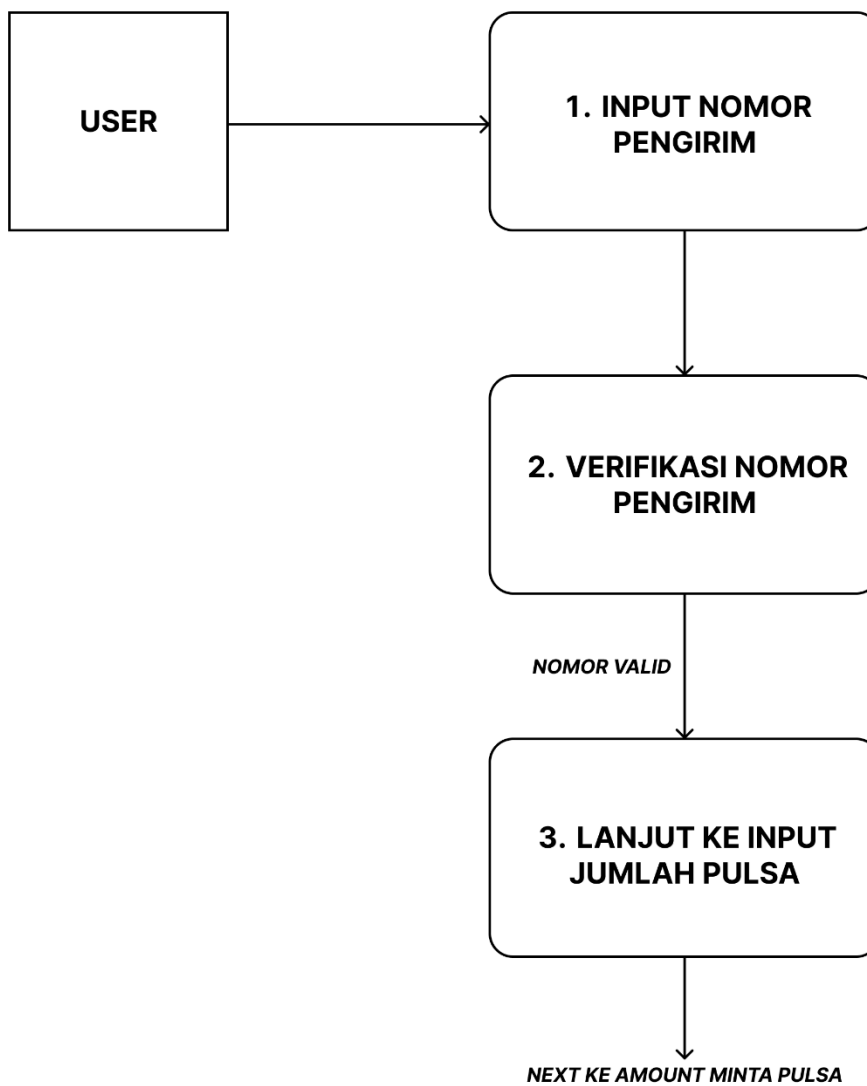


b. Minta Pulsa

Lvl 0

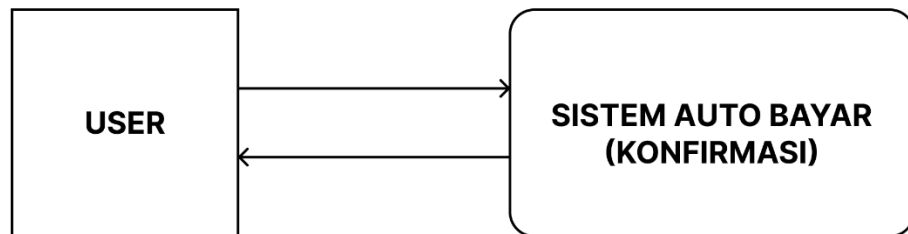


Lvl 1

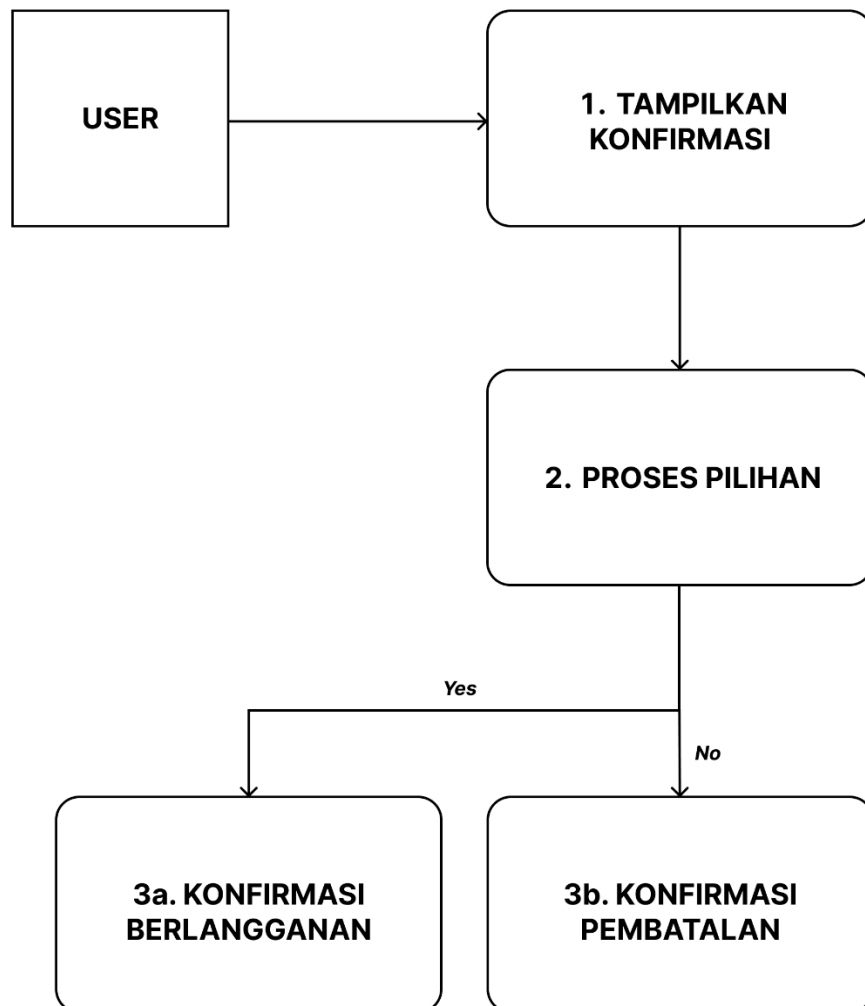


c. Auto Bayar

Lvl 0

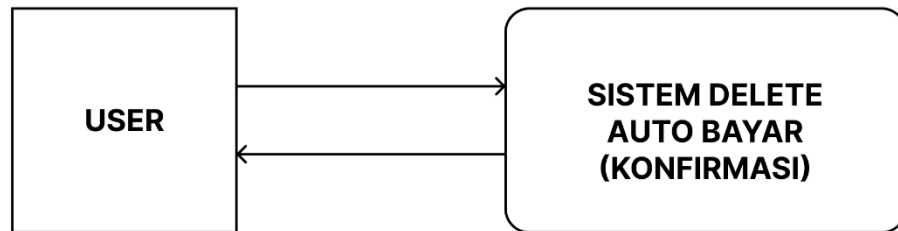


Lvl 1

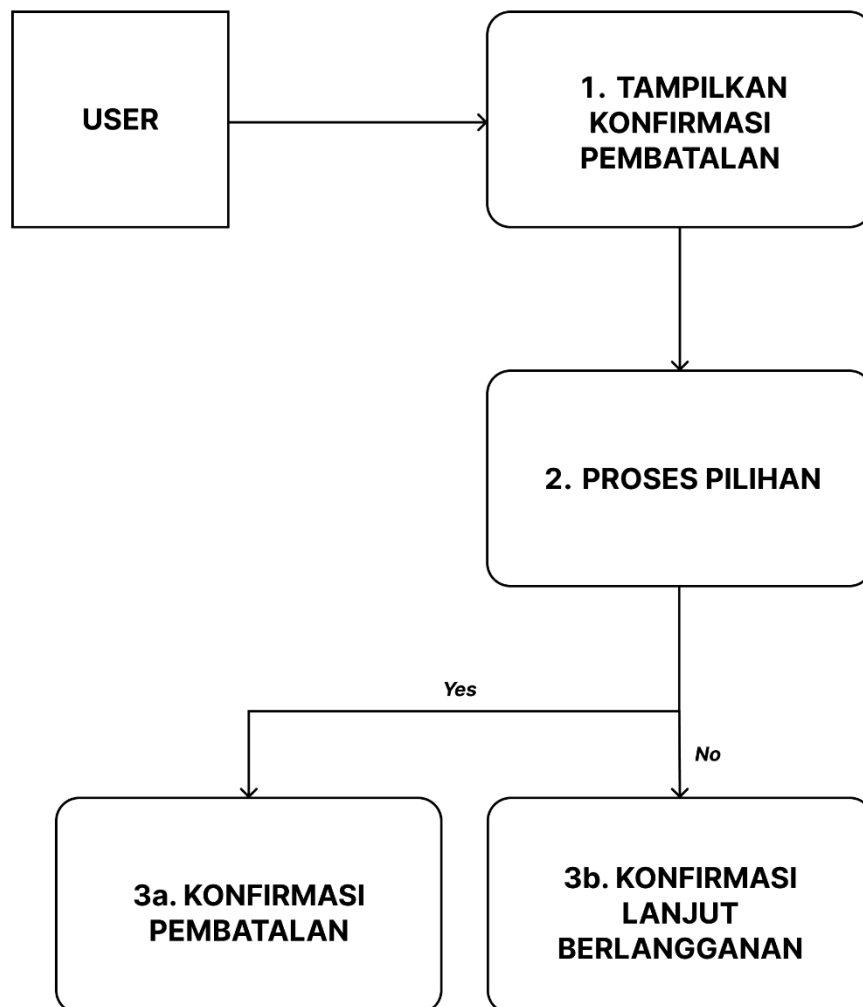


d. Delete Auto Bayar

Lvl 0

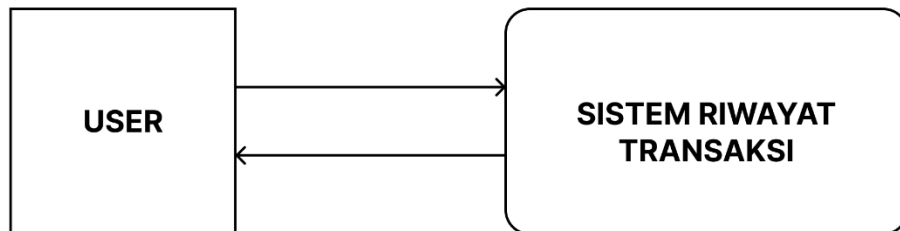


Lvl 1

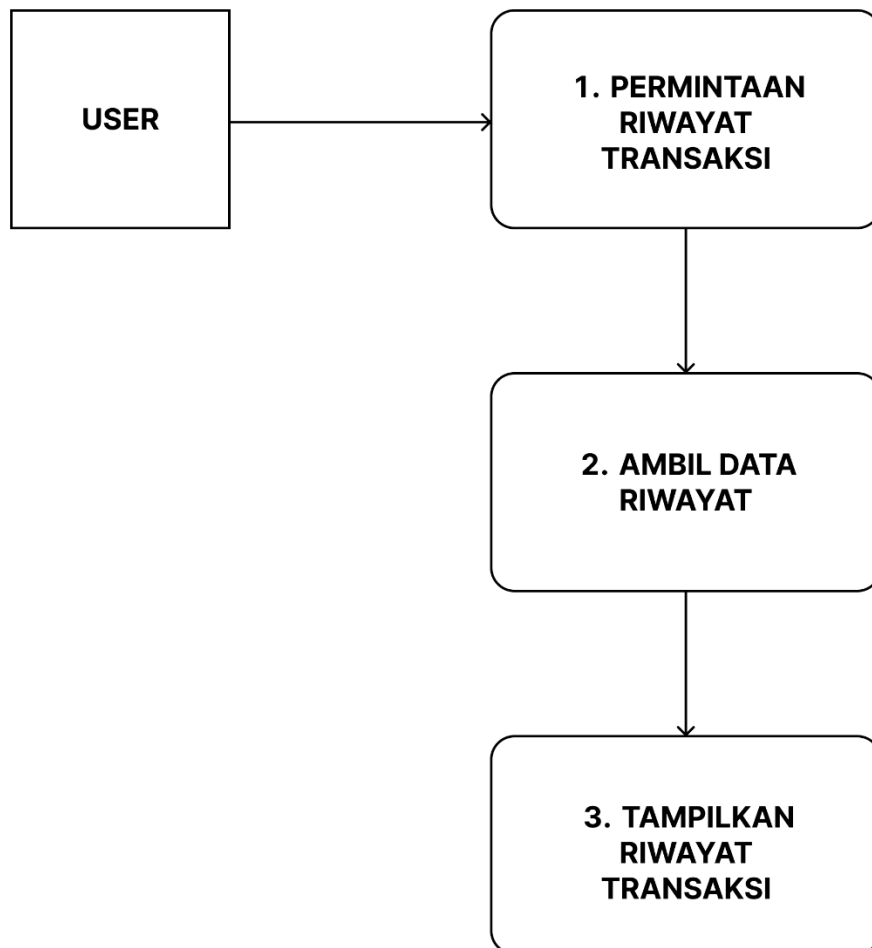


e. Riwayat

Lvl 0

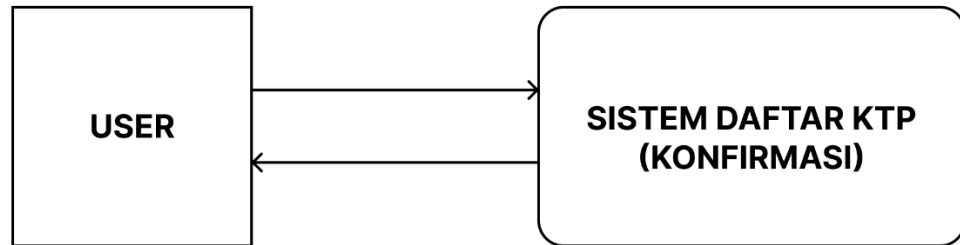


Lvl 1

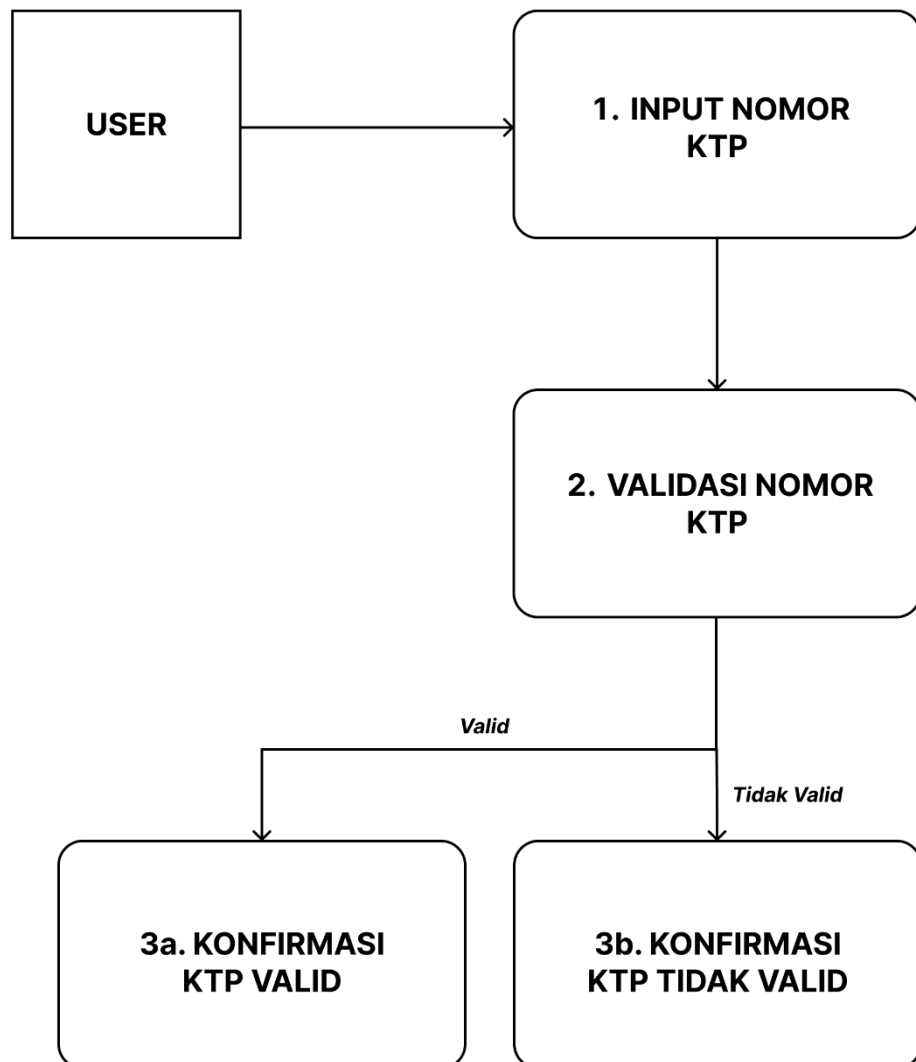


f. Daftar Ktp

Lvl 0



Lvl 1



2. Structured English – Pseudocode

a. AmountMintaPulsa.java

```
PROCESS AmountMintaPulsa WITH PARAMETERS mainMenu, recipient

    // Create the main window
    CREATE window 'amountFrame' with title "Minta Pulsa - Step 2"
    SET size of 'amountFrame' to 400x200
    MAKE 'amountFrame' non-resizable
    SET layout of 'amountFrame' to BorderLayout

    // Create a label to show recipient information
    CREATE label 'amountLabel' with text: "Masukkan jumlah pulsa yang ingin diminta dari nomor:
recipient"
    SET 'amountLabel' alignment to center
    SET font of 'amountLabel' to Arial, Bold, 14
    ADD 'amountLabel' to 'amountFrame' at NORTH

    // Create input field for the amount
    CREATE text field 'amountField' for user to enter amount
    SET 'amountField' alignment to center
    SET font of 'amountField' to Arial, Plain, 14
    ADD gray border to 'amountField'
    ADD 'amountField' to a panel called 'centerPanel'
    ADD 'centerPanel' to the CENTER of 'amountFrame'

    // Create buttons for sending or canceling the request
    CREATE panel 'buttonPanel' with grid layout (1 row, 2 columns, 10px gap)

    CREATE button 'Cancel'
    ADD 'Cancel' to 'buttonPanel'

    CREATE button 'Send'
    ADD 'Send' to 'buttonPanel'

    ADD 'buttonPanel' to the SOUTH of 'amountFrame'

    // Handle the cancel button click
    IF 'Cancel' button is clicked THEN
        CLOSE 'amountFrame'
    END IF

    // Handle the send button click
    IF 'Send' button is clicked THEN
        GET the text from 'amountField' AS 'amount'

        IF 'amount' is empty THEN
            DISPLAY an error message: "Please enter a valid amount."
        ELSE
            ADD a transaction to the main menu history: "Minta Pulsa sebesar amount dari nomor
recipient"
            DISPLAY a confirmation message: "Permintaan Pulsa sebesar amount dari nomor recipient
telah dikirim."
            CLOSE 'amountFrame'
        END IF
    END IF

    // Display the window
    MAKE 'amountFrame' visible

END
```


b. AmountPulsa.java

```
PROCESS AmountPulsa WITH PARAMETERS mainMenu, recipient

// Create the main window
CREATE window 'amountFrame' with title "Transfer Pulsa - Step 2"
SET size of 'amountFrame' to 400x200
MAKE 'amountFrame' non-resizable
SET layout of 'amountFrame' to BorderLayout

// Create a label to show recipient information
CREATE label 'amountLabel' with text: "Masukkan jumlah pulsa untuk nomor: recipient"
SET 'amountLabel' alignment to center
SET font of 'amountLabel' to Arial, Bold, 14
ADD 'amountLabel' to 'amountFrame' at NORTH

// Create input field for the amount
CREATE text field 'amountField' for user to enter amount
SET 'amountField' alignment to center
SET font of 'amountField' to Arial, Plain, 14
ADD gray border to 'amountField'
ADD 'amountField' to a panel called 'centerPanel'
ADD 'centerPanel' to the CENTER of 'amountFrame'

// Create buttons for sending or canceling the transfer
CREATE panel 'buttonPanel' with grid layout (1 row, 2 columns, 10px gap)

CREATE button 'Cancel'
ADD 'Cancel' to 'buttonPanel'

CREATE button 'Send'
ADD 'Send' to 'buttonPanel'

ADD 'buttonPanel' to the SOUTH of 'amountFrame'

// Handle the cancel button click
IF 'Cancel' button is clicked THEN
    CLOSE 'amountFrame'
END IF

// Handle the send button click
IF 'Send' button is clicked THEN
    GET the text from 'amountField' AS 'amount'

    IF 'amount' is empty THEN
        DISPLAY an error message: "Please enter a valid amount."
    ELSE
        ADD a transaction to the main menu history: "Transfer Pulsa sebesar amount ke nomor
recipient"
        DISPLAY a confirmation message: "Transfer Pulsa sebesar amount ke nomor recipient
berhasil."
        CLOSE 'amountFrame'
    END IF
END IF

// Display the window
MAKE 'amountFrame' visible

END
```

c. AutoBayar.java

```
PROCESS AutoBayar

// Prompt the user with a subscription confirmation dialog
DISPLAY a confirmation dialog with the message:
    "Apakah Anda ingin mendaftar untuk Akses Internet Unlimited sebesar 88.000 pulsa/bulan?"
SET options to YES or NO
STORE the user's response as 'choice'

// Check user's choice
IF 'choice' IS YES THEN
    DISPLAY a message: "Anda telah berhasil mendaftar Auto Bayar sebesar 88.000 pulsa/bulan."
ELSE IF 'choice' IS NO THEN
    DISPLAY a message: "Auto Bayar dibatalkan."
END IF

END
```

d. DaftarKTP.java

```
MAIN PROCESS DaftarKTP

CREATE JFrame ktpFrame with title "Daftar KTP"
SET size of ktpFrame to 400x200
SET ktpFrame to non-resizable
SET layout of ktpFrame to BorderLayout

CREATE JLabel ktpLabel with message "Silahkan masukkan nomor KTP Anda (16 digit):"
SET horizontal alignment of ktpLabel to CENTER
SET font of ktpLabel to Arial, Bold, size 14
ADD ktpLabel to ktpFrame at position NORTH

CREATE JTextField ktpField with width 20
SET horizontal alignment of ktpField to CENTER
SET font of ktpField to Arial, Plain, size 14
SET border of ktpField to Gray, 2 pixels
CREATE JPanel centerPanel
ADD ktpField to centerPanel
ADD centerPanel to ktpFrame at position CENTER

CREATE JPanel buttonPanel
SET layout of buttonPanel to GridLayout(1, 2, 10, 10)

CREATE JButton cancelButton with label "Cancel"
CREATE JButton submitButton with label "Submit"

ADD cancelButton to buttonPanel
ADD submitButton to buttonPanel

ADD buttonPanel to ktpFrame at position SOUTH

SET ActionListener for cancelButton to close ktpFrame

SET ActionListener for submitButton:
    GET text from ktpField and STORE in ktpNumber

    IF ktpNumber matches regex "\\d{16}" THEN
        DISPLAY message dialog "KTP Anda valid. Terima kasih telah mendaftar!"
        CLOSE ktpFrame
    ELSE
        DISPLAY message dialog "Nomor KTP tidak valid. Pastikan 16 digit dan hanya angka."

SET ktpFrame visibility to true

END PROCESS
```

e. DeleteAutoBayar.java

```
MAIN PROCESS DeleteAutoBayar

    DISPLAY confirmation dialog
        MESSAGE: "Apakah Anda ingin membatalkan langganan Auto Bayar untuk Akses Internet Unlimited
sebesar 80.000 pulsa/bulan?"
        TITLE: "Hapus Auto Bayar"
        OPTIONS: YES, NO

    STORE user choice in variable choice

    IF choice is YES_OPTION THEN
        DISPLAY message dialog
            MESSAGE: "Anda telah berhasil membatalkan langganan Auto Bayar."
    ELSE IF choice is NO_OPTION THEN
        DISPLAY message dialog
            MESSAGE: "Auto Bayar tetap aktif."

END PROCESS
```

f. KeypadGUI.java

MAIN PROCESS KeypadGUI

```
CREATE window with title "Dialer"
SET window to close on exit
SET window size to 350x500
PREVENT window resizing
USE BorderLayout for the window

CREATE a bottom panel with BorderLayout

CREATE a text field for input
    ALIGN text field to center
    SET font size to 16
    ADD gray border to text field
ADD text field to the north of the bottom panel

CREATE a button panel with GridLayout (1 row, 2 columns, 10px gap)
CREATE "Cancel" button
CREATE "Send" button
ADD buttons to the button panel
ADD button panel to the south of the bottom panel
ADD bottom panel to the south of the window

CREATE a number grid panel with GridLayout (4 rows, 3 columns, 10px gap)

FOR each number from 1 to 9
    CREATE a button with the number
    SET font size to 20
    SET ActionListener for button
        WHEN button is clicked
            CALL appendToInputField with the number
    ADD button to number grid

CREATE button for "*"
SET font size to 20
SET ActionListener for "*" button
    WHEN button is clicked
        CALL appendToInputField with "*"
ADD "*" button to number grid

CREATE button for "0"
SET font size to 20
SET ActionListener for "0" button
    WHEN button is clicked
        CALL appendToInputField with "0"
ADD "0" button to number grid

CREATE button for "#"
SET font size to 20
SET ActionListener for "#" button
    WHEN button is clicked
        CALL appendToInputField with "#"
ADD "#" button to number grid

ADD number grid to the north of the window

SET ActionListener for "Cancel" button
    WHEN button is clicked
        PRINT "Operation canceled."
        CLOSE the window

SET ActionListener for "Send" button
    WHEN button is clicked
        GET text from input field
        IF text equals "#000*"
            CLOSE the window
            CALL MainMenuGUI
        ELSE
            DISPLAY message "Invalid input. Please enter #000* to proceed."

DISPLAY the window

SUBPROCESS appendToInputField WITH num
    APPEND num to the current text in the input field

MAIN PROCESS
    CALL KeypadGUI
```

g. MainMenuGUI.java

```
PROCESS MainMenuGUI

    CREATE a list to store transaction history

    CREATE a new window called "Selamat Datang Di Menu 808"
    SET the window size to 350x400
    DISABLE resizing of the window
    USE a BorderLayout for the window

    CREATE a label with text "Mau Dapat Uang Cepat? Daftarkan KTP Anda di 808"
    ALIGN the label to the center
    SET the label font to Arial, plain, size 9
    ADD an empty border around the label
    ADD the label to the top section of the window

    CREATE a text area with the following menu options:
        "1. Transfer Pulsa"
        "2. Minta Pulsa"
        "3. Auto Bayar"
        "4. Delete Auto Bayar"
        "5. Riwayat"
        "6. Daftar KTP"
        "7. Exit"
    SET the text area to non-editable
    SET the font of the text area to Arial, plain, size 16
    ADD margins to the text area
    ADD the text area to the center of the window

    CREATE a bottom panel with a BorderLayout
    CREATE a text field for user input
    ALIGN the text field to the center
    SET the text field font to Arial, plain, size 16
    ADD a border to the text field
    ADD the text field to the top section of the bottom panel

    CREATE a panel for buttons with a GridLayout (1 row, 2 columns, 10px gap)

    CREATE a "Cancel" button
    CREATE a "Send" button
    ADD both buttons to the button panel
    ADD the button panel to the bottom of the bottom panel

    ADD the bottom panel to the bottom section of the window

    WHEN "Cancel" button is clicked
        CLOSE the window

    WHEN "Send" button is clicked
        CALL handleMenuSelection WITH input from the text field

    DISPLAY the window

---

SUBPROCESS handleMenuSelection WITH input

    IF input IS "1"
        CALL TransferPulsa WITH reference to this mainMenu

    IF input IS "2"
        CALL MintaPulsa WITH reference to this mainMenu

    IF input IS "3"
        CALL AutoBayar
        ADD "Auto Bayar subscribed" to transaction history

    IF input IS "4"
        CALL DeleteAutoBayar
        ADD "Auto Bayar canceled" to transaction history

    IF input IS "5"
        IF history is empty
            DISPLAY a message "Riwayat kosong. Tidak ada transaksi yang dilakukan."
        ELSE
            CALL Riwayat WITH transaction history

    IF input IS "6"
        CALL DaftarKTP

    IF input IS "7"
        DISPLAY a message "Exiting the application."
        EXIT the application

    IF input is not valid
        DISPLAY a message "Invalid selection."

---

SUBPROCESS addHistory WITH transaction

    ADD transaction to the history list
```

h. MintaPulsa.java

```
PROCESS MintaPulsa WITH input mainMenu

    CREATE a window called "Minta Pulsa - Step 1" with size 400x200 and no resizing allowed
    SET the layout of the window to "BorderLayout"

    DISPLAY a label with the text "Silahkan masukkan nomor dari siapa Anda minta pulsa"
    SET the label to be centered and use bold font size 14

    CREATE an input text field for recipient number with 20-character width
    CENTER the text inside the input field and set the font to size 14

    ADD the input field to the center of the window

    CREATE two buttons labeled "Cancel" and "Next"
    ARRANGE the buttons in a horizontal layout

    ADD an event listener for the "Cancel" button
        WHEN the "Cancel" button is clicked:
            CLOSE the window

    ADD an event listener for the "Next" button
        WHEN the "Next" button is clicked:
            IF the input field is empty THEN
                SHOW a message saying "Please enter a valid recipient number."
            ELSE
                CLOSE the current window
                CALL the "AmountMintaPulsa" function WITH the recipient number and mainMenu

    SHOW the window

END PROCESS
```

i. Riwayat.java

```
PROCESS Riwayat WITH input history (an array of transaction history)

    CREATE a window called "Riwayat Transaksi" with size 400x300
    SET the window to close only when explicitly closed (DISPOSE_ON_CLOSE)

    CREATE a text area for displaying history
    SET the text area to be non-editable
    SET the font of the text area to Arial, size 14

    FOR each entry in the history array DO
        APPEND the entry to the text area FOLLOWED BY a new line

    CREATE a scrollable pane containing the text area
    ADD the scrollable pane to the window

    SHOW the window

END PROCESS
```

j. TransferPulsa.java

```

PROCESS TransferPulsa WITH input mainMenu (reference to the main menu)

CREATE a window called "Transfer Pulsa - Step 1" with size 480x200
SET the window to be non-resizable
USE a BorderLayout for the window layout

CREATE a label with text "Silahkan masukkan nomor tujuan Transfer Pulsa:"
ALIGN the label to the center horizontally
SET the font of the label to Arial, bold, size 14
ADD the label to the top section of the window

CREATE a text field for recipient input with size 20
ALIGN the text field to the center
SET the font of the text field to Arial, plain, size 14
ADD a border to the text field
ADD the text field to the center section of the window

CREATE a panel for buttons with a grid layout (1 row, 2 columns, 10px gap)

CREATE a "Cancel" button
CREATE a "Next" button
ADD both buttons to the button panel

ADD the button panel to the bottom section of the window

WHEN "Cancel" button is clicked DO
    CLOSE the window

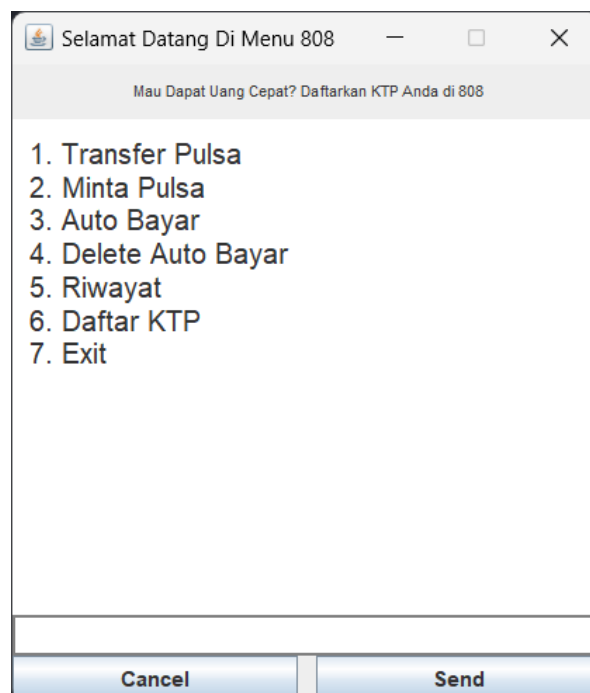
WHEN "Next" button is clicked DO
    GET the input from the recipient text field
    IF recipient input is empty THEN
        SHOW an error message "Please enter a valid recipient number."
    ELSE
        CLOSE the current window
        CALL AmountPulsa with mainMenu and recipient as arguments

SHOW the window

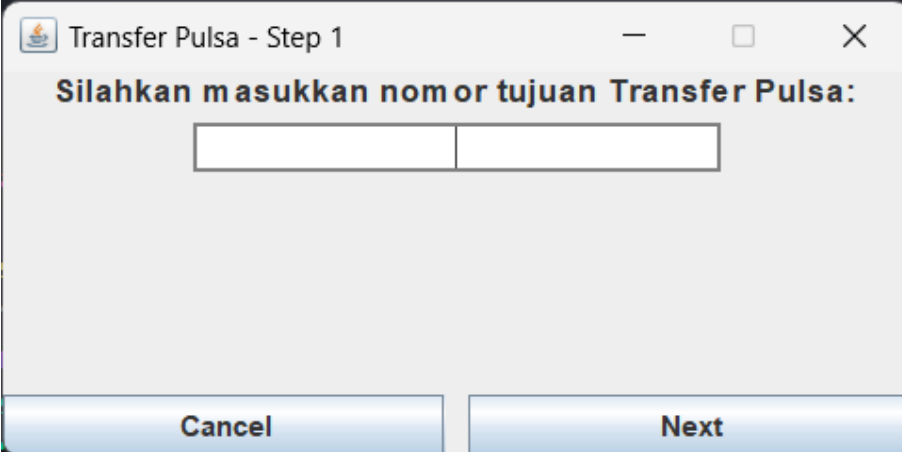
END PROCESS

```

3. Tampilan



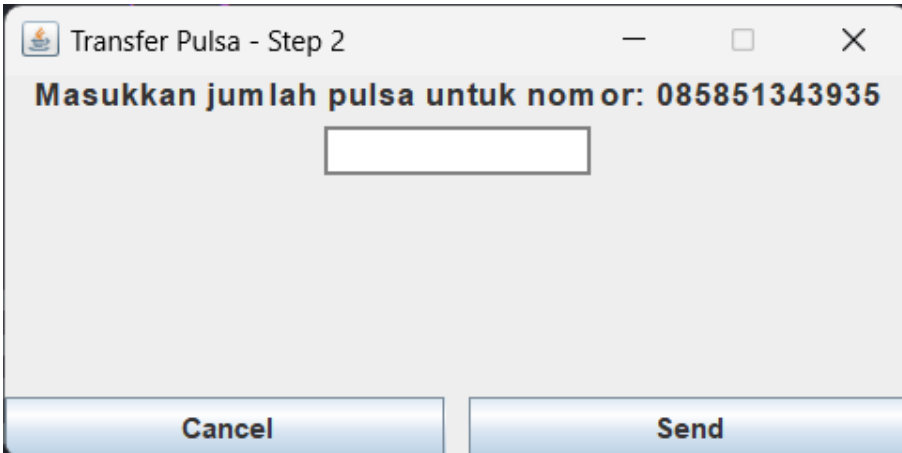
a. Tranfer Pulsa



Transfer Pulsa - Step 1

Silahkan masukkan nomor tujuan Transfer Pulsa:

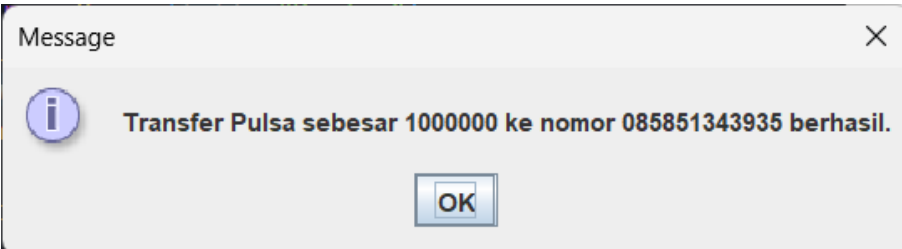
Cancel Next




Transfer Pulsa - Step 2

Masukkan jumlah pulsa untuk nomor: 085851343935

Cancel Send

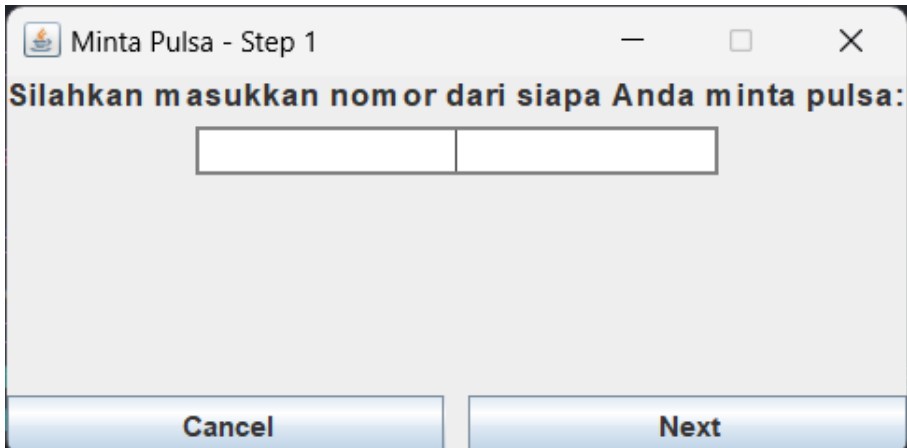


Message

 Transfer Pulsa sebesar 1000000 ke nomor 085851343935 berhasil.

OK

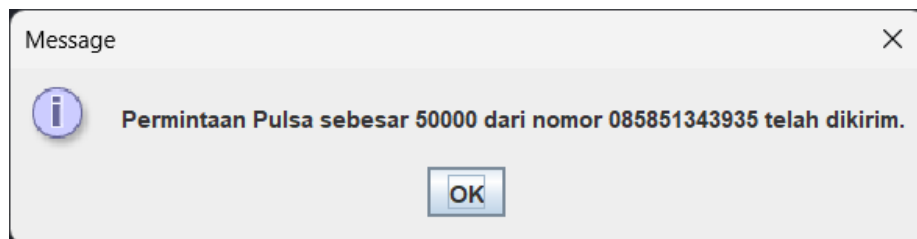
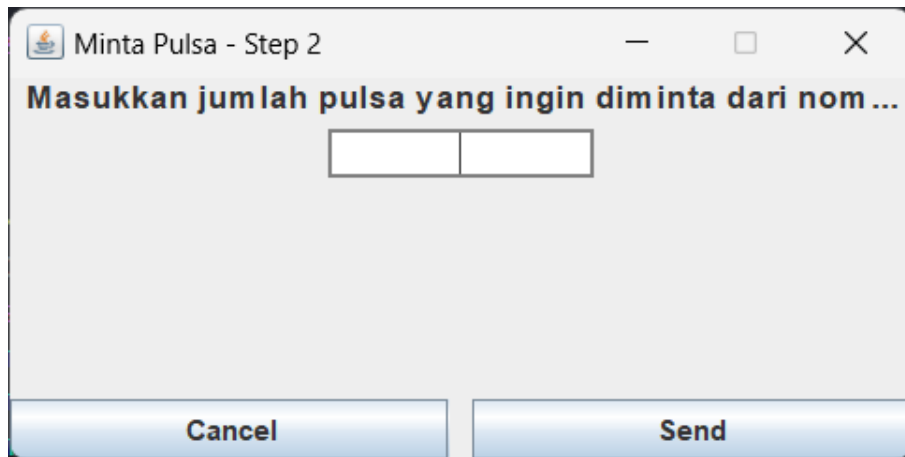
b. Minta Pulsa



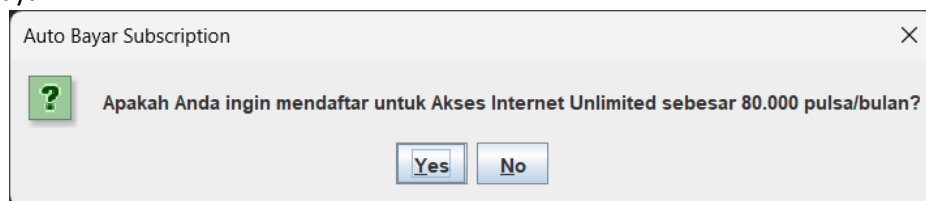
Minta Pulsa - Step 1

Silahkan masukkan nomor dari siapa Anda minta pulsa:

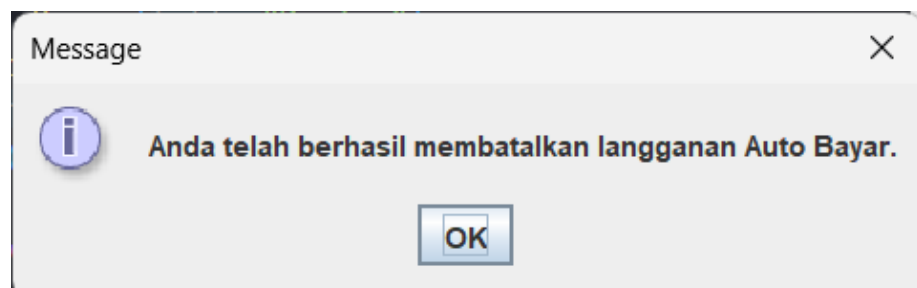
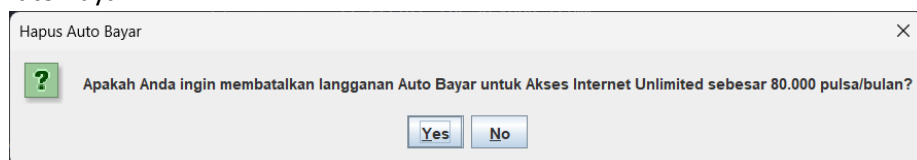
Cancel Next



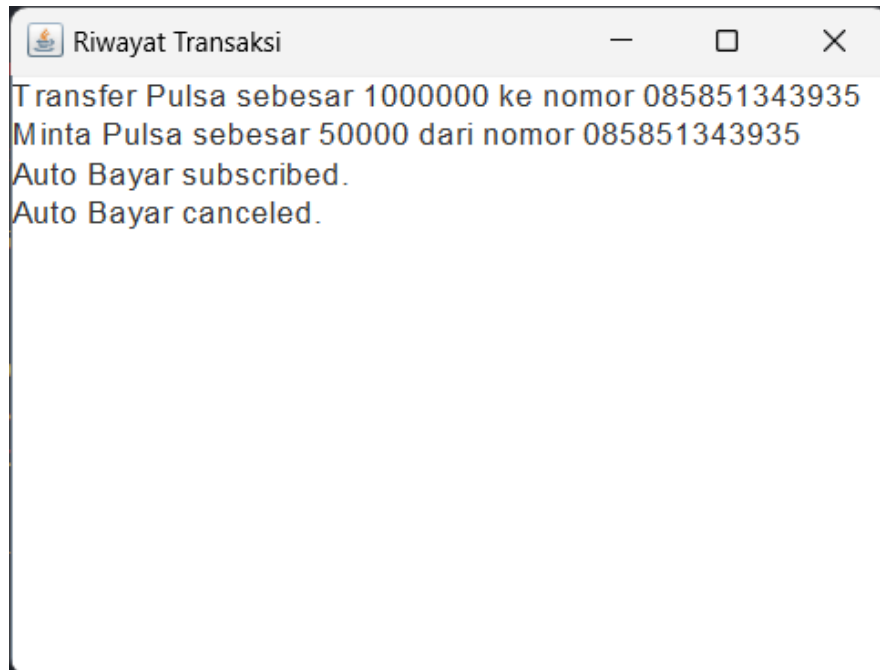
c. Auto Bayar



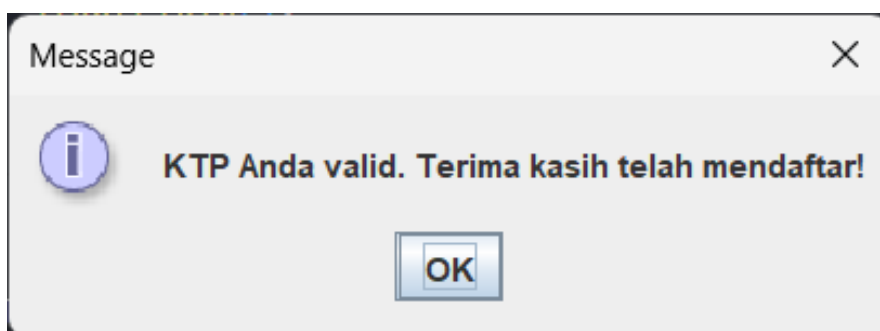
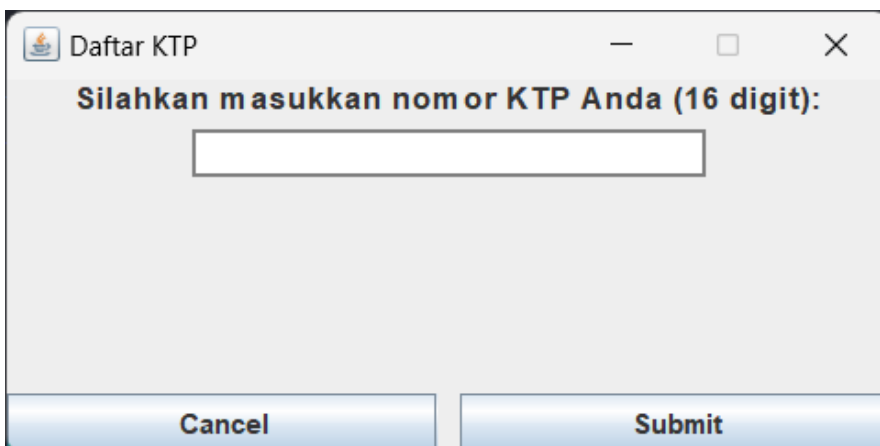
d. Delete Auto Bayar



e. Riwayat



f. Daftar KTP



Link GitHub : <https://github.com/Azizidimas/Impal>