

# TRAC<sup>⚡</sup>TY

TRACK YOUR ELECTRICITY



TARA NUR LITA  
TIARA TYAGITA A  
YULIA PUSPA N R  
KHANSA BASHIRA P  
M HAFIZH FATIHURRIZQI

## DAFTAR ISI

<b>BAB I.....</b>	<b>2</b>
<b>DESKRIPSI MASALAH.....</b>	<b>2</b>
1.1    Latar Belakang Masalah.....	2
1.2    Rumusan Masalah .....	3
1.3    Tujuan.....	3
<b>BAB II .....</b>	<b>4</b>
<b>FLOWCHART PROGRAM.....</b>	<b>4</b>
<b>BAB III.....</b>	<b>12</b>
<b>FUNGSI KODE PROGRAM.....</b>	<b>12</b>
3.1    Penggunaan Modul.....	12
3.2    Program Input Data Pengguna .....	14
3.3    Program Utama .....	17
3.4    Program Pembayaran .....	18
3.5    Program Riwayat.....	20
<b>BAB IV .....</b>	<b>24</b>
<b>HASIL RUNNING PROGRAM .....</b>	<b>24</b>

## **BAB I**

### **DESKRIPSI MASALAH**

#### **1.1 Latar Belakang Masalah**

Dalam kehidupan sehari-hari, banyak pengguna listrik menghadapi kesulitan dalam mengukur besar penggunaan listrik mereka setiap bulannya. Hal ini dapat menjadi masalah serius, terutama ketika datang ke pembayaran tagihan listrik dengan metode pascabayar. Proses pembayaran yang rumit dan kurang praktis sering kali menyulitkan pengguna untuk mengelola keuangan mereka dengan baik. Selain itu, kurangnya informasi yang jelas mengenai tagihan listrik yang dimiliki oleh pengguna juga menjadi hambatan. Pengguna membutuhkan gambaran yang jelas tentang seberapa besar penggunaan listrik mereka selama sebulan agar dapat mengatur keuangan dengan lebih efisien.

Masalah lain yang dihadapi adalah kurangnya kontrol terhadap penggunaan listrik, yang mengakibatkan peningkatan yang tidak terkendali dalam tagihan bulanan. Tanpa pencatatan yang akurat, pengguna sering kali tidak menyadari pola penggunaan listrik yang tidak efisien dan tidak dapat mengidentifikasi cara untuk menghemat energi secara signifikan.

Oleh karena itu, dibutuhkan solusi yang dapat membantu pengguna dalam menyelesaikan masalah ini. Solusi tersebut harus mencakup pencatatan yang akurat mengenai besar penggunaan listrik setiap bulan, mempermudah proses pembayaran dengan metode prabayar dan pascabayar, memberikan informasi yang jelas mengenai tagihan listrik, serta membantu pengguna dalam mengontrol dan menghemat penggunaan listrik mereka. Sebagai tawaran solusi, penulis membuat suatu program yang berfungsi untuk mempermudah pengguna dalam mengurus hal mengenai penggunaan listriknya untuk kebutuhan sehari-hari.

Dengan mengatasi tantangan ini, pengguna akan memiliki pemahaman yang lebih baik tentang penggunaan energi mereka, dapat mengelola pembayaran listrik dengan lebih efisien, dan meningkatkan kesadaran akan penghematan energi yang dapat dilakukan.

## 1.2 Rumusan Masalah

1. Bagaimana suatu program dapat mencatat besar pemakaian dan tagihan listrik?
2. Bagaimana suatu program dapat menginformasikan sisa *credit* token listrik?
3. Bagaimana suatu program mampu membantu pembayaran listrik dengan praktis?
4. Bagaimana suatu program dapat membantu pengguna dalam menghemat listrik?

## 1.3 Tujuan

1. Membuat program yang melakukan pencatatan besar pemakaian dan tagihan listrik dengan baik.
2. Membuat program *reminder* kuantitas penggunaan listrik.
3. Membuat program yang membantu pengguna membayar listrik dengan praktis.

## BAB II

### FLOWCHART PROGRAM

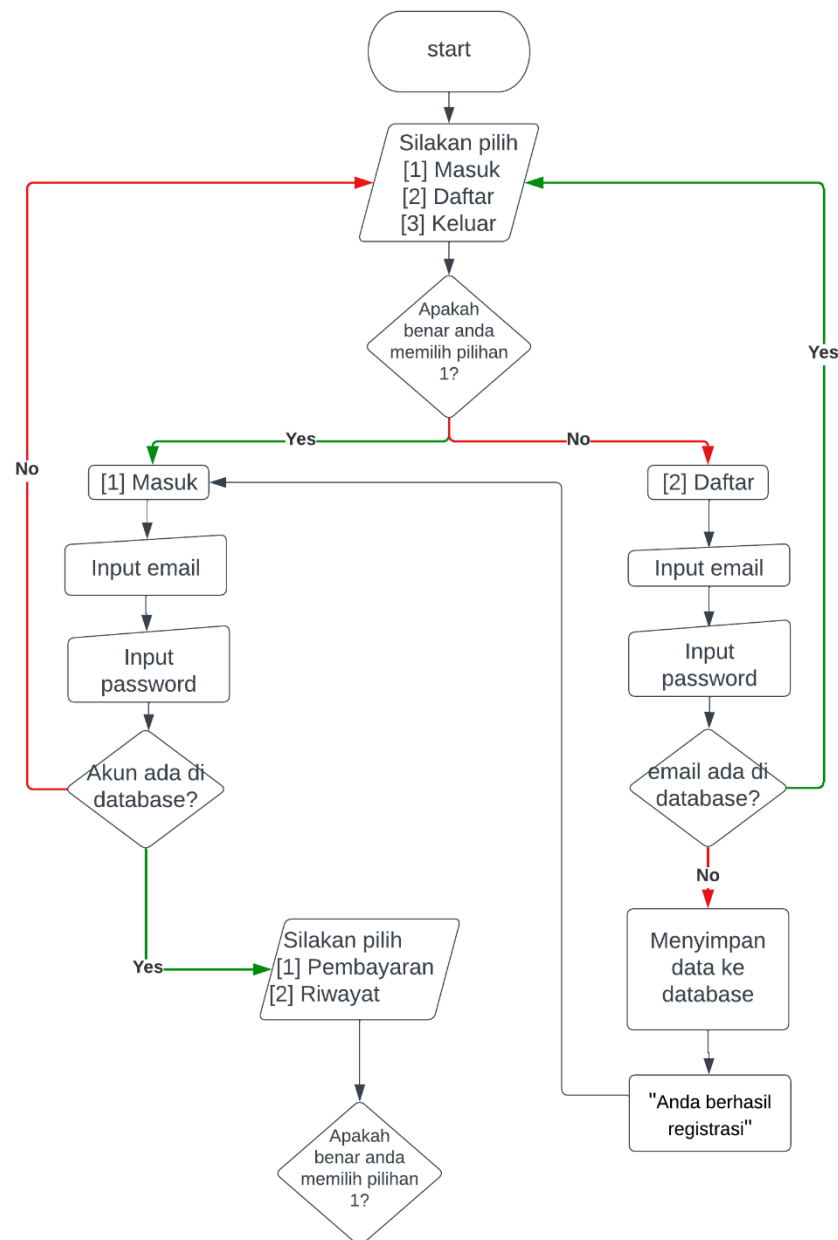
*Flowchart* adalah sebuah representasi visual secara simbolik yang digunakan untuk menggambarkan urutan langkah-langkah atau alur proses dari sebuah sistem, prosedur atau algoritma. *Flowchart* banyak digunakan untuk memvisualisasikan alur logika atau langkah-langkah yang harus diikuti dalam suatu proses secara sistematis.

Pada dasarnya, *flowchart* menggunakan simbol-simbol geometris yang saling dihubungkan oleh panah sehingga menjadi suatu rangkaian alur proses, prosedur, ataupun algoritma dari sebuah sistem. Dengan menggunakan simbol-simbol dan panah ini, *flowchart* akan memberikan gambaran visual yang jelas tentang proses secara keseluruhan, memudahkan pemahaman dan analisis terhadap alur logika dan algoritma yang terlibat.

*Flowchart* kelompok kami menggambarkan tentang aplikasi Tracity atau “*Track Your Electricity*” yang menyediakan fitur-fitur yang akan membantu pengguna dalam menggunakan listrik dengan efisien, diantaranya adalah pencatatan yang akurat mengenai besar penggunaan listrik setiap bulan, mempermudah proses pembayaran dengan metode prabayar maupun pascabayar, memberikan informasi yang jelas mengenai tagihan listrik, serta membantu pengguna dalam mengontrol dan menghemat penggunaan listrik mereka.

*Flowchart* pada program kami diawali dengan pendaftaran data pengguna dengan memasukkan *email* dan *password* melalui beberapa pilihan. Pilihan tersebut terdiri dari [1] untuk masuk, [2] untuk daftar dan [3] untuk keluar. Jika pengguna memilih daftar akan diarahkan untuk membuat *email* dan *password* kemudian jika ternyata *email* yang dimasukkan sudah terdaftar pengguna akan diarahkan kembali ke halaman awal untuk memilih antara [1] masuk, [2] daftar atau [3] keluar. Tetapi jika *email* dan *password* belum terdaftar program akan menyimpan *email* dan *password* ke dalam *database* sehingga muncul “Anda berhasil registrasi” dan setelah itu akan diarahkan ke pilihan masuk. Pada pilihan masuk pengguna akan diarahkan untuk memasukkan *email* dan *password* yang tadi sudah didaftarkan.

Setelah pengguna selesai memasuki halaman awal, pengguna akan diarahkan ke halaman berikutnya (halaman utama) untuk memilih apakah ingin melakukan pembayaran atau ingin melihat riwayat. Pengguna akan disediakan pilihan berupa [1] untuk pembayaran dan [2] untuk riwayat.

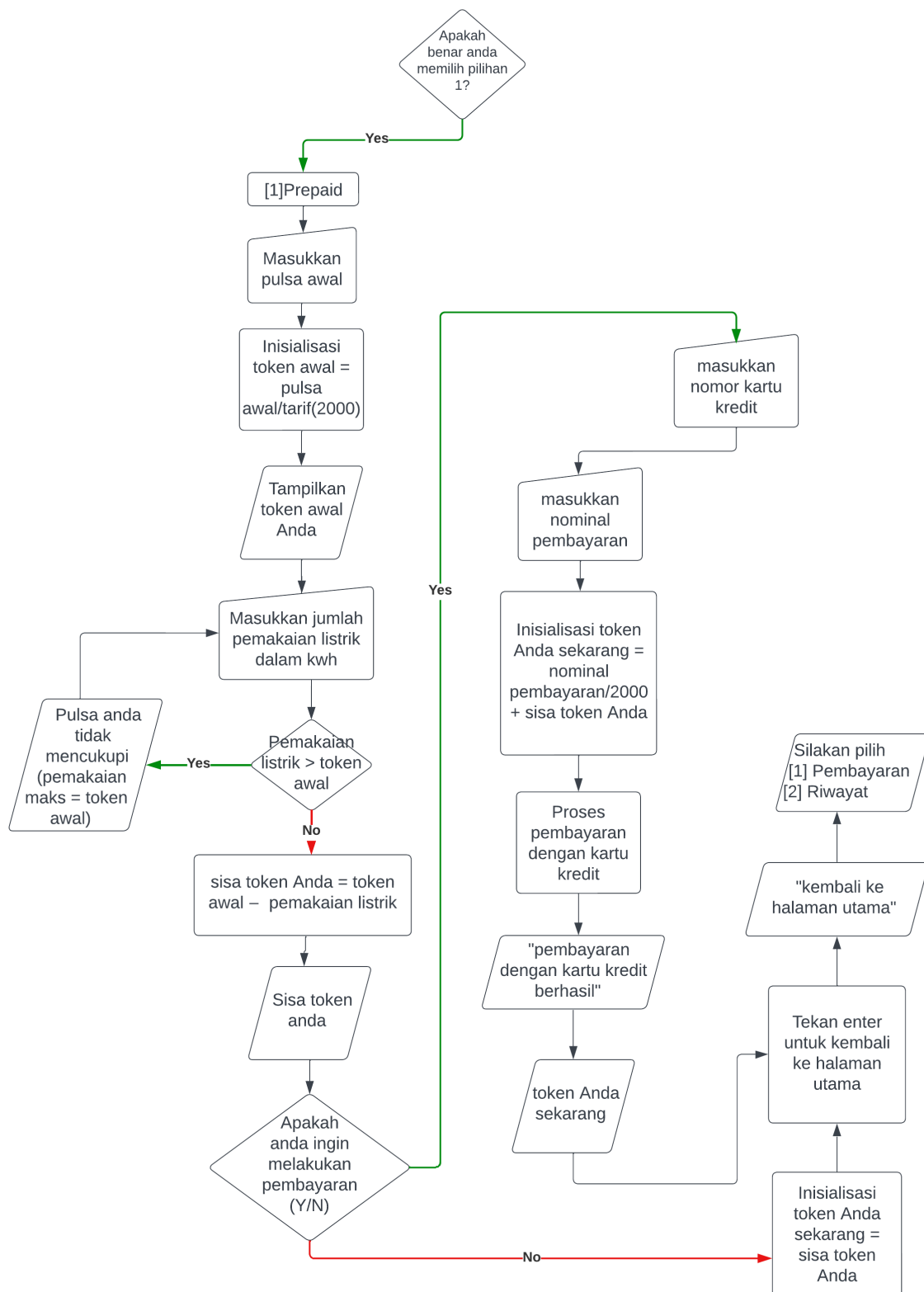


Jika pengguna memilih [1] Pembayaran akan diarahkan untuk memilih apakah ingin melakukan pembayaran listrik menggunakan metode *postpaid* atau *prepaid*. Pengguna akan disediakan pilihan berupa [1] untuk *prepaid* dan [2] untuk *postpaid*. *Prepaid* atau yang biasa dikenal listrik Prabayar adalah sistem pembayaran dimana pengguna listrik harus membeli kredit listrik sebelum menggunakannya. *Postpaid* atau yang biasa dikenal listrik Pascabayar adalah sistem pembayaran dimana pengguna listrik menggunakan listrik terlebih dahulu kemudian akan menerima tagihan bulanan berdasarkan konsumsi listrik yang terjadi.

Jika pengguna memilih [2] Riwayat, pengguna akan disediakan beberapa pilihan lagi untuk melihat riwayat pembayaran maupun riwayat pemakaian. Pengguna dapat memilih [1] untuk riwayat pembayaran dan [2] untuk riwayat pemakaian.

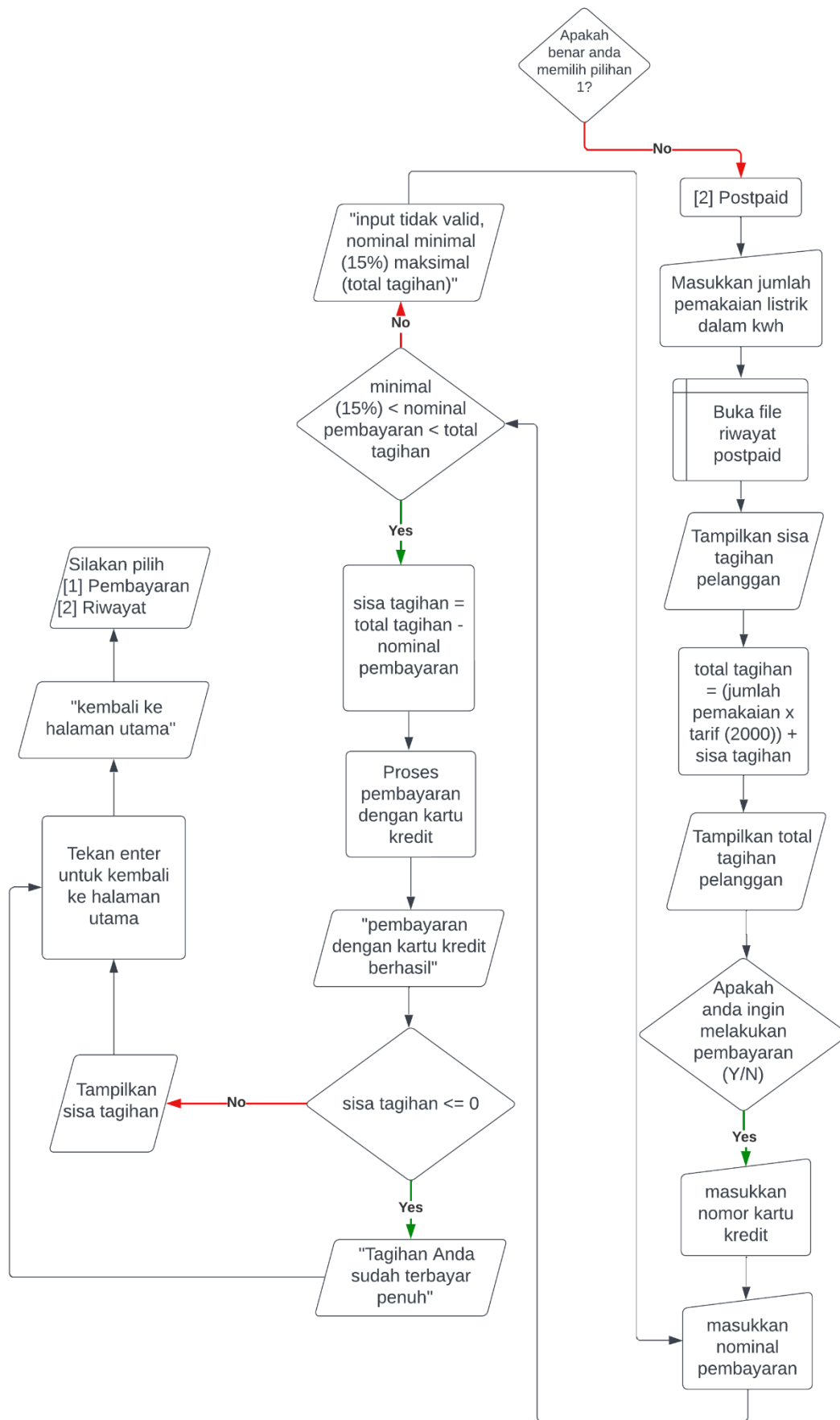


Pada opsi pembayaran jika pengguna memilih [1] *Prepaid* akan diarahkan memasukkan pulsa awal kemudian pulsa awal tersebut akan diinisialisasikan menjadi token awal dengan mengalikan pulsa awal dengan tarif listrik (Rp 2.000/kwh) sehingga sistem akan memunculkan token awal anda. Setelah itu pengguna memasukkan jumlah pemakaian listrik dalam kwh dengan syarat jika pemakaian listrik > token awal akan muncul “Pulsa anda tidak mencukupi” sehingga pengguna harus memasukkan ulang jumlah pemakaian listrik yang digunakan. Akan tetapi jika pemakaian listrik < token awal program akan menghitung sisa token dengan mengurangi token awal dengan pemakaian listrik kemudian memunculkan sisa token pengguna. Setelah itu pengguna akan diarahkan kepada pilihan “Apakah Anda ingin melakukan pembayaran? Dengan opsi [Y] untuk ya dan [N] untuk tidak. Jika memilih [Y] akan diarahkan untuk memasukkan nomor kartu kredit dan nominal pembayaran, setelah itu program akan memproses pembayaran dan menginisialisasi token pengguna sekarang dengan membagi nominal pembayaran dengan tarif listrik (Rp 2.000/kwh) kemudian ditambahkan dengan sisa token pengguna. Setelah itu program akan mengarahkan pengguna untuk memencet ‘Enter’ pada *keyboard* sehingga akan diarah kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat). Jika memilih [N] program akan menginisialisasi token pengguna sekarang sama dengan sisa token pengguna, setelah itu pengguna akan diarahkan untuk memencet ‘Enter’ pada *keyboard* untuk kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat).



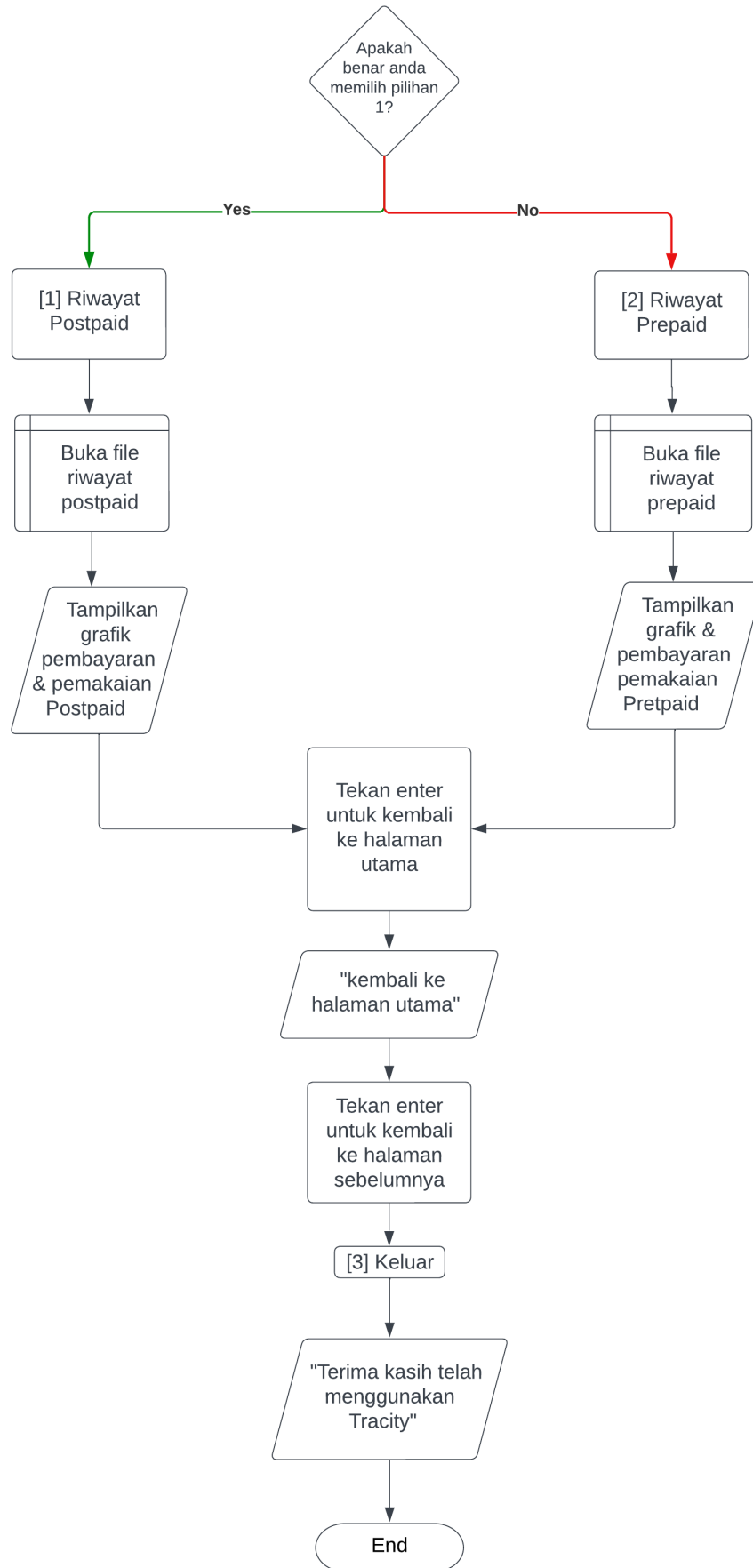


Pada opsi pembayaran jika pengguna memilih [2] *Postpaid* akan diarahkan untuk memasukkan jumlah pemakaian listrik dalam kwh kemudian program akan membuka file riwayat *postpaid* untuk menampilkan sisa tagihan pelanggan pada transaksi sebelumnya. Setelah itu program akan memproses total tagihan dengan mengalikan jumlah pemakaian listrik dengan tarif listrik (Rp 2.000/kwh) kemudian memunculkan total tagihan pelanggan tersebut. Setelah proses ini akan disediakan pilihan “Apakah anda ingin melakukan pembayaran?” dengan opsi [Y] untuk ya dan [N] untuk tidak. Jika pengguna memilih [Y] pengguna akan diarahkan untuk memasukkan nomor kartu kredit dan nominal pembayaran dengan syarat nominal pembayaran > minimal (15% dari total tagihan) dan nominal pembayaran < total tagihan, jika syarat tidak terpenuhi akan muncul “Input tidak valid, nominal minimal Rp (15% dari tagihan) maksimal Rp (total tagihan) sehingga akan diarahkan kembali untuk memasukkan nominal pembayaran. Jika syarat terpenuhi program akan menginisialisasi sisa tagihan dengan mengurangi total tagihan dengan nominal pembayaran kemudian program akan memproses pembayaran dengan kartu kredit sehingga akan muncul “Pembayaran dengan kartu kredit berhasil”. Berikutnya jika ternyata sisa tagihan lebih dari 0 program akan menampilkan sisa tagihan dan menyimpan sisa tagihan tersebut ke *database* sehingga akan muncul pada transaksi selanjutnya, setelah itu pengguna akan diarahkan untuk memencet ‘Enter’ pada *keyboard* untuk kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat). Akan tetapi jika sisa tagihan adalah 0 pengguna akan diarahkan untuk memencet ‘Enter’ pada *keyboard* untuk kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat).



Pada opsi Riwayat jika pengguna memilih [1] Riwayat *Postpaid* program akan membuka file riwayat *postpaid* setelah itu program akan menampilkan grafik pembayaran dan grafik pemakaian listrik dengan metode *postpaid*. Setelah itu pengguna akan diarahkan untuk memencet ‘Enter’ pada *keyboard* untuk kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat) kemudian diarah untuk memencet ‘Enter’ pada *keyboard* lagi untuk kembali ke halaman awal (pemilihan antara [1] Masuk, [2] Daftar, dan [3] Keluar) pada halaman awal pengguna dapat memilih opsi [3] Keluar jika ingin keluar dari program sehingga akan muncul “Terima kasih telah menggunakan Tracity” yang menandakan berakhirnya program.

Pada opsi Riwayat jika pengguna memilih [2] Riwayat *Prepaid* program akan membuka file riwayat *prepaid* setelah itu program akan menampilkan grafik pembayaran dan grafik pemakaian listrik dengan metode *prepaid*. Setelah itu pengguna akan diarahkan untuk memencet ‘Enter’ pada *keyboard* untuk kembali ke halaman utama (pemilihan antara ingin melakukan pembayaran atau melihat riwayat) kemudian diarah untuk memencet ‘Enter’ pada *keyboard* lagi untuk kembali ke halaman awal (pemilihan antara [1] Masuk, [2] Daftar, dan [3] Keluar) pada halaman awal pengguna dapat memilih opsi [3] Keluar jika ingin keluar dari program sehingga akan muncul “Terima kasih telah menggunakan Tracity” yang menandakan berakhirnya program.



## BAB III

### FUNGSI KODE PROGRAM

#### 3.1 Penggunaan Modul

Program ini menggunakan beberapa fungsi yang disatukan dalam suatu modul untuk mempermudah jalannya program. Fungsi yang digunakan ada 7 fungsi, berikut merupakan penjelasannya.

##### 1) Fungsi “Fonts”

Merupakan fungsi yang dibuat untuk dapat menampilkan teks dengan font atau warna-warna tertentu. Disediakan 3 jenis font dasar diantaranya bold, italic, dan underline. Sedangkan untuk warna sendiri tersedia 15 warna. Fungsi ini digunakan paling banyak pada *welcoming message*.

```
1 def fonts(value, color:str="", style:str=""):
2     color, style = color.lower(), style.lower()
3     fmt = {
4         "reset" : "\033[0m",
5         "bold" : "\033[1m",
6         "italic" : "\033[3m",
7         "underline" : "\033[4m",
8         "" : "",
9         "black": "\033[30m",
10        "red": "\033[31m",
11        "green": "\033[32m",
12        "yellow": "\033[33m",
13        "blue": "\033[34m",
14        "magenta": "\033[35m",
15        "cyan": "\033[36m",
16        "white": "\033[37m",
17        "orange": "\033[38;2;255;165;0m",
18        "pink": "\033[38;2;255;105;180m",
19        "purple": "\033[38;2;128;0;128m",
20        "sky": "\033[38;2;135;206;250m",
21        "teal": "\033[38;2;0;128;128m",
22        "lime": "\033[38;2;0;255;0m",
23        "brown": "\033[38;2;165;42;42m",
24    }
25     return f"{fmt[style]}{fmt[color]}{value}{fmt['reset']}"
26
```

##### 2) Fungsi “input email”

Fungsi ini berguna untuk menginput email dengan syarat email tidak boleh kosong dan harus memiliki domain gmail.com, yahoo.com, dan outlook.com. Fungsi ini digunakan ketika mendaftarkan akun dan *login*.

```
def input_email(prompt):
    while True:
        try:
            user_input = input(prompt)
            if len(user_input) == 0:
                raise ValueError("Email tidak boleh kosong.")
            elif not user_input.lower().endswith(("gmail.com", "yahoo.com", "outlook.com")):
                raise ValueError("Email tidak valid. Email harus diakhiri dengan gmail.com, yahoo.com,")
            return user_input
        except ValueError as error:
            print(f"Error: {str(error)}")
```

### 3) Fungsi “input password”

Sesuai dengan namanya, fungsi ini dibuat untuk kepentingan user dalam memasukkan password. Password yang dimasukkan harus berupa 5 digit. Penggunaan fungsi ini di *main* program terletak pada daftar akun dan *login*.

```
def password(prompt):
    while True:
        try:
            password = input(prompt)
            if len(password) != 5 or not password.isdigit():
                raise ValueError("Password harus terdiri dari 5 digit.")
            return password
        except ValueError as e:
            print(f"Error: {str(e)}")
```

### 4) Fungsi “Yes or No”

Fungsi ini digunakan untuk memasukkan input hanya berupa “Y” dan “N”. Pada *main* program, fungsi ini digunakan untuk pembayaran baik prepaid dan postpaid

```
def input_yn(prompt):
    while True:
        user_input = input(prompt).lower()
        if user_input == 'y' or user_input == 'n':
            return user_input
        else:
            print("Masukkan hanya dapat berupa 'Y' atau 'N'. Silakan coba lagi.\n")
```

### 5) Fungsi “Input kredit”

Fungsi ini mengharuskan user untuk input 14 hingga 16 digit angka. Jika angka yang dimasukkan sejumlah 14-16 digit, program akan berlanjut. Fungsi ini paling banyak digunakan dalam pembayaran baik prepaid maupun postpaid sebelum menginput nominal pembayaran.

```
def input_kredit(prompt):
    while True:
        try:
            kartu_kredit = input(prompt)
            if len(kartu_kredit) < 14 or len(kartu_kredit) > 16 or not kartu_kredit.isdigit():
                raise ValueError("Nomor kartu kredit harus terdiri dari 14 hingga 16 digit angka.\n")
            return kartu_kredit
        except ValueError as e:
            print(f"Error: {str(e)}")
```

## 6) Fungsi “nominal”

Fungsi ini mengharuskan user untuk input angka selain 0. Fungsi ini dapat digunakan untuk keperluan input nominal pembayaran, dan lain-lain. Contoh penggunaannya adalah ketika user harus memasukkan jumlah pemakaian listrik dalam bentuk kwh.

```
def nominal(prompt):  
    while True:  
        try:  
            variable = float(input(prompt))  
            if variable != 0:  
                return variable  
            else:  
                print('Input tidak boleh 0.')        except ValueError:  
            print('Input harus berupa angka.')
```

## 7) Fungsi “pulsa”

Fungsi ini mengharuskan user untuk input pulsa dengan batas minimal 50.000,00.

```
77  
78 def pulsa(prompt):  
79     while True:  
80         try:  
81             variable = float(input(prompt))  
82             if variable >= 50000:  
83                 return variable  
84             else:  
85                 print('Nominal minimal Rp50.000\n')  
86         except ValueError:  
87             print('Input harus berupa angka.\n')  
88
```

## 3.2 Program Input Data Pengguna

Program input data pengguna adalah program pada halaman pertama dimana pengguna akan mengisi data pribadi mereka sebagai syarat untuk mengakses program. Data pribadi yang dimaksud berupa *email* dan *password* untuk menjaga keamanan data pengguna. Selain itu, pada program ini juga tersaji deskripsi singkat mengenai program dan kegunaannya. Pada program ini terdapat 5 fungsi, yaitu fungsi “Welcome Message”, fungsi “run”, fungsi “home”, fungsi “daftar”, dan fungsi “masuk”. Berikut adalah penjelasannya.

### 1) Fungsi “Welcome Message”

Adalah fungsi yang dibuat untuk menampilkan pesan selamat datang dan pesan singkat mengenai deskripsi program.

```
8 def welcome_message():
9     print()
10    garis = fonts('=*80, color='blue',style='bold')
11    text1 = fonts("SELAMAT DATANG DI PROGRAM TRACITY!", color='yellow', style='bold')
12    text2 = ('Kami adalah Platform untuk membantu payment listrik Anda secara online!')
13    text3 = fonts('Selamat menggunakan program Track Your Electricity!', color='yellow')
14    print(fonts(garis.center(80)))
15    print(fonts(text1.center(90)))
16    print(fonts(text2.center(80)))
17    print(fonts(garis.center(80)))
18    print('''Tracity atau Track Your Electricity ini telah dirancang khusus untuk
19 membantu Anda melacak penggunaan listrik Anda. Dengan program ini, Anda dapat dengan
20 mudah memantau konsumsi listrik Anda dan mengidentifikasi area di mana Anda dapat
21 melakukan penghematan energi''')
22    print(fonts(garis.center(80)))
23    print('''Kami menyediakan berbagai opsi pembayaran untuk kenyamanan Anda dan riwayat
24 yang bisa diakses kapan saja!
25 1. Prepaid
26 2. Postpaid''')
27    print(fonts(garis.center(80)))
28    print(fonts(text3.center(90)))
29    print(fonts(garis.center(80)))
```

## 2) Fungsi “run”

Fungsi “run” adalah fungsi yang dibuat untuk memanggil fungsi “Welcome Message” sehingga fungsi “Welcome Message” dapat ditampilkan saat pengguna melakukan *run* program. Selain itu, fungsi “run” juga digunakan untuk melanjutkan program pada fungsi “home” yang berisi program utama dari program Tracity ini.

```
def run():
    welcome_message()
    print('\n\n===== Track Your Electricity =====')
    home()
```

## 3) Fungsi “home”

Fungsi “home” adalah fungsi yang dibuat untuk menampilkan pilihan terkait pendaftaran data pengguna, memasukkan data pengguna untuk mengakses program, atau memilih untuk keluar dari program.



```
36 def home():
37     while True:
38         print(f"{fonts('[1]', color='pink')} Masuk")
39         print(f"{fonts('[2]', color='pink')} Daftar")
40         print(f"{fonts('[3]', color='pink')} Keluar")
41         try:
42             pilih = input('Silakan pilih : ')
43             option = int(pilih)
44             if option == 1:
45                 masuk()
46                 break
47             elif option == 2:
48                 daftar()
49             elif option == 3:
50                 print("Terima kasih telah menggunakan Tracity!")
51                 exit()
52             else:
53                 raise ValueError
54         except ValueError:
55             print(f"Maaf, pilihan {pilih} tidak tersedia.")
56             print('Silakan coba lagi\n')
```

#### 4) Fungsi “daftar”

Fungsi “daftar” adalah fungsi yang dibuat untuk menjalankan program mengenai pendaftaran data pengguna yang selanjutnya digunakan sebagai persyaratan untuk mengakses program. Data pengguna yang di-*input*-kan akan dimasukkan kedalam sebuah *file.txt* sebagai tempat untuk menampung data-data Pengguna yang lainnya

```
def daftar():
    print('\n===== Daftar Akun =====')
    print('\nIsi data-data berikut dengan benar')
    email_daftar = input_email('Masukkan email :')
    with open("Daftar akun.txt", "r") as file:
        for line in file:
            data = line.strip().split(",")
            if len(data) == 2 and data[0] == email_daftar:
                print("Pendaftaran gagal. Email sudah terdaftar.")
                return
    pw_daftar = password('Masukkan password (5 digit) :')
    with open("Daftar akun.txt", "a") as file:
        file.write(f"{email_daftar},{pw_daftar}\n")
    print('Akun berhasil didaftarkan\n')
    home()
```

#### 5) Fungsi “masuk”

Fungsi “masuk” adalah fungsi yang dibuat untuk menangani proses masuk sebelum pengguna dapat mengakses program secara keseluruhan. Pengguna akan diminta untuk memasukkan *email* dan *program* yang sudah pernah didaftarkan sebelumnya pada

program. Apabila *email* ataupun *programnya* valid, artinya sudah terdaftar dan ada di *file.(txt)* atau *file* tempat penyimpanan data pengguna, pengguna kemudian dapat masuk dan mengakses semua fitur program yang ada.

```
74 def masuk():
75     print('\n===== Masuk =====')
76     print('Silakan Masukkan Akun Anda yang Sudah Terdaftar')
77     global email_login
78     email_login = input_email('Masukkan email yang terdaftar : ')
79     coba = 0
80     while coba < 3:
81         with open("Daftar akun.txt", "r") as file:
82             email_terdaftar = False
83             for line in file:
84                 data = line.strip().split(",")
85                 if len(data) == 2 and data[0] == email_login:
86                     email_terdaftar = True
87                     break
88             if email_terdaftar:
89                 password = input("Masukkan password (5 digit) : ")
90                 with open("Daftar akun.txt", "r") as file:
91                     for line in file:
92                         data = line.strip().split(",")
93                         if len(data) == 2 and data[0] == email_login and data[1] == password:
94                             print("Login berhasil!")
95                             time.sleep(2)
96                             home2()
97                             break
98             else:
99                 print("Password salah. Silakan coba lagi.")
100         else:
101             print("Email tidak terdaftar. Silakan coba lagi.")
102             email_login = input_email('Masukkan email yang terdaftar : ')
103         coba += 1
104     if coba == 3:
105         print("Gagal login. Terlalu banyak percobaan.\n")
106         home()
```

### 3.3 Program Utama

Program utama adalah program yang dibuat agar pengguna dapat memilih fitur yang ingin dijalankan. Pada program utama terdapat satu fungsi yang langsung menghubungkan dengan fungsi-fungsi lainnya yaitu fungsi “home2”.

#### 1) Fungsi “home2”

Fungsi “home2” adalah fungsi utama yang menghubungkan kepada fungsi pada setiap fitur yang ada di program. Sehingga pengguna hanya perlu memasukan perintah berupa hasil pilihan dan program akan melanjutkannya ke fungsi selanjutnya.

```
108 def home2():
109     while True:
110         print()
111         print(f"{fonts('[1]', color='pink')} Pembayaran")
112         print(f"{fonts('[2]', color='pink')} Riwayat")
113         print(f"{fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic')}")
114         try:
115             pilih = input('Silakan pilih : ')
116             if pilih == '1':
117                 pembayaran()
118             elif pilih == '2':
119                 riwayat()
120             elif pilih == '':
121                 os.system('cls')
122                 time.sleep(2)
123                 home()
124             else :
125                 raise ValueError
126         except ValueError:
127             print(f"Maaf, pilihan {pilih} tidak tersedia.")
128             print('Silakan coba lagi\n')
```

### 3.4 Program Pembayaran

Program pembayaran adalah program lanjutan dari fungsi “home2”. Pada program ini, pengguna dapat melakukan pembayaran secara prabayar atau pascabayar. Sebelum masuk ke salah satu metode pembayaran, pengguna dapat memilih metode pembayaran mana yang akan digunakan.

```
130 def pembayaran():
131     while True:
132         print('\n===== Pembayaran =====')
133         print(f"{fonts('[1]', color='pink')} Prepaid")
134         print(f"{fonts('[2]', color='pink')} Postpaid")
135         print(f"{fonts('Tekan enter untuk kembali ke page sebelumnya', color='yellow', style='italic')}")
136         try:
137             pilih = input('Silakan pilih : ')
138             if pilih == '1':
139                 prepaid()
140             elif pilih == '2':
141                 postpaid()
142             elif pilih == '':
143                 os.system('cls')
144                 time.sleep(2)
145                 home2()
146             else :
147                 raise ValueError
148         except ValueError:
149             print(f"Maaf, pilihan {pilih} tidak tersedia.")
150             print('Silakan coba lagi\n')
```

#### 1) Fungsi “Prepaid”

Fungsi “Prepaid” adalah fungsi yang dibuat agar pengguna dapat melakukan pembayaran listrik prabayar, sistem pembayaran dimana pengguna listrik harus membeli kredit listrik sebelum menggunakannya. Pengguna membeli token sesuai dengan tarif kemudian setiap kali pengguna menggunakan listrik, nilai token akan berkurang sesuai dengan konsumsi listrik yang digunakan. Pada fungsi ini, pengguna nantinya akan diminta untuk memasukkan besar pembayaran atas listrik yang ingin dibeli dan program akan

menghitung berapa besar kWh yang didapatkan dari pembayaran tersebut. Pemasukkan data pada program nantinya akan dimasukkan kedalam sebuah *file.(txt)* untuk menyimpan data pengguna.

```
174 def prepaid():
175     email = email_login
176     tanggal = datetime.date.today()
177     pulsa_awal = pulsa("Masukkan pulsa awal : ")
178     token_awal = pulsa_awal/2000
179     print("Token awal Anda : ", token_awal)
180     pemakaian_listrik = nominal("Masukkan jumlah pemakaian listrik dalam kwh : ")
181     bayar = 0
182     sisa_token = 0
183     while pemakaian_listrik > token_awal:
184         print(f"Pulsa anda tidak mencukupi. Pembelian maksimal untuk {token_awal} kwh\n")
185         time.sleep(1)
186         pemakaian_listrik = nominal("Masukkan jumlah pemakaian listrik dalam kwh : ")
187     sisa_token = token_awal - pemakaian_listrik
188     print("Sisa token Anda : ", sisa_token)
189     pembayaran = input_yn("Apakah Anda ingin melakukan pembelian token? (Y/N): ")
190     if pembayaran == "y":
191         nomor_kartu_kredit = input_kredit("Masukkan nomor kartu kredit: ")
192         bayar = pulsa("Masukkan nominal pembayaran: ")
193         token_sekarang = (bayar/2000) + sisa_token
194         print(fonts("\nPembayaran Anda sedang diproses...", color='yellow', style='italic'))
195         time.sleep(2)
196         print(f"Pembayaran kartu kredit {fonts('berhasil', color='blue')}")
197         print("Token anda sekarang :", token_sekarang)
198         ask = input(fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic'))
199         if ask == '':
200             print(fonts("Kembali ke halaman utama...", color='yellow', style='italic'))
201             time.sleep(2)
202             os.system('cls')
203             home2()
204     else:
205         token_sekarang = sisa_token
206         print('Token anda sekarang :', token_sekarang)
207         ask = input(fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic'))
208         if ask == '':
209             print(fonts("Kembali ke halaman utama...", color='yellow', style='italic'))
210             time.sleep(2)
211             os.system('cls')
212             home2()
213     with open('Prepaid.txt', 'a') as file:
214         file.write(f"\n{email},{tanggal},{bayar},{token_sekarang}")
215     exit()
```

## 2) Fungsi “Postpaid”

Fungsi “Postpaid” adalah fungsi yang dibuat agar pengguna dapat melakukan pembayaran listrik secara pascabayar, sistem pembayaran dimana pengguna dapat menggunakan listrik dahulu yang kemudian akan dibayar sesuai dengan tagihan yang diberikan. Pada fungsi ini, pengguna nantinya akan diminta untuk memasukkan besar pemakaian listrik dalam kWh, kemudian sistem akan menghitung besar tagihan yang harus pengguna bayarkan dengan mengalikan besar pemakaian listrik dengan tarif per kWh yang telah ditentukan. Pemasukkan besar penggunaan listrik yang nantinya akan disimpan ke dalam sebuah *file.(txt)* untuk menyimpan data pengguna.

```

217 sisa_tagihan = 0
218
219 def postpaid():
220     email = email_login
221     global sisa_tagihan
222     tanggal = datetime.date.today()
223     tagihan = sisa_tagihan
224     if tagihan > 0:
225         print('=' * 30)
226         print('Tagihan Anda Rp', tagihan)
227         print('=' * 30)
228     kwh = nominal("\nMasukkan jumlah pemakaian listrik dalam kwh      : ")
229     total = kwh * 2000
230     if tagihan > 0:
231         total += tagihan
232     bayar = 0
233     print('Sisa tagihan sebelumnya      Rp', sisa_tagihan)
234     print(fonts(f"Total tagihan Anda      Rp {total}", style='underline'))
235     pembayaran = input_yn("Apakah Anda ingin melakukan pembayaran? (Y/N) : ")
236     if pembayaran == "y":
237         nomor_kartu_kredit = input_kredit("Masukkan nomor kartu kredit : ")
238         minimal = total*(15/100)
239         bayar = nominal('Masukkan nominal pembayaran : ')
240         while bayar < minimal or bayar > total:
241             print(f"Input tidak valid! Nominal minimal Rp{minimal} maksimal Rp{total}\n")
242             time.sleep(1)
243             bayar = nominal('Masukkan nominal pembayaran : ')
244         sisa_tagihan = total - bayar
245         print(fonts("\nPembayaran Anda sedang diproses...", color='yellow', style='italic'))
246         time.sleep(2)
247         print(f"Pembayaran kartu kredit {fonts('berhasil', color='blue')}")
248         if sisa_tagihan <= 0 :
249             print("Tagihan Anda sudah terbayar penuh")
250             ask = input(fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic'))
251
252             if ask == '':
253                 print(fonts("Kembali ke halaman utama...", color='yellow', style='italic'))
254                 time.sleep(2)
255                 os.system('cls')
256                 home2()
257             else:
258                 print("Sisa tagihan Anda sebesar Rp", sisa_tagihan)
259                 ask = input(fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic'))
260                 if ask == '':
261                     print(fonts("Kembali ke halaman utama...", color='yellow', style='italic'))
262                     time.sleep(2)
263                     os.system('cls')
264                     home2()
265                 else:
266                     print("Sisa tagihan Anda sebesar Rp", total)
267                     ask = input(fonts('Tekan enter untuk kembali ke halaman utama', color='yellow', style='italic'))
268                     if ask == '':
269                         print(fonts("Kembali ke halaman utama...", color='yellow', style='italic'))
270                         time.sleep(2)
271                         os.system('cls')
272                         home2()
273                     with open('Postpaid.txt', 'a') as file:
274                         file.write(f"\n{email},{tanggal},{total},{kwh}")
275
276

```

### 3.5 Program Riwayat

Program riwayat adalah program lanjutan dari fungsi “home2”. Pada program ini, pengguna akan disajikan aktivitas pembayaran maupun aktivitas penggunaan listrik baik prepaid maupun postpaid yang ditampilkan dalam bentuk grafik terhadap waktu. Sebelum masuk ke salah satu riwayat metode pembayaran tertentu, pengguna dapat memilih riwayat apa yang akan mereka lihat baik prepaid maupun postpaid.

```

152 def riwayat():
153     while True:
154         print('\n===== Riwayat =====')
155         print(f'{fonts("[1]", color="pink")}: Riwayat Prepaid')
156         print(f'{fonts("[2]", color="pink")}: Riwayat Postpaid')
157         print(fonts('Tekan enter untuk kembali ke halaman sebelumnya', color='yellow', style='italic'))
158         try:
159             pilih = input('Silakan pilih : ')
160             if pilih == '1':
161                 riwayat_prepaid()
162             elif pilih == '2':
163                 riwayat_postpaid()
164             elif pilih == '':
165                 os.system('cls')
166                 time.sleep(2)
167                 home2()
168             else:
169                 raise ValueError
170         except ValueError:
171             print(f'Maaf, pilihan {pilih} tidak tersedia.')
172             print('Silakan coba lagi\n')

```

## 1) Fungsi “Riwayat Prepaid”

Fungsi “Riwayat Pembayaran” adalah fungsi dimana pengguna dapat melihat grafik terkait pemakaian dan pembayaran yang pernah dilakukan oleh pengguna. Fungsi ini dibuat dengan menghubungkan data pada fungsi “Prepaid”. Grafik yang disajikan adalah grafik pembayaran terhadap waktu tertentu dan grafik pemakaian terhadap waktu tertentu.

```

226 def riwayat_prepaid():
227     data1 = []
228     with open('Prepaid.txt', 'r') as file:
229         for line in file:
230             line_data = line.strip().split(',')
231             data1.append(line_data)
232     tanggal = []
233     pembayaran = []
234     for item in data1:
235         tanggal.append(item[1][5:10])
236         pembayaran.append(float(item[2]))
237     plt.subplot(1, 2, 1)
238     plt.plot(tanggal, pembayaran)
239     plt.title("Riwayat Pembayaran")
240     plt.xlabel("Tanggal")
241     plt.ylabel("Pembayaran (dalam rupiah)")
242     plt.show()
243     data2 = []
244     with open('Prepaid.txt', 'r') as file:
245         for line in file:
246             line_data = line.strip().split(',')
247             data2.append(line_data)

```

```
248 tanggal2 = []
249 pemakaian = []
250 for item in data2:
251     tanggal2.append(item[1][5:10])
252     pemakaian.append(float(item[3]))
253 plt.subplot(1, 2, 2)
254 plt.plot(tanggal2, pemakaian)
255 plt.title("Riwayat Pemakaian")
256 plt.xlabel("Tanggal")
257 plt.ylabel("Pemakaian (dalam kwh)")
258 plt.tight_layout()
259 plt.show()
```

## 2) Fungsi “Riwayat Postpaid”

Fungsi “Riwayat Postpaid” adalah fungsi dimana pengguna dapat melihat grafik terkait pemakaian dan pembayaran yang pernah dilakukan oleh pengguna. Fungsi ini dibuat dengan menghubungkan data pada fungsi “Postpaid”. Grafik yang disajikan adalah grafik pembayaran terhadap waktu tertentu dan grafik pemakaian terhadap waktu tertentu.

```
261 def riwayat_postpaid():
262     data1 = []
263     with open('Postpaid.txt', 'r') as file:
264         for line in file:
265             line_data = line.strip().split(',')
266             data1.append(line_data)
267     tanggal = []
268     pembayaran = []
269     for item in data1:
270         tanggal.append(item[1][5:10])
271         pembayaran.append(float(item[2]))
272     plt.subplot(1, 2, 1)
273     plt.plot(tanggal, pembayaran)
274     plt.title("Riwayat Pembayaran")
275     plt.xlabel("Tanggal")
276     plt.ylabel("Pembayaran (dalam rupiah)")
277     plt.show()
278
279     data2 = []
280     with open('Postpaid.txt', 'r') as file:
281         for line in file:
282             line_data = line.strip().split(',')
283             data2.append(line_data)
```



```
284 tanggal2 = []
285 pemakaian = []
286 for item in data2:
287     tanggal2.append(item[1][5:10])
288     pemakaian.append(float(item[3]))
289 plt.subplot(1, 2, 2)
290 plt.plot(tanggal2, pemakaian)
291 plt.title("Riwayat Pemakaian")
292 plt.xlabel("Tanggal")
293 plt.ylabel("Pemakaian (dalam kwh)")
294 plt.tight_layout()
295 plt.show()
```



## BAB IV

### HASIL RUNNING PROGRAM

Program di atas adalah program bernama "Tracity" yang merupakan platform untuk membantu pembayaran listrik secara online. Berikut adalah langkah-langkah dan fungsionalitas program ini:

1. Program dimulai dengan menampilkan pesan selamat datang dan menjelaskan tentang fitur-fitur yang disediakan oleh program ini.

#### SELAMAT DATANG DI PROGRAM TRACITY!

Kami adalah Platform untuk membantu payment listrik Anda secara online!

Tracity atau Track Your Electricity Program ini dirancang untuk membantu Anda melacak penggunaan listrik Anda. Dengan program ini, Anda dapat dengan mudah memantau konsumsi listrik Anda dan mengidentifikasi area di mana Anda dapat menghemat energi

Kami menyediakan berbagai opsi pembayaran untuk kenyamanan Anda dan riwayat yang bisa diakses kapan saja!

1. Prepaid
2. Postpaid

Selamat menggunakan program Track Your Electricity!

2. Pengguna akan diminta untuk memilih opsi untuk masuk, mendaftar, atau keluar.

```
===== Track Your Electricity =====
[1] Masuk
[2] Daftar
[3] Keluar
Silakan pilih      :
```

3. Jika pengguna memilih untuk mendaftar, mereka akan diminta untuk mengisi *email* dan *password* untuk membuat akun baru. Program akan memeriksa apakah *email* sudah terdaftar sebelumnya.

```
[1] Masuk
[2] Daftar
[3] Keluar
Silakan pilih      : 2
```

```
===== Daftar Akun =====
Isi data-data berikut dengan benar
Masukkan email           : taralita@gmail.com
Masukkan password (5 digit) : 24680
Akun berhasil didaftarkan
```

4. Jika pengguna memilih untuk masuk, mereka akan diminta untuk memasukkan email dan password yang sudah terdaftar. Program akan memeriksa kecocokan data dengan data yang tersimpan. Pengguna akan memiliki 3 kesempatan untuk memasukkan password yang benar sebelum gagal login.

```
[1] Masuk
[2] Daftar
[3] Keluar
Silakan pilih      : 1
```

```
===== Masuk =====
Silakan Masukkan Akun Anda yang Sudah Terdaftar
Masukkan email yang terdaftar      : taralita@gmail.com
Masukkan password (5 digit) : 12345
Password salah. Silakan coba lagi.
Masukkan password (5 digit) : 12346
Password salah. Silakan coba lagi.
Masukkan password (5 digit) : 12347
Password salah. Silakan coba lagi.
```

Gagal login. Terlalu banyak percobaan.

```
[1] Masuk
[2] Daftar
[3] Keluar
Silakan pilih      : █
```

5. Setelah berhasil masuk, pengguna akan dibawa ke menu utama.

===== Masuk =====

Silakan Masukkan Akun Anda yang Sudah Terdaftar

Masukkan email yang terdaftar : taralita@gmail.com

Masukkan password (5 digit) : 24680

Login berhasil!

[1] Pembayaran

[2] Riwayat

*Tekan enter untuk kembali ke halaman utama*

Silakan pilih :

6. Di menu utama, pengguna dapat memilih opsi pembayaran atau melihat Riwayat dan juga terdapat opsi untuk kembali ke halaman utama.

[1] Pembayaran

[2] Riwayat

*Tekan enter untuk kembali ke halaman utama*

Silakan pilih :

7. Jika pengguna memilih pembayaran, mereka dapat memilih antara metode pembayaran prepaid atau postpaid.

===== Transaksi =====

[1] Pembayaran

[2] Riwayat

*Tekan enter untuk kembali ke halaman utama*

Silakan pilih : 1

===== Pembayaran =====

[1] Prepaid

[2] Postpaid

*Tekan enter untuk kembali ke page sebelumnya*

Silakan pilih :

8. Jika pengguna memilih prepaid, mereka akan diminta untuk memasukkan pulsa awal. Jika jumlah pulsa awal kurang dari Rp50.000, program akan mengulang kembali dan meminta pengguna untuk memasukkan pulsa awal kembali.

===== Pembayaran =====

[1] Prepaid

[2] Postpaid

*Tekan enter untuk kembali ke page sebelumnya*

Silakan pilih : 1

Masukkan pulsa awal : 100

Nominal minimal Rp50.000

Masukkan pulsa awal :

Selanjutnya program akan meminta pengguna untuk memasukkan jumlah listrik yang digunakan. Kemudian program akan menyertakan sisa token pengguna. Setelah itu program akan menawarkan pembayaran untuk membeli token lagi. Jika pengguna setuju (menjawab: y atau Y) maka program akan meminta nomor kartu kredit dan meminta untuk memasukkan nominal pembayaran. Setelah itu program akan menambahkan sisa token sebelumnya dengan token yang baru dibeli.

```
Masukkan pulsa awal      : 100000
Token awal Anda          : 50.0
Masukkan jumlah pemakaian listrik dalam kwh      : 33
Sisa token Anda: 17.0
Apakah Anda ingin melakukan pembelian token? (Y/N): y
Masukkan nomor kartu kredit: 12638137263726
Masukkan nominal pembayaran: 250000
```

```
Pembayaran Anda sedang diproses...
Pembayaran kartu kredit berhasil
Token anda sekarang: 142.0
```

Setelah pembayaran berhasil, pengguna dapat menekan enter untuk kembali ke halaman utama yaitu halaman “Transaksi”.

*Tekan enter untuk kembali ke halaman utama*  
*Kembali ke halaman utama...*

9. Jika pengguna memilih postpaid, mereka akan diminta untuk memasukkan jumlah pemakaian listrik dalam kWh. Kemudian program akan menampilkan sisa tagihan sebelumnya dan tagihan sekarang yang kemudian akan dijumlahkan oleh program.

```
===== Pembayaran =====
[1] Prepaid
[2] Postpaid
Tekan enter untuk kembali ke page sebelumnya
Silakan pilih      : 2

Masukkan jumlah pemakaian listrik dalam kwh      : 10000
Sisa tagihan sebelumnya      Rp 0
Total tagihan Anda           Rp 20000000.0
```

Saat melakukan pembayaran terdapat syarat bahwa nominal pembayaran yang dibayar minimal 15% dari total tagihan dan maksimal sesuai total tagihan. Jika

ada sisa tagihan, pengguna dapat melakukan pembayaran lagi hingga tagihan pengguna lunas terbayar semua.

```
===== Pembayaran =====
[1] Prepaid
[2] Postpaid
Tekan enter untuk kembali ke page sebelumnya
Silakan pilih      : 2

Masukkan jumlah pemakaian listrik dalam kwh      : 100
Total tagihan Anda                               Rp 200000.0
Apakah Anda ingin melakukan pembayaran? (Y/N)    : y
Masukkan nomor kartu kredit : 162316241365142
Masukkan nominal pembayaran : 10000
Input tidak valid! Nominal minimal Rp30000.0 maksimal Rp200000.0

Masukkan nominal pembayaran : 100000

Pembayaran Anda sedang diproses...
Pembayaran kartu kredit berhasil
Sisa tagihan Anda sebesar Rp 100000.0
Tekan enter untuk kembali ke halaman utama
```

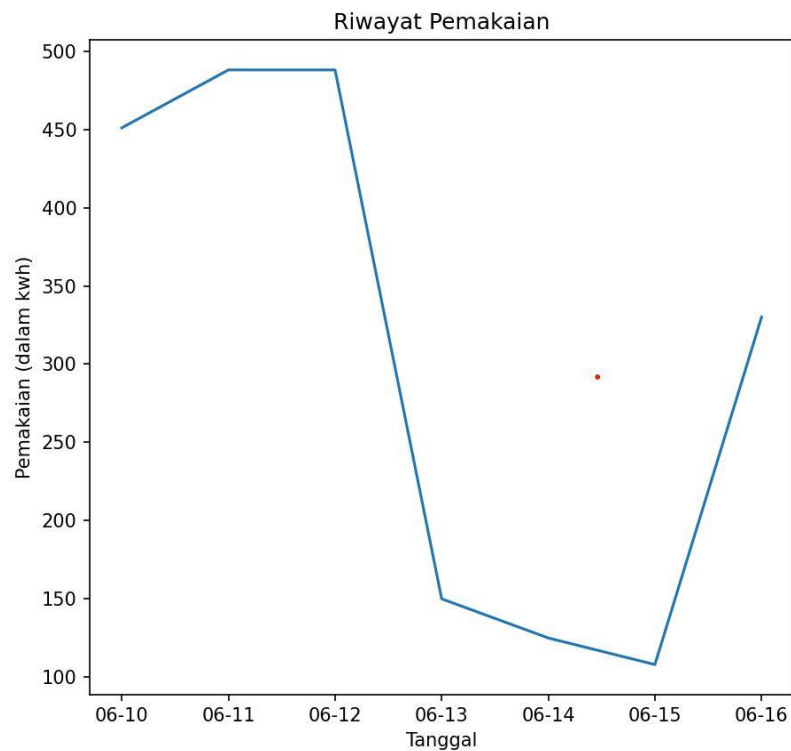
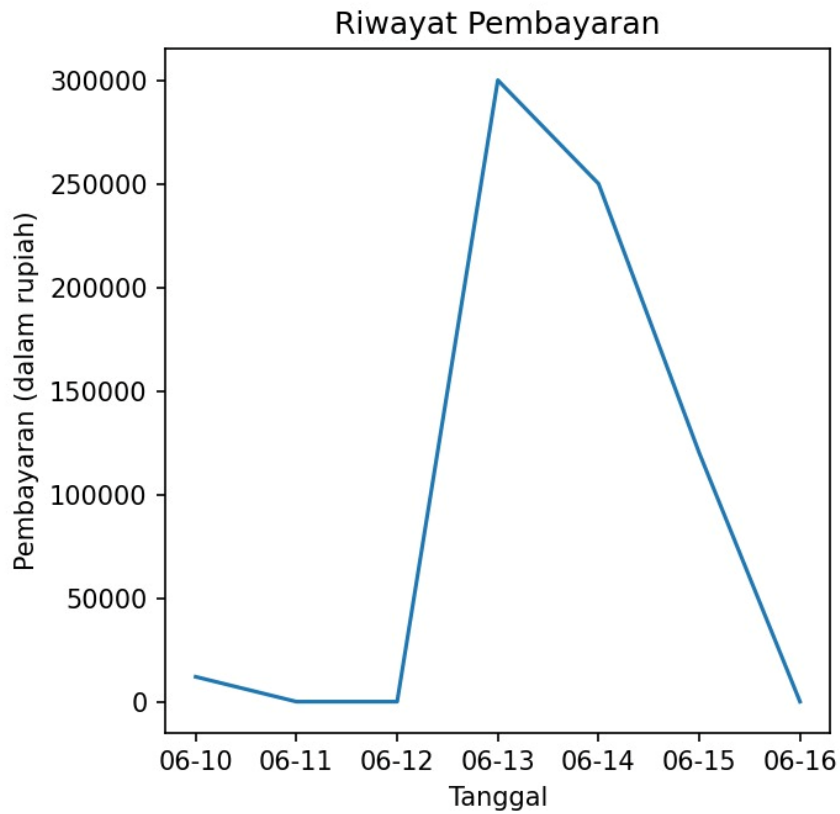
Jika ada sisa tagihan pada transaksi sebelumnya, sistem akan mencatatnya sehingga pada transaksi selanjutnya pengguna dapat melakukan pembayaran hingga tagihan pengguna lunas terbayar semua.

```
===== Pembayaran =====
[1] Prepaid
[2] Postpaid
Tekan enter untuk kembali ke page sebelumnya
Silakan pilih      : 2
=====
Tagihan Anda Rp 100000.0
=====

Masukkan jumlah pemakaian listrik dalam kwh      : 10
Sisa tagihan sebelumnya                          Rp 100000.0
Total tagihan Anda                               Rp 120000.0
Apakah Anda ingin melakukan pembayaran? (Y/N)    : y
Masukkan nomor kartu kredit : 162316241365142
Masukkan nominal pembayaran : 120000

Pembayaran Anda sedang diproses...
Pembayaran kartu kredit berhasil
Tagihan Anda sudah terbayar penuh
```

10. Pengguna juga dapat melihat riwayat pembayaran dan riwayat pemakaian listrik. Riwayat akan ditampilkan dalam bentuk grafik menggunakan library matplotlib.



Itulah gambaran umum tentang hasil *running* program “Tracity”. Program ini dapat membantu pengguna dalam melacak penggunaan listrik dan melakukan pembayaran dengan lebih mudah dan efisien secara *online*.