

APLIKASI CEK MUTASI BANK BERBASIS *WEB*
(STUDI KASUS: BCA DAN BTN)

LAPORAN PROYEK AKHIR



Oleh:
NINGRUM FADILLAH
NIM. 361755401162

PROGRAM STUDI DIPLOMA III
TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
2021

**APLIKASI CEK MUTASI BANK BERBASIS *WEB*
(STUDI KASUS: BCA DAN BTN)**

LAPORAN PROYEK AKHIR



**Proyek Akhir Ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat
Kelulusan Program Studi Diploma III Teknik Informatika dan Mencapai
Gelar Ahli Madya (A.Md)**

**NINGRUM FADILLAH
NIM. 361755401162**

**PROGRAM STUDI DIPLOMA III
TEKNIK INFORMATIKA
POLITEKNIK NEGERI BANYUWANGI
2021**

--Halaman ini sengaja dikosongkan--

PERSEMBAHAN

Alhamdulillah puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga Proyek Akhir ini dapat diselesaikan pada waktu yang telah ditentukan.

Proyek akhir ini saya persembahkan untuk:

1. Bapak Paisol dan Ibu Nurhidayah selaku kedua orang tua yang telah senantiasa mendoakan serta memberikan dukungan dan semangat.
2. Kakak Nurul Habibi dan Adik Badarudin Yusuf yang selalu mendukung dan menyemangati saya.
3. Ibu Dianni Yusuf, S.Kom., M.Kom. selaku Dosen pembimbing 1 dan Bapak Sepyan Purnama K, S.Kom., M.Kom. selaku Dosen pembimbing 2. Terima kasih telah meluangkan waktu, memberikan ilmu, membagikan ilmu dan pengalamannya, dan kesabaran untuk membimbing saya demi kelancara Proyek Akhir ini.
4. Dosen-dosen Politeknik Negeri Banyuwangi khususnya Dosen Program Studi Teknik Informatika yang telah ikut membimbing dalam pengerjaan proyek Proyek Akhir.
5. Almamater Politeknik Negeri Banyuwangi yang saya banggakan.
6. Seluruh teman-teman mahasiswa Politeknik Negeri Banyuwangi khususnya Program Studi Teknik Informatika seperjuangan.

--Halaman ini sengaja dikosongkan--

MOTTO

“Banyak hal yang bisa menjatuhkanmu. Tapi satu-satunya hal yang benar-benar dapat menjatuhkanmu adalah sikapmu sendiri”

-RA. Kartini-

--Halaman ini sengaja dikosongkan--

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Ningrum Fadillah

NIM : 361755401162

Menyatakan dengan sesungguhnya bahwa proyek akhir yang berjudul “Aplikasi Cek Mutasi Bank Berbasis *Web* (Studi Kasus: BCA dan BTN)” adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan atau plagiat. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Banyuwangi,

Yang Menyatakan

Ningrum Fadillah

NIM. 361755401162

--Halaman ini sengaja dikosongkan--

APLIKASI CEK MUTASI BANK BERBASIS WEB
(STUDI KASUS: BCA, DAN BTN)

Proyek Akhir ini disusun untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Ahli Madya (A.Md)
Politeknik Negeri Banyuwangi

Oleh :
NINGRUM FADILLAH
NIM. 361755401162

Tanggal Ujian : 25 FEB 2021

Menyetujui,

Pembimbing 1 : Dianni Yusuf, S.Kom., M.Kom.

()

Pembimbing 2 : Sepyan Purnama K, S.Kom., M.Kom.

()

Penguji 1 : Lutfi Hakim, S.Pd., M.T.

()

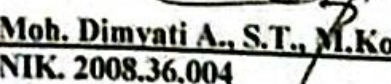
Penguji 2 : Muh Fuad Al Haris, S.T., M.T

()

Mengesahkan,
Ketua Jurusan
Teknik Informatika,

Mengetahui,
Koordinator Program Studi
D3 Teknik Informatika,


Eka Misfiko Rini, S.Kom., M.kom
NIP. 198310202014042001


Moh. Dimyati A., S.T., M.Kom.
NIK. 2008.36.004

--Halaman ini sengaja dikosongkan--

APLIKASI CEK MUTASI BANK BERBASIS WEB (STUDI KASUS: BCA DAN BTN)

Nama Mahasiswa : Ningrum Fadillah
NIM : 361755401162
Dosen Pembimbing : 1. Dianni Yusuf, S.Kom., M.Kom.
2. Sepyan Purnama K, S.Kom., M.Kom.

ABSTRAK

Perbankan merupakan lembaga keuangan yang bergerak dalam mengelola jasa manajemen keuangan masyarakat. Kecepatan, kemudahan dan keamanan merupakan salah satu bentuk layanan yang harus diberikan oleh lembaga perbankan kepada nasabahnya. Perubahan mendasar dari industri perbankan adalah perpindahan dari bank tradisional menjadi *electronic banking*. *Internet banking* merupakan salah satu layanan dari *electronic banking* yang banyak digunakan oleh nasabah karena dapat melakukan hampir semua jenis transaksi perbankan. Situs *web Internet banking* yang digunakan pada saat ini hanya dapat digunakan untuk satu akun bank saja. Untuk nasabah yang memiliki dua akun bank yang berbeda harus membuka halaman *web* sesuai dengan bank yang digunakan. Oleh karena itu, dibuatnya Aplikasi Cek Mutasi Bank Berbasis Web yang dapat digunakan nasabah untuk mengecek transaksi debit dan kredit dua akun bank yang berbeda dalam satu halaman *website*. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP dan *framework* laravel 7 dengan basis data MySQL. Teknik pengambilan data mutasi bank menggunakan metode *scraping* dengan *library* selenium dan beautifulsoup.

Kata kunci: Mutasi, *Internet Banking*, Perbankan

--Halaman ini sengaja dikosongkan--

WEB-BASED BANK MUTATION CHECK APPLICATION
(CASE STUDY: BCA AND BTN)

By : Ningrum Fadillah
Student Identify Number : 361755401162
Supervisor : 1. Dianni Yusuf, S.Kom., M.Kom.
2. Sepyan Purnama K, S.Kom., M.Kom.

ABSTRACT

Banking is a financial institution engaged in managing public financial management services. Speed, convenience and security are among the forms of services that banking institutions must provide to their customers. A fundamental change from the banking industry is the move from traditional banks to electronic banking. Internet banking is one of the services of electronic banking that is widely used by customers because it can do almost any type of banking transaction. Internet banking websites used at this time can only be used for one bank account only. For customers who have two different bank accounts must go to the web page according to the bank used. Therefore, a Web-Based Bank Mutation Check Application is created that customers can use to check debit and credit transactions of two different bank accounts in one page of the website. The application was created using the PHP programming language and the laravel 7 framework with the MySQL database. Bank mutation data retrieval technique using scraping method with selenium and beautifulsoup library.

Keywords: *Mutation, Internet Banking, Banking*

--Halaman ini sengaja dikosongkan—

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Proyek Akhir yang berjudul “Aplikasi Cek Mutasi Bank Berbasis *Web* (Studi Kasus: BCA dan BTN)”. Proyek Akhir ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Diploma Tiga (D3) pada Program Studi Teknik Informatika Politeknik Negeri Banyuwangi.

Penyusunan Proyek Akhir ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terimakasih kepada :

1. Bapak Paisol dan Ibu Nurhidayah selaku kedua orang tua saya, yang tidak putus mendoakan dan memberi dukungan selama ini.
2. Bapak Son Kuswadi, Dr. Eng. selaku direktur Politeknik Negeri Banyuwangi.
3. Ibu Dianni Yusuf, S.kom., M.Kom. dan Bapak Sepyan Purnama K, S.Kom., M.Kom. selaku pembimbing yang telah memberikan banyak ilmu dan dukungan kepada penulis.
4. Sahabat dan relasi yang mendoakan dan memberi dukungan selama proses pengerjaan proyek akhir.
5. Serta semua pihak yang terkait yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari kata sempurna, oleh karena itu penulis sangat mengharapkan segala kritik dan saran dari semua pihak yang bersifat membangun. Penulis berharap Proyek Akhir ini dapat bermanfaat khususnya bagi penulis dan umumnya bagi pembaca.

Banyuwangi,

Ningrum Fadillah

NIM. 361755401162

--Halaman ini sengaja dikosongkan--

DAFTAR ISI

SAMPUL LUAR.....	i
HALAMAN JUDUL.....	i
PERSEMBAHAN	iii
MOTTO.....	v
PERNYATAAN.....	vii
PENGESAHAN	ix
ABSTRAK	xi
ABSTRACT.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.5 Batasan Masalah.....	2
BAB 2 TINJAUAN PUSTAKA	3
2.1 Dasar Teori Pendukung	3
2.1.1 <i>Internet Banking</i>	3
2.1.2 Mutasi rekening.....	3
2.1.3 <i>Web Scraping</i>	4
2.1.4 <i>Framework Laravel</i>	5
2.1.5 <i>Library Selenium</i>	8
2.1.6 <i>Library BeautifulSoup</i>	8
2.1.7 XAMPP	9
2.1.8 <i>My Structured Query Language (MySQL)</i>	9
2.1.9 <i>Unified Modeling Language (UML)</i>	10
2.1.10 <i>Metode Extrame Programing (XP)</i>	12
2.1.11 <i>Black Box Testing</i>	13
2.2 Penelitian Terkait.....	14
BAB 3 METODE PENELITIAN.....	17
3.1 Waktu dan Tempat Penelitian	17
3.1.1 Jadwal Penelitian.....	17
3.1.2 Tempat Penelitian.....	17
3.2 Metode Pengembangan Aplikasi	17
3.2.1 Perencanaan (<i>Planning</i>)	18
3.2.2 Desain sistem (<i>Design</i>)	19
3.2.3 Pembuatan Aplikasi (<i>Coding</i>).....	19
3.2.4 Pengujian (<i>Testing</i>)	20
3.2.5 <i>Software Increment</i> (Peningkatan Perangkat Lunak).....	20
3.3 Gambaran Umum Sistem	20
3.3.1 Gambaran Umum Saat Ini.....	20
3.3.2 Gambaran Umum yang diusulkan.....	21
3.4 Spesifikasi Aplikasi	21
3.4.1 Aturan Sistem.....	21

3.4.2	Struktur Menu.....	22
3.4.3	<i>Use Case Diagram</i>	23
3.4.4	<i>Activity Diagram</i>	25
3.5	Mock-up.....	26
3.5.1	Halaman Utama Aplikasi.....	26
3.5.2	Halaman <i>Login</i> Aplikasi.....	27
3.5.3	Halaman <i>Register</i>	27
3.5.4	Halaman Beranda Nasabah.....	28
3.5.5	Halaman <i>Login Internet Banking</i>	28
3.5.6	Halaman Cek Mutasi	29
3.5.7	Halaman Data Nasabah	30
3.5.8	Halaman Riwayat	30
3.6	<i>Entity Relationship Diagram</i> (ERD)	31
BAB 4 HASIL DAN PEMBAHASAN		33
4.1	Hasil	33
4.1.1	Tampilan Halaman Awal Aplikasi	33
4.1.2	Tampilan Halaman <i>Login</i> Aplikasi.....	34
4.1.3	Tampilan Halaman <i>Register</i>	35
4.1.4	Tampilan Halaman Beranda Nasabah	35
4.1.5	Tampilan Halaman <i>Login Internet Banking</i>	36
4.1.6	Tampilan Halaman Cek Mutasi	37
4.1.7	Tampilan Halaman ATM Terdekat	38
4.1.8	Tampilan Halaman Data Nasabah	38
4.1.9	Tampilan Halaman Riwayat	39
4.2	Pembahasan.....	40
4.2.1	Proses <i>Scraping</i> BCA	41
4.2.2	Proses <i>Scraping</i> BTN	49
4.2.3	<i>Cleaning</i> Data.....	60
4.2.4	Menampilkan Data Mutasi	62
4.2.5	Presentase Mutasi	67
4.2.6	Unduh PDF	68
4.2.7	Unduh Excel	70
4.2.8	Waktu Terakhir <i>Login</i> Nasabah.....	72
4.2.9	ATM Terdekat	74
4.3	Pengujian.....	75
4.3.1	Pengujian <i>Login</i> Aplikasi	75
4.3.2	Pengujian Cek Mutasi Bank	78
4.3.3	Tabel Pengujian <i>Black Box</i>	81
BAB 5 PENUTUP		85
5.1	Kesimpulan	85
5.2	Saran	85
DAFTAR PUSTAKA.....		87

DAFTAR GAMBAR

Gambar 2.1 Struktur MVC.....	6
Gambar 2.2 <i>User Interface XAMPP</i>	9
Gambar 2.3 <i>eXtremme Programming (XP)</i>	12
Gambar 3.1 Alur Pengambilan Data BCA	18
Gambar 3.2 Alur Pengambilan Data BTN	19
Gambar 3.3 Gambaran Umum yang Berjalan.....	20
Gambar 3.4 Gambaran Umum yang Diusulkan.....	21
Gambar 3.5 Struktur Menu Aplikasi Cek Mutasi Bank.....	22
Gambar 3.6 <i>Use Case</i> Aplikasi Cek Mutasi Bank	23
Gambar 3.7 <i>Activity Diagram Login</i>	25
Gambar 3.8 <i>Activity Diagram</i> Cek Mutasi Bank	26
Gambar 3.9 Halaman Utama Aplikasi	27
Gambar 3.10 Tampilan Halaman <i>Login</i>	27
Gambar 3.11 Tampilan Halaman <i>Register</i>	28
Gambar 3.12 Tampilan Halaman Beranda Nasabah	28
Gambar 3.13 Tampilan Halaman <i>Login Internet Banking</i>	29
Gambar 3.14 Tampilan Halaman Cek Mutasi.....	29
Gambar 3.15 Tampilan Halaman Data Nasabah.....	30
Gambar 3.16 Tampilan Halaman Riwayat	30
Gambar 3.17 <i>ER-Diagram</i>	31
Gambar 4.1 Halaman Awal Aplikasi	33
Gambar 4.2 Halaman <i>Login</i>	34
Gambar 4.3 Halaman Register	35
Gambar 4.4 Halaman Beranda Nasabah	36
Gambar 4.5 Halaman <i>Login Internet Banking</i> BCA.....	36
Gambar 4.6 Halaman Cek Mutasi.....	37
Gambar 4.7 ATM Terdekat.....	38
Gambar 4.8 Halaman Data Nasabah	39
Gambar 4.9 Halaman Riwayat Nasabah	39
Gambar 4.10 <i>Login Internet Banking</i> BCA	41
Gambar 4.11 <i>Routing Login Internet Banking</i> BCA.....	41
Gambar 4.12 Fungsi cek pada <i>BCAController</i>	42
Gambar 4.13 Menangkap Nilai yang Dikirim <i>Controller</i>	43
Gambar 4.14 <i>login Internet Banking</i> BCA	44
Gambar 4.15 Pengisian Form <i>Login Internet Banking</i> BCA	45
Gambar 4.16 Menekan Beberapa Menu.....	45
Gambar 4.17 <i>Account Information</i>	46
Gambar 4.18 <i>Account Statement</i>	46
Gambar 4.19 <i>Monthly Statemen</i>	47
Gambar 4.20 <i>View Account Statement</i>	47
Gambar 4.21 Tabel Mutasi.....	48
Gambar 4.22 <i>Scraping</i> Data Mutasi dan <i>Export Data</i> Json	48
Gambar 4.23 <i>Logout Internet Banking</i> BCA	49
Gambar 4.24 Halaman <i>Logout Internet Banking</i> BCA.....	49
Gambar 4.25 <i>Login Internet Banking</i> BTN.....	50

Gambar 4.26 <i>Routing Login Internet Banking</i> BTN	50
Gambar 4.27 Fungsi Cek Pada BTNController	51
Gambar 4.28 Menangkap Nilai yang Dikirim <i>Controller</i>	52
Gambar 4.29 <i>Login Internet Banking</i> BTN	53
Gambar 4.30 Pengisian <i>form login Internet Banking</i> BTN	54
Gambar 4.31 Mengambil Pertanyaan	55
Gambar 4.32 <i>Source Code</i> Menjawab Pertanyaan <i>Pop-up</i>	56
Gambar 4.33 Menjawab Pertanyaan <i>Pop-up</i>	57
Gambar 4.34 Menekan Beberapa Menu, <i>Scraping</i> dan <i>Export Data</i> Json	57
Gambar 4.35 Menu Rekening	58
Gambar 4.36 Menu Semua Rekening	58
Gambar 4.37 Nama Pengguna	59
Gambar 4.38 Tabel Mutasi BTN	59
Gambar 4.39 <i>Logout</i> BTN	60
Gambar 4.40 Data Awal Mutasi BCA	60
Gambar 4.41 <i>Source Code Cleaning</i> Data Mutasi BCA	60
Gambar 4.42 Hasil <i>Cleaning</i> Data Mutasi BCA	61
Gambar 4.43 Data Awal Mutasi BTN	61
Gambar 4.44 <i>Source Code Cleaning</i> Data Mutasi BTN	61
Gambar 4.45 Hasil <i>Cleaning</i> Data Mutasi BTN	62
Gambar 4.46 Memasukkan Data Json BCA Pada <i>Database</i>	62
Gambar 4.47 Memasukkan Data Json BTN Pada <i>Database</i>	63
Gambar 4.48 Data Mutasi Pada <i>Database</i>	64
Gambar 4.49 <i>Controller</i> CekMutasiController	65
Gambar 4.50 <i>View</i> cekmutasi.blade.php	67
Gambar 4.51 Data yang Dapat Dilihat Nasabah	67
Gambar 4.52 <i>Controller</i> CekMutasiController Membuat Persentase Mutasi	68
Gambar 4.53 <i>View</i> cekmutasi.blade.php untuk Menampilkan Pie Chart	68
Gambar 4.54 <i>Controller</i> PDF	69
Gambar 4.55 Hasil Tampilan Unduh PDF	70
Gambar 4.56 Konfigurasi <i>Model Export Excel</i>	71
Gambar 4.57 Konfigurasi <i>Controller Export Excel</i>	71
Gambar 4.58 Hasil Tampilan Unduh Excel	72
Gambar 4.59 Konfigurasi <i>Event Listener</i>	73
Gambar 4.60 Konfigurasi <i>EventServiceProvider</i>	73
Gambar 4.61 Lokasi Pencarian ATM BCA dan BTN	
Gambar 4.62 titik koordinat ATM BCA dan BTN	
Gambar 4.63 titik lokasi ATM BCA dan BTN	
Gambar 4.64 Tampilan <i>Password</i> Kosong	74
Gambar 4.65 Tampilan <i>Email</i> Kosong	74
Gambar 4.66 Tampilan <i>Email</i> dan <i>Password</i> Salah	75
Gambar 4.67 Tampilan Halaman Utama Admin	75
Gambar 4.68 Tampilan Halaman Utama Nasabah	76
Gambar 4.69 Tampilan Tidak Ada Data Mutasi	76
Gambar 4.70 Tampilan <i>Login Internet Banking</i> BCA	77
Gambar 4.71 Tampilan Data Mutasi BCA	77
Gambar 4.72 Tampilan <i>Login Internet Banking</i> BTN	78
Gambar 4.73 Tampilan Data Mutasi BTN	78

Gambar 4.74 Tampilan Data Mutasi BCA dan BTN	79
----------------------------------------------------	----

--Halaman ini sengaja dikosongkan--

DAFTAR TABEL

Tabel 2.1 Komponen-Komponen <i>Use Case</i> Diagram	11
Tabel 2.2 Komponen-Komponen <i>Activity</i> Diagram.....	12
Tabel 2.3 Penelitian Terkait	14
Tabel 3.1 Jadwal Penelitian Tugas Akhir	17
Tabel 3.2 Aturan Sistem Aplikasi Cek Mutasi Bank	21
Tabel 3.3 Penjelasan Aktor	23
Tabel 3.4 Keterangan <i>Use Case Login</i> Aplikasi Cek Mutasi.....	23
Tabel 3.5 Keterangan <i>Use Case Login Internet Banking</i>	23
Tabel 3.6 Keterangan <i>Use Case</i> Cek Mutasi	24
Tabel 3.7 Keterangan <i>Use Case</i> Melihat Data Nasabah	24
Tabel 3.8 Keterangan <i>Use Case</i> Melihat Riwayat Nasabah	24
Tabel 3.9 Keterangan <i>Use Case scraping</i>	24
Tabel 4.1 Pengujian <i>Black Box</i>	79

--Halaman ini sengaja dikosongkan--

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perbankan merupakan lembaga keuangan yang bergerak dalam mengelola jasa manajemen keuangan masyarakat. Kecepatan, kemudahan dan keamanan merupakan salah satu bentuk layanan yang harus mampu diberikan oleh lembaga perbankan kepada nasabahnya. Perubahan mendasar dari industri perbankan adalah perpindahan dari bank tradisional menjadi *electronic banking*. *Electronic banking* pada dasarnya memiliki sejumlah layanan yaitu Anjungan Tunai Mandiri (ATM), kartu debit, layanan pesan singkat (*mobile banking*), telepon (*call center/ phone banking*) dan internet (*internet banking*). Salah satu layanan *electronic banking* yang menjadi andalan sebuah bank kini adalah *internet banking*. Hadirnya *internet banking* harus diakui banyak membantu masalah pelayanan. Layanan *internet banking* memungkinkan nasabah melakukan hampir semua jenis transaksi perbankan melalui internet, khususnya melalui situs *website*. Lewat sarana ini, setiap orang dapat melakukan pengecekan rekening, transfer dana, pembelian voucher telepon seluler, hingga pembayaran tagihan rekening listrik, telepon, dan air (Hartawan, 2017).

Situs *web Internet banking* yang digunakan pada saat ini hanya dapat digunakan untuk satu akun bank saja. Untuk nasabah yang memiliki dua akun bank yang berbeda harus membuka halaman *web* sesuai dengan bank yang digunakan. Sedangkan untuk mengakumulasi semua transaksi debit dan kredit dari beberapa akun bank yang berbeda tersebut nasabah harus menghitungnya satu persatu. Hal ini cukup menyulitkan nasabah yang memerlukan data akumulasi debit dan kredit secara langsung dari beberapa akun bank berbeda yang dimilikinya.

Bedasarkan permasalahan yang telah diuraikan, maka diperlukan adanya solusi yang dapat digunakan untuk dapat mengecek transaksi debit dan kredit dua akun bank yang berbeda dalam satu halaman *website*. Maka pada proyek akhir ini diangkat sebuah judul “Aplikasi Cek Mutasi Bank Berbasis *Web* (Studi Kasus: BCA dan BTN)”. Aplikasi ini dibuat dengan menggunakan metode *scraping*, yang merupakan teknik pengambilan data dengan masuk ke suatu halaman

website dan mengambil data tertentu dengan memanfaatkan element-element HTML atau XHTML.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka terdapat masalah yang dapat dirumuskan yaitu bagaimana cara membuat aplikasi cek mutasi bank pada bank BCA dan BTN?

1.3 Tujuan

Tujuan dari pembuatan aplikasi cek mutasi bank adalah untuk mengecek informasi debit dan kredit dari akun bank yang berbeda.

1.4 Manfaat

Adapun manfaat dari adanya aplikasi ini adalah untuk mempermudah nasabah yang memiliki dua akun bank yang berbeda dalam mengecek informasi debit dan kredit tanpa perlu membuka dua halaman *web internet banking*.

1.5 Batasan Masalah

Adapun batasan masalah dari proyek akhir ini adalah sebagai berikut:

1. Aplikasi ini hanya digunakan untuk dua bank yaitu : BCA dan BTN.
2. Aplikasi ini hanya memberikan informasi transaksi debit dan kredit.
3. Pengambilan data menggunakan Teknik *scraping* dengan *library* selenium dan beautifulsoup.
4. Data mutasi yang dapat diakses untuk BCA hanya pada interval satu bulan sebelumnya dan BTN pada interval 10 transaksi terakhir.
5. Sistem aplikasi menggunakan Bahasa pemrograman PHP (*Framework* Laravel) dan menggunakan *database* MySQL.
6. Autentifikasi *login* aplikasi menggunakan *email* dan *password*.

BAB 2

TINJAUAN PUSTAKA

2.1 Dasar Teori Pendukung

2.1.1 *Internet Banking*

Internet Banking merupakan salah satu bentuk produk jasa yang mulai banyak ditawarkan oleh bank-bank komersial di Indonesia. Pada saat ini penggunaan *Internet Banking* bagi nasabah meluas karena memberikan manfaat yang sangat besar bagi kelancaran proses-proses transaksi. proses transfer, penarikan, dan pembayaran transaksi yang dilakukan melalui *Internet Banking* akan mempermudah dan mempercepat penyelesaian transaksi tersebut. Salah satu produk berbasis internet yang saat ini menjadi perhatian kalangan perbankan dunia adalah *internet banking* (Yan Evi, 2018).

Internet banking memberikan solusi penghematan biaya operasional (*cost effective*) dalam penggunaannya dibandingkan dengan saluran lainnya. Dikarenakan *internet banking* mampu mengurangi biaya transaksi ke titik terendah yaitu dapat menghemat 79% biaya dibandingkan dengan biaya transaksi perbankan yang lainnya (Sanusi, 2000). Banyak layanan yang telah ditawarkan pihak bank kepada nasabah dalam hal ini seperti sms *banking*, *mobile banking*. Berdasarkan OJK (Otoritas Jasa Keuangan) mencatat bahwa jumlah pengguna *e-banking* meningkat sebesar 270 % dari 13,6 juta nasabah pada tahun 2012 menjadi 50,4 juta nasabah pada tahun 2016. Peningkatan terjadi dikarenakan adanya kebutuhan masyarakat dalam memanfaatkan teknologi (Yan Evi, 2018).

BCA memiliki layanan jasa perbankan internasional sedangkan BTN merupakan himpunan bank negara yang memiliki banyak nasabah sehingga dimungkinkan pengguna *internet banking*-nya juga besar.

2.1.2 Mutasi rekening

Pengertian mutasi rekening adalah sebuah laporan atas seluruh aktifitas yang terjadi di dalam rekening tersebut. Selama kurun waktu tertentu. Aktifitas ini dapat berupa dana masuk maupun dana keluar. Beberapa fungsi atau manfaat dari mutasi rekening bagi pemilik rekening bank:

1. Sebagai alat yang membantu berlangsungnya kelancaran sebuah transaksi dalam sebuah bisnis (ini berlaku bagi mereka yang melakukan bisnis online dan menerima pembayaran melalui transfer bank).
2. Sebagai kontrol atas semua transaksi yang telah dilakukan atau dijalankan selama kurun waktu tertentu. Terutama bagi mereka yang banyak melakukan transaksi melalui *internet banking* dan *mobile banking* hal ini sekaligus sebagai indikator pengaman sebagai transaksi yang mungkin dilakukan oleh pihak lain terhadap rekening bank anda tanpa sepengetahuan anda (Nafisan, 2018).

2.1.3 *Web Scraping*

Web Scraping adalah Teknik yang digunakan dalam proses pengambilan sebuah dokumen semi ter-stuktur dari internet, umumnya berupa halaman *web* dalam bahasa markup seperti HTML atau XHTML, dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain. *Web Scraping* dikenal sebagai *screen scraping*. *Web Scraping* tidak dapat dimasukkan dalam bidang data *mining* karena data *mining* menyiratkan upaya untuk memahami pola semantik atau tren dari sejumlah besar data yang telah diperoleh. Aplikasi *web scraping* (juga disebut *intelligent, automated, or autonomous agents*) hanya fokus pada cara memperoleh data melalui pengambilan dan ekstraksi data dengan ukuran data yang bervariasi (Ahmat Josi, 2014).

Implementasi Ekstraksi *Web (Web Scraping)* memiliki sejumlah langkah :

1. *Create Scraping Template:*
Pembuat program mempelajari dokumen HTML dari *website* yang akan diambil informasinya untuk *tag* HTML yang mengapit informasi yang akan diambil.
2. *Explore Site Navigation:*
Pembuat program mempelajari teknik *navigasi* pada *website* yang akan diambil informasinya untuk ditirukan pada aplikasi *web scraper* yang akan dibuat.

3. *Automate Navigation and Extraction :*

Berdasarkan informasi yang didapat pada langkah 1 dan 2 diatas, aplikasi *web scraper* dibuat untuk mengotomatisasi pengambilan informasi dari *website* yang ditentukan.

4. *Extracted Data and Package History:*

Informasi yang didapat dari langkah 3 (tiga) disimpan dalam tabel-tabel *database*.

Perbedaan *scraping* dan *crawling* yaitu, *scraping* merupakan teknik yang digunakan untuk pengambilan data dari sebuah *website* setelah masuk kedalam halaman *web* dan hanya dapat difokuskan pada satu halaman *web*. Sedangkan *crawling* merupakan teknik yang digunakan untuk memindai beberapa halaman *web* yang kemudian digunakan untuk membuat index dari data yang dicarinya.

Beberapa bank di Indonesia telah membuka API untuk dapat digunakan pihak ketiga sebagai pengembang, salah satu bank dengan API yang dapat digunakan adalah BCA, sedangkan untuk bank lainnya harus bekerjasama dengan pihak bank agar dapat menggunakan API. Dalam satu kali pengambilan data yang dilakukan pada BCA dikenakan tarif sebesar Rp 4.000. Selain membayar API yang disediakan bank umumnya terbatas dan harus melewati keamanan seperti API key sedangkan untuk *scraping* hanya cukup menggunakan *username* dan *password internet banking* untuk pengambilan data pada halaman cek mutasi.

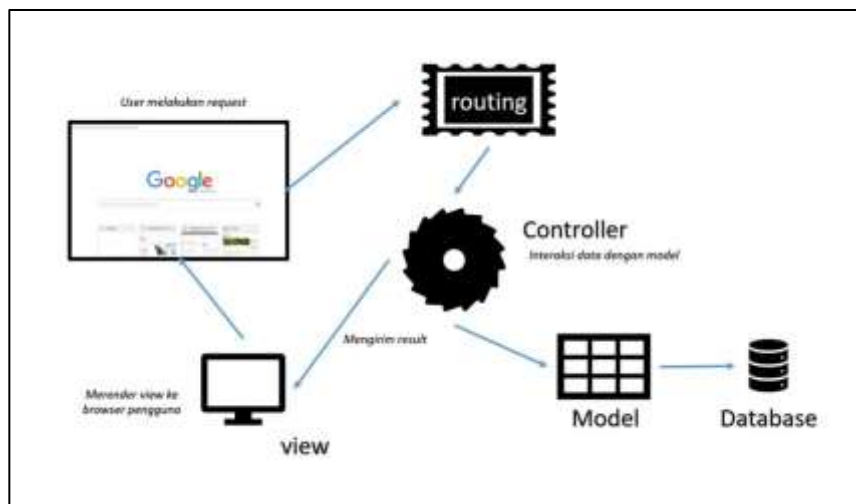
2.1.4 Framework Laravel

Framework adalah kerangka kerja. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer/programmer* dalam menangani berbagai masalah-masalah dalam pemrograman, seperti koneksi ke *database*, pemanggilan *variable*, *file*, dan lain-lain sehingga pekerjaan *developer* lebih fokus dan lebih cepat dalam membangun aplikasi. *Framework* adalah komponen pemrograman yang siap digunakan ulang kapan saja sehingga *programmer* tidak harus membuat *script* yang sama untuk tugas yang sama (Yudhanto Yudho, 2018).

Pada penelitian ini menggunakan *framework* Laravel. Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*). Laravel adalah pengembangan *website*

berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu. Sebagai sebuah *framework* PHP, Laravel hadir sebagai *platform* yang bersifat *open source*. Yang menarik dari Laravel adalah sintaks ekspresif dan elegan, serta dirancang khusus untuk memudahkan dan mempercepat proses *web development* (Yudhanto Yudho, 2018).

Model View Controller (MVC) berguna untuk memisahkan *logic* untuk manipulasi data, antarmuka pengguna dan kontrol aplikasi.



Gambar 2.1 Struktur MVC

- **Model**
Bagian ini digunakan sebagai penghubung antara *database* dan juga *controller*.
- **Controller**
Bagian ini yang mengatur hubungan antara model dan *view*, yang berfungsi menerima *request* dan data dari *user* yang kemudian akan diproses.
- **View**
Bagian ini yang menerima dan merepresentasikan data kepada *user*.

Berikut kekurangan *framework* laravel:

1. Ukuran *file* besar

2. Membutuhkan koneksi Internet untuk instalasi dan mendownload *library*
3. Tidak semua *hosting* cocok dengan Laravel

Berikut alasan mengapa memilih dan menggunakan *framework* laravel:

1. Mudah dan dokumentasi lengkap

Platform menarik dan mudah digunakan. Mudah dipahami strukturnya karena dokumentasi yang disediakan tergolong kedalam dokumentasi yang sangat baik, rapi, mudah dan jelas.

2. *Open source*

Laravel adalah *framework open source* yang dapat digunakan secara bebas, gratis, dan memungkinkan kita untuk membuat *web* aplikasi yang besar dan kompleks dengan mudah.

3. Arsitektur MVC

Dapat membuat struktur kode yang lebih rapi di mana pola tersebut memisahkan antara logika dan *view*. Arsitektur MVC dapat meningkatkan *performance*, memiliki beberapa fungsi *build-in* (fungsi yang dapat langsung digunakan), dan penulisan dokumentasi yang lebih baik.

4. *Blade Template*

Memiliki template yang unik juga sangat cantik sehingga menarik perhatian untuk digunakan. Di dalam dokumentasi resminya, laravel menyebutkan 2 manfaat menggunakan *blade template*, yakni: *inheritance* (dapat diwariskan) dan *section*.

5. Memiliki *fitur migration*

Dengan migrasi memungkinkan untuk mempertahankan struktur *database* yang kita miliki tanpa harus membuatnya kembali. *Migration* memungkinkan kita untuk menuliskan kode PHP untuk mengatur *database*.

6. Keamanan

Laravel memberikan beberapa pilihan penting yang dapat digunakan untuk membuat aplikasi agar tetap aman. ORM Laravel menggunakan PDO yang dapat mencegah *SQL Injection*, memiliki proteksi csrf token, dan banyak hal lainnya.

7. Komunitas yang besar

Merupakan *framework* yang telah didukung komunitas yang besar sehingga memudahkan dalam belajar dan mencari solusi atas setiap permasalahan yang mungkin ditemukan.

8. Hemat waktu

Jika ada *developer* baru yang masuk ke *project*, cukup mempelajari dokumentasi resmi Laravel. Dengan begini akan sangat menghemat waktu.

9. *Quality*

Laravel yang berkualitas baik akan, akan menopang aplikasi yang kita buat dengan kokoh dan kuat (Yudhanto Yudho, 2018).

2.1.5 *Library Selenium*

Selenium merupakan sebuah *library* Python yang memungkinkan kita untuk membuka *browser* yang dipilih dan melakukan beberapa hal seperti halnya manusia, seperti mengklik tombol, meng-*input*-kan data pada form, serta mencari suatu informasi dalam halaman *web*. Meskipun penggunaan selenium dapat menyebabkan keterlambatan dalam waktu *scraping*, namun selenium memiliki kelebihan. Selenium mendukung banyak *browser* dan bahasa pemrograman. Selenium juga mendukung untuk menggunakan *browser web* yang berjalan pada mesin jarak jauh menggunakan selenium grid dan dapat mengatasi javascript yang diberikan laman dinamis (Ram Sharan Chaulagain, 2017).

2.1.6 *Library BeautifulSoup*

BeautifulSoup merupakan sebuah *library* Python yang memungkinkan kita untuk melakukan *scraping* dengan mudah dan cepat. BeautifulSoup bukan *library default* yang ada di Python, jadi harus *diinstall* terlebih dahulu. BeautifulSoup dapat membantu memformat dan mengatur *web* yang berantakan dengan memperbaiki HTML (Mitchell, 2015).

Keunggulan BeautifulSoup adalah;

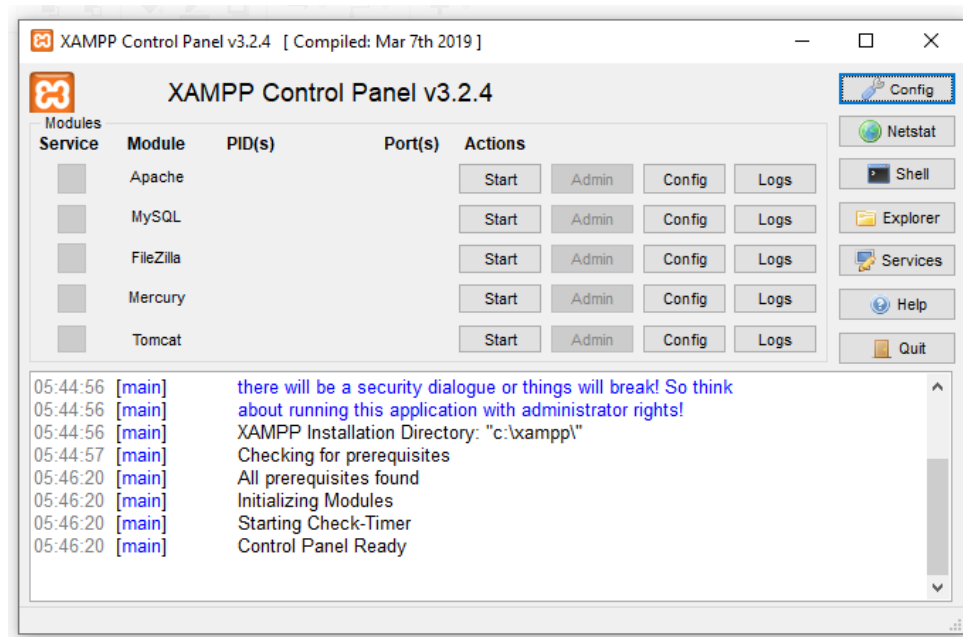
1. BeautifulSoup hadir dengan metode-metode yang sederhana sehingga kita dengan mudah melakukan proses navigasi, pencarian dan juga modifikasi struktur data situs yang kita *scraping*.
2. BeautifulSoup secara otomatis mampu mengkonversi dokumen ke dalam format UTF-8.

3. Bekerja dengan baik dengan *library* Python lxml dan html5lib untuk melakukan parsing dokumen.

2.1.7 XAMPP

XAMPP adalah kompilasi program aplikasi gratis terfavorit di kalangan *developer/programmer* yang berguna untuk pengembangan *website* berbasis PHP dan MySQL. Dengan satu aplikasi ini kita akan mendapatkan paket *software* komplet yang bisa dijalankan pada Windows ataupun Linux sehingga *programmer* dapat dengan mudah melakukan simulasi pada komputer lokal sebelum diunggah ke internet. (Yudhanto Yudho, 2018)

Perangkat lunak ini memiliki kelebihan, yaitu berperan sebagai *server web apache* untuk simulasi pengembangan *website*. Tool pengembangan *web* ini mendukung teknologi *web* populer seperti PHP, MySQL dan Perl.



Gambar 2.2 User Interface XAMPP

2.1.8 My Structured Query Language (MySQL)

MySQL adalah suatu RDBMS (*Relational Database Management System*) yang menjalankan fungsi pengolahan data (*Placeholder1*). MySQL merupakan *software database* yang paling populer saat ini. Program ini berjalan sebagai *server* yang menyediakan *multi-user*, mengakses ke sejumlah *database* baik, *multithread* maupun *multi-user*, dan telah *diinstal* oleh sekitar 6 juta kali diseluruh dunia (Yudhanto Yudho, 2018).

Dalam perancangan aplikasi menggunakan RDBMS MySQL karena dari segi penggunaan dan performa yang cukup cepat. Beberapa perancangan aplikasi yang telah ada sebelumnya banyak menggunakan MySQL karena didukung dengan *tool manager* yang banyak yang salah satunya adalah *phpmyadmin* yang dapat diakses melalui *platform web*.

Kelebihan MySQL:

1. *Free*, stabil, dan tangguh.
2. Fleksibel dengan berbagai pemrograman.
3. *Security* atau keamanan yang baik.
4. Dukungan dari banyak komunitas.
5. Kemudahan management *database*.
6. Mendukung transaksi.
7. Perkembangan *software* cukup cepat.

Kekurangan MySQL:

1. Kurang mendukung koneksi ke Bahasa visual seperti VB, Delphi, Foxpro dikarenakan koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program visual tersebut.
2. Data yang ditangani belum begitu besar.
3. Lambat untuk *query* yang kompleks, seperti LEFT JOIN yang banyak dan penggunaan *SubQuery*.
4. Belum mendukung *Windowing Function*.



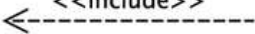
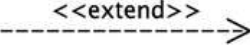



2.1.9 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML mendefinisikan notasi dan *syntax/semantic* seperti bahasa-bahasa lainnya. Membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun yaitu dengan menggunakan UML. UML mempunyai beberapa diagram diantaranya adalah *Use Case Diagram* dan *Activity Diagram* (Hendini, 2016).

1. Use Case Diagram

Dalam sebuah sistem *use case* diagram diharapkan dapat menggambarkan fungsionalitas sebuah sistem. Yang mana menekankan “apa” yang dikerjakan sistem, dan bukan “bagaimana”. Jadi dalam sebuah *use case* itu merepresentasikan sebuah interaksi antara aktor dengan sistem. Pada *Use Case Diagram* terdapat komponen/symbol-simbol yang ditunjukkan pada Tabel 2.1. (Hendini, 2016)






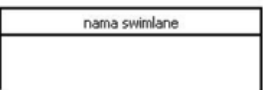
Tabel 2.1 Komponen-Komponen *Use Case Diagram*

	Gambar	Nama	Keterangan
1		<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil terukur bagi suatu aktor.
3		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
4		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
5		<i>Association</i>	Abstraksi dari penghubung antara aktor dan <i>use case</i> .
6		<i>Generalization</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
		<i>System</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

2. Activity Diagram

Activity diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat di modelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas kedalam keadaan sesaat (*state*). Beberapa simbol-simbol yang ada pada diagram *activity diagram* ditunjukkan dalam Tabel 2.2.

Tabel 2.2 Komponen-Komponen *Activity* Diagram

No	Gambar	Nama	Keterangan
1		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
3		Percabangan/ <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		Penggabungan <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.1.10 Metode *Extrame Programing* (XP)

Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang sederhana dan mencakup salah satu metode tangkas yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP adalah salah satu metode tangkas yang paling banyak digunakan dan menjadi pendekatan yang sangat terkenal. Sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium (Supriyatna, 2018).



Gambar 2.3 *eXtremme Programming* (XP)

Pada Gambar 2.3 menunjukkan kerangka kerja dari metode XP. Berikut penjelasannya:

a. *Planning* / Perencanaan

Tahapan ini dimulai dengan pemahaman konteks dari aplikasi, mendefinisikan keluaran (*output*), *fitur* yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, penentuan waktu dan mempertimbangkan biaya pengembangan aplikasi, serta alur proses pengembangan aplikasi.

b. *Design* / Desain

Tahap ini melakukan pembuatan desain aplikasi, meliputi pembuatan basisdata dan *mockup* aplikasi.

c. *Coding* / Penulisan Kode Program

Melakukan pembuatan aplikasi dengan berdasar pada *planning* dan *design* yang telah dilakukan. Dalam kegiatan ini nantinya menghasilkan sebuah aplikasi yang siap untuk dilakukan pengujian.

d. *Testing* / Percobaan dan Pengujian

Tahap ini memfokuskan pada pengujian *fitur – fitur* yang ada pada aplikasi dengan melibatkan calon pengguna sehingga tidak ada kesalahan (*error*), serta aplikasi yang dibuat akan dapat digunakan.

e. *Software Increment* / Peningkatan Perangkat Lunak.

Melakukan proses pengembangan sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambahkan layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem.

2.1.11 Black Box Testing

Black-Box adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing* di bagian luar” (Hendra Rahmadi, 2015).

Beberapa teknik *testing* yang termasuk dalam tipe *Black-Box Testing* antara lain :

1. *Equivalence Partitioning*

Pada teknik ini, tiap inputan data dikelompokkan kedalam *grup* tertentu, yang kemudian dibandingkan *output*-nya.

2. *Boundary Value Analysis*

Pada teknik *Boundary Value Analysis*, dilakukan *input*-an yang melebihi dari batasan sebuah data. Sebagai contoh, untuk sebuah harga barang, maka dapat dilakukan *testing* dengan menggunakan angka negatif (yang tidak diperbolehkan dalam sebuah harga). Jika perangkat lunak berhasil mengatasi *input*-an yang salah tersebut, maka dapat dikatakan teknik ini telah selesai dilakukan.

3. *Cause Effect Graph* Dalam teknik ini, dilakukan proses *testing* yang menghubungkan sebab dari sebuah input dan akibatnya pada *output* yang dihasilkan.

Kelebihan *BlackBox Testing*:

- Tidak perlu melihat *source code* secara detail
- Mendeteksi kesalahan pengetikan / *i*
- Mendeteksi kesalahan *Design / User Interface* dari sebuah *software / website*
- Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
- Seorang *Tester* tidak harus *Programmer*

Kekurangan *BlackBox Testing*:

- Ketergantungan dengan dokumen dan design *software* tersebut
- Tidak sampai *level code*, sehingga tester tidak mengetahui *level security* dari *software* tersebut

2.2 Penelitian Terkait

Tabel 2.3 Penelitian Terkait

No	Nama	Judul	Fitur dan Pembahasan
1.	Vivensius Mitra, Herry Sujaini, Arif Bijaksana Putra Negara (2017)	Rancang Bangun Aplikasi <i>Web Scraping</i> untuk Korpus Paralel Indonesia Inggris dengan Metode HTML	a. Mengambil data dari situs (http://berita2bahasa.com/) b. Pemrosesan data mentah melalui tahap cleaning.

		DOM	tokenisasi, dan lowercased. c. Menghasilkan dokumen yang memiliki isi sama dan ditulis dalam bahasa yang berbeda.
2.	Irlita Tsani Mala Fishiyam (2017)	Penerapan Teknik <i>Web Scraping</i> pada <i>Website</i> Tiket Pesawat Sebagai Perbandingan Harga Tiket Pesawat Berbasis <i>Web Responsive</i>	a. Mengambil data dari situs (traveloka,tiket.com dan Ezytravel.co.id) b. Menerapkan teknik <i>web scraping</i> untuk proses perbandingan harga dan untuk mengintegrasikan data-data tiap <i>website</i> tiket pesawat pada satu halaman <i>web</i> c. <i>Website</i> pencarian tiket pesawat bersifat <i>real time</i>
3.	I Putu Arditya Darmawan, I Nyoman Piarsa, I Putu Arya Dharmaadi (2017)	Ekstrak Hirarki Data Dari Situs <i>Web A-Z Animals</i> Menggunakan <i>Web Scraping</i>	a. Mengambil data dari situs a-z-animal.com b. Aplikasi yang dibuat berupa <i>web base</i> yang diimplementasikan dengan metode <i>web scraping</i> pada aplikasi yang telah dibuat, hasil dari pencarian disimpan ke dalam tabel menggunakan <i>database MySQL</i>
4.	Adentya Maryo Santosa (2016)	Aplikasi Pengecekan Transaksi <i>Internet Banking</i> Menggunakan Metode <i>Web Scraping</i>	a. mengambil data mutasi rekening masing-masing pengguna dan menyimpan data tersebut ke <i>database</i> . b. Aplikasi telah mampu memfasilitasi fitur manajemen pengguna seperti proses <i>register</i> , <i>login</i> , <i>reset</i> dan ubah <i>password</i> , <i>register</i> bank, <i>delete</i> bank, dll.

--Halaman ini sengaja dikosongkan--

BAB 3

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Jadwal Penelitian

Penelitian akan dilakukan selama kurang lebih 9 bulan, mulai bulan Februari 2020 sampai dengan Januari 2021 yang mencakup tahap proses perancangan sampai implementasi sistem. Adapun jadwal pengerjaan tugas akhir tahun akademik 2020/2021 ditunjukkan pada tabel 3.1.

Tabel 3.1 Jadwal Penelitian Tugas Akhir

No	Kegiatan	2020-2021											
		Feb	Mar	Apr	Mei	Jun	Jul	Agu	Sep	Okt	Nop	Des	Jan
1	Perencanaan (<i>Planning</i>)												
2	Desain sistem (<i>Design</i>)												
3	Pembuatan Aplikasi (<i>Coding</i>)												
4	Pengujian (<i>Testing</i>)												
5	<i>Software Increment</i>												
6	Laporan Tugas Akhir												

3.1.2 Tempat Penelitian

Tempat penelitian tugas akhir ini akan dilakukan di kampus Politeknik Negeri Banyuwangi Jl. Raya Jember KM. 13 Kabat, Labanasem Banyuwangi, dengan objek Bank Cetral Asia (BCA) dan Bank Tabungan Negara (BTN).

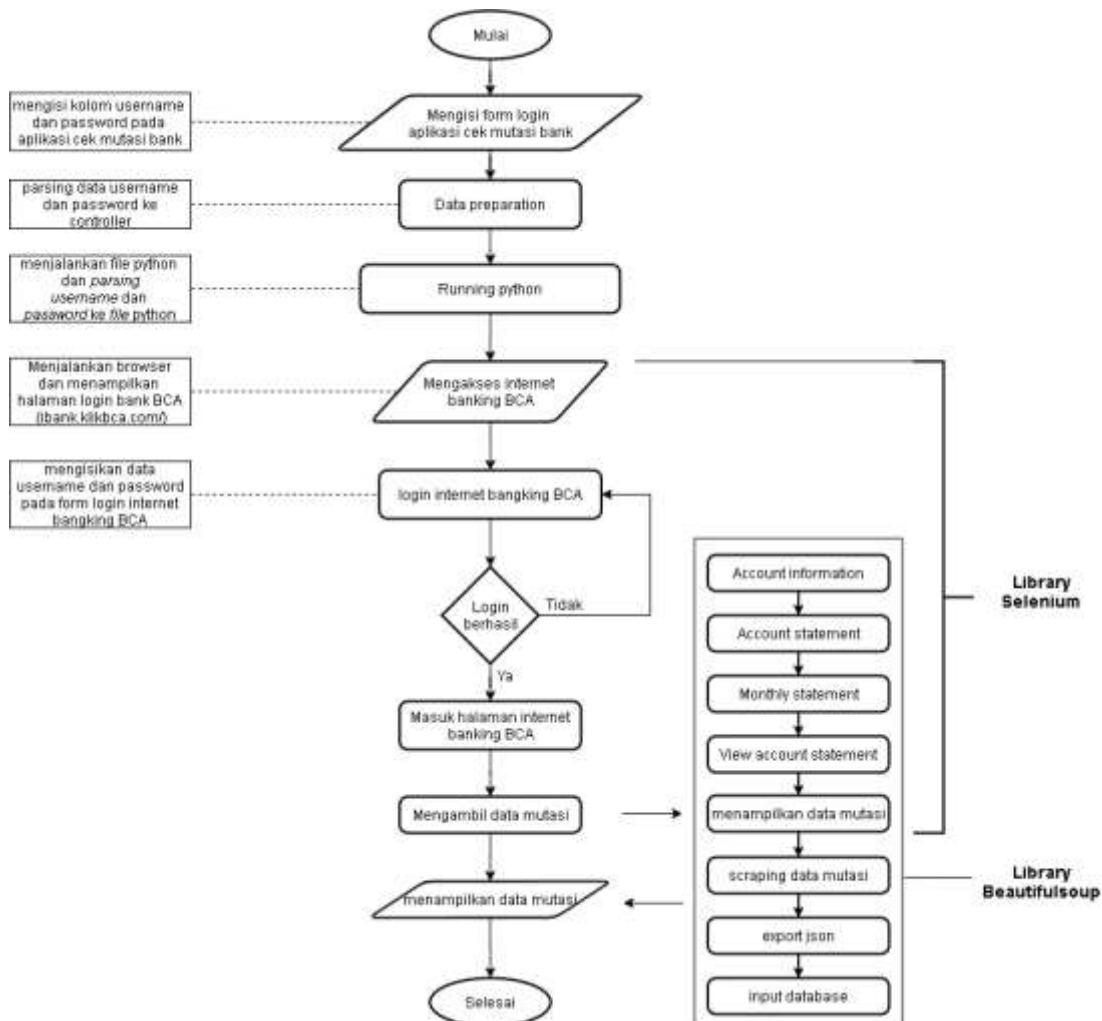
3.2 Metode Pengembangan Aplikasi

Tugas akhir aplikasi cek mutasi bank berbasis web dalam pelaksanaannya menggunakan salah satu metode *Extreme Programming* (XP). Metode ini menjadi sebuah pendekatan yang cocok untuk pengembangan awal sebuah aplikasi, dengan melihat batas waktu pengerjaannya. Metode XP dikembangkan dengan tujuan untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif. XP juga bertujuan untuk mengurangi biaya selama ada perubahan

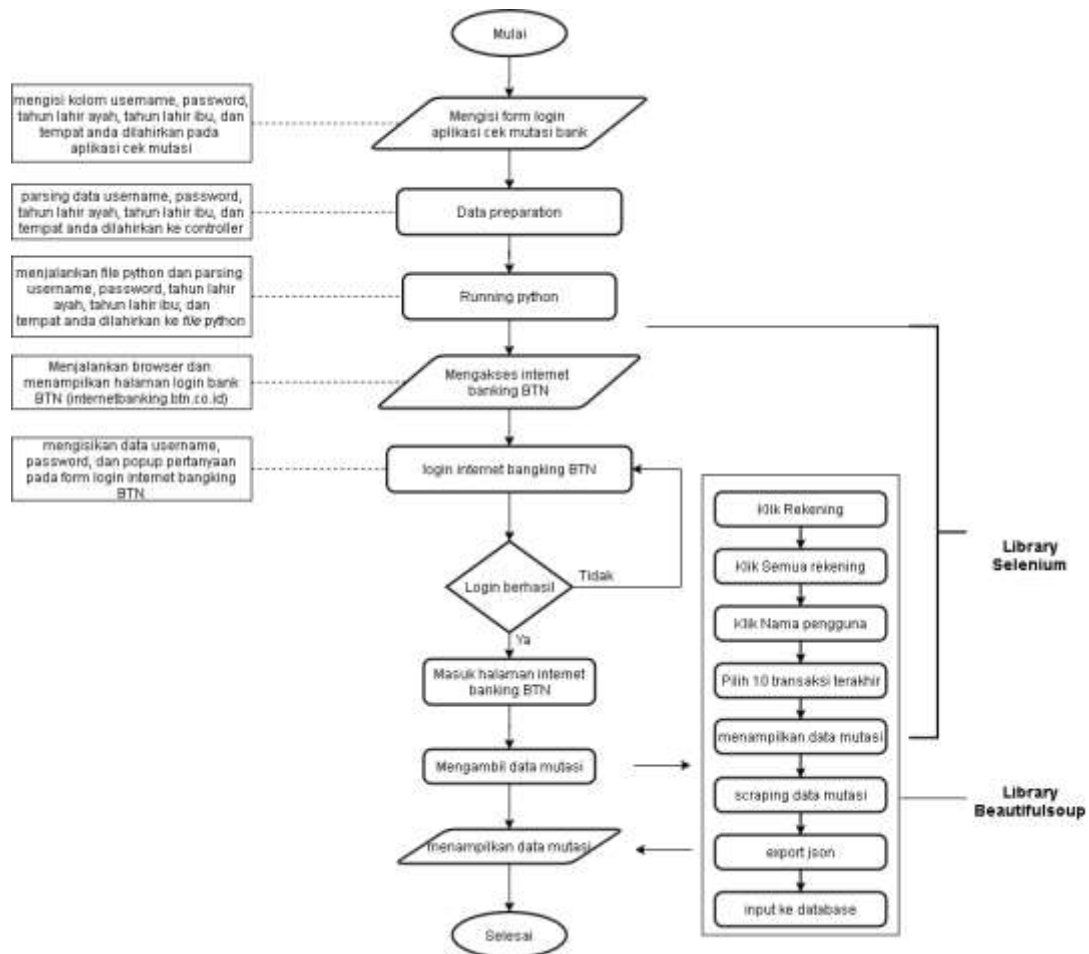
dalam pengembangan perangkat lunak menggunakan siklus (tahapan) pengembangan perangkat lunak yang singkat. XP melihat perubahan perangkat lunak merupakan suatu yang alami, tak terhindarkan selama pengembangan perangkat lunak (Suryantara, 2017).

3.2.1 Perencanaan (*Planning*)

Tahapan ini dimulai dengan mencari data dan informasi. Informasi didapatkan dari wawancara pada nasabah yang menggunakan *internet banking*. Informasi yang didapatkan berupa pengalaman dari para pengguna yang harus membuka halaman *website* berbeda untuk mengetahui data rekeningnya dari akun bank yang berbeda. Data didapatkan dari jurnal mengenai *internet banking* dan metode *scraping*. Alur pengambilan data dengan metode *scraping* dapat dilihat pada gambar 3.1 dan gambar 3.2.



Gambar 3.1 Alur Pengambilan Data BCA



Gambar 3.2 Alur Pengambilan Data BTN

Pengambilan data dengan metode *scraping* menggunakan bahasa pemrograman python dengan *library* selenium dan BeautifulSoup, *library* selenium dan BeautifulSoup digunakan karena lebih cocok dan mendukung untuk mengatasi masalah keamanan *internet banking* pada proyek pembuatan aplikasi ini.

3.2.2 Desain sistem (Design)

Pada tahapan perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Proses ini memudahkan untuk memperkirakan aplikasi seperti apa yang akan dibuat. Pemodelan sistem yang digunakan yaitu *Use-Case Diagram*, *Activity Diagram*. Sedangkan untuk pemodelan basis data menggunakan *Entity Relationship Diagram* (ERD).

3.2.3 Pembuatan Aplikasi (Coding)

Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program. Dalam pembangunan aplikasi cek mutasi

bank menggunakan bahasa pemrograman PHP dan implementasi basis data menggunakan MySQL.

3.2.4 Pengujian (*Testing*)

Tahapan ini merupakan tahapan pengujian terhadap aplikasi yang sudah dibangun. Setelah proses perencanaan, desain sistem, dan pengkodean selesai dilanjutkan dengan proses pengujian. Metode yang digunakan dalam melakukan pengujian terhadap aplikasi cek mutasi bank adalah *Black-Box Testing* dengan melakukan pengujian terhadap masukan dan keluaran yang dihasilkan sistem.

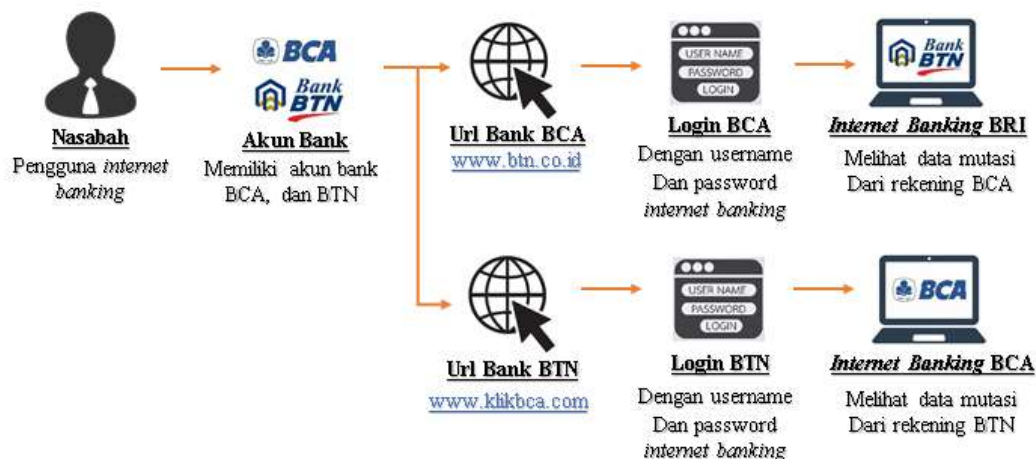
3.2.5 *Software Increment* (Peningkatan Perangkat Lunak).

Tahapan ini merupakan tahap pengembangan sistem yang sudah dibuat secara bertahap setelah aplikasi dibuat. Pada tahapan ini dapat dilakukan penambahan *fitur* agar meningkatkan kemampuan fungsional aplikasi.

3.3 Gambaran Umum Sistem

Gambaran umum sangatlah diperlukan dalam melakukan pembuatan ataupun pengembangan suatu sistem. Pada gambaran umum sistem terdapat dua pokok bahasan yaitu gambaran umum sistem yang berjalan dan gambaran umum sistem yang diusulkan. Tujuan dari pembahasan gambaran umum sistem ini yaitu untuk mengetahui pembaharuan atau pengembangan yang akan kita lakukan terhadap sistem yang telah berjalan sebelumnya.

3.3.1 Gambaran Umum Saat Ini

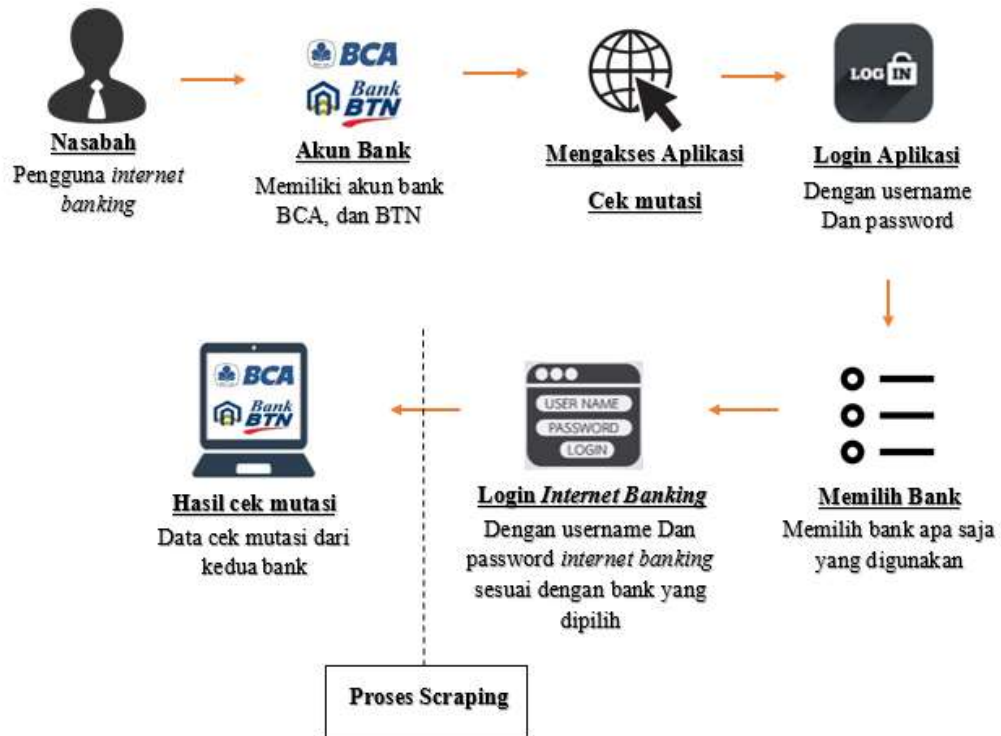


Gambar 3.3 Gambaran Umum yang Berjalan

Pada gambar 3.3 Menjelaskan bahwa nasabah pengguna akun *internet banking* yang memiliki dua rekening berbeda dari bank BCA dan BTN. Nasabah

tersebut harus mengakses web dari kedua bank tersebut secara satu persatu. Selanjutnya nasabah *login* dengan akun *internet banking* sesuai dengan *website* bank yang dibuka. Barulah nasabah dapat melihat mutasi bank dari masing-masing bank pada halaman *web* yang berbeda.

3.3.2 Gambaran Umum yang diusulkan



Gambar 3.4 Gambaran Umum yang Diusulkan

Pada gambar 3.4 Menjelaskan bahwa nasabah yang memiliki dua rekening berbeda dari bank BCA dan BTN, cukup membuka dan *login* pada aplikasi cek mutasi, setelah *login* nasabah memilih bank yang ingin dicek data mutasinya. Selanjutnya nasabah *login* ke *internet banking* dari bank yang telah dipilih dan hasil cek mutasi bank dapat dilihat pada halaman cek mutasi.

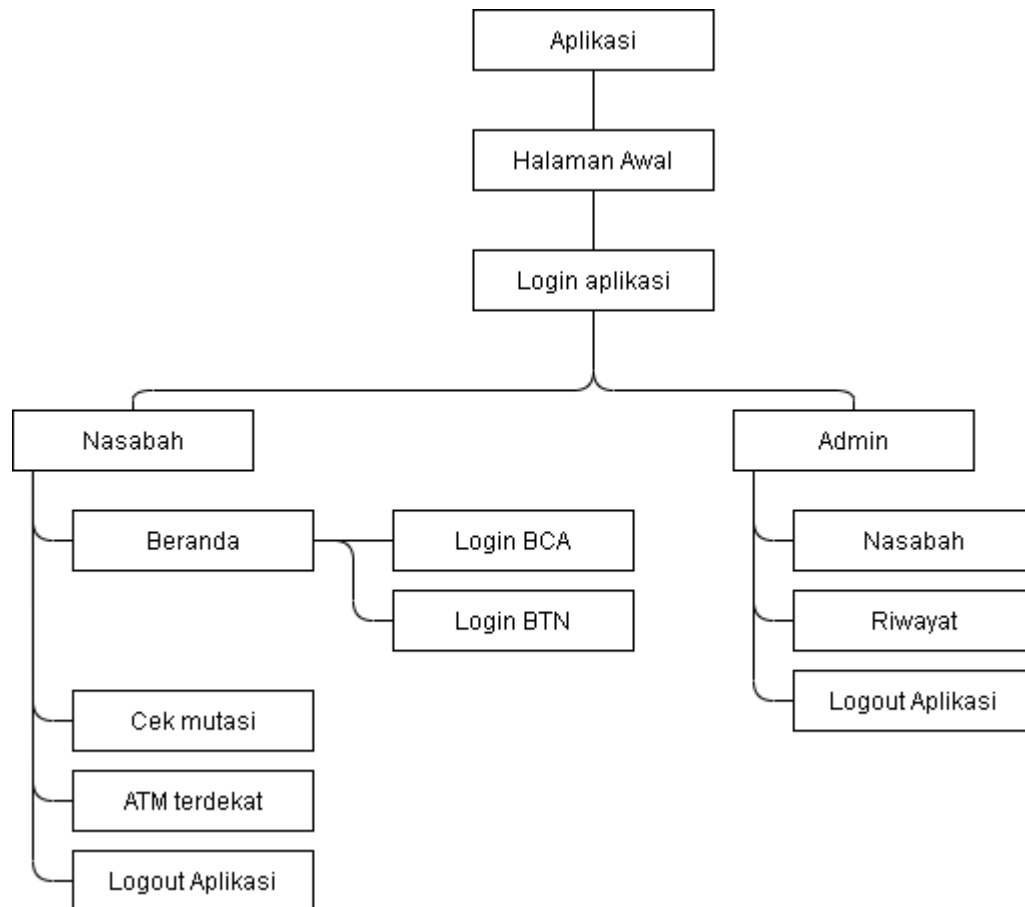
3.4 Spesifikasi Aplikasi

3.4.1 Aturan Sistem

Tabel 3.2 Aturan Sistem Aplikasi Cek Mutasi Bank

No	KETERANGAN	PENGUNA
1	Melihat nasabah yang terdaftar pada aplikasi.	Admin Aplikasi
2	Melihat riwayat dari <i>user</i> yang menggunakan aplikasi.	Admin Aplikasi
3	<i>Login internet banking</i> dan cek mutasi.	Nasabah
4	Cetak hasil cek mutasi.	Nasabah

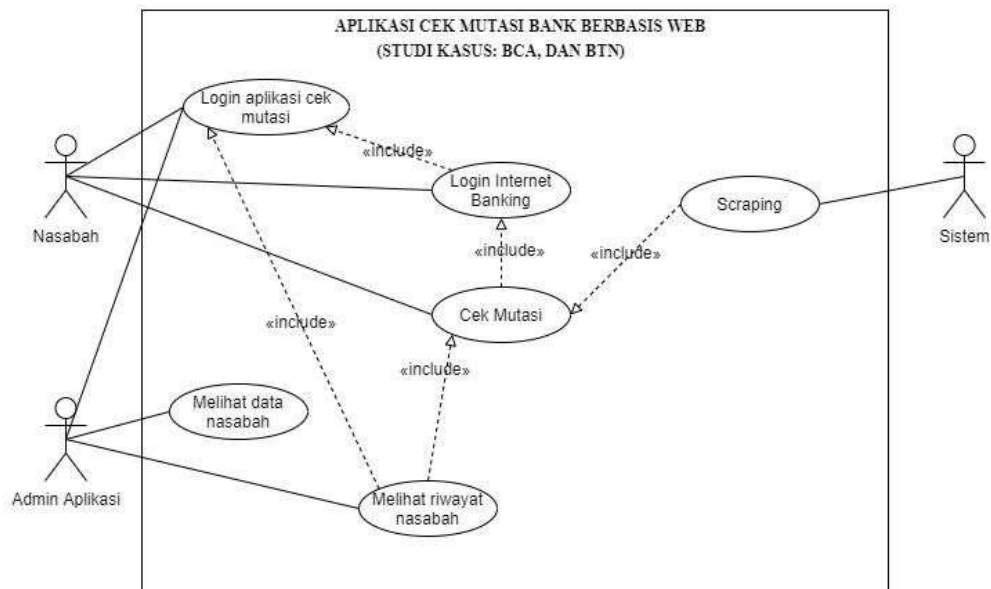
3.4.2 Struktur Menu



Gambar 3.5 Struktur Menu Aplikasi Cek Mutasi Bank

Gambar 3.5 menunjukkan struktur menu pada Aplikasi cek mutasi bank berbasis *web*. Aplikasi mengharuskan pengguna melakukan *login* karena menu-menu dapat diakses setelah pengguna melakukan *login*. Beberapa menu nasabah yang dapat diakses adalah beranda (*login internet banking*), cek mutasi, ATM terdekat dan *logout*. Menu yang terdapat pada admin aplikasi meliputi nasabah, riwayat, dan *logout*.

3.4.3 Use Case Diagram



Gambar 3.6 Use Case Aplikasi Cek Mutasi Bank

Pada Gambar 3.6 ditunjukkan pemodelan *Use Case* dari Aplikasi Cek Mutasi Bank Berbasis *Web*. Terdapat 3 aktor dalam aplikasi ini untuk lebih detail dijelaskan pada Tabel 3.3 dan Penjelasan tiap *use case* ditunjukkan pada Tabel 3.4, Tabel 3.5, Tabel 3.6, Tabel 3.7, Tabel 3.8, dan Tabel 3.9.

Tabel 3.3 Penjelasan Aktor

No	Aktor	Deskripsi
1	Nasabah	Pemilik akun perbankan (BCA dan BTN)
2	Admin aplikasi	Bertugas mengeloh user dan riwayat aplikasi
3	Sistem	Sistem yang melakukan proses pengambilan data mutasi dari <i>internet banking</i>

Tabel 3.4 Keterangan Use Case Login Aplikasi Cek Mutasi

<i>Use Case Name</i>	Login Aplikasi Cek Mutasi
<i>Actor</i>	Nasabah, Admin Aplikasi
<i>Description</i>	Validasi untuk masuk kesistem dengan menggunakan <i>username</i> dan <i>password</i>
<i>Pre-Condition</i>	Membuka aplikasi
<i>Post-Condition</i>	Masuk halaman beranda sesuai hak akses
<i>Alternative Course</i>	Login tidak berhasil jika <i>username</i> dan <i>password</i> tidak terdaftar pada <i>database</i>

Tabel 3.5 Keterangan Use Case Login Internet Banking

<i>Use Case Name</i>	Login Internet Banking
<i>Actor</i>	Nasabah
<i>Description</i>	Validasi untuk masuk keinternet banking dengan menggunakan <i>username</i> dan <i>password</i>
<i>Pre-Condition</i>	Masuk ke halaman login internet banking

<i>Post-Condition</i>	Melakukan cek mutasi
<i>Alternative Course</i>	<i>Login Internet Banking</i> tidak dapat dilakukan sebelum <i>Login Aplikasi Cek Mutasi</i> <i>Login</i> tidak berhasil jika <i>username</i> dan <i>password</i> tidak memiliki akun pada <i>internet banking</i>

Tabel 3.6 Keterangan *Use Case* Cek Mutasi

<i>Use Case Name</i>	Cek Mutasi
<i>Actor</i>	Nasabah
<i>Description</i>	Mengecek mutasi dari bank yang dimiliki
<i>Pre-Condition</i>	<i>Login internet banking</i>
<i>Post-Condition</i>	Mencetak mutasi
<i>Alternative Course</i>	Cek mutasi tidak dapat dilakukan jika keluar dari <i>internet banking</i> .

Tabel 3.7 Keterangan *Use Case* Melihat Data Nasabah

<i>Use Case Name</i>	Melihat Data Nasabah
<i>Actor</i>	Admin Aplikasi
<i>Description</i>	Melihat nama nasabah dan tanggal dibuatnya akun aplikasi cek mutasi bank milik nasabah.
<i>Pre-Condition</i>	<i>Login</i> sebagai Admin aplikasi
<i>Post-Condition</i>	-
<i>Alternative Course</i>	Tidak dapat melihat data nasabah jika tidak ada <i>login</i>

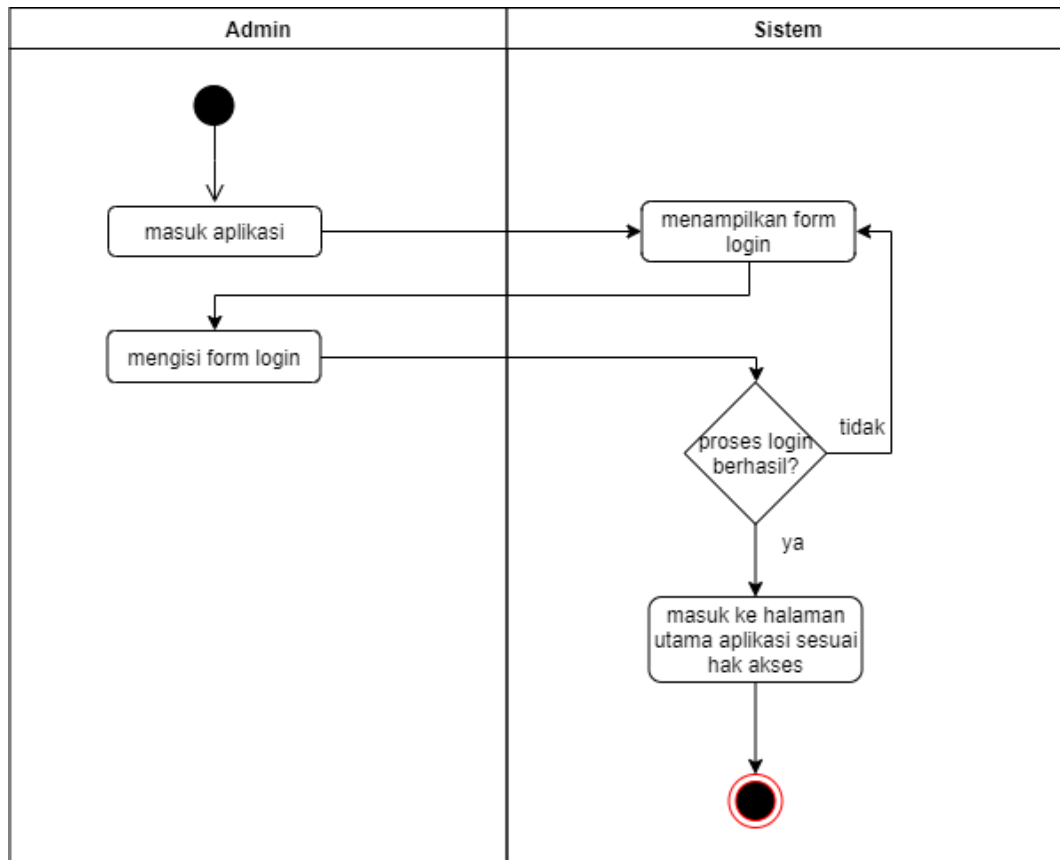
Tabel 3.8 Keterangan *Use Case* Melihat Riwayat Nasabah

<i>Use Case Name</i>	Melihat Data Riwayat Nasabah
<i>Actor</i>	Admin Aplikasi
<i>Description</i>	Melihat riwayat <i>login</i> nasabah pengguna aplikasi cek mutasi bank
<i>Pre-Condition</i>	<i>Login</i> sebagai Admin aplikasi
<i>Post-Condition</i>	-
<i>Alternative Course</i>	Tidak dapat melihat riwayat jika tidak ada nasabah yang <i>login</i> aplikasi cek mutasi bank

Tabel 3.9 Keterangan *Use Case* scraping

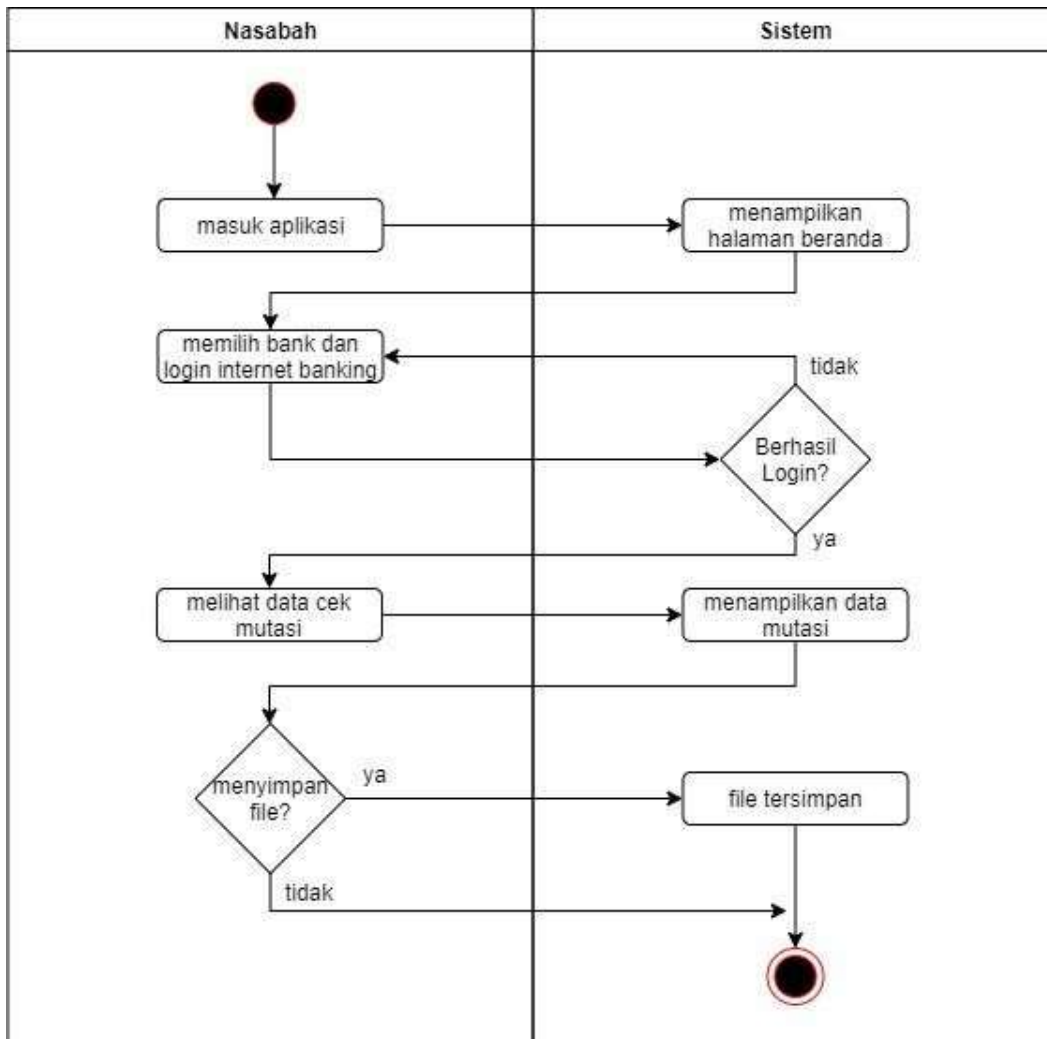
<i>Use Case Name</i>	Scraping
<i>Actor</i>	Sistem
<i>Description</i>	Monitoring aktivitas yang dilakukan oleh <i>users</i> (debit dan kredit)
<i>Pre-Condition</i>	<i>Login internet banking</i>
<i>Post-Condition</i>	Mengambil data debit dan kredit
<i>Alternative Course</i>	Cek mutasi tidak berhasil apabila belum <i>login internet banking</i>

3.4.4 Activity Diagram



Gambar 3.7 Activity Diagram *Login*

Pada gambar 3.7 menunjukkan alur *login*, apabila nasabah sudah memiliki akun maka langsung mengisi form *login*. Untuk nasabah yang belum memiliki akun maka harus mendaftar terlebih dahulu baru dapat melakukan proses *login*. Apabila proses *login* berhasil maka akan masuk ke halaman beranda dan apabila *login* gagal maka akan ditampilkan form *login* ulang. Setelah berhasil *login* maka akan masuk ke halaman sesuai dengan hak akses.



Gambar 3.8 Activity Diagram Cek Mutasi Bank

Pada gambar 3.8 menunjukkan alur penggunaan aplikasi yang dimulai dari proses *login* jika sudah memiliki akun dan proses pendaftaran apabila belum memiliki akun. Setelah *login* masuk kehalaman utama aplikasi untuk *login* ke akun *internet banking* bank. Setelah proses *login internet banking* berhasil maka mutasi berupa data debit dan kredit dari kedua bank tersebut sudah dapat dilihat. Terdapat juga fitur *download* file excel dan PDF dari data hasil cek mutasi.

3.5 Mock-up

3.5.1 Halaman Utama Aplikasi

Gambar 3.9 merupakan racangan halaman utama aplikasi cek mutasi bank. Halaman ini ditampilkan pada awal saat *user* sebelum melakukan proses *login*. Halaman ini hanya digunakan sebagai pembuka.



Gambar 3.9 Halaman Utama Aplikasi

3.5.2 Halaman *Login* Aplikasi

Gambar 3.10 merupakan rancangan tampilan *login*. *Login* bertujuan untuk autentikasi *user* sebelum masuk ke aplikasi. *Login* dilakukan untuk pengguna yang telah terdaftar pada aplikasi.

The image shows a dark blue background with the word 'Login' in white text at the top center. Below it, there are two input fields. The first field is labeled 'Nama Pengguna' and has a placeholder text 'Nama Pengguna'. The second field is labeled 'Kata Sandi' and has a placeholder text 'Kata Sandi'. Below the input fields, there are two red buttons with white text. The first button is labeled 'Masuk' and the second button is labeled 'Daftar'.

Gambar 3.10 Tampilan Halaman *Login*

3.5.3 Halaman *Register*

Gambar 3.11 merupakan rancangan tampilan *register*. *Register* digunakan untuk nasabah yang belum memiliki akun yang terdaftar pada aplikasi cek mutasi. Setelah melakukan *register* maka langsung dapat melakukan proses *login* aplikasi.

Register

Nama Lengkap	<input type="text" value="Nama Lengkap"/>
Nama Pengguna	<input type="text" value="Nama Pengguna"/>
Kata Sandi	<input type="password" value="Kata Sandi"/>
Konfirmasi Kata Sandi	<input type="password" value="Konfirmasi password"/>

Daftar

[Sudah punya akun? login](#)

Gambar 3.11 Tampilan Halaman *Register*

3.5.4 Halaman Beranda Nasabah

Gambar 3.12 merupakan rancangan tampilan halaman beranda nasabah. Halaman ini digunakan untuk *login* pada kedua akun *internet banking*. *Login* dilakukan satu-persatu dari kedua akun *internet banking*.

Ningrum Fadillah
Logout →

Beranda

Cek Mutasi

Masuk Akun E-Banking

Masuk Akun E-Banking

Gambar 3.12 Tampilan Halaman Beranda Nasabah

3.5.5 Halaman *Login Internet Banking*

Gambar 3.13 merupakan rancangan tampilan *login internet banking*. *Login* menggunakan *username* dan *password*. *Login* dilakukan sebelum melakukan cek

mutasi. *Login internet banking* hanya dapat dilakukan setelah *login* aplikasi cek mutasi.



Login Internet Banking BCA

Nama Pengguna

Kata Sandi

Masuk

Gambar 3.13 Tampilan Halaman *Login Internet Banking*

3.5.6 Halaman Cek Mutasi

Gambar 3.14 merupakan rancangan tampilan cek mutasi. Pada halaman berisi data mutasi dari akun *internet banking* yang sudah dimasukkan pada halaman *login internet banking* sebelumnya pada gambar 3.13.



Ningrum Fadillah

Beranda

Cek Mutasi

Logout →

Cek Mutasi

Pengguna: Ningrum

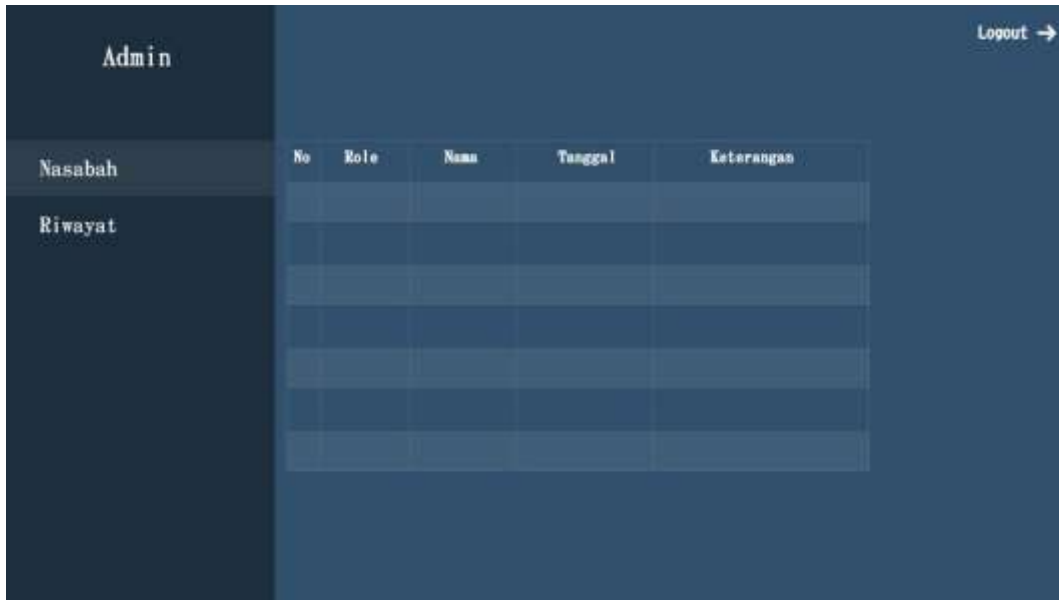
Status: online

No	Tanggal	Nominal	Nominal Akhir	Bank
1	senin, 15 agustus 2020	0.000	7.000	BCA
2	rabu, 17 agustus 2020	20.000	20.000	BTN
3	kamis, 18 agustus 2020	50.000	50.000	BTN

Gambar 3.14 Tampilan Halaman Cek Mutasi

3.5.7 Halaman Data Nasabah

Gambar 3.15 merupakan rancangan tampilan data nasabah. Halaman ini hanya dapat diakses oleh admin aplikasi. Halaman ini terdapat data nasabah yang terdaftar pada aplikasi cek mutasi bank.

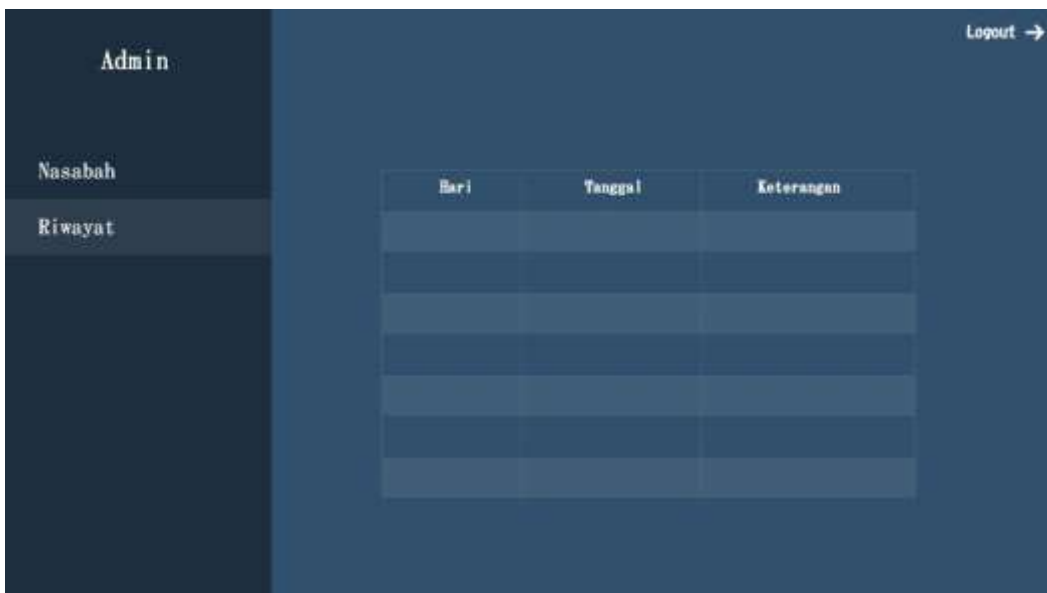


Admin	Logout →				
Nasabah	No	Role	Nama	Tanggal	Keterangan
Riwayat					

Gambar 3.15 Tampilan Halaman Data Nasabah

3.5.8 Halaman Riwayat

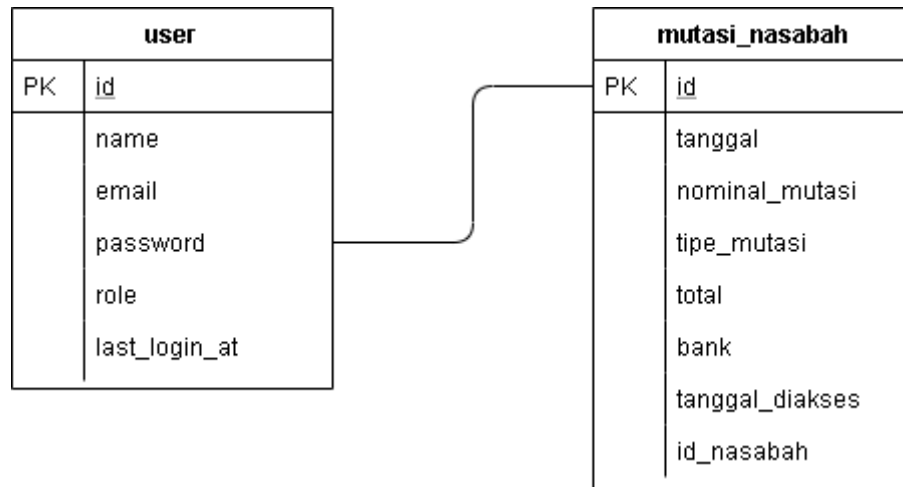
Gambar 3.16 merupakan rancangan tampilan riwayat. Halaman ini hanya dapat diakses oleh admin aplikasi. Halaman ini terdapat data riwayat nasabah yang mengakses aplikasi cek mutasi bank.



Admin	Logout →		
Nasabah			
Riwayat	Hari	Tanggal	Keterangan

Gambar 3.16 Tampilan Halaman Riwayat

3.6 Entity Relationship Diagram (ERD)



Gambar 3.17 ER-Diagram

Pada gambar 3.17 terdapat 2 tabel yang saling berelasi. Pertama ada tabel *user* dan kedua ada tabel *mutasi_nasabah*. Dimana *id* pada tabel *user* digunakan untuk *id* nasabah pada tabel *mutasi_nasabah*, ini berfungsi untuk menampilkan data mutasi nasabah sesuai dengan *login user*.

--Halaman ini sengaja dikosongkan--

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil

Aplikasi cek mutasi bank merupakan aplikasi yang digunakan untuk mengecek transaksi debit dan kredit dari dua akun bank yang berbeda dalam satu halaman *website*. Nasabah yang telah menggunakan aplikasi ini tidak perlu membuka dua halaman *web* sesuai dengan bank yang digunakan seperti BCA dan BTN. Terdapat dua pengguna pada aplikasi ini yaitu: admin aplikasi dan nasabah pengguna *internet banking*. Admin aplikasi memonitoring data nasabah dan riwayat nasabah pengguna aplikasi. Nasabah melakukan *login internet banking* untuk mengetahui data mutasi dari rekening yang dimilikinya.

Penggunaan BCA dan BTN dikarenakan kemudahan untuk melakukan *scraping* pada bank tersebut. BCA tidak menggunakan captcha sebagai keamanan untuk *login internet banking*, sedangkan BTN menggunakan keamanan tiga pertanyaan pribadi milik nasabah yang akan ditanyakan secara acak pada saat *login internet banking*. Sebelumnya telah dilakukan berbagai percobaan pada beberapa bank yaitu: Mandiri Syariah, BNI, BRI, dan CIMB. Pada Mandiri Syariah, BNI, dan BRI terkendala adanya captcha kombinasi angka dan huruf dalam bentuk gambar pada saat *login* yang tidak bisa dilalui proses *scraping*. Pada CIMB terkendala pada saat melakukan *login website internet banking* CIMB mendeteksi adanya aktivitas yang bukan dilakukan oleh manusia pada saat *login* melainkan oleh robot (komputer untuk otomatis *login*).

4.1.1 Tampilan Halaman Awal Aplikasi

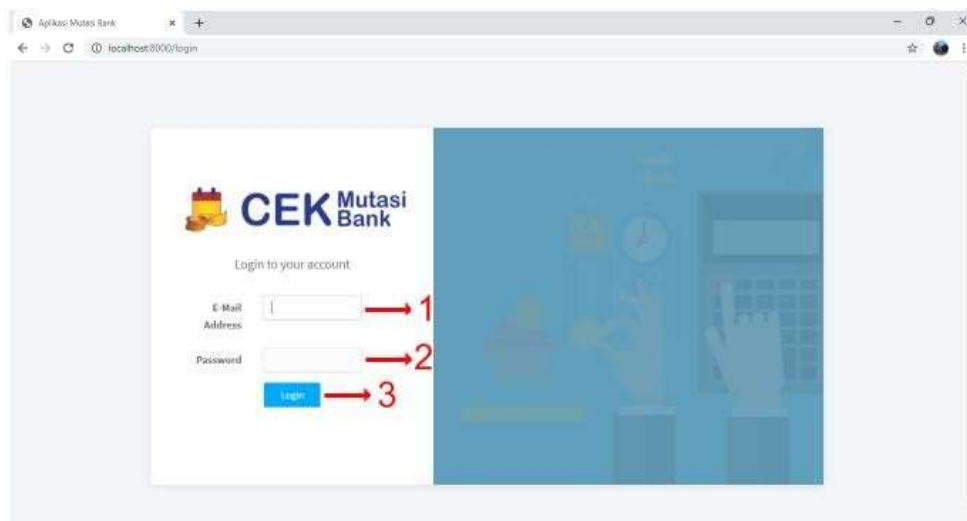
Gambar 4.1 merupakan halaman awal saat aplikasi pertama kali dibuka. Pada halaman ini terdapat dua *button* yaitu *login* dan *register*. Halaman *login* digunakan untuk pengguna yang sudah memiliki akun, sedangkan halaman *register* digunakan untuk pendaftaran akun bagi nasabah yang belum memiliki akun pada aplikasi cek mutasi bank ini.



Gambar 4.1 Halaman Awal Aplikasi

4.1.2 Tampilan Halaman *Login* Aplikasi

Gambar 4.2 merupakan halaman *login* aplikasi yang digunakan untuk admin aplikasi dan nasabah untuk masuk ke aplikasi. *Login* hanya dapat digunakan bagi pengguna yang sudah memiliki akun yang terdaftar pada aplikasi cek mutasi bank bank saja. Pada halaman *login* aplikasi terdapat *form login* yang harus diisi terlebih dahulu.



Gambar 4.2 Halaman *Login*

Pada gambar 4.2 menunjukkan tampilan halaman *login* dengan penjelasan *fitur* sebagai berikut :

1. Kolom *email address* yang disikan berdasarkan akun yang sudah terdaftar pada basis data.

2. Kolom *password* yang diisikan sesuai dengan akun yang diisikan pada kolom pertama.
3. *Button login* digunakan untuk masuk ke aplikasi.

4.1.3 Tampilan Halaman *Register*

Gambar 4.3 merupakan halaman *register* yang digunakan nasabah untuk membuat akun. Akun tersebut yang digunakan untuk *login* aplikasi cek mutasi bank.



Gambar 4.3 Halaman *Register*

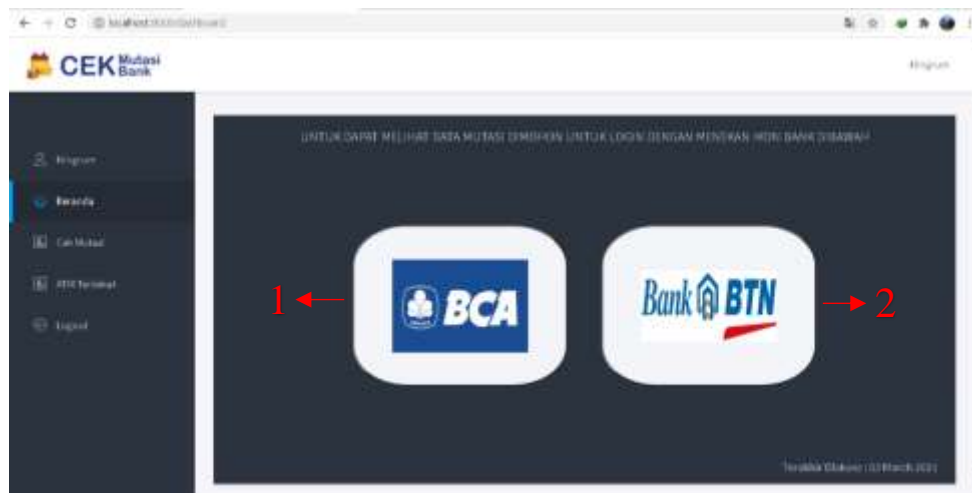
Pada gambar 4.3 menunjukkan tampilan halaman *register* dengan penjelasan fitur sebagai berikut :

1. Kolom *name* yang diisikan untuk nama pengguna.
2. Kolom *email address* diisikan untuk email yang akan digunakan.
3. Kolom *password* diisikan untuk kata sandi yang akan digunakan saat *login*.
4. Kolom *confirm password* diisikan untuk mengulang *password* yang telah diisikan pada kolom nomor 3.
5. *Button register* digunakan untuk menyimpan data yang telah diisikan ada pada halaman *register*

4.1.4 Tampilan Halaman Beranda Nasabah

Gambar 4.4 merupakan halaman beranda nasabah yang didalamnya terdapat ikon dari tiga bank yang dapat digunakan pada aplikasi cek mutasi bank. Halaman ini berfungsi untuk milih bank yang akan digunakan dan untuk *login* akun

bank yang akan digunakan, sehingga sebelum masuk ke halaman *login internet banking* nasabah harus memilih banknya terlebih dahulu.



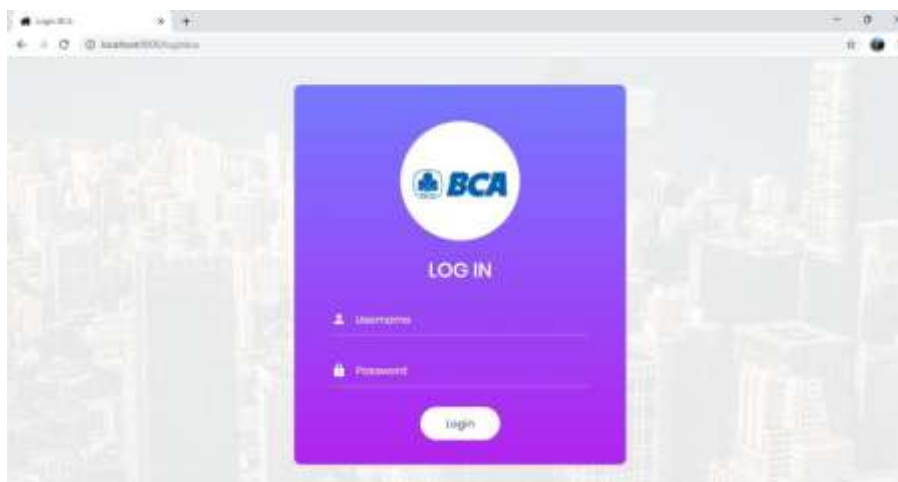
Gambar 4.4 Halaman Beranda Nasabah

Pada gambar 4.4 menunjukkan tampilan halaman beranda nasabah dengan penjelasan fitur sebagai berikut :

1. Ikon BCA digunakan untuk *login* pada akun *internet banking* BCA.
2. Ikon BTN digunakan untuk *login* pada akun *internet banking* BTN.

4.1.5 Tampilan Halaman Login Internet Banking

Gambar 4.5 merupakan halaman *login internet banking* yang digunakan nasabah untuk memasukkan akun *internet banking* yang dimilikinya. *Login* hanya dapat digunakan bagi nasabah yang sudah memiliki akun *internet banking* yang terdaftar pada bank. Pada halaman *login* terdapat form *login* yang harus diisi terlebih dahulu.



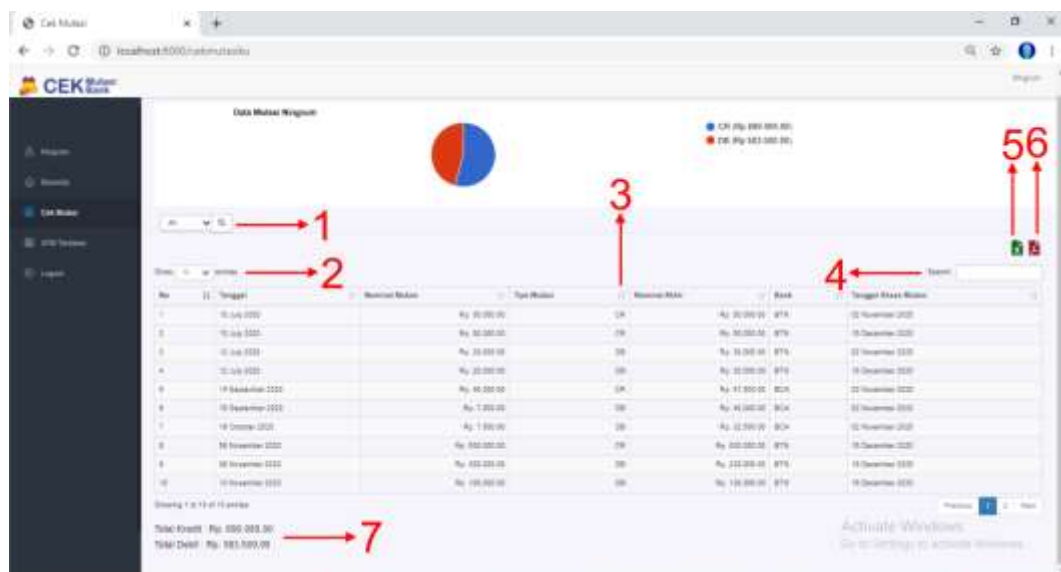
Gambar 4.5 Halaman *Login Internet Banking* BCA

Pada gambar 4.5 menunjukkan tampilan halaman *login* dengan penjelasan fitur sebagai berikut :

1. Kolom *username* yang disikan *username* sesuai akun *internet banking* yang dimiliki nasabah.
2. Kolom *password* yang diisikan *password* sesuai akun *internet banking* yang dimiliki nasabah.
3. *Button login* digunakan untuk masuk ke *web internet banking* milik bank.

4.1.6 Tampilan Halaman Cek Mutasi

Gambar 4.6 merupakan halaman cek mutasi bank. Halaman ini berisi data mutasi dari akun bank milik nasabah yang telah melakukan *login internet banking* sebelumnya pada gambar 4.5. halaman ini berisi semua data transaksi debit dan kredit beserta dengan jumlah masing-masing dari transaksi debit dan kredit tersebut.



Gambar 4.6 Halaman Cek Mutasi

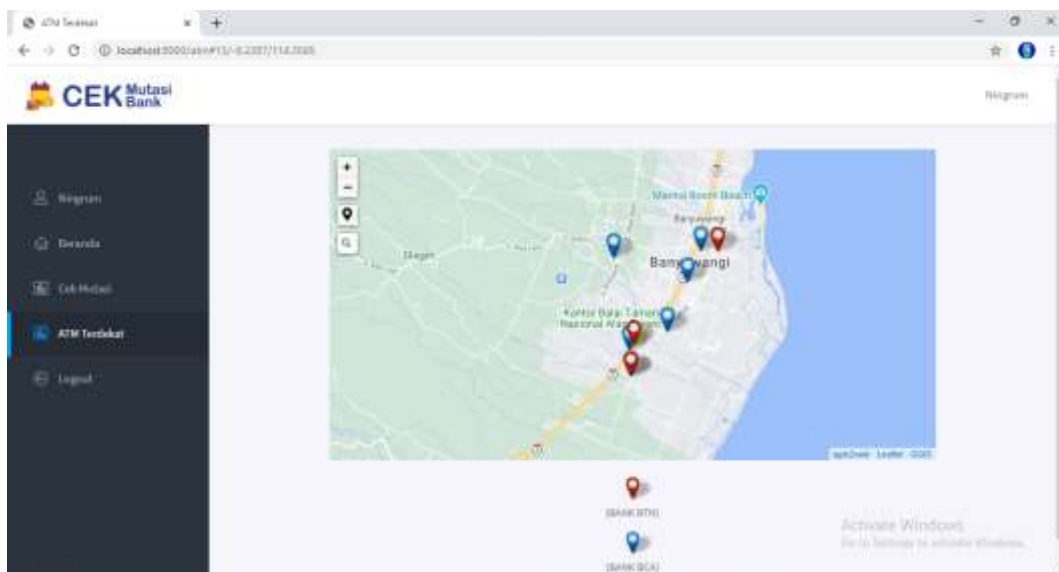
Pada gambar 4.6 menunjukkan tampilan halaman *login* dengan penjelasan fitur sebagai berikut :

1. *Filter* digunakan untuk mencari bank atau tanggal tertentu.
2. Kolom *Show* digunakan untuk mengatur banyak baris data yang ditampilkan pada halaman cek mutasi.
3. Kolom *ShortBy* digunakan untuk mengurutkan data pada halaman cek mutasi.

4. Kolom *Search* digunakan untuk melakukan pencarian data pada halaman cek mutasi.
5. Ikom Excel digunakan untuk mengunduh data mutasi dalam format excel.
6. Ikom PDF digunakan untuk mengunduh data mutasi dalam format PDF.
7. Akumulasi data kredit dan debit.

4.1.7 Tampilan Halaman ATM Terdekat

Gambar 4.7 merupakan halaman lokasi ATM terdekat. Pada halaman ini berisi informasi lokasi ATM BCA dan BTN terdekat dengan nasabah pengguna aplikasi cek mutasi bank.



Gambar 4.7 ATM Terdekat

4.1.8 Tampilan Halaman Data Nasabah

Gambar 4.8 merupakan halaman data nasabah. Halaman ini berisi data nasabah yang menggunakan aplikasi cek mutasi bank. halaman ini berisi nama nasabah dan tanggal pembuatan akun pada aplikasi cek mutasi bank.

No	Nama Pengguna	Status	Tanggal Registrasi
1	Nisgram	nasabah	Monday, 21 September 2020
2	Tamara	nasabah	Tuesday, 22 September 2020
3	Nala	nasabah	Wednesday, 23 September 2020
4	Nila	nasabah	Thursday, 03 October 2020
5	Nita	nasabah	Thursday, 03 November 2020

Gambar 4.8 Halaman Data Nasabah

4.1.9 Tampilan Halaman Riwayat

Gambar 4.9 merupakan halaman riwayat nasabah. Halaman ini berisi data riwayat aktivitas nasabah yang menggunakan aplikasi cek mutasi bank. halaman ini berisi nama nasabah dan tanggal terakhir akses aplikasi cek mutasi bank.

No	Nama Pengguna	Status	Terakhir Login
1	Nisgram	nasabah	Sunday, 03 January 2021
2	Tamara	nasabah	Sunday, 04 October 2020
3	Nala	nasabah	Saturday, 27 October 2020
4	Nila	nasabah	Saturday, 10 October 2020
5	Nita	nasabah	Thursday, 03 November 2020

Gambar 4.9 Halaman Riwayat Nasabah

4.2 Pembahasan

Aplikasi cek mutasi bank berbasis *web* adalah aplikasi pengecekan data mutasi (debit dan kredit) untuk nasabah bank BCA dan BTN pengguna *internet banking*. Dengan adanya aplikasi ini nasabah bank tidak perlu membuka dua halaman *internet banking* dari bank-bank tersebut, nasabah hanya perlu membuka satu yaitu aplikasi cek mutasi bank berbasis *web* saja. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP dan *framework* laravel 7 dengan basis data MySQL. Teknik pengambilan data mutasi bank menggunakan metode *scraping* dengan *library* selenium dan beautifulsoup.

Proses pengambilan data yang berlangsung pada aplikasi cek mutasi bank dimulai dari nasabah melakukan *login internet banking* pada aplikasi cek mutasi bank yang selanjutnya saat tombol *login* ditekan maka file python dijalankan dan data login tersebut di parsing ke python. Menjalankan browser chrome dengan *web driver* dan membuka alamat *web internet banking* yang dituju hingga masuk pada halaman login internet banking dan selanjutnya data *login* yang telah diparsing di isikan pada form *login internet banking*. Setelah berhasil *login* maka menekan menu-menu hingga masuk pada halaman yang berisikan data debit dan kredit. Selanjutnya *scraping* data debit dan kredit yang dibutuhkan dengan memanfaatkan *xpath* dan menyimpan data sementara pada variabel yang telah disediakan. Selanjutnya agar data dapat dimasukkan ke dalam database data tersebut di *export* terlebih dahulu menjadi file json. Barulah setelah dalam bentuk json file diinputkan pada database.

Melakukan proses login, menekan menu-menu, hingga masuk pada halaman yang berisi data mutasi dilakukan menggunakan *library* selenium. Sedangkan *scraping* data mutasi menggunakan *library* beautifulsoup. Kedua *library* tersebut digunakan karena keduanya memiliki kelebihan dan kekurangan masing-masing, seperti halnya *library* selenium tidak dapat melakukan *scraping* data mutasi dan *library* beautifulsoup tidak dapat melakukan proses *login* dan menekan menu-menu. Sehingga keduanya harus digunakan pada pembuatan aplikasi cek mutasi bank ini agar didapatkan hasil yang dibutuhkan.

Aplikasi ini dioperasikan oleh admin aplikasi sebagai pihak yang memonitoring siapa saja nasabah yang menggunakan aplikasi cek mutasi bank

berbasis *web* dan riwayat aktivitas nasabah tersebut (terakhir *login aplikasi* cek mutasi bank berbasis *web*). Aplikasi ini dapat digunakan juga oleh nasabah yang hanya memiliki salah satu akun *internet banking* bank BCA atau BTN.

4.2.1 Proses Scraping BCA

Sebelum melakukan proses *scraping* untuk mengambil data mutasi BCA, nasabah harus melakukan *login* pada halaman *login internet banking* BCA yang ada pada aplikasi cek mutasi bank.

```

41 <form class="login100-form validate-form" autocomplete="off" action="{{url('/bca/scrap')}}" method="POST">
42   @csrf
43   <span class="login100-form-logo" style="">
44
45   
46   </span>
47
48   <span class="login100-form-title" p-b-34 p-t-27">
49     @if(session('alert'))
50     <div class="alert" role="alert" id="myAlert">
51       <div class="alert alert-danger alert-dismissible fade show" role="alert">
52         <span class="badge badge-pill badge-danger"></span> {{ session('alert') }}
53         <button type="button" onclick="closeAlert()" class="close" data-dismiss="alert" aria-label="Close">
54           <span aria-hidden="true">&times;</span></button>
55       </div>
56     </div>
57   @endif
58   </span>
59
60   <div class="wrap-input100 validate-input" data-validate="Enter username">
61     <input class="input100" required type="text" name="username" placeholder="Username">
62     <span class="focus-input100" data-placeholder="&#x20;"></span>
63   </div>
64
65   <div class="wrap-input100 validate-input" data-validate="Enter password">
66     <input class="input100" required type="password" name="password" placeholder="Password">
67     <span class="focus-input100" data-placeholder="&#x20;"></span>
68   </div>
69
70   <div class="container-login100-form-btn">
71     <button type="submit" class="login100-form-btn">
72       Login
73     </button>
74   </div>
75 </form>

```

Gambar 4.10 Login Internet Banking BCA

Gambar 4.10 merupakan *source code* form *login internet banking* BCA. *Source code* digunakan untuk membuat tampilan *form login*, menampung dan mengirimkan data yang diisikan pada *form login* oleh nasabah.

```
Route::POST('/bca/scrap', 'BCAController@cek');
```

Gambar 4.11 Routing Login Internet Banking BCA

Gambar 4.11 merupakan *source code* yang digunakan untuk mengarahkan dari *form login* ke fungsi cek yang ada pada *controller* (BCAController).

```

32     public function cek(Request $request)
33     {
34         $username = $request->username;
35         $password = $request->password;
36         $nasabah = Auth::user()->id;
37         $id_nasabah = Auth::user()->id;
38         $bulan = date("M-Y");
39         // echo shell_exec("python C:/xampp/htdocs/PANINGnabil/public/bca-scrapping-master/bca.py 2>&1 $username $password
40
41
42         if (file_exists(public_path() . "/" . $bulan . "_" . $nasabah . "_data_bca_new.json")) {
43             return redirect()->back()->with('alert', 'Koneksi Gagal');
44         }
45
46         else{
47
48             $timeA = time();
49
50             echo shell_exec(public_path() . "/Python27/python.exe " . public_path() . "/bca-scrapping-master/bca.py 2>&1 $username
51             $password $nasabah $bulan"); //hosting
52             // dd($timeA);
53             $timeB = time();
54             // dd($timeB);
55             $interval = $timeB - $timeA;
56
57             // dd($interval);
58             // $nasabah = Auth::user()->name;
59             // dd($timeB);
60             // $hasilTime = $timeB - $timeA;
61             // $hasil = floor($hasilTime / (24 * 60 * 60 ));
62             // dd($hasil);
63
64             // NingrumF0212
65
66             if ($interval >= 20) {
67
68                 $url_bca = public_path() . "/" . $bulan . "_" . $nasabah . "_data_bca_new.json";
69                 $datos_bca = file_get_contents($url_bca);
70                 $data_bca = json_decode($datos_bca, true);
71
72                 $data_bca = array_filter($data_bca);
73                 $data_bca = collect($data_bca)->all();
74
75                 #foreach ($data_bca as $d) {
76                     $servername = "localhost";
77                     $username = "root";
78                     $password = "";
79                     $dbname = "cekMutasi-Batamane";
80                     $tanggal = str($d[0], 'Y-m-d');
81                     // $tanggal = date('d-m', strtotime($tanggal));
82                     $tanggal = $tanggal . '-' . date('Y');
83                     $tanggal_diakses = date('Y-m-d');
84                     $tanggal = date("Y-m-d", strtotime($tanggal));
85
86                 // Create connection
87                 $conn = mysqli_connect($servername, $username, $password, $dbname);
88                 // $sql = "INSERT INTO mutasi_bca (tanggal,nominal_mutasi,tipu_mutasi,total,bank,tanggal_diakses,id_nasabah) VALUES
89                 ($tanggal, '$d[3]', '$d[4]', '$d[5]', 'bca', '$tanggal_diakses', '$nasabah')";
90                 $sql = "INSERT INTO mutasi_nasabah (tanggal,nominal_mutasi,tipu_mutasi,total,bank,tanggal_diakses,id_nasabah) VALUES
91                 ($tanggal, '$d[3]', '$d[4]', '$d[5]', 'bca', '$tanggal_diakses', '$nasabah')";
92
93                 if (!$conn) {
94                     die("Connection failed: " . mysqli_connect_error());
95                 }
96                 echo "Connected successfully";
97                 if (mysqli_query($conn, $sql)) {
98                     echo "New record created successfully";
99                 } else {
100                     echo "Error: " . $sql . "<br>" . mysqli_error($conn);
101                 }
102                 mysqli_close($conn);
103             }
104
105             return redirect('/cekmutasiku')->with('alert', 'Berhasil Menambahkan Data');
106         }
107
108         else{
109             return redirect()->back()->with('alert', 'Koneksi Gagal');
110         }
111     }
112 }
113 }
114 }
115 }
116 }
117 }
118 }

```

Gambar 4.12 Fungsi Cek Pada BCAController

Gambar 4.12 merupakan *source code* fungsi cek yang ada pada BCAController. *Source code* ini digunakan untuk parsing data dari *form login* ke BCAController, menjalankan *file* python, mengambil data dari *file* json dan memasukkannya pada *database*.

```

1 import io
2 import sys
3 import json
4 import time
5 import urllib
6 import getpass
7 from selenium import webdriver
8 # from selenium.webdriver.chrome_options import Options
9 from selenium.webdriver.common.by import By
10 from selenium.webdriver.support.ui import WebDriverWait
11 from selenium.webdriver.support import expected_conditions as EC
12 from selenium.common.exceptions import TimeoutException
13 from selenium.webdriver.support.ui import Select
14 from bs4 import BeautifulSoup
15
16
17 x = str(sys.argv[1])
18
19 y = str(sys.argv[2])
20
21 z = str(sys.argv[3])
22
23 b = str(sys.argv[4])
24 class bca(object):
25
26
27     __url = 'https://ibank.klikbca.com/'
28     def __init__(self):
29
30         self.formLogin()

```

Gambar 4.13 Menangkap Nilai yang Dikirim *Controller*

Gambar 4.13 merupakan bagian *source code* pada *file* python untuk *scraping* BCA. Penjelasan gambar 4.13 sebagai berikut:

1. *Import library.*
2. Menangkap nilai yang dikirim *controller* dan menyimpannya kedalam variabel-variabel.
3. Penamaan class.
4. Membuat variabel untuk menampung alamat *internet banking* BCA.

```
def formlogin(self):
    # webdriver = Chrome() Firefox() PhantomJS()
    # webdriver = webdriver.Chrome(executable_path=r"C:\xampp\htdocs\csp\test\chromedriver.exe")
    # webdriver = webdriver.Firefox(executable_path=r"C:\xampp\htdocs\csp\test\geckodriver.exe")
    # webdriver = webdriver.PhantomJS(executable_path=r"C:\xampp\htdocs\csp\test\phantomjs.exe")
    username = self.gettext('Masukkan username')
    password = self.gettext('Masukkan password')
    username = x
    password = x
    self.__driver = webdriver.Chrome(executable_path=r"C:\xampp\htdocs\csp\test\chromedriver.exe")
    self.__driver.get('http://192.168.1.100:8080')
    self.__driver.wait = WebDriverWait(self.__driver, 3)
    self.__username = username
    self.__password = password
    self.__session = z

    self.authlogin()
```

```

def authLogin(self):
    try:
        self.__driver.get(self.__url)
        username = self.__driver.wait.until(EC.presence_of_element_located((By.ID, "user_id")))
        password = self.__driver.wait.until(EC.element_to_be_clickable((By.ID, "pswd")))
        loginBTN = self.__driver.wait.until(EC.presence_of_element_located((By.NAME, "value(Submit)")))
        username.send_keys(self.__username)
        password.send_keys(self.__password)
        loginBTN.send_keys(webdriver.common.keys.Keys.SPACE)

        try:
            self.__driver.switch_to.frame(self.__driver.find_element_by_xpath("//frame[@name=\"header\"]"))
            self.__driver.switch_to_default_content()
            self.cekMutasi()
            self.logout()
            time.sleep(10)
        except:
            alert = self.__driver.switch_to_alert()
            print(alert.text)
            alert.accept()
    except:
        print("please check your connection...")

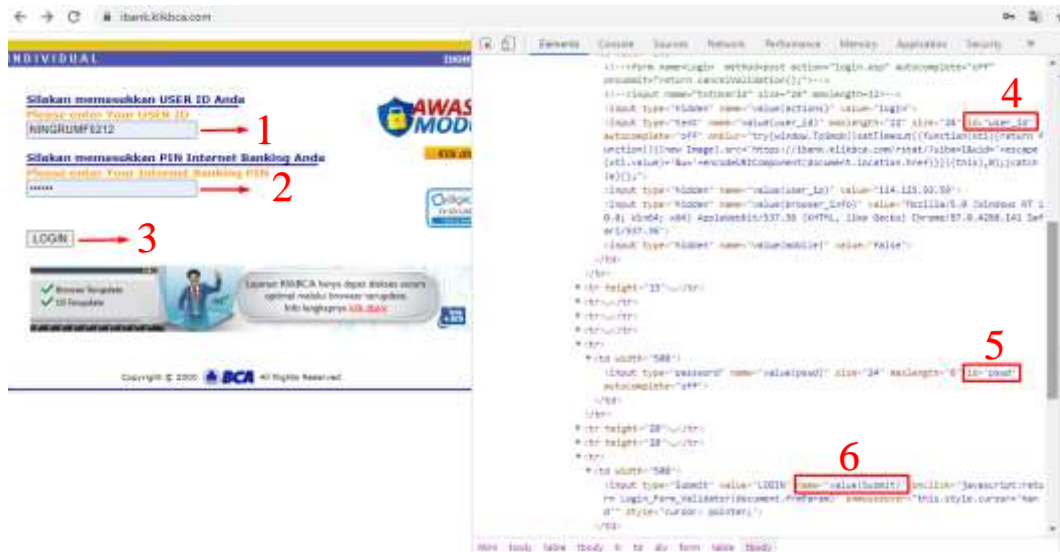
```

Gambar 4.14 login Internet Banking BCA

Gambar 4.14 merupakan bagian *file* python untuk *scraping* BCA dan digunakan untuk *login internet banking* BCA. Penjelasan gambar 4.14 sebagai berikut:

1. Menjalankan *browser* chrome yang digunakan untuk membuka dan *login* halaman *internet banking* BCA.
2. Mengarahkan pada *field username* yang ada pada *internet banking* BCA dengan memanfaatkan *id* (*user_id*) dari *field username* yang ada pada *form login internet banking* BCA. Dapat dilihat pada gambar 4.15.
3. Mengarahkan pada *field password* yang ada pada *internet banking* BCA dengan memanfaatkan *id* (*pswd*) dari *field password* yang ada pada *form login internet banking* BCA. Dapat dilihat pada gambar 4.15.
4. Mengarahkan pada tombol *login* yang ada pada *internet banking* BCA dengan memanfaatkan *name* (*value(submit)*). Dapat dilihat pada gambar 4.15.
5. Mengisikan *username* dari aplikasi cek mutasi bank ke *username* yang ada pada *form login internet banking* BCA sesuai dengan yang telah diarahkan.
6. Mengisikan *password* dari aplikasi cek mutasi bank ke *password* yang ada pada *form login internet banking* BCA sesuai dengan yang telah diarahkan.
7. Menekan tombol *login* yang ada pada *form login internet banking* BCA setelah *field username* dan *password* terisi.
8. Menjalankan fungsi cek mutasi.
9. Menjalankan fungsi *logout*.

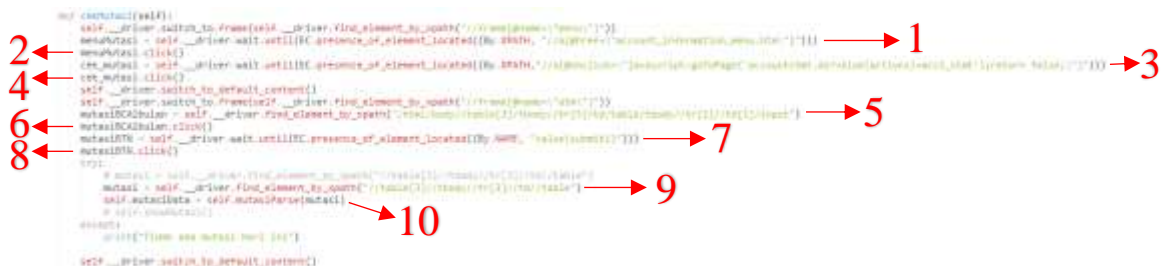
10. Menjalankan fungsi *sleep*.



Gambar 4.15 Pengisian Form Login Internet Banking BCA

Gambar 4.15 merupakan tampilan halaman *form login internet banking* BCA. Penjelasan gambar 4.15 sebagai berikut:

1. *Field username internet banking* BCA.
2. *Field password internet banking* BCA.
3. Tombol *login internet banking* BCA.
4. *Id* dari *field username* yang digunakan untuk mengarahkan pengisian data *username* dari aplikasi cek mutasi ke *internet banking* BCA.
5. *Id* dari *field password* yang digunakan untuk mengarahkan pengisian data *password* dari aplikasi cek mutasi ke *internet banking* BCA.
6. *Name* dari tombol *login* yang digunakan untuk mengarahkan agar setelah pengisian *username* dan *password* dapat langsung menekan tombol *login* yang ada pada *internet banking* BCA.



Gambar 4.16 Menekan Beberapa Menu

Gambar 4.16 merupakan bagian *file* python untuk *scraping* BCA dan merupakan *source code* pada *file* python yang digunakan untuk membuka menu. Penjelasan gambar 4.16 sebagai berikut:

1. Mengarahkan pada menu *account information* dengan memanfaatkan href="account_information_menu.htm" seperti yang dapat dilihat pada gambar 4.17.



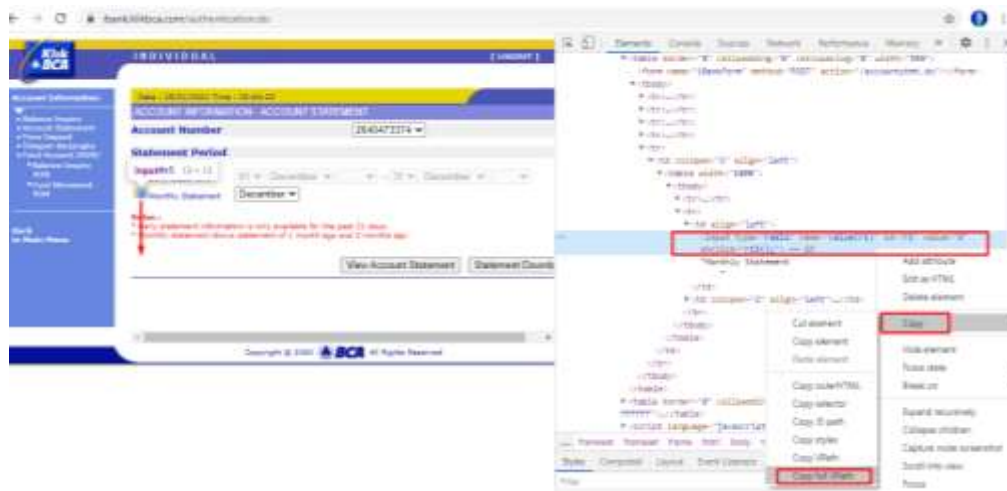
Gambar 4.17 Account Information

2. Menekan menu *account information* sesuai yang telah diarahkan.
3. Mengarahkan pada menu *account statement* dengan memanfaatkan onclick="javascript:goToPage('accountstmt.do?value(actions)=acct_stmt');return false;" seperti yang dapat dilihat pada gambar 4.18.



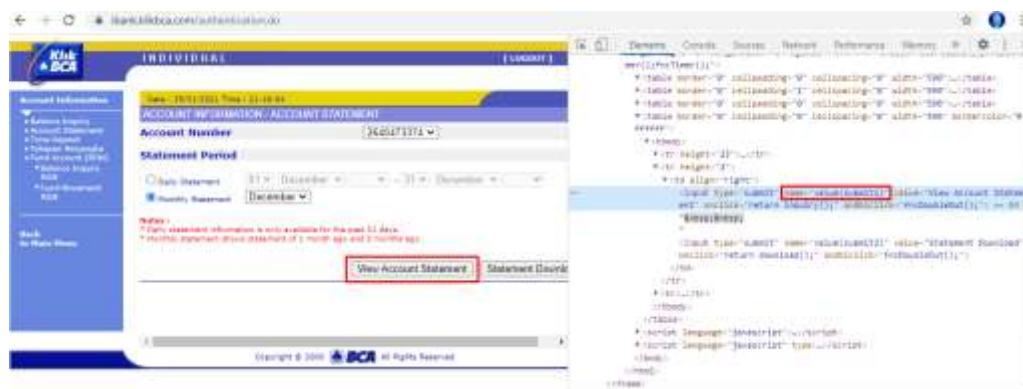
Gambar 4.18 Account Statement

4. Menekan menu *account statement* sesuai yang telah diarahkan.
5. Menekan *radio-button monthly statement* dengan memanfaatkan *Xpath* seperti yang terlihat pada gambar 4.19.



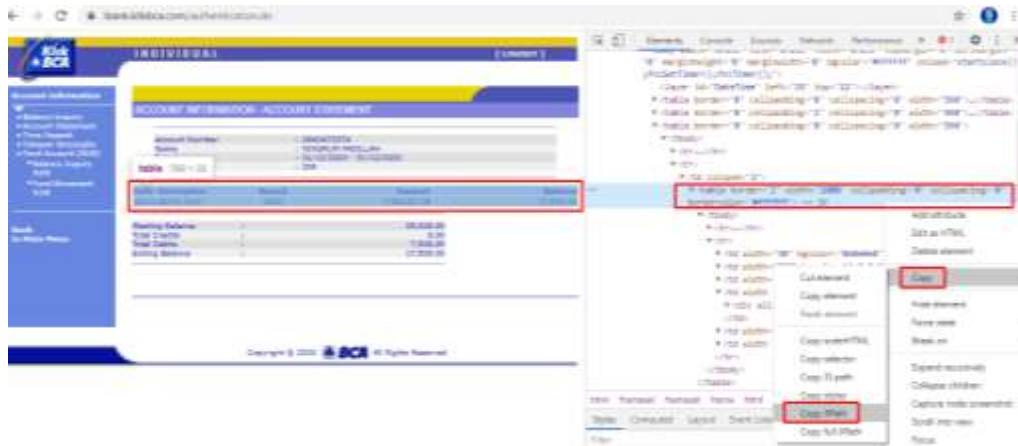
Gambar 4.19 *Monthly Statement*

6. Menekan *radio-button monthly statement* sesuai yang telah diarahkan.
7. Mengarahkan pada tombol *view account statement* dengan memanfaatkan `name="vaue(submit1)"` seperti yang dapat dilihat pada gambar 4.20.



Gambar 4.20 *View Account Statement*

8. Menekan *view account statement* sesuai yang telah diarahkan.
9. Mengarahkan pada tabel yang akan di *scraping* dengan memanfaatkan *Xpath* seperti yang terlihat pada gambar 4.21.



Gambar 4.21 Tabel Mutasi

10. Menjalankan fungsi `mutasiParse`.

```
def mutasiParse(self, table):
    table_mutasi = BeautifulSoup(table.get_attribute('innerHTML'), "html.parser") → 1
    # print(table_mutasi.prettify())
    data = [] → 2
    table_body = table_mutasi.find('tbody') → 3
    rows = table_body.find_all('tr') → 4
    for row in rows :
        cols = row.find_all('td') → 5
        cols = [ele.text.strip().replace(",",".") for ele in cols] → 6
        data.append([ele for ele in cols if ele]) → 7
    # print(data)
    file = open(b+"-"+z+"_data_bca_new.json", "w") → 8
    json.dump(data[1:], file) → 9
```

Gambar 4.22 *Scraping* Data Mutasi dan *Export* Data Json

Gambar 4.22 merupakan bagian *file* python untuk *scraping* BCA dan merupakan bagian *source code* pada *file* python yang digunakan untuk *scraping* data mutasi BCA. Penjelasan gambar 4.16 sebagai berikut:

1. untuk mengekstrak elemen-elemen html pada tabel yang telah arahkan lalu disimpan pada variabel `table_mutasi`.
2. Variabel `data` untuk menampung data dalam bentuk *array*.
3. Mengambil semua `tbody` dari variabel `table_mutasi` dan disimpan pada variabel `table_body`.
4. Mengambil semua `tr` dari variabel `table_body` dan disimpan pada variabel `rows`.
5. Mengambil semua `td` dari variabel `rows` dan disimpan pada variabel `cols`.
6. Menghapus karakter *default* (*whitespace* ataupun *tab*) awal dan akhir pada *string* dan mengubah karakter “,” menjadi “.”.
7. Mengisikan data yang telah diproses ke dalam variabel `data`.

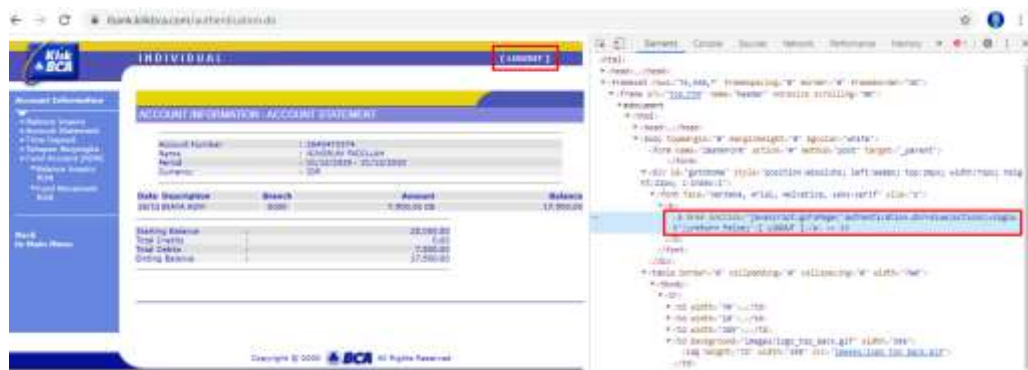
8. Membuat *file* json dengan nama *bulan-tahun_id nasabah_data_bca_new.json*.
9. Mengisikan data yang ada pada variabel data ke dalam *file* json yang dibuat sebelumnya dimulai dari *array* ke-1 (*default*-nya dari *array* ke-0).

```
def logout(self):
    try :
        self.__driver.switch_to.frame(self.__driver.find_element_by_xpath("//frame[@name=\"header\"]")) → 2
        logout = self.__driver.wait.until(EC.presence_of_element_located(
            (By.XPATH, "//a[@onclick=\"javascript:goToPage('authentication.do?value(actions)=logout');return false;\" ]")) → 2
        logout.click() → 3
        print("Anda berhasil logout")
    except TimeoutException:
        print("Session timeout. please login again")
```

Gambar 4.23 Logout Internat Banking BCA

Gambar 4.23 merupakan bagian *file* python untuk *scraping* BCA dan digunakan untuk *logout* dari *internet banking* BCA. Penjelasan gambar 4.23 sebagai berikut:

1. Untuk menampilkan *frame header*.
2. Mengarahkan pada tombol *logout* dengan memanfaatkan *onclick="javascript:goToPage('authentication.do?value(actions)=logout');return false;"* seperti yang dapat dilihat pada gambar 4.24.
3. Menekan tombol *logout* yang telah diarahkan.



Gambar 4.24 Halaman Logout Internet Banking BCA

4.2.2 Proses Scraping BTN

Sebelum melakukan proses *scraping* untuk mengambil data mutasi BTN, nasabah harus melakukan *login* pada halaman *login internet banking* BTN yang ada pada aplikasi cek mutasi bank.

```

41 <form class="login100-form validate-form" autocomplete="off" action="{{url('/btn/scrap')}}" method="POST">
42     @csrf
43     <span class="login100-form-label">
44
45
46     
48
49     <span class="login100-form-title p-b-34 p-t-17">
50         @if(session('alert'))
51         <div class="alert" role="alert" id="myAlert">
52             <div class="alert alert-danger alert-dismissible fade show" role="alert">
53                 <span class="badge badge-pill badge-danger"></span> {{ session('alert') }}
54                 <button type="button" onclick="closeAlert()" class="close" data-dismiss="alert" aria-label="Close">
55                     <span aria-hidden="true">&times;</span></button>
56             </div>
57         </div>
58     </span>
59
60     @else
61     log in
62     @endif
63
64     </span>
65
66     <div class="wrap-input100 validate-input" data-validate = "Enter username">
67         <input class="input100" required type="text" name="username" placeholder="Username">
68         <span class="focus-input100" data-placeholder="&#x20;"></span>
69     </div>
70
71     <div class="wrap-input100 validate-input" data-validate="Enter password">
72         <input class="input100" required type="password" name="password" placeholder="Password">
73         <span class="focus-input100" data-placeholder="&#x20;"></span>
74     </div>
75
76     <div class="wrap-input100 validate-input">
77         <input class="input100" required type="text" name="remember" placeholder="Ingat Berapa Hari Anda Login?">
78         <span class="focus-input100" data-placeholder="&#x20;"></span>
79     </div>
80
81     <div class="wrap-input100 validate-input">
82         <input class="input100" required type="text" name="password2" placeholder="Ingat Berapa Hari Anda Login?">
83         <span class="focus-input100" data-placeholder="&#x20;"></span>
84     </div>
85
86     <div class="wrap-input100 validate-input">
87         <input class="input100" required type="text" name="password3" placeholder="Ingat Berapa Hari Anda Login?">
88         <span class="focus-input100" data-placeholder="&#x20;"></span>
89     </div>
90
91     <div class="container-login100-form-btn">
92         <button class="login100-form-btn">
93             Login
94         </button>
95     </div>
96
97 </form>

```

Gambar 4.25 Login Internet Banking BTN

Gambar 4.25 merupakan *source code form login internet banking* BTN. *Source code* digunakan untuk membuat tampilan *form login*, menampung dan mengirimkan data yang diisi pada *form login* oleh nasabah.

```
Route::POST('/btn/scrap', 'BTNController@cek');
```

Gambar 4.26 Routing Login Internet Banking BTN

Gambar 4.26 merupakan *source code* yang digunakan untuk mengarahkan dari *form login* ke fungsi cek yang ada pada *controller* (BTNController).


```

1  import io
2  import sys
3  import json
4  import time
5  import urllib
6  import getpass
7  from selenium import webdriver
8  from selenium.webdriver.common.by import By
9  from selenium.webdriver.support.ui import WebDriverWait
10 from selenium.webdriver.support import expected_conditions as EC
11 from selenium.common.exceptions import TimeoutException
12 from selenium.webdriver.support.ui import Select
13 import unicodedata
14 from bs4 import BeautifulSoup
15
16
17 x = str(sys.argv[1])
18
19 y = str(sys.argv[2])
20
21 a = str(sys.argv[3])
22
23 b = str(sys.argv[4])
24
25 c = str(sys.argv[5])
26
27 z = str(sys.argv[6])
28
29 bulan = str(sys.argv[7])
30
31 class btn(object):
32
33     __url = 'https://internetbanking.btn.co.id/retail/login/loginRequest'
34     # __urlC = 'https://internetbanking.btn.co.id/retail/login/checkCookie'
35     def __init__(self):
36         self.formLogin()

```

1

2

3

4

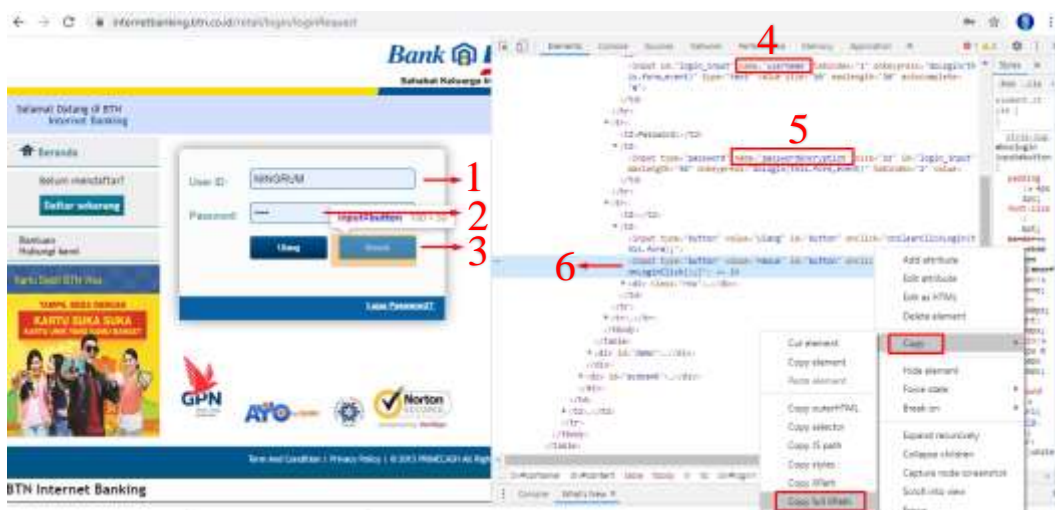
5

Gambar 4.28 Menangkap Nilai yang Dikirim *Controller*

Gambar 4.28 merupakan bagian *source code* pada *file* python untuk *scraping* BTN. Penjelasan gambar 4.28 sebagai berikut:

1. *Import library.*
2. Menangkap nilai yang dikirim *controller* dan menyimpannya kedalam variabel-variabel.
3. Penamaan *class*.
4. Membuat variabel untuk menampung alamat *internet banking* BTN.
5. Menjalankan fungsi *formLogin*.

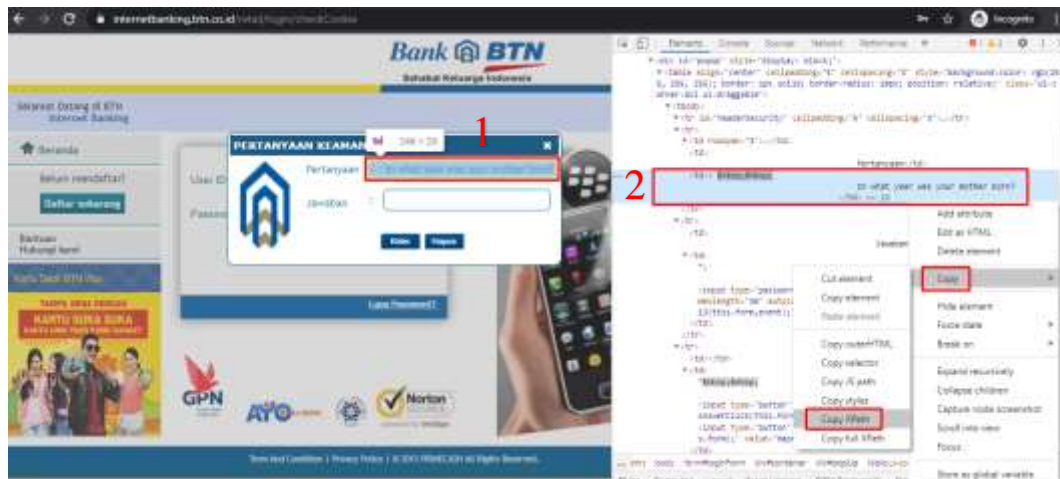
6. Mengisikan *username* dari aplikasi cek mutasi bank ke *username* yang ada pada *form login internet banking* BTN sesuai dengan yang telah diarahkan.
7. Mengisikan *password* dari aplikasi cek mutasi bank ke *password* yang ada pada *form login internet banking* BTN sesuai dengan yang telah diarahkan.
8. Menekan tombol *login* yang ada pada *form login internet banking* BTN setelah *field username* dan *password* terisi.
9. Mengarahkan pada pertanyaan *pop-up* dengan memanfaatkan *Xpath* dan menyimpan pertanyaan pada variabel *pop-up*. Dapat dilihat pada gambar 4.31.
10. Mengubah nilai dari variabel *pop-up* menjadi *string* dan disimpan pada variabel *pq*.
11. Menghapus tanda “.” pada pertanyaan yang ada pada variabel *pq* dan menyimpannya pada variabel *string*.
12. Memotong spasi atau karakter *string* yang ditentukan di sebelah kiri.
13. Menjalankan fungsi *validation*.
14. Menjalankan fungsi cek mutasi.
15. Menjalankan fungsi *logout*.
16. Menjalankan fungsi *sleep*.
17. Menjalankan fungsi *logout* apabila terjadi masalah pada saat menjalankan fungsi-fungsi sebelumnya.



Gambar 4.30 Pengisian *form login Internet Banking* BTN

Gambar 4.30 merupakan tampilan halaman *form login internet banking* BTN. Penjelasan gambar 4.30 sebagai berikut:

1. *Field username internet banking* BTN.
2. *Field password internet banking* BTN.
3. Tombol *login internet banking* BTN.
4. *Name* dari *field username* yang digunakan untuk mengarahkan pengisian data *username* dari aplikasi cek mutasi ke *internet banking* BTN.
5. *Name* dari *field password* yang digunakan untuk mengarahkan pengisian data *password* dari aplikasi cek mutasi ke *internet banking* BTN.
6. *Inspect element* dari tombol *login* yang *Xpath*-nya digunakan untuk mengarahkan agar setelah pengisian *username* dan *password* dapat langsung menekan tombol *login* yang ada pada *internet banking* BTN.



Gambar 4.31 Mengambil Pertanyaan

Gambar 4.31 merupakan tampilan halaman *pop-up* setelah mengisi *form login internet banking* BTN. Penjelasan gambar 4.31 sebagai berikut:

1. Pertanyaan *pop* yang diambil.
2. *Inspect element* dan mengambil *Xpath* yang ada pada gambar 4.31 yang digunakan untuk mengambil nilai dari pertanyaan (pertanyaan apa yang muncul).

```

def validation(self):
    # self.__driver.find_element_by_name("securityAnswer")
    # loginBTN = self.__driver.find_element_by_xpath("//html/body/form/div/div[4]/table/tr[4]/td/input[1]")
    # AccBTN = self.__driver.find_element_by_xpath("//html/body/form/div/div[4]/table/tr[4]/td/input[2]")
    loginBTN = self.__driver.wait.until(EC.presence_of_element_located((By.XPATH, "securityAnswer")))
    AccBTN = self.__driver.wait.until(EC.presence_of_element_located((By.XPATH, "//html/body/form/div/div[4]/table/tr[4]/td/input[2]")))
    print(self.pertanyaan)
    if self.pertanyaan == "In what city were you born?":
        # print(self.pertanyaan)
        # print ("BANYUWANG")
        loginBTN.send_keys(self.__jawab1)
        AccBTN.click()
    elif self.pertanyaan == "In what year was your mother born?":
        # print(self.pertanyaan)
        # print ("1978")
        loginBTN.send_keys(self.__jawab2)
        AccBTN.click()
    elif self.pertanyaan == "In what year was your father born?":
        # print(self.pertanyaan)
        # print ("1988")
        loginBTN.send_keys(self.__jawab3)
        AccBTN.click()
    else:
        keluar = "http://internetbanking.bni.co.id/retail/login/logout"
        self.__driver.get(keluar)
        print("sudah selesai")

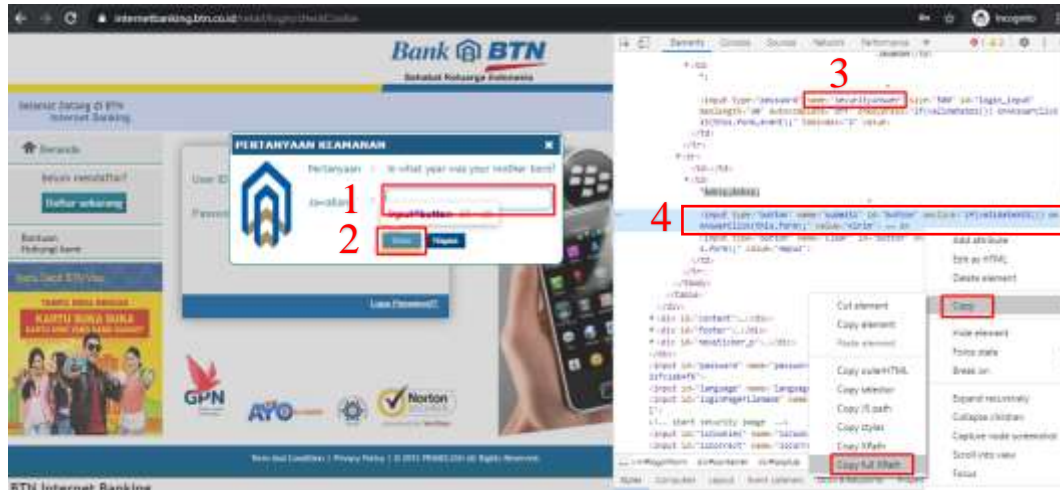
```

Gambar 4.32 Source Code Menjawab Pertanyaan Pop-up

Gambar 4.32 merupakan bagian *source code* pada *file* python untuk *scraping* BTN dan digunakan untuk menjawab pertanyaan *pop-up* setelah mengisi *form* login internet banking BTN. Penjelasan gambar 4.32 sebagai berikut:

1. Mengarahkan pada *field* jawaban dengan memanfaatkan *name* (securityAnswer) seperti yang dapat dilihat pada gambar 4.33.
2. Mengarahkan pada tombol kirim dengan memanfaatkan *Xpath* seperti yang dapat dilihat pada gambar 4.33.
3. Jika pertanyaan dikenali “*In what city were you born?*” maka jawaban yang dikirimkan pada *field* jawaban *pop-up* adalah jawaban pertanyaan pertama yang telah diisikan pada saat *login* aplikasi cek mutasi bank dan akan menekan tombol kirim seperti yang dapat dilihat pada gambar 4.33.
4. Jika pertanyaan dikenali “*In what year was your mother born?*” maka jawaban yang dikirimkan pada *field* jawaban *pop-up* adalah jawaban pertanyaan kedua yang telah diisikan pada saat *login* aplikasi cek mutasi bank dan akan menekan tombol kirim seperti yang dapat dilihat pada gambar 4.33.
5. Jika pertanyaan dikenali “*In what year was your father born?*” maka jawaban yang dikirimkan pada *field* jawaban *pop-up* adalah jawaban pertanyaan ketiga yang telah diisikan pada saat *login* aplikasi cek mutasi bank dan akan menekan tombol kirim seperti yang dapat dilihat pada gambar 4.33.

6. Jika tidak ada pertanyaan yang dikenali sama dengan tiga pertanyaan tersebut maka akan dijalankan fungsi untuk *logout* dari *internet banking* BTN.



Gambar 4.33 Menjawab Pertanyaan *Pop-up*

Gambar 4.33 merupakan tampilan halaman *pop-up* setelah mengisi form login internet banking BTN. Penjelasan gambar 4.33 sebagai berikut:

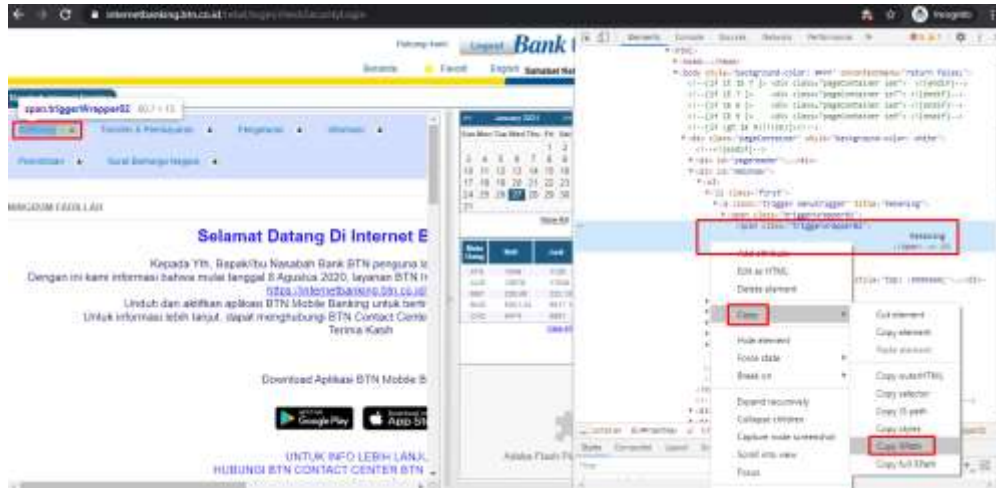
1. *Field* untuk menjawab pertanyaan.
2. Tombol kirim digunakan untuk mengirimkan jawaban pertanyaan *pop-up*.
3. Diambil *name* dari *field* jawaban yang digunakan untuk mengarahkan pengisian data jawaban dari aplikasi cek mutasi ke *internet banking* BTN.
4. *Inspect element* dari tombol kirim yang *Xpath*-nya digunakan untuk mengarahkan pada tombol kirim.



Gambar 4.34 Menekan Beberapa Menu, *Scraping* dan *Export* Data *Json*

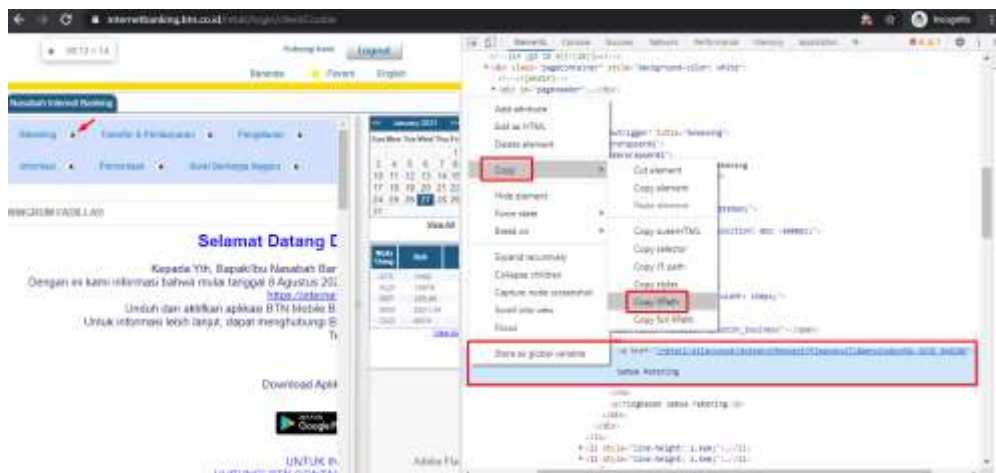
Gambar 4.34 merupakan bagian *source code* pada file python untuk *scraping* BTN dan digunakan untuk membuka menu dan *scraping* data mutasi internet banking BTN. Penjelasan gambar 4.34 sebagai berikut:

1. Mengarahkan selenium pada *frame* menu.
2. Mengarahkan pada menu rekening dengan memanfaatkan *id* (mainNav) dan *Xpath* lalu menekan rekening seperti pada gambar 4.35.



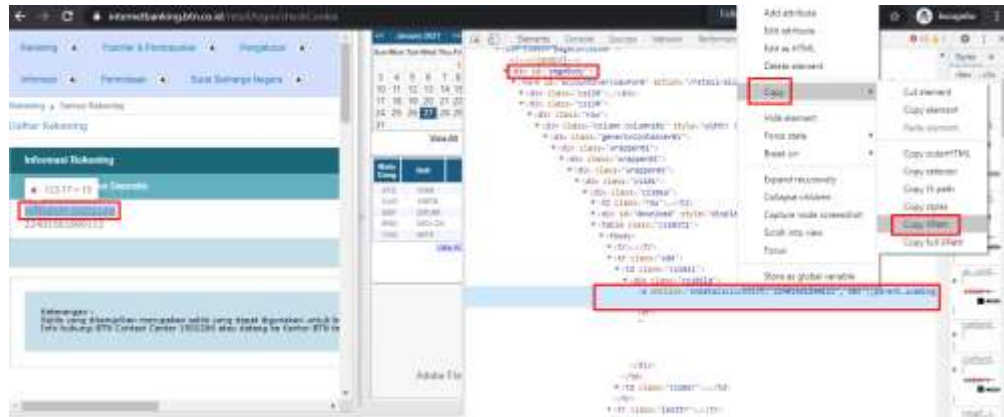
Gambar 4.35 Menu Rekening

3. Mengarahkan pada menu semua rekening dengan memanfaatkan *id* (mainNav) dan *Xpath* lalu menekan semua rekening seperti pada gambar 4.36.



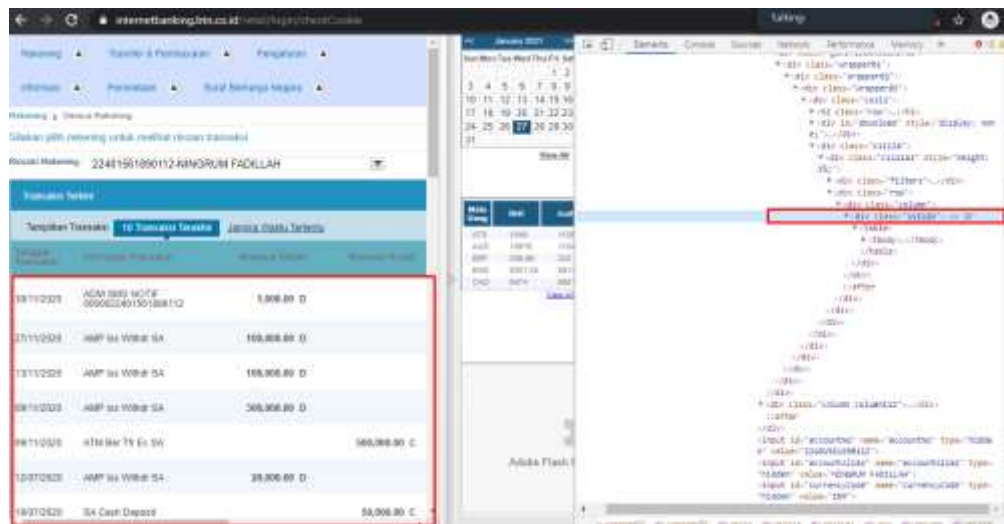
Gambar 4.36 Menu Semua Rekening

4. Mengarahkan selenium pada *frame* daftar rekening (mainFrame).
5. Mengarahkan pada nama pengguna (NINGRUM FADILLAH) dengan memanfaatkan *id* (pageBody) dan *Xpath* lalu menekan nama pengguna (NINGRUM FADILLAH) seperti pada gambar 4.37.



Gambar 4.37 Nama Pengguna

6. Menekan nama pengguna yang telah diarahkan pada nomor 5.
7. Mengarahkan pada tabel mutasi dengan memanfaatkan *class* (cs111b) seperti pada gambar 4.38.



Gambar 4.38 Tabel Mutasi BTN

8. Membuat Variabel data yang berfungsi untuk menampung data *array* untuk *file* json yang akan dibuat.
9. Untuk mengekstrak elemen-elemen html pada tabel yang telah arahkan lalu disimpan pada variabel *table_mutasi*.
10. Mengambil semua *tbody* dari variabel *tabel_mutasi* dan disimpan pada variabel *table_body*.
11. Mengambil semua *tr* dari variabel *table_body* dan disimpan pada variabel *rows*.
12. Mengambil semua *td* dari variabel *rows* dan disimpan pada variabel *cols*.

13. Menghapus karakter *default* (*whitespace* ataupun *tab*) awal pada *string* dan menghapus karakter tertentu (`\xa0` dan `\u00a0`). Serta mengubah karakter “D” menjadi “DB” dan karakter “C” menjadi “CR”.
14. Mengisikan data yang telah diproses ke dalam variabel data.
15. Membuat *file* json dengan nama `bulan_id nasabah_data_btn_new.json`
16. Mengisikan data yang ada pada variabel data ke dalam *file* json yang dibuat sebelumnya dimulai dari *array* ke-1 (*default*-nya dari *array* ke-0) hingga *array* ke-(n-1).
17. Menjalankan fungsi *logout* apabila terjadi masalah pada saat menjalankan fungsi-fungsi cekMutasi.

```
def logout(self):
    keluar = "https://internetbanking.btn.co.id/retail/login/logout" → 1
    self.__driver.get(keluar) → 2
```

Gambar 4.39 Logout BTN

Gambar 4.39 merupakan bagian *source code* pada *file* python untuk *scraping* BTN dan digunakan untuk *logout* dari *internet banking* BTN setelah data mutasi didapatkan. Penjelasan gambar 4.39 sebagai berikut:

1. Membuat variabel dengan nama `keluar` yang didalamnya berisi alamat untuk keluar dari *internet banking* BTN.
2. Memanggil variabel `keluar`.

4.2.3 Cleaning Data

1. Cleaning Data BCA

```
[[{"idData": "1", "idDescription": "1", "idBranch": "1", "idAccount": "1", "idBalance": "1"}, {"idData": "2", "idDescription": "2", "idBranch": "2", "idAccount": "2", "idBalance": "2"}]]
```

Gambar 4.40 Data Awal Mutasi BCA

Gambar 4.40 merupakan data awal yang berhasil didapatkan. Data tersebut berisi semua *field* dari tabel yang mutasi BCA yang telah di-*scraping*.

```
for row in rows :
    cols = row.find_all('td')
    1 cols = [ele.text.strip().replace(",",".") for ele in cols]
    data.append([ele for ele in cols if ele])
# print(data)
file = open(b+" "+z+" data_bca_new.json", "w")
2 json.dump(data[1:], file)
```

Gambar 4.41 Source Code Cleaning Data Mutasi BCA

Gambar 4.41 merupakan *source code* yang digunakan untuk memperbaiki atau menghapus data. Penjelasan gambar 4.41 sebagai berikut:

1. Menghapus karakter *default* (*whitespace* ataupun tab) awal dan akhir pada *string* dan mengubah karakter “,” menjadi “.”.
2. Mengisikan data yang ada pada variabel data ke dalam *file* json yang dibuat sebelumnya dimulai dari *array* ke-1 (*default*-nya dari *array* ke-0).

```
[["18/12", "BIAYA ADM", "0000", "7.500.00", "DB", "17.500.00"]]
```

Gambar 4.42 Hasil *Cleaning* Data Mutasi BCA

Gambar 4.42 merupakan data yang telah diproses dengan *source code* pada gambar 4.41. Data tersebut yang kemudian di masukkan ke dalam *database*.

2. *Cleaning* Data BTN



Gambar 4.43 Data Awal Mutasi BTN

Gambar 4.43 merupakan data awal yang berhasil didapatkan. Data tersebut berisi semua *field* dari tabel yang mutasi yang telah di-*scraping*.



Gambar 4.44 *Source Code* *Cleaning* Data Mutasi BTN

Gambar 4.44 merupakan *source code* yang digunakan untuk memperbaiki atau menghapus data. Penjelasan gambar 4.44 sebagai berikut:

1. Menghapus karakter *default* (*whitespace* ataupun tab) awal pada *string* dan menghapus karakter tertentu (`\xa0` dan `\u00a0`). Serta mengubah karakter “D” menjadi “DB” dan karakter “C” menjadi “CR”.
2. Mengisikan data yang ada pada variabel data ke dalam *file* json yang dibuat sebelumnya dimulai dari *array* ke-1 (*default*-nya dari *array* ke-0) hingga *array* ke-(n-1).

```
[["30/11/2020", "ADBMSMSNOTIF0000022401561890112", "1,000.00DB", "", "29,000.00"],
["27/11/2020", "AMPIsswithdrSA", "100,000.00DB", "", "30,000.00"],
["13/11/2020", "AMPIsswithdrSA", "100,000.00DB", "", "130,000.00"],
["09/11/2020", "AMPIsswithdrSA", "300,000.00DB", "", "230,000.00"],
["09/11/2020", "ATMBertfExSA", "", "500,000.00CR", "530,000.00"],
["12/07/2020", "AMPIsswithdrSA", "20,000.00DB", "", "30,000.00"],
["10/07/2020", "SACRashDBeposit", "", "50,000.00CR", "50,000.00"]]
```

Gambar 4.45 Hasil *Cleaning* Data Mutasi BTN

Gambar 4.45 merupakan data yang telah diproses dengan *source code* pada gambar 4.44. Data tersebut yang kemudian di masukkan ke dalam *database*.

4.2.4 Menampilkan Data Mutasi

Data mutasi yang ditampilkan pada halaman cek mutasi milik nasabah awalnya harus melalui bebarapa proses yaitu: memasukkan data json BCA dan BTN pada *database* lalu menampilkan data dari *database* ke halamn web cek mutasi, barulah data mutasi dapat dilihat oleh nasabah.

```
$url_bca = public_path() . "/" . $bulan . "-" . $nasabah . "_data_bca_new.json";
$datos_bca = file_get_contents($url_bca);
$data_bca = json_decode($datos_bca, true);

$data_bca = array_filter($data_bca);

$data_bca = collect($data_bca)->all();

foreach ($data_bca as $d) {
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "CekMutasi-Database";
    $tanggal = strtr($d[0], '/', '-');
    // $tanggal = date('d-m', strtotime($tanggal));
    $tanggal = $tanggal . '-' . date('Y');
    $tanggal_diakses = date("Y-m-d");
    $tanggal = date("Y-m-d", strtotime($tanggal));

    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // $sql = "INSERT INTO mutasi_bca (tanggal,nominal_mutasi,tipe_mutasi,total,bank,tanggal_diakses,id_nasabah)
    VALUES ('$tanggal','$d[3]','$d[4]','$d[5]','bca','$tanggal_diakses','$nasabah')";
    $sql = "INSERT INTO mutasi_nasabah (tanggal,nominal_mutasi,tipe_mutasi,total,bank,tanggal_diakses,id_nasabah)
    VALUES ('$tanggal','$d[3]','$d[4]','$d[5]','bca','$tanggal_diakses','$nasabah')";

    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Connected successfully";
    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }

    mysqli_close($conn);
}

return redirect('/cekmutasiku')->with('alert','Berhasil Menambahkan Data');
```

Gambar 4.46 Memasukkan Data Json BCA Pada *Database*

Setelah proses *scraping* data mutasi BCA berhasil akan didapatkan data dalam bentuk json yang kemudian akan di masukkan ke dalam *database*. Proses

memasukkan data json bca pada *database* dapat dilihat pada gambar 4.46 dengan penjelasan sebagai berikut:

1. *source code* yang digunakan untuk mengambil data dari *file* json yang menampung data mutasi BCA dan mengubahnya menjadi bentuk *array*.
2. *source code* yang digunakan untuk membuat koneksi ke *database*.
3. *source code* yang digunakan untuk menambahkan data dari *file* json yang telah diubah menjadi bentuk *array* yang kemudian menambahkannya ke dalam tabel *database* (*mutasi_nasabah*).
4. *source code* yang digunakan untuk membuat pesan apabila *database* data berhasil ditambahkan atau tidak.
5. *source code* yang digunakan untuk mengarahkan *URL* ke halaman “/cekmutasiku”.

```
$url_btn = public_path() . "/storage/"; $nama_btn = "data_btn_bca.json";
$data_btn = file_get_contents($url_btn);
$data_btn = json_decode($data_btn, true);

$data_btn = array_filter($data_btn);
$data_btn = array_values($data_btn);
$data_btn = collect($data_btn->all());

foreach ($data_btn as $d) {
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "cekmutasi_database";
    $tanggal = date("Y-m-d");
    $tanggal = date("d-m-Y", strtotime($tanggal));
    // $tanggal = date("Y-m-d", strtotime($tanggal));
    $tanggal_diakses = date("Y-m-d");

    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $dbname);

    $s = $d[0];
    $stipensi = substr($s, 0);
    $stipensi2 = substr($s, 10);
    $stipensi3 = substr($s, 11);
    $stipensi4 = substr($s, 12);
    // $stipensi4 = substr($s, 13);
    if ($stipensi == "CK" || $stipensi2 == "CK" || $stipensi3 == "CK" || $stipensi4 == "CK") {

        $s = substr($s[0], 0, 1);
        // $s = substr($s[0], 0, 1);

        if ($s[0] == "0") {
            $sql = "INSERT INTO mutasi_nasabah ($tanggal, nominal_mutasi, tipe_mutasi, total, bank, tanggal_diakses, id_nasabah) VALUES ('$tanggal', '$s', 'CK', '$s[0]', '$stipensi', '$tanggal_diakses', '$nama_btn')";
        } else {
            $s = substr($s[0], 0, 1);
            $sql = "INSERT INTO mutasi_nasabah ($tanggal, nominal_mutasi, tipe_mutasi, total, bank, tanggal_diakses, id_nasabah) VALUES ('$tanggal', '$s', 'NB', '$s[0]', '$stipensi', '$tanggal_diakses', '$nama_btn')";
        }

        if ($conn) {
            die("Connection failed: " . mysqli_connect_error());
        } else {
            echo "Connected successfully";
        }
        if (mysqli_query($conn, $sql)) {
            echo "New record created successfully";
        } else {
            echo "Error: " . $sql . " -> " . mysqli_error($conn);
        }
        mysqli_close($conn);
    }
}

return redirect('/cekmutasiku')->with('alert', 'Berhasil menambahkan Data');
```

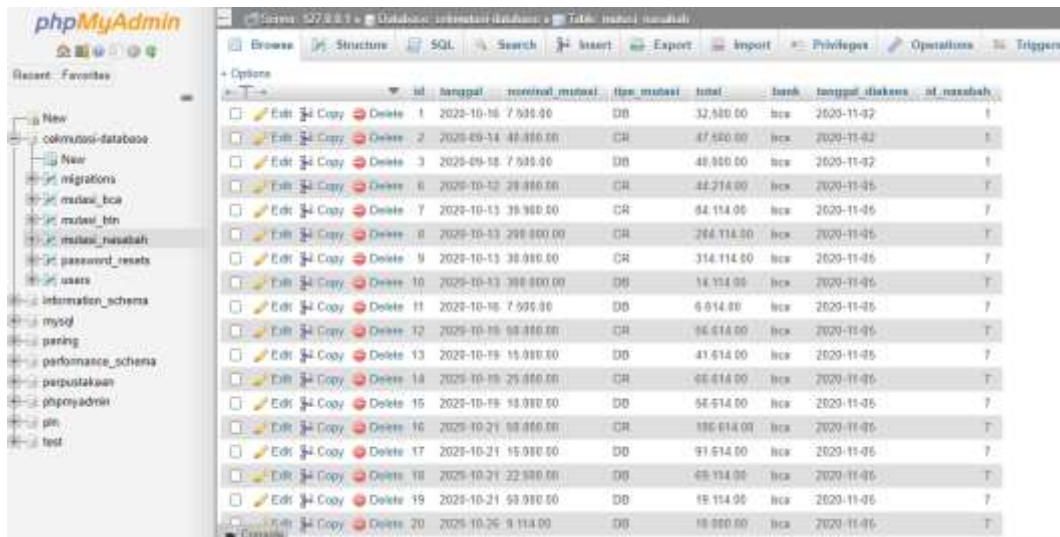
The image shows a PHP script with five red annotations: 1 points to the JSON file loading and parsing code; 2 points to the database connection setup; 3 points to the SQL query construction and execution; 4 points to the success and error handling logic; and 5 points to the final redirect statement.

Gambar 4.47 Memasukkan Data Json BTN Pada *Database*

Setelah proses *scraping* data mutasi BTN berhasil akan didapatkan data dalam bentuk json yang kemudian akan di masukkan ke dalam *database*. Proses

memasukkan data json bca pada *database* dapat dilihat pada gambar 4.47 dengan penjelasan sebagai berikut:

1. *Source code* yang digunakan untuk mengambil data dari *file* json yang menampung data mutasi BTN dan mengubahnya menjadi bentuk *array*.
2. *Source code* yang digunakan untuk membuat koneksi ke *database*.
3. *Source code* yang digunakan untuk menambahkan data dari *file* json yang telah diubah menjadi bentuk *array* yang kemudian menambahkannya ke dalam tabel *database* (*mutasi_nasabah*).
4. *Source code* yang digunakan untuk membuat pesan apabila *database* data berhasil ditambahkan atau tidak.
5. *Source code* yang digunakan untuk mengarahkan *URL* ke halaman “/cekmutasiku”.



	id	tanggal	nominal_mutasi	tipe_mutasi	total	bank	tanggal_diakses	id_nasabah
	1	2020-10-16	7.935.00	DB	32.580.00	bca	2020-11-02	1
	2	2020-09-14	48.880.00	GR	47.600.00	bca	2020-11-02	1
	3	2020-09-14	7.935.00	DB	48.980.00	bca	2020-11-02	1
	6	2020-10-12	28.880.00	GR	44.214.00	bca	2020-11-05	1
	7	2020-10-11	38.980.00	GR	64.114.00	bca	2020-11-05	1
	8	2020-10-13	288.800.00	GR	264.114.00	bca	2020-11-05	1
	9	2020-10-13	38.980.00	GR	314.114.00	bca	2020-11-05	1
	10	2020-10-13	388.800.00	DB	14.114.00	bca	2020-11-05	1
	11	2020-10-16	7.935.00	DB	6.614.00	bca	2020-11-05	1
	12	2020-10-16	68.880.00	GR	96.614.00	bca	2020-11-05	1
	13	2020-10-16	18.980.00	DB	41.614.00	bca	2020-11-05	1
	14	2020-10-16	25.880.00	GR	66.614.00	bca	2020-11-05	1
	15	2020-10-16	18.980.00	DB	54.614.00	bca	2020-11-05	1
	16	2020-10-21	98.880.00	GR	186.614.00	bca	2020-11-05	1
	17	2020-10-21	18.980.00	DB	91.614.00	bca	2020-11-05	1
	18	2020-10-21	22.880.00	DB	48.114.00	bca	2020-11-05	1
	19	2020-10-21	58.980.00	DB	15.114.00	bca	2020-11-05	1
	20	2020-10-26	9.114.00	DB	18.980.00	bca	2020-11-05	1

Gambar 4.48 Data Mutasi Pada *Database*

Setelah data mutasi berhasil diinputkan maka data akan berada pada tabel *mutasi_nasabah* yang telah disediakan pada *database* seperti yang dapat dilihat pada gambar 4.48. Tabel *mutasi_nasabah* berisi semua data dari mutasi BCA dan BTN. Data-data tersebut yang kemudian akan ditampilkan pada halaman cek mutasi sesuai dengan *id* nasabah yang menggunakan aplikasi cek mutasi bank.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Auth;
7  use PDF;
8  use Date;
9  use DB;
10 use App\Exports\MutasiExport;
11 class CekMutasiController extends Controller
12 {
13     //
14     // Display a listing of the resource.
15     //
16     // @return \Illuminate\Http\Response
17     //
18     public function index(Request $request)
19     {
20         $bulan = date("d-m-y");
21         $id_nasabah = Auth::user()->id;
22
23         // dd($bulan);
24
25         $data_bca = DB::table('mutasi_nasabah')
26             ->where('id_nasabah',$id_nasabah)
27             ->where('bank','bca')
28             ->groupBy(['tanggal','time_mutasi','nominal_mutasi'])->get();
29
30         $data_bta = DB::table('mutasi_nasabah')
31             ->where('id_nasabah',$id_nasabah)
32             ->where('bank','bta')
33             ->groupBy(['tanggal','time_mutasi','nominal_mutasi'])->take(10)->get();
34
35         $data_report = $data_bca->merge($data_bta);
36         $data_report = $data_report->sortBy('tanggal');
37
38         $data = DB::table('mutasi_nasabah')
39             ->groupBy(['time_mutasi'])
40             ->select(
41                 DB::raw('time_mutasi as a'),
42                 DB::raw('tanggal as tanggal'),
43                 DB::raw('sum(nominal_mutasi as number)')
44             )->where('id_nasabah',$id_nasabah)
45             // ->havingRaw('COUNT(*) > 4')
46             ->get();
47
48         $array[] = ['a', 'number'];
49         foreach($data as $key => $value)
50         {
51             $array[++$key] = [$value->a, ("<div>No " . number_format(((($value->number)/1000),2,'','.'),1,$value->number);
52
53         }
54
55         $totalrc = DB::table('mutasi_nasabah')
56             ->where('id_nasabah',$id_nasabah)
57             ->where('time_mutasi','CR')
58             ->get(DB::raw('sum(nominal_mutasi as Totalcredit)'));
59         $totalCR = $totalrc->sum(0,1);
60         $totalDR = DB::table('mutasi_nasabah')
61             ->where('id_nasabah',$id_nasabah)
62             ->where('time_mutasi','DR')
63             ->get(DB::raw('sum(nominal_mutasi as Totaldebit)'));
64         $totalDR = $totalDR->sum(0,1);
65
66         // dd($totalDR);
67         return view('nasabah.laporan',compact('mutasi_bca','mutasi_bta','data_report','TOTALCR','TOTALDR'))->with('data',$json_encode($array));
68     }
69 }

```

Gambar 4.49 Controller CekMutasiController

Menampilkan data mutasi dari *database* ke dalam halaman *web* cek mutasi memerlukan konfigurasi pada *controller* CekMutasiController seperti gambar 4.49. Penjelasan gambar 4.49 sebagai berikut:

1. *Source code* yang digunakan untuk mengambil data tanggal dan waktu sekarang.
2. *Source code* yang digunakan untuk mengambil *id user*.
3. *Source code* yang digunakan untuk mengambil data mutasi (hanya data mutasi bank bca) dari *database*.
4. *Source code* yang digunakan untuk mengambil data mutasi (hanya data mutasi bank bca) dari *database*.

5. *Source code* yang digunakan untuk menggabungkan data mutasi bca dari variabel “data_bca” dan data mutasi btn dari variabel “data_btn” yang kemudian diurutkan berdasarkan tanggal.
6. *Source code* yang digunakan untuk mengambil data dari tabel mutasi_nasabah dan dijumlahkan berdasarkan tipe_mutasi dan id_nasabah.
7. *Source code* yang digunakan untuk menjumlahkan data dari table mutasi_nasabah dengan tipe_mutasi adalah “CR”.
8. *Source code* yang digunakan untuk menjumlahkan data dari table mutasi_nasabah dengan tipe_mutasi adalah “DB”.
9. *Source code* yang digunakan untuk mengarahkan *URL* ke halaman “/cekmutasiku”.

```

106 <div class="panel-body" id="container">
107   @if(session('alert'))
108     <div class="alert" role="alert" id="myAlert">
109       <div class="alert alert-danger alert-dismissible fade show" role="alert">
110         <span class="badge badge-pill badge-danger"></span> {{ session('alert') }}
111         <button type="button" onclick="closeAlert()" class="close" data-dismiss="alert" aria-label="Close">
112           <span aria-hidden="true">&times;</span>
113         </button>
114       </div>
115     </div>
116   @endif
117   <table id="example" class="display nowrap table-striped table-bordered table" style="width:100%">
118     <!-- <div class="col-lg-1 col-xs-2 thumbnail" style="margin-left: 30em"><
119     <br> -->
120
121
122     <thead>
123       <tr>
124         <th>No</th>
125         <th>Tanggal</th>
126         <th>Nominal Mutasi</th>
127         <th>Tipe Mutasi</th>
128         <th>Nominal Akhir</th>
129         <th>Bank</th>
130         <th>Tanggal Akses Mutasi</th>
131       </tr>
132     </thead>
133
134     <tbody>
135
136       <?php $i = 1; ?>
137
138       @foreach($data_report as $value)
139
140         <tr >
141
142           <td>{{ $i }}</td>
143
144           <td>{{Carbon\Carbon::parse($value->tanggal)->translatedFormat('d F Y')}}</td>
145           <td style="text-align: right">Rp. {{ $value->nominal_mutasi}}</td>
146           <td style="text-align: right">{{ $value->tipe_mutasi}}</td>
147           <td style="text-align: right">Rp. {{ $value->total}}</td>
148           <td style="text-align: right">Rp. {{ $value->bank}}</td>
149           <td>{{Carbon\Carbon::parse($value->tanggal_diakses)->translatedFormat('d F Y')}}</td>

```

```

150 |
151 |         <?php $i++ ?>
152 |
153 |     </tr>
154 |     @endforeach
155 |
156 | </tbody>
157 |
158 | </table>
159 | @if($TotalCR!=" " && $TotalDB!=" ")
160 | <tr>
161 |
162 |     <td>Total Kredit : Rp. {{number_format((( $TotalCR)*1000),2,',','')}}</td>
163 |
164 | </tr>
165 | <tr>
166 |
167 |     <td>Total Debit : Rp. {{number_format((( $TotalDB)*1000),2,',','')}}</td>
168 |
169 | </tr>
170 | @else
171 | @endif
172 |
173 | @endif
174 |
175 | </div>

```

Gambar 4.50 View cekmutasi.blade.php

Gambar 4.50 merupakan *source code* yang digunakan untuk membuat tampilan cek mutasi yang menampilkan data mutasi BCA dan BTN pada aplikasi cek mutasi bank. Hasil tampilan halaman cek mutasi bank dapat dilihat pada gambar 4.51.

No	Tanggal	Nominal Mutasi	Tipe Mutasi	Nominal Akut	Bank	Tanggal Akut Mutasi
1	30 July 2020	Rp. 300,000.00	CR	Rp. 300,000.00	BTN	10 December 2020
2	30 July 2020	Rp. 200,000.00	DB	Rp. 30,000.00	BTN	10 December 2020
3	14 September 2020	Rp. 40,000.00	CR	Rp. 47,388.00	BCA	02 November 2020
4	16 September 2020	Rp. 7,500.00	DB	Rp. 40,000.00	BCA	02 November 2020
5	24 Oktober 2020	Rp. 1,000.00	DB	Rp. 52,000.00	BCA	02 November 2020
6	08 November 2020	Rp. 500,000.00	CR	Rp. 530,000.00	BTN	10 December 2020
7	08 November 2020	Rp. 300,000.00	DB	Rp. 230,000.00	BTN	10 December 2020
8	22 November 2020	Rp. 300,000.00	DB	Rp. 130,000.00	BTN	10 December 2020
9	30 November 2020	Rp. 7,500.00	DB	Rp. 25,000.00	BCA	04 Desember 2020
10	27 November 2020	Rp. 330,000.00	DB	Rp. 30,000.00	BTN	10 December 2020

Showing 1 to 10 of 12 entries

Total Kredit : Rp. 640.000,00
Total Debit : Rp. 571.000,00

Gambar 4.51 Data yang Dapat Dilihat Nasabah

4.2.5 Presentase Mutasi

Membuat persentase data debit dan kredit dari BCA dan BTN memerlukan konfigurasi pada *controller* CekMutasiController seperti pada gambar 4.52 yang digunakan untuk mengambil data dari tabel mutasi_nasabah dan dijumlahkan berdasarkan tipe_mutasi dan id_nasabah. Untuk menampilkan data hasil dari *controller* dalam bentuk pie chart menggunakan *source code* pada gambar 4.53.

```

38 $data = DB::table('mutasi_nasabah')
39     ->groupBy(['tipe_mutasi'])
40     ->select(
41         DB::raw('tipe_mutasi as t'),
42         DB::raw('tanggal as tanggal'),
43         DB::raw('sum(nominal_mutasi) as number'))
44     ->where('id_nasabah', 'id_nasabah')
45     ->orderByRaw('total')
46     ->get();
47
48 $array[] = ['t', 'number'];
49 foreach($data as $key => $value)
50 {
51     $array[$key] = [$value->t, "Rp. ", number_format(((($value->number)*1000),2),',','.', ''), ""];
52 }
53
54 $totalCR = DB::table('mutasi_nasabah')
55     ->where('id_nasabah', 'id_nasabah')
56     ->where('tipe_mutasi', 'CR')
57     ->getOneRaw('sum(nominal_mutasi) as totalcredit');
58 $totalCR = substr($totalCR,4,-2);
59 $totalDB = DB::table('mutasi_nasabah')
60     ->where('id_nasabah', 'id_nasabah')
61     ->where('tipe_mutasi', 'DB')
62     ->getOneRaw('sum(nominal_mutasi) as totaldebit');
63 $totalDB = substr($totalDB,4,-2);
64
65 // $totalCR;
66 return view('nasabah.indexmutasi', compact('mutasi_bca', 'mutasi_bni', 'data_report', 'totalCR', 'totalDB'))->with('data', json_encode($array));
67 }

```

Gambar 4.52 Controller CekMutasiController Membuat Persentase Mutasi

Gambar 4.52 merupakan *source code* yang digunakan untuk mengambil data dari tabel *mutasi_nasabah* dan dijumlahkan berdasarkan *tipe_mutasi* dan *id_nasabah* yang nantinya akan ditampilkan dalam bentuk *pie chart*. Mengambil data dari tabel *mutasi_nasabah* dan dijumlahkan berdasarkan *tipe_mutasi* (DB) dan *id_nasabah* yang nantinya akan menampilkan jumlah total debit. Mengambil data dari tabel *mutasi_nasabah* dan dijumlahkan berdasarkan *tipe_mutasi* (CR) dan *id_nasabah* yang nantinya akan menampilkan jumlah total kredit. kemudian mengarahkan *URL* ke halaman “/cekmutasiku”.

```

36 <div id="pie_chart"></div>
212 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
213 <!-- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />-->
214
215 <script type="text/javascript">
216     var analytics = <?php echo $data; ?>
217
218     google.charts.load('current', {'packages':['corechart']});
219     google.charts.setOnLoadCallback(drawChart);
220
221     function drawChart()
222     {
223         var data = google.visualization.arrayToDataTable(analytics);
224         var options = {
225             title : 'Data Mutasi {{Auth::user()->name}}'
226         };
227     };
228     var chart = new google.visualization.Piechart(document.getElementById('pie_chart'));
229     chart.draw(data , options);
230
231 }
232 </script>

```

Gambar 4.53 View cekmutasi.blade.php untuk Menampilkan Pie Chart

Gambar 4.53 merupakan *source code* yang digunakan untuk menangkap nilai yang dikirim controller dan mengkonversi menjadi bentuk *Pie Chart*.

4.2.6 Unduh PDF

Pembuatan data cek mutasi dalam bentuk PDF membutuhkan sebuah *library* *dompdf*. Untuk pemasangan *library* *dompdf* dapat menggunakan

composer dengan cara membuka Command Prompt, kemudian perintah yang dituliskan seperti berikut:

composer require dompdf/dompdf

Agar dapat di jalankan di semua function, pdf_generator perlu ditambahkan ke dalam config/autoload. function untuk menjalankan perintah tersebut di dalam *controller* seperti pada gambar 4.54 dan hasil tampilan PDF yang telah diunduh dapat dilihat pada gambar 4.55.

```
170 public function unduh_pdf_database(Request $request)
171 {
172     $id_nasabah = Auth::user()->id; } 1
173     $nasabah = Auth::user()->nama;
174     $tanggal = date("Y-m-d"); } 2
175     $filter = $request->filter; } 3
176
177     $data_report = DB::table('mutasi_nasabah')->where('id_nasabah',$id_nasabah);
178
179
180     if ($filter == 'all' ) {
181         $data_report = $data_report;
182     }
183     else if ($filter == 'bca' ) {
184         $data_report = $data_report->where('bank','bca');
185     }
186
187     else if ($filter == 'btn' ) {
188         $data_report = $data_report->where('bank','btn');
189     }
190     else if ($filter == 4 ) {
191         if (date($request->tanggal_mulai) <= date($request->tanggal_akhir)) {
192             $data_report = $data_report->whereDate('tanggal','>=', $request->tanggal_mulai_cetak);
193             $data_report = $data_report->whereDate('tanggal','<=', $request->tanggal_akhir_cetak);
194         }
195         else{
196             return redirect()->back()->with('gagal', 'Data Tidak Ada');
197         }
198     }
199
200 }
201
202
203 $data_report = $data_report->get();
204
205 $pdf = PDF::loadView('nasabah.unduh_mutasi', compact('data_report','nasabah','tanggal'))->setPaper('a4', 'landscape'); } 5
206 // dd($data_report);
207 return $pdf->stream('CekMutasi'.$nasabah.' - '.$tanggal.'.pdf');
208
209
210 }
```

Gambar 4.54 Controller PDF

Gambar 4.54 merupakan *source code* yang digunakan untuk membuat *file* PDF yang dapat diunduh. Penjelasan gambar 4.54 sebagai berikut:

1. *Source code* yang digunakan untuk mengambil *id user* dan nama.
2. *Source code* yang digunakan untuk mengambil data tanggal dan waktu sekarang.
3. *Source code* yang digunakan untuk menangkap nilai yang ditangkap *controller* dan menyimpannya kedalam variabel *filter*.
4. *Source code* yang digunakan untuk mengambil data mutasi dari *database* dan memfilternya.
5. *Source code* yang digunakan untuk mencetak data dengan format PDF.

NO	TANGGAL	NISAL	TANGGAL	NISAL
1	10/10/2020	1.000.000	10/10/2020	1.000.000
2	10/10/2020	1.000.000	10/10/2020	1.000.000
3	10/10/2020	1.000.000	10/10/2020	1.000.000
4	10/10/2020	1.000.000	10/10/2020	1.000.000
5	10/10/2020	1.000.000	10/10/2020	1.000.000
6	10/10/2020	1.000.000	10/10/2020	1.000.000
7	10/10/2020	1.000.000	10/10/2020	1.000.000
8	10/10/2020	1.000.000	10/10/2020	1.000.000
9	10/10/2020	1.000.000	10/10/2020	1.000.000
10	10/10/2020	1.000.000	10/10/2020	1.000.000
11	10/10/2020	1.000.000	10/10/2020	1.000.000
12	10/10/2020	1.000.000	10/10/2020	1.000.000
13	10/10/2020	1.000.000	10/10/2020	1.000.000
14	10/10/2020	1.000.000	10/10/2020	1.000.000
15	10/10/2020	1.000.000	10/10/2020	1.000.000

Gambar 4.55 Hasil Tampilan Unduh PDF

Gambar 4.55 merupakan hasil tampilan PDF yang telah diunduh. *File* PDF berisi data mutasi BCA dan BTN sesuai dengan data yang ditampilkan pada gambar 4.51.

4.2.7 Unduh Excel

Pembuatan data cek mutasi dalam bentuk excel kita harus memasang paket tambahan dari luar yaitu laravel excel. Paket laravel excel adalah paket yang di buat oleh *maatwebsite*. Untuk menginstall paket excel kita jalankan perintah: **composer require maatwebsite/excel**

Untuk dokumentasi lengkapnya dapat di akses pada alamat <https://github.com/Maatwebsite/Laravel-Excel> disini kita menggunakan *export* excel secara array. Konfigurasi yang harus di lakukan yaitu kita jalankan perintah: **php artisan make:export MutasiExport**

Konfigurasi *model export* excel dapat dilihat pada gambar gambar 4.56, *function* untuk menjalankan perintah tersebut di dalam *controller* dapat dilihat pada gambar 4.57 dan hasil unduh excel dapat dilihat pada gambar 4.58.


```

1  namespace App\Exports;
2
3  use Maatwebsite\Excel\Concerns\FromCollection;
4
5  use App\User;
6  use Maatwebsite\Excel\Concerns\FromArray;
7  use Maatwebsite\Excel\Concerns\WithHeadings;
8  class MutasiExport implements FromArray, WithHeadings
9  {
10
11      /**
12       * @return \Illuminate\Support\Collection
13       */
14
15      protected $data;
16
17      public function __construct(array $data)
18      {
19          $this->data = $data;
20      }
21
22      1 public function headings(): array
23      {
24          2 return ["no", "tanggal", "nominal mutasi", "tipe mutasi", "nominal akhir", "bank", "tanggal akses mutasi", "id nasabah"];
25      }
26
27      public function array(): array
28      {
29          return $this->data;
30      }
31
32 }

```

Gambar 4.56 Konfigurasi *Model Export Excel*

Gambar 4.56 merupakan *source code* yang digunakan menyiapkan file excel yang akan diunduh. Penjelasan gambar 4.56 sebagai berikut:

1. *Source code* yang digunakan untuk membuat *heading* pada excel yang akan dicetak.
2. *Source code* yang digunakan untuk mengambil data dari array yang akan dicetak menjadi excel.

```

214 public function unduh_excel_database(Request $request)
215 {
216     $id_nasabah = Auth::user()->id;
217     $nasabah = Auth::user()->nama;
218     $tanggal = date("Y-m-d");
219     $filter = $request->filter;
220
221     $data_report = DB::table('mutasi_nasabah')->where('id_nasabah', $id_nasabah);
222
223     if ($filter == 'all' ) {
224         $data_report = $data_report;
225     }
226     else if ($filter == 'bca' ) {
227         $data_report = $data_report->where('bank', 'bca');
228     }
229     else if ($filter == 'btn' ) {
230         $data_report = $data_report->where('bank', 'btn');
231     }
232     else if ($filter == 4 ) {
233         if (date($request->tanggal_mulai) <= date($request->tanggal_akhir)) {
234             $data_report = $data_report->whereDate('tanggal', '>=', $request->tanggal_mulai);
235             $data_report = $data_report->whereDate('tanggal', '<=', $request->tanggal_akhir);
236         }
237         else{
238             return redirect()->back()->with('alert', 'Data Tidak Ada');
239         }
240     }
241
242     $data_report = $data_report->get()->toArray();
243
244     $export = new MutasiExport($data_report);
245     return Excel::download( $export, 'Mutasi '.$nasabah.' Tanggal '.$tanggal.' '.date('Y-m-d').'.xlsx');
246
247 }

```

Gambar 4.57 Konfigurasi *Controller Export Excel*

Gambar 4.57 merupakan *source code* yang digunakan untuk membuat *file* excel yang dapat diunduh. Penjelasan gambar 4.57 sebagai berikut:

1. *Source code* yang digunakan untuk mengambil *id user* dan nama.
2. *Source code* yang digunakan untuk mengambil data tanggal dan waktu sekarang.
3. *Source code* yang digunakan untuk menangkap nilai yang ditangkap *controller* dan menyimpannya kedalam variabel *filter*.
4. *Source code* yang digunakan untuk mengambil data mutasi dari *database* dan memfilternya.
5. *Source code* yang digunakan untuk mencetak data dengan format excel .

No	Tanggal	Nominal Mutasi	Tipe Mutasi	Nominal Akut	Bank	Tanggal Akut Mutasi	ID Nasabah
1	2020-10-18	7.500.00	DB	12.500.00	BCA	2020-11-02	1
2	2020-09-14	40.000.00	CR	47.500.00	BCA	2020-11-02	1
3	2020-09-18	7.500.00	DB	40.000.00	BCA	2020-11-02	1
4	2020-07-10	50.000.00	CR	50.000.00	BTN	2020-11-02	1
5	2020-07-12	25.000.00	DB	30.000.00	BTN	2020-11-02	1
6	2020-11-20	7.500.00	DB	25.000.00	BCA	2020-12-14	1
7	2020-07-10	30.000.00	CR	30.000.00	BTN	2020-12-14	1
8	2020-07-12	30.000.00	DB	30.000.00	BTN	2020-12-14	1
9	2020-07-12	30.000.00	DB	30.000.00	BTN	2020-12-14	1
10	2020-07-10	50.000.00	CR	50.000.00	BTN	2020-12-14	1
11	2020-07-12	25.000.00	DB	30.000.00	BTN	2020-12-14	1
12	2020-11-09	500.000.00	CR	530.000.00	BTN	2020-12-14	1
13	2020-11-09	100.000.00	DB	230.000.00	BTN	2020-12-14	1
14	2020-11-13	100.000.00	DB	130.000.00	BTN	2020-12-14	1
15	2020-11-27	100.000.00	DB	80.000.00	BTN	2020-12-14	1
16	2020-11-30	1.000.00	DB	70.000.00	BTN	2020-12-14	1

Gambar 4.58 Hasil Tampilan Unduh Excel

Gambar 4.58 merupakan hasil tampilan excel yang telah diunduh. *File* excel berisi data mutasi BCA dan BTN sesuai dengan data yang ada pada *database* seperti gambar 4.48.

4.2.8 Waktu Terakhir Login Nasabah

Menampilkan data halaman riwayat nasabah memerlukan konfigurasi pada *event listener* seperti gambar 4.35. Konfigurasi yang harus dilakukan yaitu membuat sebuah *event listener* dengan menjalankan perintah:

php artisan make:listener RecordLoginInfo

Selanjutnya konfigurasi *EventServiceProvider* dapat dilihat pada gambar 4.59.

```

1  <?php
2
3  namespace App\Listeners;
4
5  use Illuminate\Contracts\Queue\ShouldQueue;
6  use Illuminate\Queue\InteractsWithQueue;
7  use Carbon\Carbon;
8  use Illuminate\Auth\Events\Login;
9  class RecordLoginInfo
10 {
11     /**
12      * Create the event listener.
13      *
14      * @return void
15      */
16     public function __construct()
17     {
18         //
19     }
20
21     /**
22      * Handle the event.
23      *
24      * @param object $event
25      * @return void
26      */
27     public function handle(Login $event)
28     {
29         $event->user->forceFill([
30             'last_login_at' => Carbon::now()->toDateTimeString(),
31             // 'last_login_ip' => request()->getClientIp()
32         ])->save();
33     }
34 }

```

Gambar 4.59 Konfigurasi *Event Listener*

Gambar 4.59 merupakan *source code event listener* yang berfungsi untuk menyimpan informasi *login user* dengan men-trigger event *login*.

```

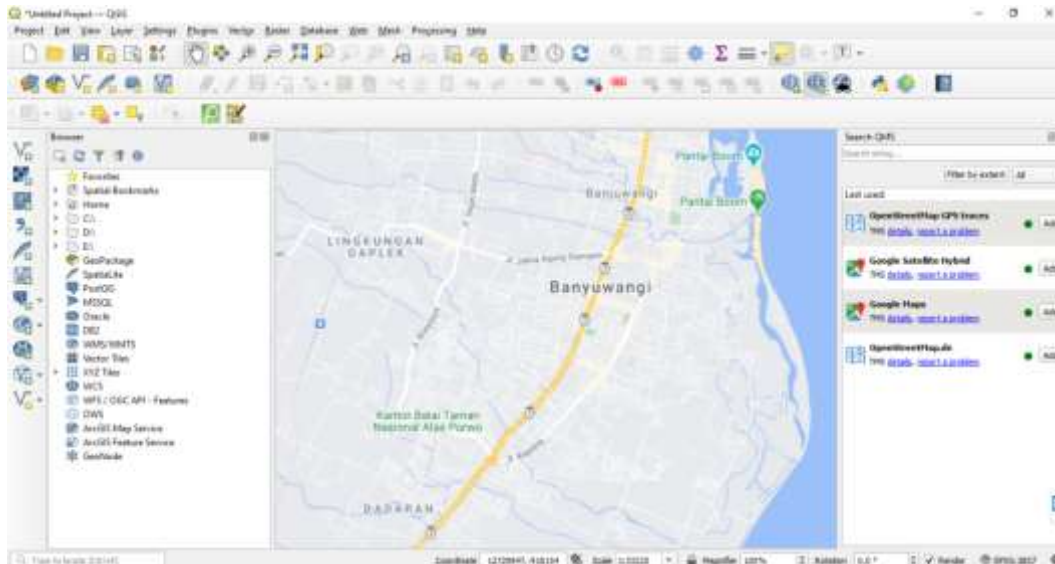
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\Facades\Event;
6  use Illuminate\Auth\Events\Registered;
7  use Illuminate\Auth\Listeners\SendEmailVerificationNotification;
8  use Illuminate\Foundation\Support\Providers\EventServiceProvider as ServiceProvider;
9  use App\Listeners\RecordLoginInfo;
10 use Illuminate\Auth\Events\Login;
11
12 class EventServiceProvider extends ServiceProvider
13 {
14     /**
15      * The event listener mappings for the application.
16      *
17      * @var array
18      */
19     // protected $listen = [
20     //     Registered::class => [
21     //         SendEmailVerificationNotification::class,
22     //     ],
23     // ];
24
25     protected $listen = [
26         Login::class => [
27             RecordLoginInfo::class
28         ]
29     ];
30
31     /**
32      * Register any events for your application.
33      *
34      * @return void
35      */
36     public function boot()
37     {
38         parent::boot();
39
40         //
41     }
42 }

```

Gambar 4.60 Konfigurasi *EventServiceProvider*

4.2.9 ATM Terdekat

Menampilkan data ATM BCA dan BTN terdekat dibuat dengan menggunakan aplikasi QGIS untuk menentukan wilayah yang akan dicari titik koordinat untuk ATM seperti pada gambar 4.61.



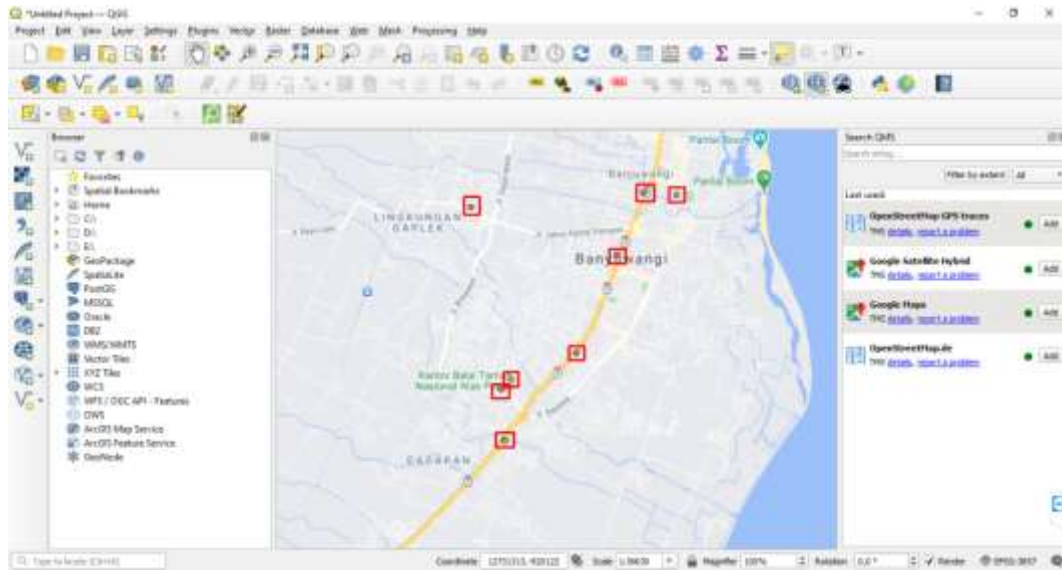
Gambar 4.61 Lokasi Pencarian ATM BCA dan BTN

Untuk mendapatkan titik lokasi ATM BCA dan BTN harus menggunakan cara manual seperti mencari lokasi ATM terdekat dengan lokasi penelitian menggunakan google maps lalu menyalin titik koordinat ATM dan menyimpannya pada file dengan format CSV seperti pada gambar 4.62.

	A	B	C	D		A	B	C	D	E
1	x,y	titik BCA			1	x,y	titik BCA			
2	114.35690349090474,-8.23259232366237				2	114.29324826816817,-8.308507365628211				
3	114.37541643895925,-8.212150578065076				3	114.29651991519846,-8.303838433971169				
4	114.35614325798672,-8.239371585162983				4	114.29237923250413,-8.309282349629397				
5					5	114.35578684218629,-8.233727498395336				
6					6	114.36875852684486,-8.218808112860817				
7					7	114.35610772869235,-8.239318474867282				
8					8	114.36418032869237,-8.229699850459765				
9					9	114.37169008451559,-8.211915932903455				
10					10	114.35233451519845,-8.213526821071191				
11					11					

Gambar 4.62 titik koordinat ATM BCA dan BTN

Agar titik ATM BCA dan BTN tampil pada lokasi pada gambar 4.61 maka data titik koordinat pada gambar 4.62 harus di import pada proyek yang telah dibuat pada aplikasi QGIS. Hasilnya dilihat pada gambar 4.63.



Gambar 4.63 titik lokasi ATM BCA dan BTN

Selanjutnya diberi marker penanda lokasi ATM BCA dan BTN agar dapat terlihat dengan jelas seperti pada gambar 4.7. setelah marker dibuat selanjutnya projek QGIS di export dan di masukkan pada aplikasi cek mutasi bank agar dapat tampil pada menu ATM terdekat dan dapat dilihat oleh nasabah pengguna aplikasi cek mutasi bank.

4.3 Pengujian

Metode yang digunakan dalam pengujian ini yaitu metode *Blackbox testing*. Tahap pengujian dilakukan untuk menjamin kualitas dan fungsional sehingga dapat berjalan dengan lancar sesuai dengan yang diharapkan. Dilakukan pengujian dengan memasukkan data yang sesuai dan tidak sesuai untuk melihat respon sistem. Pengujian meliputi proses *login* aplikasi dan cek mutasi bank.

4.3.1 Pengujian Login Aplikasi

- a. Pengguna (admin aplikasi atau nasabah) memasukkan *email* dan tidak memasukkan *password*. Hasil dapat dilihat pada gambar 4.64 dengan pesan “Please fill out this field”.



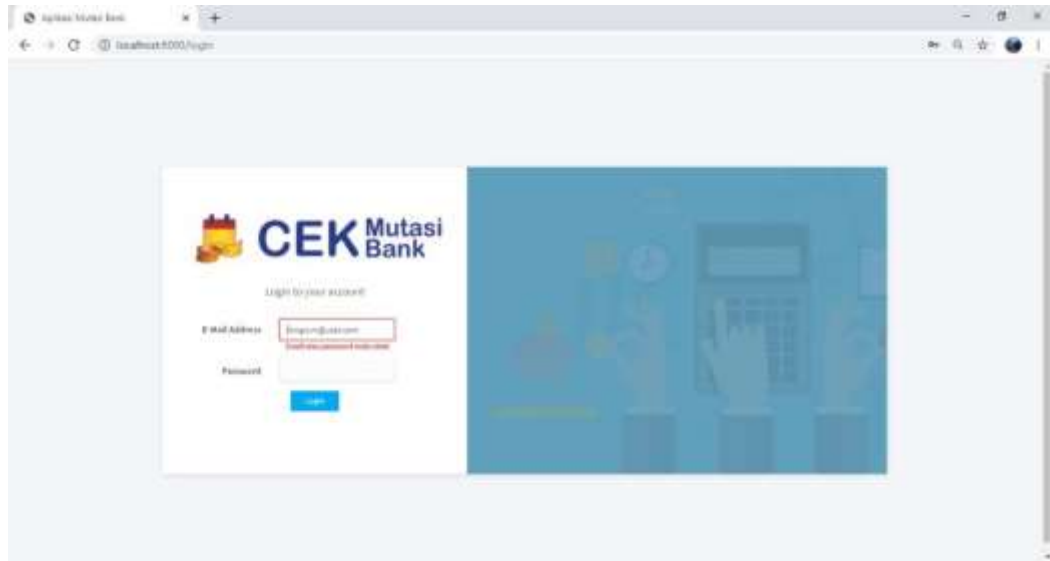
Gambar 4.64 Tampilan *Password Kosong*

- b. Pengguna (admin aplikasi atau nasabah) memasukkan *password* dan tidak memasukkan *email*. Hasil dapat dilihat pada gambar 4.65 dengan pesan “*Please fill out this field*”.



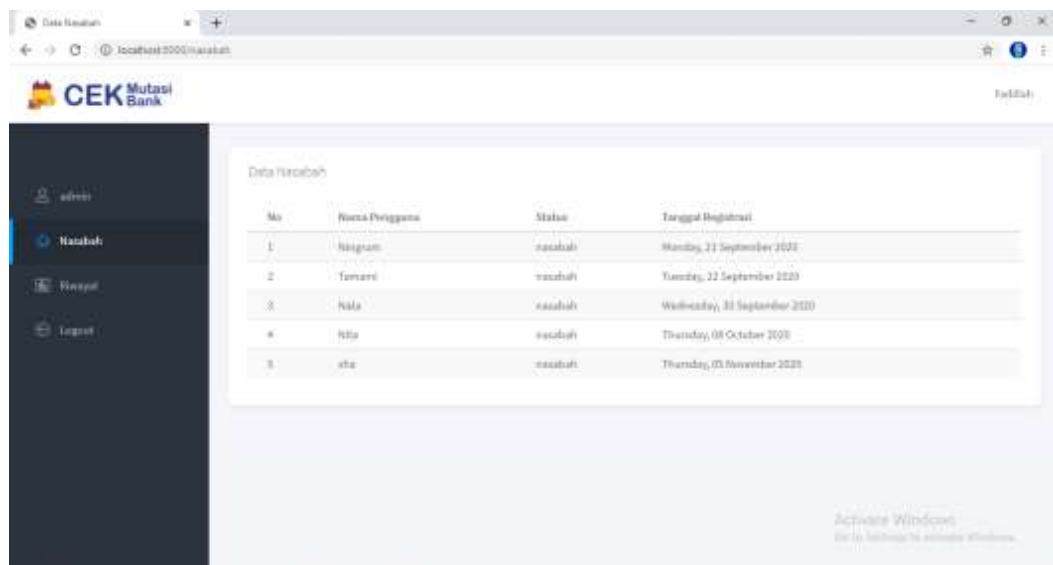
Gambar 4.65 Tampilan *Email Kosong*

- c. Pengguna (admin aplikasi atau nasabah) memasukkan *email* dan *password* dengan data yang tidak terdaftar pada basis data. Hasil dapat dilihat pada Gambar 4.66 dengan menampilkan pesan “*email dan password anda salah*”.

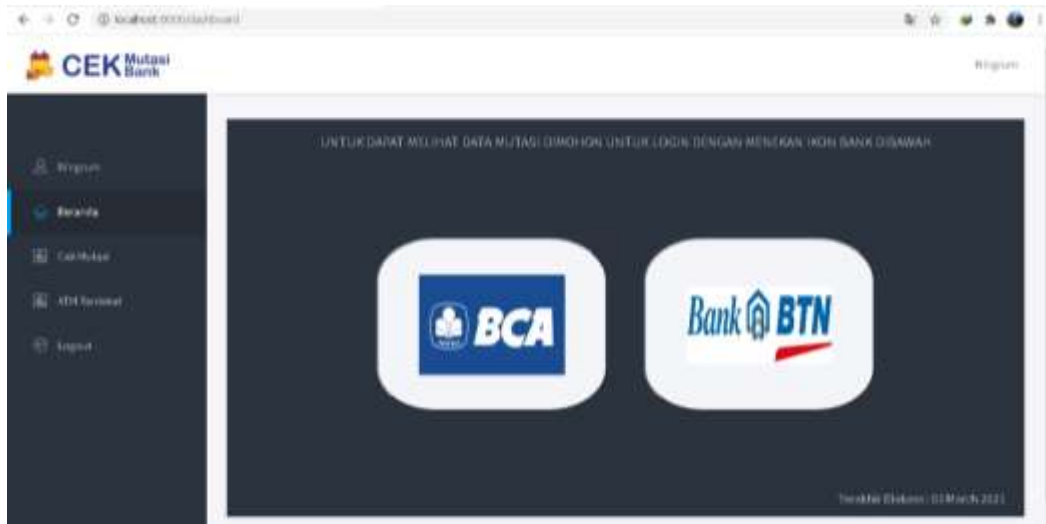


Gambar 4.66 Tampilan *Email* dan *Password* Salah

- d. Pengguna (admin aplikasi atau nasabah) memasukkan *email* dan *password* dengan data yang terdaftar pada basis data. Hasil dapat dilihat pada gambar 4.67 Halaman utama admin aplikasi yaitu halaman data nasabah dan gambar 4.68 halaman utama nasabah yaitu halaman beranda nasabah.



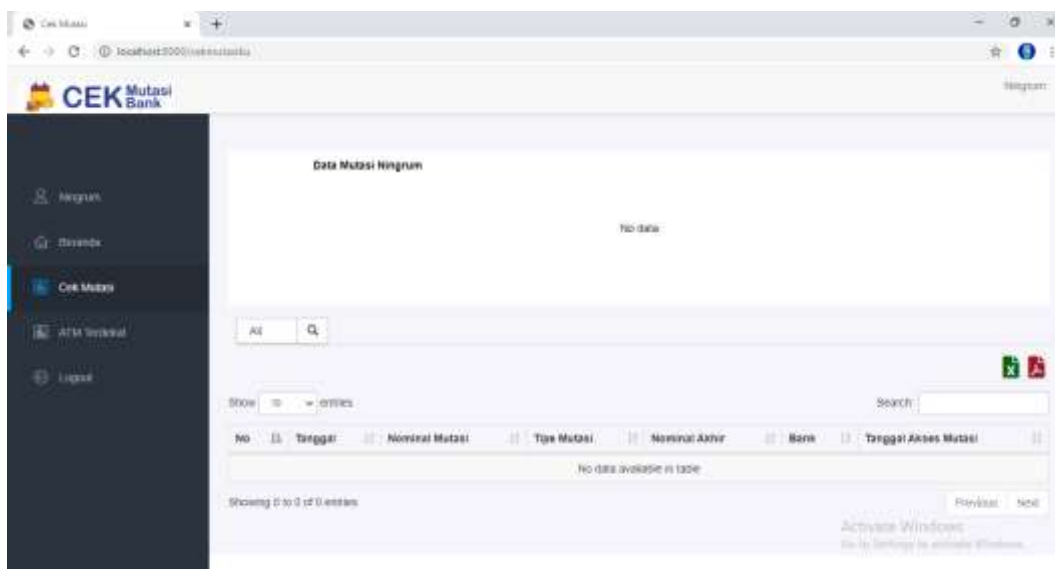
Gambar 4.67 Tampilan Halaman Utama Admin



Gambar 4.68 Tampilan Halaman Utama Nasabah

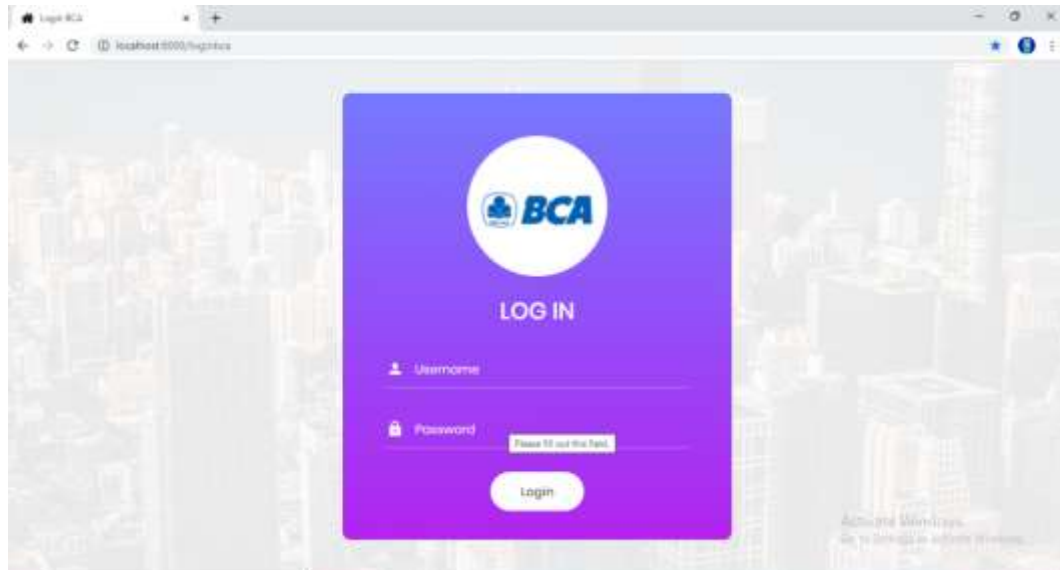
4.3.2 Pengujian Cek Mutasi Bank

- a. Nasabah belum memasukkan *username* dan *password internet banking* pada halaman beranda nasabah. maka pada saat memilih menu cek mutasi tidak akan ada data mutasi yang ditampilkan. Hasil dapat dilihat pada gambar 4.69 dengan pesan “No data available in table”.

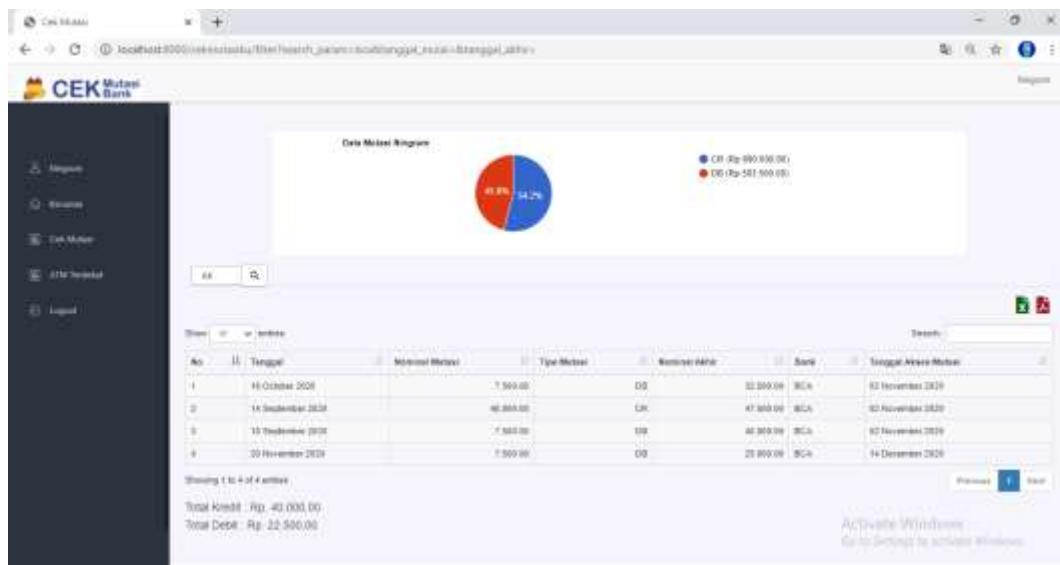


Gambar 4.69 Tampilan Tidak Ada Data Mutasi

- b. Nasabah *login internet banking* BCA dengan memasukkan *username* dan *password* seperti pada gambar 4.70 dengan tidak *login internet banking* BTN. maka pada saat memilih menu cek mutasi hanya akan menampilkan data mutasi BCA, hasil dapat dilihat pada gambar 4.71.

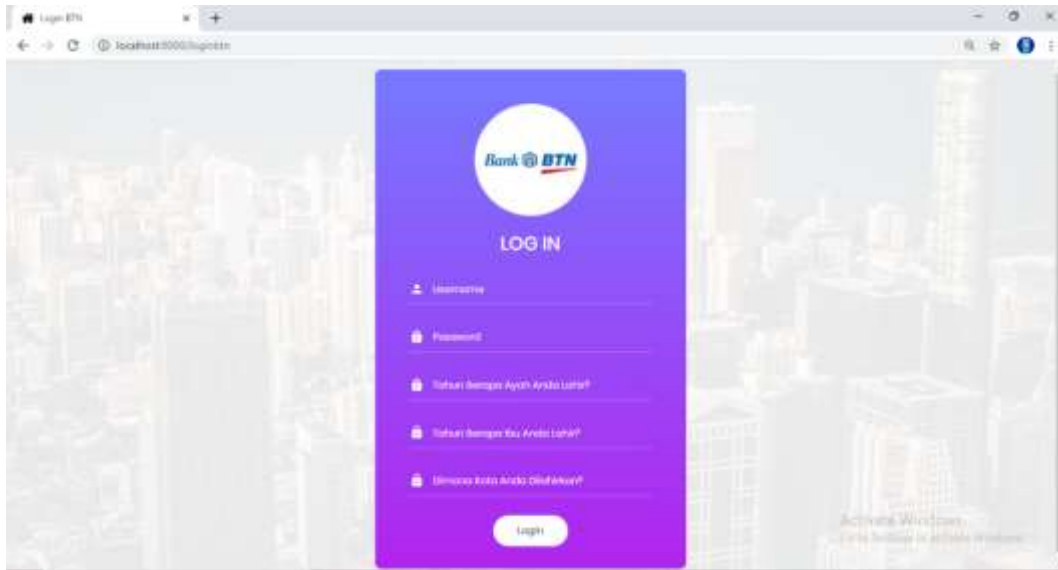


Gambar 4.70 Tampilan *Login Internet Banking BCA*

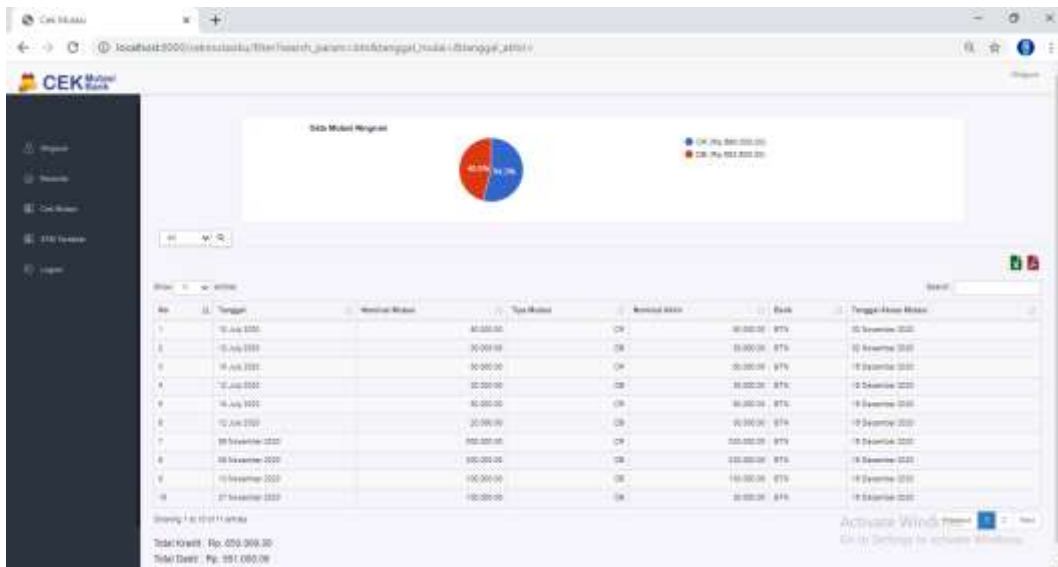


Gambar 4.71 Tampilan Data Mutasi BCA

- c. Nasabah *login internet banking* BTN dengan memasukkan *username*, *password*, tahun lahir ayah, tahun lahir ibu, dan tempat anda lahir seperti pada gambar 4.72 dengan tidak *login internet banking* BCA. maka pada saat memilih menu cek mutasi hanya akan menampilkan data mutasi BTN, hasil dapat dilihat pada gambar 4.73.

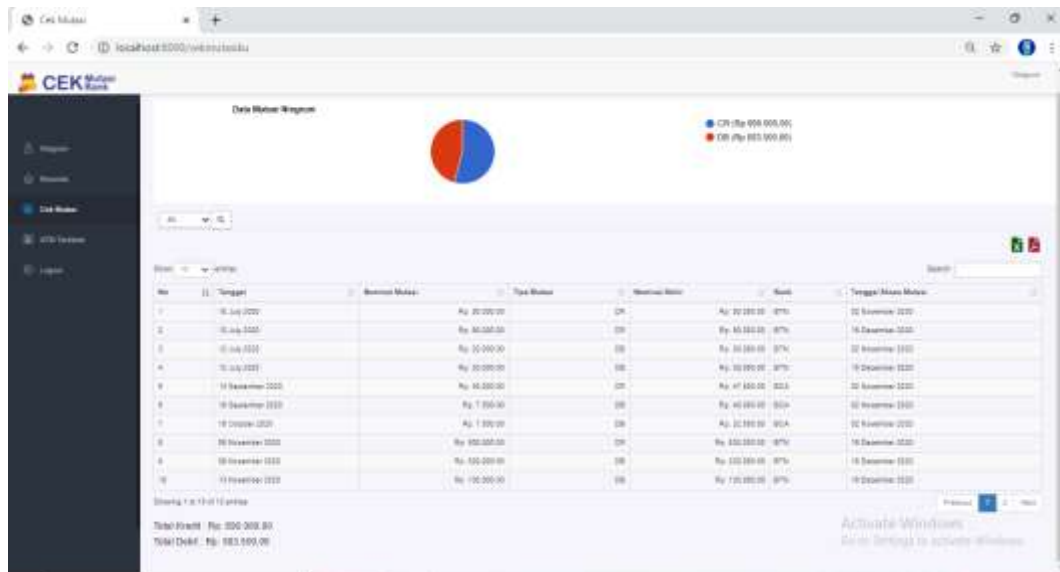


Gambar 4.72 Tampilan *Login Internet Banking* BTN



Gambar 4.73 Tampilan Data Mutasi BTN

- d. Nasabah telah *login internet banking* BCA dan BTN seperti pada gambar 4.70 dan 4.72, maka semua data mutasi dari kedua bank akan ditampilkan pada halaman cek mutasi bank, hasil dapat dilihat pada gambar 4.74.



Gambar 4.74 Tampilan Data Mutasi BCA dan BTN

4.3.3 Tabel Pengujian *Black Box*

Berdasarkan hasil pengujian yang dilakukan untuk melihat kesesuaian pada aplikasi, apakah aplikasi berjalan sesuai dengan yang diharapkan atau tidak, maka hasil pengujian menghasilkan data yang diperoleh ditunjukkan pada Tabel 4.1.

Tabel 4.1 Pengujian *Black Box*

No	Features	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Status	Bukti Pengujian
1.	Login	Melakukan login dengan memasukkan email dan tidak memasukkan password	Email: ningrum@user.com Password: -	Menampilkan pesan peringatan —Please fill out this field!	Sesuai harapan	Valid	Gambar 4.64
		Melakukan login dengan tidak memasukkan email dan memasukkan password	Email: - Password: 123456789	Menampilkan pesan peringatan —Please fill out this field!	Sesuai harapan	Valid	Gambar 4.65

		Melakukan <i>login</i> dengan memasukkan <i>email</i> dan <i>password</i> yang tidak terdaftar pada basis data	<i>Email:</i> ningrum@admin.com <i>Password:</i> 123456789	Menampilkan pesan peringatan “ <i>Email</i> dan <i>password</i> salah”	Sesuai harapan	Valid	Gambar 4.66
		Memasukkan email dan password dengan data yang terdaftar pada basis data	<i>Email:</i> ningrum@user.com <i>Password:</i> 123456789	Menampilkan halaman utama nasabah aplikasi yaitu halaman Beranda nasabah	Sesuai harapan	Valid	Gambar 4.68
	2. Cek Mutasi	Melakukan cek mutasi namun belum memasukkan <i>username</i> dan <i>password</i> internet banking pada halaman beranda nasabah	BCA <i>Username:</i> - <i>Password:</i> - BTN <i>Username:</i> - <i>Password:</i> - Tahun lahir ayah: - Tahun lahir ibu: - Tempat anda lahir: -	Menampilkan pesan “ <i>No data available in table</i> ”	Sesuai harapan	Valid	Gambar 4.69
		Melakukan cek mutasi dengan <i>login internet banking</i> BCA namun tidak <i>login internet banking</i> BTN	<i>Username:</i> NINGRUM0212 <i>Password:</i> *****	Menampilkan data mutasi dari bank BCA	Sesuai harapan	Valid	Gambar 4.71
		Melakukan cek mutasi dengan <i>login internet banking</i> BTN namun tidak <i>login internet banking</i> BCA	<i>Username:</i> Ningrum <i>Password:</i> ***** Tahun lahir ayah: 1958 Tahun lahir ibu:	Menampilkan data mutasi dari bank BTN	Sesuai harapan	Valid	Gambar 4.73

			1974 Tempat anda lahir: Batumarta				
		Melakukan cek mutasi dengan <i>login internet</i> <i>banking</i> BCA dan BTN	<i>Username:</i> NINGRUM0212 <i>Password:</i> ***** <i>Username:</i> Ningrum <i>Password:</i> ***** Tahun lahir ayah: 1958 Tahun lahir ibu: 1974 Tempat anda lahir: Batumarta	Menampilkan data mutasi dari bank BCA dan BTN	Sesuai harapan	Valid	Gambar 4.74

--Halaman ini sengaja dikosongkan--

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya mengenai Aplikasi cek mutasi bank Berbasis *Web* dapat disimpulkan bahwa:

1. Aplikasi Cek Mutasi Bank Berbasis *Web* dapat memberikan informasi mutasi bank BCA dan BTN berupa data debit dan kredit kepada nasabah dengan menu cek mutasi.
2. Nasabah dapat melihat data mutasi dari kedua bank maupun salah satu bank pada menu cek mutasi dengan *fitur filter*.
3. Nasabah dapat mengunduh data mutasi bank dari bank BCA dan BTN dalam bentuk PDF dan excel dengan fitur unduh PDF dan Excel yang terdapat pada menu cek mutasi.

5.2 Saran

Aplikasi cek mutasi bank berbasis *web* ini dapat dikembangkan dengan menambahkan bank lainnya selain BCA dan BTN yang sejenis. Selain menggunakan *library* selenium dan beautifulsoup dapat juga melakukan percobaan dengan menggunakan *library* scrapy. Untuk data ATM terdekat sebaiknya dibuat secara *realtime* agar dapat mengikuti pembaruan lokasi ATM. Selain itu untuk lebih memudahkan nasabah mengetahui informasi mutasi yang terbaru ditambahkan fitur notifikasi. Agar aplikasi ini lebih mudah digunakan untuk pengguna *smartphone* sebaiknya aplikasi ini juga diintegrasikan dengan android.

--Halaman ini sengaja dikosongkan--

DAFTAR PUSTAKA

- Agustino Halim, d. (2017). Perancangan Aplikasi Web Crawler untuk Menghasilkan Dokumen Teks pada Domain Tertentu. *Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol. 5, No. 2, (2017).*
- Ahmat Josi, L. A. (2014). Penerapan Teknik Web Scraping Pada Mesin Pencari Artikel Ilmiah. *Jurnal Sistem Informatika (JSI), 5(2), 159- 164. .*
- Hartawan, D. (2017). Pengaruh Internet Banking Terhadap Kepuasan Nasabah (Studi Kasus pada Nasabah Bank Rakyat Indonesia Syariah Cabang Palembang). *I-Economic Vol.3. No 2.*
- Mitchell, R. (2015). Web Scraping With Python Collecting Data From the Modern Web. *United States of America: O'Reilly Media inc.*
- Nafisan, Y. (2018). Pemanfaatan Aplikasi Banknesia pada Laporan Mutasi Keuangan Layanan Internet Banking. *IKRAITH-INFORMATIKA, VOL. 2, NO. 1, MARET 2018.*
- Ram Sharan Chaulagain, d. (2017). Cloud Based Web Scraping for Big Data Applications. *IEEE International Conference on Smart Cloud.*
- Supriyatna, A. (2018). Metode Extreme Programming pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja . *JURNAL TEKNIK INFORMATIKA VOL 11 NO. 1.*
- Yan Evi, d. (2018). Pengaruh Internet Banking Terhadap Minat Nasabah Dalam Bertransaksi Dengan Technology Acceptance Model. *JURNAL INFORMATIKA, Vol.5 No.1.*
- Yudhanto Yudho, H. A. (2018). *Panduan Mudah Belajar Framework Laravel.* Jakarta: PT Alex Media Komputindo.

--Halaman ini sengaja dikosongkan--

BIODATA DIRI

Nama : Ningrum Fadillah
Alamat : Dusun Wonorejo RT.01/RW.02
Desa Marga Bakti
Kecamatan Sinar Peninjauan
Kabupaten Ogan Komering Ulu
Tempat, Tanggal Lahir : Batumarta XI, 02 Desember 1998
Agama : Islam
Kewarganegaraan : WNI
Jenis Kelamin : Wanita
Status : Mahasiswa
Tinggi Badan : 153 cm
Berat Badan : 50 kg



RIWAYAT PENDIDIKAN

Institusi	Jurusan	Tahun
RA Babbussalam	-	2004
SDN 164 OKU	-	2010
SMP N 36 OKU	-	2013
SMA N 8 OKU	IPA	2016
Politeknik Negeri Banyuwangi	D-I Teknik Jaringan	2017
Politeknik Negeri Banyuwangi	D-III Teknik Informatika	2021