



**PENERAPAN METODE *SINGLE MOVING AVERAGE* DAN *ECONOMIC ORDER QUANTITY* (EOQ) PADA SISTEM PERSEDIAAN BAHAN BAKU DI  
ROOTS CAFE**

**SKRIPSI**

diajukan guna memenuhi salah satu syarat untuk  
menyelesaikan tugas akhir

Oleh :

**Bagus Windi Prakoso**

**NIM. 122410101072**

**PROGAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2017**



**PENERAPAN METODE *SINGLE MOVING AVERAGE* DAN *ECONOMIC ORDER QUANTITY* (EOQ) PADA SISTEM PERSEDIAAN BAHAN BAKU DI  
ROOTS CAFE**

**SKRIPSI**

diajukan guna memenuhi salah satu syarat untuk  
menyelesaikan tugas akhir

Oleh :

**Bagus Windi Prakoso**

**NIM. 122410101072**

**PROGAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2017**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Anik Andayani dan Ayahanda Mohammad Gimam yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
3. Kakak dan adikku tercinta yang selalu memberi dukungan;
4. Seluruh keluarga besar Mapala Balwana dan MAPALA se-Indonesia;
5. Seluruh keluarga besar KSR PMI unit UNEJ;
6. dulur Info Warga Jember(IWJ) semua korwil terutama korwil VI(KAPAS);
7. Sahabat kontrakan puri bunga nirwana yaitu Gertak, Puput, Seftya, Dwi, Zalfa, Lintang, Maya, Gausilia dan Gulay yang selalu meberikan dukungan dan motivasi;
8. Sahabatku Agil, Barok, Dion, A'la dan Rahmat yang selalu membantu proses kuliahku;
9. Keluarga besar Roots Cafe yang sudah banyak membantu dalam proses penelitian;
10. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
11. FORMATION yang sudah selalu membuat kita kompak;
12. Almamater program studi sistem informasi universitas jember.

**MOTTO**

“Karena inovasi tanpa adanya kolaborasi hanyalah halusinasi yang tidak akan pernah terwujud”



## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Bagus Windi Prakoso

NIM : 122410101072

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan *Single Moving Average* dan *Economic Order Quantity*(EOQ) pada Sistem Persediaan Bahan Baku di Roots Cafe”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 19 Mei 2017

Yang menyatakan,

Bagus Windi Prakoso

NIM 122410101072

**SKRIPSI**

**PENERAPAN METODE *SINGLE MOVING AVERAGE* DAN *ECONOMIC ORDER QUANTITY* (EOQ) PADA SISTEM PERSEDIAAN BAHAN BAKU  
DI ROOTS CAFE**

Oleh  
**Bagus Windi Prakoso**  
**NIM 122410101072**

Pembimbing :

Dosen Pembimbing Utama : Dr. Saiful Bukhori., S.T., M.Kom.

Dosen Pembimbing Pendamping : Windi Eka Yulia Retnani.,S.Kom.,M.T.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Penerapan *Single Moving Average* dan *Economic Order Quantity*(EOQ) pada Sistem Persediaan Bahan Baku di Roots Cafe” telah diuji dan disahkan pada:

hari, tanggal :Jumat, 19 Mei 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori., S.T., M.Kom

NIP. 196811131994121001

Windi Eka Yulia Retnani.,S.Kom.,M.T.

NIP. 19803052010122002



**PENGESAHAN PENGUJI**

Skripsi berjudul “Penerapan *Single Moving Average* dan *Economic Order Quantity*(EOQ) pada Sistem Persediaan Bahan Baku di Roots Cafe” telah diuji dan disahkan pada:

hari, tanggal :Jumat, 19 Mei 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Nelly Oktavia A, S.Si., MT.

NIP. 198410242009122008

Oktalia Juwita, S.Kom., M.MT.

NIP. 198110202014042001

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP 196704201992011001



## RINGKASAN

**Penerapan *Single Moving Average* dan *Economic Order Quantity*(EOQ) pada Sistem Persediaan Bahan Baku di Roots Cafe;** Bagus Windi Prakoso, 122410101072; 2017: 234 halaman; Progam Studi Sistem Informasi Universitas Jember.

Roots Cafe merupakan usaha perorangan yang bergerak dibidang restoran atau cafe. Roots cafe menjual banyak menu makanan dan minuman yang menggunakan banyak bahan baku. Sehingga pemilik kesulitan dalam memanajemen kebutuhan bahan baku yang dapat mengurangi jumlah keuntungan pada Roots Cafe. Pemilik emprediksi secara manual dalam menentukan jumlah kebutuhan bahan baku yang harus dibelanjakan pada periode selanjutnya sehingga tingkat keakuratan yang dihasilkan kuarng akurat, hal tersebut membuat banyak bahan baku yangterbuang karena melampaui batas kadaluarsa dan mungkin kekurangan bahan baku sebelum jadwal belanja yang membuat menu berbahan baku yang habis menjadi kosong. Untuk mengatasi permasalahan tersebut dapat memanfaatkan metode peramalan untuk menentukan prediksi penjualan menu makanan dan minuman dan metode *economic order quantity* untuk menentukan kebutuhan bahan baku secara ekonomis dari hasil prediksi penjualan pada periode selajutnya.

Pada penelitian ini dibangun sistem yang menerapkan metode *single moving average* sebagai metode peramalan dan *economic order quantity* untuk menentukan kebutuhan bahan baku pada periode berikunya, sehingga dapat membantu dalam proses penentuan jumlah kebutuhanbahan baku secara efisien. Dalam proses peramalan dengan menggunakan metode *single moving average* digunakan data mulai dari penjualan bulan Mei 2015 sampai bulan Juli 2016 dengan rata-rata nilai *mean absolute precantage error* dibawah 10% yang artinya prediksi akurat. Hasil prediksi penjualan tersebut digunakan dalam menghitung kebutuhan bahan baku dengan

menerapkan metode *economic order quantity* dari hasil perhitungan metode EOQ akan dikurangi dengan jumlah persediaan bahan baku, apakah bahan baku tersebut layak untuk dibelanjakan atau tidak. Sistem yang dibangun juga dapat memajemen data menu makanan dan minuman, data bahan baku, data keuangan, data pegawai dan memiliki hak akses, sehingga data yang terdapat didalam sistem terjamin keamanannya.



## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Penerapan *Single Moving Average* dan *Economic Order Quantity*(EOQ) pada Sistem Persediaan Bahan Baku di Roots Cafe. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori.,S.T., M.Kom selaku Dosen Pembimbing Utama dan Windi Eka Yulia Retnani.,S.Kom.,M.T selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Windi Eka Yulia Retnani.,S.Kom.,M.T sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di program studi sistem informasi;
5. Ibunda Anik Andayani dan Ayahanda Mohammad Giman yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
6. Mas Wahyu, Wisnu(nyong) , Galuh dan Putra yang selalu memberi dukungan;
7. Seluruh keluarga besar tersayang;
8. Mas Ichank pemilik Roots Cafe sebagai lokasi penelitian yang telah meluangkan waktu untuk membantu dalam melancarkan skripsi ini;
9. Keluarga besar FORMATION angkatan 2012 yang telah menjadi keluarga selama menempuh pendidikan S1;
10. Seluruh keluarga besar Mapala Balwana dan MAPALA se-Indonesia

11. Seluruh keluarga besar KSR PMI unit UNEJ;
12. dulur Info Warga Jember(IWJ) semua korwil terutama korwil VI(KAPAS);
13. Sahabat kontrakan puri bunga nirwana yaitu Gertak, Puput, Seftya, Dwi, Zalfa, Lintang, Maya, Gausilia dan Gulay yang selalu meberikan dukungan dan motivasi;
14. Sahabatku Agil, Barok, Dion dan Rahmat yang selalu membantu proses kuliahku;
15. Keluarga besar Roots Cafe yang sudah banyak membantu dalam proses penelitian;
16. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
17. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapakan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember,19 Mei 2017

Penulis

**DAFTAR ISI**

|  | <b>Halaman</b> |
|--|----------------|
| PERSEMBAHAN.....                                 | iii            |
| MOTTO .....                                      | iv             |
| PERNYATAAN.....                                  | v              |
| SKRIPSI.....                                     | vi             |
| PENGESAHAN PEMBIMBING.....                       | vii            |
| PENGESAHAN PENGUJI.....                          | viii           |
| RINGKASAN .....                                  | ix             |
| PRAKATA.....                                     | xi             |
| DAFTAR ISI.....                                  | xiii           |
| DAFTAR GAMBAR .....                              | xviii          |
| DAFTAR TABEL.....                                | xxiii          |
| BAB 1. PENDAHULUAN .....                         | 25             |
| 1.1 Latar Belakang .....                         | 25             |
| 1.2 Rumusan Masalah .....                        | 26             |
| 1.3 Tujuan dan Manfaat.....                      | 27             |
| 1.3.1 Tujuan dari penelitian ini adalah : .....  | 27             |
| 1.3.2 Manfaat dari Penelitian ini adalaah :..... | 27             |
| 1.4 Batasan Masalah.....                         | 27             |
| 1.5 Sistematika Penulisan.....                   | 28             |
| BAB 2. TINJUAN PUSTAKA .....                     | 29             |

|  |           |
|--|-----------|
| 2.1 Pola Data .....                            | 29        |
| 2.2 Metode Single Moving Average .....         | 31        |
| 2.3 Ukuran Ketepatan Nilai Peramalan .....     | 32        |
| 2.4 Metode Economic Order Quantity (EOQ) ..... | 33        |
| <b>BAB 3. METODOLOGI PENELITIAN.....</b>       | <b>36</b> |
| 3.1 Tahap Penelitian .....                     | 36        |
| 3.2 Metode Pengumpulan Data .....              | 37        |
| 3.3 Tahapan Analisis .....                     | 37        |
| 3.4 Tahapan Desain .....                       | 40        |
| 3.5 Tahapan Implementasi .....                 | 41        |
| 3.6 Tahapan Testing .....                      | 41        |
| 3.7 Tahapan Supporting atau maintance .....    | 42        |
| <b>BAB 4. PENGEMBANGAN SISTEM.....</b>         | <b>43</b> |
| 4.1 Deskripsi Umum Sistem.....                 | 43        |
| 4.1.1 <i>Statement of Purpose (SOP)</i> .....  | 43        |
| 4.1.2 Fungsi Sistem.....                       | 44        |
| 4.2 Analisis Kebutuhan Sistem .....            | 45        |
| 4.2.1 Kebutuhan Fungsional .....               | 46        |
| 4.2.2 Kebutuhan Non Fungsional.....            | 46        |
| 4.3 Desain Sistem .....                        | 47        |
| 4.3.1 <i>Business Process</i> .....            | 47        |
| 4.3.2 <i>Use Case Diagram</i> .....            | 48        |
| 1) <i>Use Case Login</i> .....                 | 54        |



|   |     |
|---|-----|
| 2) Menambahkan akun User .....                | 58  |
| 3) Merubah akun user .....                    | 62  |
| 4) Merubah status User .....                  | 66  |
| 5) Merubah Biodata .....                      | 69  |
| 6) Menambahkan Biodata .....                  | 72  |
| 7) Menampilkan peramalan .....                | 75  |
| 8) Menampilkan Detail Peramalan .....         | 77  |
| 9) MenampilkanKebutuhan Ekonomis .....        | 80  |
| 10) Menampilkan Daftar Belanja.....           | 83  |
| 11) Menampilkan Keuangan .....                | 86  |
| 12) Menambahkan data menu .....               | 89  |
| 13) Merubah data Menu .....                   | 95  |
| 14) Menghapus data Menu.....                  | 101 |
| 15) Menampilkan Menu .....                    | 106 |
| 16) Menampilkan total bayar setiap meja ..... | 108 |
| 17) Menampilkan Pesanan .....                 | 111 |
| 18) Menambahkan dataTransaksi .....           | 113 |
| 19) Merubah Status Pesanan .....              | 117 |
| 20) Menambahkan Bahan Baku .....              | 119 |
| 21) MenampilkanBahan Baku.....                | 123 |
| 22) Merubah Bahan Baku .....                  | 125 |
| 23) Menghapus Bahan Baku .....                | 129 |
| 24) Menambahkan Penggunaan Bahan Baku .....   | 133 |



|  |     |
|--|-----|
| 25) Menghapus penggunaan bahan baku .....  | 138 |
| 26) Menambahkan persediaan Bahan Baku .....  | 143 |
| 27) <i>Logout</i> .....  | 147 |
| 4.3.3 <i>Class Diagram</i> .....   | 149 |
| 4.3.4 <i>Entity Relationship Diagram (ERD)</i> .....   | 151 |
| 4.4 Implementasi .....   | 151 |
| 4.5 Pengujian Sistem .....   | 158 |
| 4.5.1 Metode <i>White Box</i> .....  | 159 |
| 4.5.2 Metode <i>Black Box</i> .....  | 172 |
| <b>BAB 5. HASIL DAN PEMBAHASAN</b> .....   | 174 |
| 5.1 Hasil implementasi code program pada Sistem Manajemen Kebutuhan<br>Ekonomis roots Cafe ..... | 174 |
| 5.1.1 Fitur Login .....  | 174 |
| 5.1.2 Halaman Dashboard .....  | 175 |
| 5.1.3 Fitur manajemen pegawai .....  | 176 |
| 5.1.4 Fitur Manajemen Data Diri .....  | 178 |
| 5.1.5 Fitur Menampilkan Hasil Peramalan .....  | 179 |
| 5.1.6 Fitur Menampilkan Kebutuhan Ekonomis .....   | 180 |
| 5.1.7 Fitur Keuangan .....   | 181 |
| 5.1.8 Fitur Mengelola data menu .....  | 183 |
| 5.1.9 Fitur Total Bayar .....  | 184 |
| 5.1.10 Fitur Menampilkan Pesanan .....   | 186 |
| 5.1.11 Menambahkan Transaksi .....   | 187 |

|  |            |
|--|------------|
| 5.1.12 Fitur Mengelola Bahan Baku .....  | 188        |
| 5.1.13 Fitur Mengelola Penggunaan Bahan Baku .....   | 190        |
| 5.1.14 Fitur Logout .....  | 192        |
| 5.2 Hasil Penerapan Metode <i>Single Moving Average</i> (SMA) dan <i>Economic order Quantity</i> (EOQ) ..... | 193        |
| 5.3 Pengujian Sistem Manajemen Kebutuhan Ekonomis Roots Cafe.....  | 198        |
| <b>BAB 6. PENUTUP</b> .....  | <b>216</b> |
| 6.1 Kesimpulan.....  | 216        |
| 6.2 Saran.....   | 217        |
| <b>DAFTAR PUSTAKA</b> .....  | <b>218</b> |
| Lampiran .....   | 219        |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 Pola data Horisontal (sumber : Makridakis 1992 ) .....                          | 29 |
| Gambar 2.2 Pola data musiman (sumber : Makridakis 1992 ) .....                             | 30 |
| Gambar 2.3 Pola data siklus (sumber : Makridakis 1992 ) .....                              | 30 |
| Gambar 2.4 Pola data siklus (sumber : Makridakis 1992 ) .....                              | 31 |
| Gambar 3. 1 System Development Life Cycle water fall (Sumber : Bassil 2012)                | 36 |
| Gambar 3. 2 Alur proses penerapan metode <i>Single Moving Average</i> dan EOQ..            | 39 |
| Gambar 4. 1 <i>Business Process</i> Sistem pengelola kebutuhan persediaan bahan baku ..... | 48 |
| Gambar 4. 2 <i>Use Case</i> Sistem pengelola kebutuhan persediaan bahan baku .....         | 49 |
| Gambar 4. 3 <i>Activity Diagram Login</i> .....  | 57 |
| Gambar 4. 4 <i>Sequence Diagram Login (Admin)</i> .....                                    | 58 |
| Gambar 4. 5 <i>Activity Diagram</i> Menambahkan akun user .....                            | 61 |
| Gambar 4. 6 <i>Sequence</i> menambahkan akun user .....                                    | 62 |
| Gambar 4. 7 <i>Activity Diagram</i> merubah akun user .....                                | 65 |
| Gambar 4. 8 <i>Sequence Diagram</i> merubah akun user .....                                | 66 |
| Gambar 4. 9 <i>Activity Diagram</i> merubah status user .....                              | 68 |
| Gambar 4. 10 <i>Sequence Diagram</i> merubah status user .....                             | 69 |
| Gambar 4. 11 <i>Activity Diagram</i> merubah biodata .....                                 | 71 |
| Gambar 4. 12 <i>Sequence Diagram</i> Merubah biodata .....                                 | 72 |
| Gambar 4. 13 <i>Activity</i> menambahkan biodata .....                                     | 74 |
| Gambar 4. 14 <i>Sequence Diagram</i> menambahkan biodata .....                             | 75 |
| Gambar 4. 15 <i>Activity Diagram</i> menampilkan peramalan .....                           | 76 |
| Gambar 4. 16 <i>Sequence Diagram</i> menampilkan peramalan .....                           | 77 |
| Gambar 4. 17 <i>Activity Diagram</i> menampilkan detail peramalan .....                    | 79 |
| Gambar 4. 18 <i>Sequence Diagram</i> menampilkan detail peramalan .....                    | 80 |
| Gambar 4. 19 <i>Activity Diagram</i> menampilkan kebutuhan ekonomis .....                  | 82 |

|   |     |
|---|-----|
| Gambar 4. 20 Sequence Diagram menampilkan kebutuhan ekonomis .....      | 83  |
| Gambar 4. 21 Activity Diagram menampilkan daftar belanja.....           | 85  |
| Gambar 4. 22 Sequence Diagram menampilkan daftar belanja .....          | 86  |
| Gambar 4. 23 Activity Diagram menampilkan keuangan .....                | 88  |
| Gambar 4. 24 Sequence Diagram menampilkan keuangan .....                | 89  |
| Gambar 4. 25 Activity Diagram menambahkan menu makanan .....            | 92  |
| Gambar 4. 26 Activity Diagram menambahkan menu minuman .....            | 93  |
| Gambar 4. 27 Sequence Diagram menambahkan menu makanan .....            | 94  |
| Gambar 4. 28 Sequence Diagram menambahkan menu minuman.....             | 95  |
| Gambar 4. 29 Activity Diagram merubah data menu makanan .....           | 98  |
| Gambar 4. 30 Activity Diagram merubah data menu minuman .....           | 99  |
| Gambar 4. 31 Sequence Diagram merubah data menu makanan .....           | 100 |
| Gambar 4. 32 Sequence Diagram merubah data menu minuman .....           | 101 |
| Gambar 4. 33 Activity Diagram menghapus data menu makanan .....         | 103 |
| Gambar 4. 34 Activity Diagram menghapus data menu minuman .....         | 104 |
| Gambar 4. 35 Sequence Diagram menghapus data menu makanan.....          | 105 |
| Gambar 4. 36 Sequence Diagram menghapus data menu minuman .....         | 105 |
| Gambar 4. 37 Activity Diagram menampilkan menu makanan .....            | 107 |
| Gambar 4. 38 Activity Diagram menampilkan menu minuman .....            | 107 |
| Gambar 4. 39 Sequence Diagram menampilkan menu makanan .....            | 108 |
| Gambar 4. 40 Sequence Diagram menampilkan menu minuman.....             | 108 |
| Gambar 4. 41 Activity Diagram menampilkan total bayar setiap meja ..... | 110 |
| Gambar 4. 42 Sequence Diagram menampilkan total bayar setiap meja ..... | 111 |
| Gambar 4. 43 Activity Diagram menampilkan pesanan .....                 | 112 |
| Gambar 4. 44 Sequence Diagram menampilkan pesanan.....                  | 113 |
| Gambar 4. 45 Activity Diagram menambahkan data transaksi.....           | 115 |
| Gambar 4. 46 Sequence Diagram menambahkan transaksi.....                | 116 |
| Gambar 4. 47 Activity Diagram merubah status pesanan .....              | 118 |
| Gambar 4. 48 Sequence Diagram merubah status pesanan.....               | 119 |

|  |            |
|--|------------|
| <b>Gambar 4. 49 Activity Diagram menambahkan data bahan baku .....</b>       | <b>122</b> |
| <b>Gambar 4. 50 Sequence Diagram menambahkan data bahan baku.....</b>        | <b>123</b> |
| <b>Gambar 4. 51 Activity Diagram menampilkan bahan baku.....</b>             | <b>125</b> |
| <b>Gambar 4. 52 Sequence Diagram menampilkan bahan baku .....</b>            | <b>125</b> |
| <b>Gambar 4. 53 Activity Diagram merubah bahan baku.....</b>                 | <b>128</b> |
| <b>Gambar 4. 54 Sequence Diagram merubah bahan baku .....</b>                | <b>129</b> |
| <b>Gambar 4. 55 Activity Diagram menghapus bahan baku.....</b>               | <b>132</b> |
| <b>Gambar 4. 56 Sequence Diagram menghapus bahan baku .....</b>              | <b>133</b> |
| <b>Gambar 4. 57 Activity Diagram menambahkan penggunaan .....</b>            | <b>136</b> |
| <b>Gambar 4. 58 Activity Diagram menambahkan penggunaan .....</b>            | <b>136</b> |
| <b>Gambar 4. 59 Sequence Diagram menambahkan penggunaan.....</b>             | <b>137</b> |
| <b>Gambar 4. 60 Sequence Diagram menambahkan penggunaan.....</b>             | <b>138</b> |
| <b>Gambar 4. 61 Activity Diagram menghapus penggunaan .....</b>              | <b>141</b> |
| <b>Gambar 4. 62 Activity Diagram menghapus penggunaan .....</b>              | <b>141</b> |
| <b>Gambar 4. 63 Sequence Diagram menghapus penggunaan.....</b>               | <b>142</b> |
| <b>Gambar 4. 64 Sequence Diagram menghapus penggunaan.....</b>               | <b>143</b> |
| <b>Gambar 4. 65 Activity Diagram menambahkan persediaan bahan baku .....</b> | <b>146</b> |
| <b>Gambar 4. 66 Sequence Diagram menambahkan persediaan baha baku .....</b>  | <b>147</b> |
| <b>Gambar 4. 67 Activity Diagram Logout .....</b>                            | <b>149</b> |
| <b>Gambar 4. 68 Sequence Diagram Logout .....</b>                            | <b>149</b> |
| <b>Gambar 4. 69 Class Diagram Sistem manajemen kebutuhan .....</b>           | <b>150</b> |
| <b>Gambar 4. 70 Entity Relationship Diagram SiMaKERoots .....</b>            | <b>151</b> |
| <b>Gambar 4. 71 Function mulai() pada class hitung.....</b>                  | <b>161</b> |
| <b>Gambar 4. 72 Function forcest() pada class hitung .....</b>               | <b>162</b> |
| <b>Gambar 4. 73 Function mulai() pada class hasil .....</b>                  | <b>164</b> |
| <b>Gambar 4. 74 Function belanja pada class hasil.....</b>                   | <b>165</b> |
| <b>Gambar 4. 75 Data alir function mulai() pada class hitung.....</b>        | <b>166</b> |
| <b>Gambar 4. 76 Data alir function forcest() pada class hitung .....</b>     | <b>166</b> |
| <b>Gambar 4. 77 Data alir function mulai() pada class hasil.....</b>         | <b>167</b> |



|   |            |
|---|------------|
| <b>Gambar 4. 78 Data alir function belanja() pada class hasil .....</b>     | <b>167</b> |
| <b>Gambar 5. 1 Fitur Login .....</b>  | <b>175</b> |
| <b>Gambar 5. 2 Halaman dashboard Admin/Owner.....</b>                       | <b>176</b> |
| <b>Gambar 5. 3 Halaman dashboard waiters,chef,barkeeper dan kasir .....</b> | <b>176</b> |
| <b>Gambar 5. 4 Halaman data pegawai.....</b>                                | <b>177</b> |
| <b>Gambar 5. 5 Halaman tambah pegawai .....</b>                             | <b>177</b> |
| <b>Gambar 5. 6 Form edit pegawai .....</b>                                  | <b>178</b> |
| <b>Gambar 5. 7 Halaman Data Diri User .....</b>                             | <b>178</b> |
| <b>Gambar 5. 8 Halaman Peramalan.....</b>                                   | <b>179</b> |
| <b>Gambar 5. 9 Halaman Detail Peramalan.....</b>                            | <b>180</b> |
| <b>Gambar 5. 10 Halaman Economic Order Quantity.....</b>                    | <b>181</b> |
| <b>Gambar 5. 11 Tampilan Daftar Belanja .....</b>                           | <b>181</b> |
| <b>Gambar 5. 12 Halaman Keuangan.....</b>                                   | <b>182</b> |
| <b>Gambar 5. 13 Halaman Data Menu Makanan.....</b>                          | <b>183</b> |
| <b>Gambar 5. 14 Halaman Data Menu Minuman .....</b>                         | <b>183</b> |
| <b>Gambar 5. 15 Halaman Tambah Menu .....</b>                               | <b>184</b> |
| <b>Gambar 5. 16 Halaman Edit Menu .....</b>                                 | <b>184</b> |
| <b>Gambar 5. 17 Halaman Transaksi Bayar.....</b>                            | <b>185</b> |
| <b>Gambar 5. 18 Total Bayar.....</b>  | <b>185</b> |
| <b>Gambar 5. 19 Halaman Pesanan Aktor Waiters .....</b>                     | <b>186</b> |
| <b>Gambar 5. 20 Halaman Pesanan Aktor Chef.....</b>                         | <b>186</b> |
| <b>Gambar 5. 21 Halaman Pesanan Aktor Barkeeper .....</b>                   | <b>187</b> |
| <b>Gambar 5. 22 POP UP pilih nomer Meja .....</b>                           | <b>187</b> |
| <b>Gambar 5. 23 Form Tambah Transaksi .....</b>                             | <b>188</b> |
| <b>Gambar 5. 24 Halaman Bahan Baku .....</b>                                | <b>188</b> |
| <b>Gambar 5. 25 Halaman Tambah Bahan Baku.....</b>                          | <b>189</b> |
| <b>Gambar 5. 26 POP Up Edit Bahan Baku.....</b>                             | <b>189</b> |
| <b>Gambar 5. 27 POP Up Hapus Bahan Baku.....</b>                            | <b>190</b> |
| <b>Gambar 5. 28 POP Up Tambah Persediaan Bahan Baku.....</b>                | <b>190</b> |

|  |            |
|--|------------|
| <b>Gambar 5. 29 Halaman Olah Bahan Baku Makanan .....</b>                      | <b>191</b> |
| <b>Gambar 5. 30 Halaman Olah Bahan Baku Minuman .....</b>                      | <b>191</b> |
| <b>Gambar 5. 31 POP Up Tambah Penggunaan Bahan Baku .....</b>                  | <b>192</b> |
| <b>Gambar 5. 32 POP Up Hapus Penggunaan Bahan Baku.....</b>                    | <b>192</b> |
| <b>Gambar 5. 33 Tombol untuk Logout .....</b>                                  | <b>193</b> |
| <b>Gambar 5. 34 Memilih Menu Untuk prediksi .....</b>                          | <b>194</b> |
| <b>Gambar 5. 35 Kode Program Perhitungan metode Single Moving Average ....</b> | <b>195</b> |
| <b>Gambar 5. 36 Kode Program Penerapan EOQ.....</b>                            | <b>197</b> |
| <b>Gambar 5. 37 Grafik Penjualan menu Fried Rice .....</b>                     | <b>199</b> |
| <b>Gambar 5. 38 Grafik nilai MAPE menu Fried Rice .....</b>                    | <b>210</b> |



## DAFTAR TABEL

|             |  |     |
|-------------|--|-----|
| Tabel 4. 1  | Deskripsi aktor sistem pengelola kebutuhan persediaan bahan baku                 | 50  |
| Tabel 4. 2  | Deskripsi <i>use case</i> sistem pengelola kebutuhan persediaan bahan baku ..... | 51  |
| Tabel 4. 3  | <i>Scenario Login</i> .....  | 54  |
| Tabel 4. 4  | <i>Scenario</i> menambahkan data user .....                                      | 59  |
| Tabel 4. 5  | <i>Scenario</i> merubah data user .....  | 63  |
| Tabel 4. 6  | <i>Scenario</i> merubah status user .....  | 67  |
| Tabel 4. 7  | <i>Scenario</i> merubah biodata .....  | 69  |
| Tabel 4. 8  | <i>Scenario</i> menambahkan biodata .....  | 72  |
| Tabel 4. 9  | Menampilkan peramalan .....  | 75  |
| Tabel 4. 10 | <i>Scenario</i> menampilkan detail peramalan .....                               | 77  |
| Tabel 4. 11 | <i>Scenario</i> menampilkan kebutuhan ekonomis .....                             | 80  |
| Tabel 4. 12 | <i>Scenario</i> menampilkan daftar belanja .....                                 | 83  |
| Tabel 4. 13 | <i>Scenario</i> menampilkan keuangan .....                                       | 86  |
| Tabel 4. 14 | Tabel menambahkan data menu .....  | 89  |
| Tabel 4. 15 | <i>Scenario</i> merubah data menu .....  | 96  |
| Tabel 4. 16 | <i>Scenario</i> menghapus data menu .....  | 102 |
| Tabel 4. 17 | <i>Scenario</i> menampilkan menu .....   | 106 |
| Tabel 4. 18 | <i>Scenario</i> menampilkan total bayar setiap meja .....                        | 109 |
| Tabel 4. 19 | <i>Scenario</i> menampilkan pesanan .....  | 111 |
| Tabel 4. 20 | <i>Scenario</i> menambahkan data transaksi .....                                 | 113 |
| Tabel 4. 21 | <i>Scenario</i> merubah status pesanan .....                                     | 117 |
| Tabel 4. 22 | <i>Scenario</i> menambahkan bahan baku .....                                     | 119 |
| Tabel 4. 23 | <i>Scenario</i> menampilkan bahan baku .....                                     | 124 |
| Tabel 4. 24 | <i>Scenario</i> merubah bahan baku .....   | 126 |
| Tabel 4. 25 | <i>Scenario</i> menghapus bahan baku .....                                       | 130 |

|                    |   |            |
|--------------------|---|------------|
| <b>Tabel 4. 26</b> | <b>Scenario menambahkan bahan baku .....</b>            | <b>133</b> |
| <b>Tabel 4. 27</b> | <b>Scenario menghapus penggunaan bahan baku .....</b>   | <b>139</b> |
| <b>Tabel 4. 28</b> | <b>Scenario menambahkan penggunaan bahan baku .....</b> | <b>143</b> |
| <b>Tabel 4. 29</b> | <b>Scenario logout .....</b>                            | <b>147</b> |
| <b>Tabel 4. 30</b> | <b>Test case function mulai() class hitung.....</b>     | <b>170</b> |
| <b>Tabel 4. 31</b> | <b>Test case function forcest() class hitung.....</b>   | <b>171</b> |
| <b>Tabel 4. 32</b> | <b>Test case function mulai() class hasil.....</b>      | <b>171</b> |
| <b>Tabel 4. 33</b> | <b>Test case function belanja() class hasil .....</b>   | <b>172</b> |
| <b>Tabel 5. 1</b>  | <b>Data Penjualan Fried rice.....</b>                   | <b>198</b> |
| <b>Tabel 5. 2</b>  | <b>Prediksi Baris Pertama .....</b>                     | <b>200</b> |
| <b>Tabel 5. 3</b>  | <b>Prediksi Baris Kedua .....</b>                       | <b>200</b> |
| <b>Tabel 5. 4</b>  | <b>Prediksi Baris Ketiga .....</b>                      | <b>200</b> |
| <b>Tabel 5. 5</b>  | <b>Prediksi Baris Keempat .....</b>                     | <b>201</b> |
| <b>Tabel 5. 6</b>  | <b>Prediksi Baris Kelima .....</b>                      | <b>201</b> |
| <b>Tabel 5. 7</b>  | <b>Prediksi Baris Keenam .....</b>                      | <b>202</b> |
| <b>Tabel 5. 8</b>  | <b>Prediksi Baris Ketujuh .....</b>                     | <b>203</b> |
| <b>Tabel 5. 9</b>  | <b>Prediksi Baris Kedelapan .....</b>                   | <b>203</b> |
| <b>Tabel 5. 10</b> | <b>Prediksi Baris Kesembilan .....</b>                  | <b>204</b> |
| <b>Tabel 5. 11</b> | <b>Prediksi Baris Kesepuluh .....</b>                   | <b>204</b> |
| <b>Tabel 5. 12</b> | <b>Prediksi Baris Kesebelas.....</b>                    | <b>205</b> |
| <b>Tabel 5. 13</b> | <b>Prediksi Baris Kedua belas.....</b>                  | <b>206</b> |
| <b>Tabel 5. 14</b> | <b>Prediksi Baris Ketiga belas.....</b>                 | <b>207</b> |
| <b>Tabel 5. 15</b> | <b>Prediksi Baris Keempat belas .....</b>               | <b>207</b> |
| <b>Tabel 5. 16</b> | <b>Prediksi Baris Kelima belas .....</b>                | <b>208</b> |
| <b>Tabel 5. 17</b> | <b>Prediksi untuk Bulan Agustus 2016 .....</b>          | <b>208</b> |

## BAB 1. PENDAHULUAN

Bab ini merupakan awal untuk penulisan skripsi. Pada bab ini akan dibahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian.

### 1.1 Latar Belakang

Roots Cafe merupakan usaha bisnis perorangan yang bergerak dibidang restoran atau cafe dan resmi dibuka sejak tanggal 12 April 2015. Konsep yang diusung Roots cafe merupakan classic cafe yang menawarkan suasana tempo dulu, pada 1 maret 2016 dibukanya lantai 2 dengan konsep yang diusung yaitu bengkel motor tua yang sangat menarik para pengunjung cafe. Roots Cafe menjual 2 menu makanan berat, 13 menu makanan ringan dan 10 makanan penutup, sedangkan untuk minuman sendiri terdiri dari 52 macam jenis minuman. Dalam mengelolah menu makanan dan minuman Roots Cafe belum menemukan metode yang sesuai sehingga untuk persediaan bahan baku terkadang terlalu banyak dan bahkan sampai kehabisan karena tidak memprioritaskan menu makanan dan minuman yang sering terjual.

Masalah yang sering dihadapi oleh Roots Cafe yaitu masalah mengelolah persediaan bahan baku, setiap bulan roots cafe selalu membuang bahan baku dikarenakan masa simpan bahan baku tersebut. Hal tersebut membuat kerugian terhadap penghasilan perusahaan, pengelolaan persediaan bahan baku sangat diperlukan untuk mengurangi nilai kerugian terhadap cafe. Metode yang paling banyak digunakan dan terbukti memiliki nilai akurat yang baugs yaitu metode *Economic Order Quantity*, menurut (Fransisco, 2013) penerapan EOQ sangat efisien dala menentukan perencanaan kebutuhan bahan baku pada PT.Bumi Asri Pasaman Buntok, selain itu juga terdapat penelitian yang dilakukan oleh (Tunjung, 2010) yang menerapkan metode EOQ dalam pengelolaan persediaan bahan baku kedelai pada

perusahaan kecap yang mana hasil dari penerapan EOQ memiliki nilai persediaan pengamanan lebih kecil daripada sebelum menggunakan metode EOQ. Untuk itu penggunaan metode *Economic Order Quantity* (EOQ) perlu diterapkan dalam mengelolah persediaan bahan baku disebuah restoran ataupun cafe, untuk penerapan metode EOQ diperlukan nilai kebutuhan pada periode selanjutnya, untuk itu perlunya penerapan metode peramalan yang dapat menentukan nilai kebutuhan pada periode selanjutnya, pemilihan metode peramalan dilihat dari pola data yang digambarkan oleh datasebelumnya guna memperoleh nilai yang akurat.

Melihat sangat pentingnya dalam mengelolah persediaan bahan baku, Roots Cafe memerlukan sebuah sistem yang dapat menampilkan persediaan bahan baku yang perlu disediakan secara akurat berdasarkan permintaan konsumen pada periode sebelumnya. Sehingga mengurangi terjadinya kekurangan bahan baku maupun kelebihan bahan baku. Dalam melakukan peramalan (*forecast*) terhadap data pada periode sebelumnya, peneliti menggunakan metode *Single Moving Average* karena pola data penjualan yang terdapat pada roots cafe yaitu pola data horisontal yang mana data penjualan menunya setiap periode mengalami kenaikan dan penurunan tidak menentu. Peramalan ini bertujuan untuk menentukan prediksi penjualan menu makanan dan minuman yang nilainya akan digunakan dalam perhitungan metode *economic order quantity*(EOQ) sebagai permintaan dalam periode waktu untuk menentukan kebutuhan setiap bahan bakunya pada periode selanjutnya .

## 1.2 Rumusan Masalah

Bedasarkan latar belakang diatas, dapat dirumuskan permasalahan-permasalahan yang muncul yaitu :

1. Bagaimana menerapkan perhitungan menggunakan persamaan *Single Moving Average* dan *Ekonomic Order Quantity* pada sebuah sistem pengelola persediaan bahan baku di Roots Cafe ?

2. Bagaimana membangun sistem pengelolaan persediaan bahan baku di Roots Cafe menggunakan metode *Single Moving Average* dan *Economic Order Quantity* (EOQ)?

### 1.3 Tujuan dan Manfaat

#### 1.3.1 Tujuan dari penelitian ini adalah :

- a. Merancang sebuah sistem yang mengimplementasikan metode *Single Moving Average* dan *Economic Order Quantity* (EOQ) pada sebuah sistem pengelolaan persediaan bahan baku guna efisiensi persediaan bahan bakudi sebuah cafe berbasis web.
- b. Membangun Sistem informasi yang dapat menyimpan data penjualan menu makanan dan minuman, persediaan bahan baku yang kemudian akan disimpan dalam bentuk database berbasis web.

#### 1.3.2 Manfaat dari Penelitian ini adalaah :

- a. Membantu *owner* cafe dalam menyiapkan persedian bahan baku dari menu makanan dan minuman paling banyak terjual dalam periode sebelumnya.
- b. Untuk periode selanjutnya.Sistemyang dihasilkan dapat digunakan sebagai acuan untuk ilmu pengetahuan.

### 1.4 Batasan Masalah

Batasan Masalah dalam penelitian ini yaitu :

1. Objek yang diteliti adalah bahan baku, menu makanan dan minuman di Roots Cafe.
2. Data transaksi penjualan pada bulan april 2015- Juli 2016
3. User yang dapat mengakses yaitu *owner, chef, barkeeper, waiters*, kasir.
4. Sistem berbasis web dengan bahasa pemrograman PHP.
5. Database yang digunakan sebagai penyimpanan data adalah MY SQL.



### 1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah:

1. Pendahuluan

Bab ini terdiri dari latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini berisi tentang teori-teori, materi dan informasi yang digunakan dalam penelitian. Dimulai dari kajian pustaka mengenai sistem informasi manajemen menu makanan dan minuman berbasis web, café, dan Metode *single moving average* dan *economic order quantity*.

3. Metodologi Penelitian

Bab ini menguraikan tentang metode yang akan digunakan selama penelitian.

Meliputi pendekatan penelitian, studi literatur, sumber data penelitian, dan tahapan analisis hingga model perancangan sistem.

4. Analisis dan Perancangan Sistem

Bab ini menguraikan tentang analisis dan perancangan sistem informasi yang akan dibangun.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan.

6. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya

## BAB 2. TINJUAN PUSTAKA

Bab ini menjelaskan teori apa saja yang dipakai untuk mendukung proses mengerjakan penelitian ini. Tinjauan pustaka dan teori diambil dari jurnal, tesis, dan internet

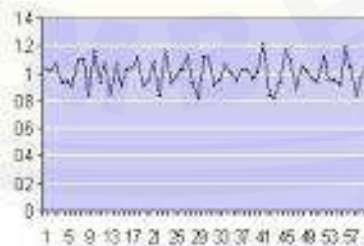
### 2.1 Pola Data

Metode peramalan merupakan suatu cara untuk menentukan nilai data pada periode berikutnya berdasarkan nilai data dari periode-periode sebelumnya, nilai data pada periode-periode sebelumnya akan membentuk sebuah pola untuk menentukan peramalan yang cocok untuk digunakan. Pola data dibedakan dalam empat jenis siklus (*cyclical*) dan trend yaitu sebagai berikut:

#### 1. Pola Data Horisontal/Stasioner

Data berpola stasioner adalah deret data yang berfluktuasi di sekitar nilai rata-rata yang konstan yaitu tidak berubah sepanjang waktu. Keadaan seperti ini dapat terjadi jika pola permintaan yang mempengaruhi data tersebut relatif stabil.

Menurut (Makridakis et al., 1992) metode peramalan yang dianggap tepat untuk data berpola stasioner seperti metode rata-rata bergerak (*moving average*) dan penghalusan eksponensial tunggal (*single exponential smoothing*). Pola data stasioner atau horisontal dapat dilihat pada gambar 2.1.

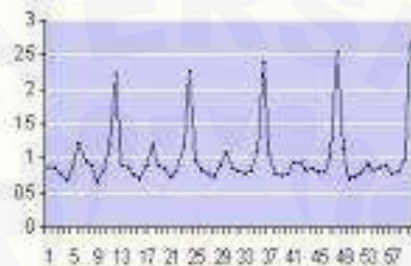


Gambar 2.1 Pola data Horisontal  
(sumber : Makridakis 1992 )



## 2. Pola Data Musiman

Data berpola musiman adalah deret data yang dipengaruhi oleh faktor musiman seperti kuartal tahun tertentu, bulan tertentu, atau hari tertentu. Pengaruh dari pola data ini dapat berupa iklim cuaca ataupun pengaruh masa-masa yang berulang tiap tahunnya seperti masa liburan dan hari raya. Pola data musiman dapat dilihat pada gambar 2.2.



Gambar 2.2 Pola data musiman  
(sumber : Makridakis 1992 )

## 3. Pola Data Siklus

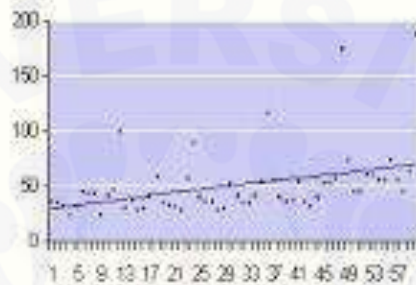
Pola data siklus merupakan deret data yang menunjukkan gerakan naik turun dalam jangka panjang dari suatu kurva trend. Terjadi bila datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Pola data siklis dapat dilihat pada gambar 2.3



Gambar 2.3 Pola data siklus  
(sumber : Makridakis 1992 )

#### 4. Pola data Trend

Pola data trend merupakan deret data yang menunjukkan kenaikan atau penurunan jangka panjang dalam data. Suatu data pengamatan yang mempunyai trend disebut data nonstasioner. Pola data trend dapat dilihat pada gambar 2.4



Gambar 2.4 Pola data siklus  
(sumber : Makridakis 1992 )

#### 2.2 Metode Single Moving Average

Metode peramalan *Single Moving Average* merupakan metode peramalan yang menggunakan rata-rata bergerak berdasarkan data sebelumnya, metode ini tidak cocok bila digunakan pada pola data trend atau pola data yang Bergeraknya cenderung keatas atau menurun. (Heizer dkk ,2005) mengatakan bahwa kelebihan dari metode *Single Moving Average* merupakan model peramalan objektif yang paling efektif dan efisien dari segi biaya. Metode *Single Moving Average* mempunyai karakteristik khusus yaitu:

1. Untuk menentukan ramalan pada periode yang akan datang memerlukan data historis selama jangka waktu tertentu. Misalnya, dengan 4 (empat) bulan moving average, maka ramalan bulan ke 5 (lima) baru dibuat setelah bulan ke 4 (empat) selesai/berakhir. Jika bulan moving averages ramalan bulan ke 7 (tujuh) baru bisa dibuat setelah bulan ke 6 (enam) berakhir.

2. Semakin panjang jangka waktu moving average , efek pelicinan semakin terlihat dalam ramalan atau menghasilkan moving average.

Persamaan yang digunakan untuk menghitung nilai peramalan mdengan metode *Single Moving Average* dapat dilihat dalam persamaan (1).

$$S_{t+1} = \frac{x_t + x_{t-1} + x_{t-2} + \dots + x_{t-n+1}}{n} \dots \dots \dots \text{Persamaan (1)}$$

Keterangan :

$S_{t+1}$  = Nilai predikdisi pada periode ke- t+1

$x_t$  = Nilai real data pada periode ke t

$n$  = Jumlah batas dalam moving average

Dari Persamaan (1) akan didapat nilai peramalan untuk data pada periode selajutnya. Metode ini disebut rata - rata bergerak karena setiap kali data observasi baru tersedia, maka angka - angka baru dihitung dan digunakan sebagai ramalan (forecast) untuk periode yang akan datang.

### 2.3 Ukuran Ketepatan Nilai Peramalan

Dalam semua situasi peramalan mengandung derajat ketidakpastian. Kita mengenali fakta ini dengan memasukkan unsur kesalahan (error) dalam perumusan sebuah peramalan deret waktu. Sumber penyimpangan dalam peramalan bukan hanya disebabkan oleh unsur error , tetapi ketidakmampuan suatu model peramalan mengenali unsur yang lain dalam deret data juga mempengaruhi besarnya penyimpangan dalam peramalan. Jadi besarnya penyimpangan hasil peramalan bisa disebabkan oleh besarnya faktor yang tidak diduga (*outliers*) dimana tidak ada metode peramalan yang mampu menghasilkan peramalan yang akurat, atau bisa juga disebabkan metode peramalan yang digunakan tidak dapat memprediksi dengan tepat komponen trend, komponen musiman, komponen horisontal atau komponen siklus

yang mungkin terdapat dalam deret data, yang berarti metode yang digunakan tidak tepat. MAPE (*Mean Absolute Percentage Error*) merupakan ukuran ketetapan relatif yang digunakan untuk mengetahui persentase penyimpangan hasil peramalan, dengan persamaan(2):

$$MAPE = \sum \frac{\frac{x_t - S_t}{x_t}}{p} \cdot 100\% \dots \dots \dots \text{Persamaan (2)}$$

Dari persamaan (2) diketahui bahwa:

MAPE = *Mean Absolute Percentage Error* (%)

$x_t$  = Nilai aktual data pada periode ke-t

$S_t$  = Nilai prediksi pada periode ke-t

$p$  = Banyaknya periode yang dihitung

Suatu model mempunyai kinerja sangat bagus jika nilai MAPE berada di bawah 10%, dan mempunyai kinerja bagus jika nilai MAPE berada di antara 10% dan 20%.

#### 2.4 Metode Economic Order Quantity (EOQ)

Dalam menentukan kebijakan dalam menyediakan bahan baku yang tepat berdasarkan permintaan dapat diterapkan metode EOQ. Menurut Rangkuti (Freddy, Manajemen Persediaan, 2007), *Economic Order Quantity* (EOQ) adalah sejumlah pembelian bahan mentah pada setiap kali pesan dengan biaya yang paling rendah.

Tujuan dari sebagian besar metode persediaan adalah meminimumkan biaya total. Dengan asumsi-asumsi yang diberikan, biaya-biaya yang signifikan adalah biaya pemesanan (set up cost) dan biaya penyimpanan (holding cost/ carrying cost). Biaya-biaya lain seperti biaya satuan itu sendiri adalah konstan. Sehingga dengan meminimalkan jumlah pemesanan dan penyimpanan berarti dapat meminimalkan biaya total.

Keuntungan menggunakan teknik EOQ adalah pemesanan dilakukan lebih dari kebutuhan bersihnya, sehingga apabila terjadi perubahan kualitas produksi menjadi lebih besar, maka persediaan bahan baku tersedia. Kekurangan teknik ini memberikan biaya penyimpanan terlalu besar bila dibandingkan dengan teknik *Lot for Lot* atau metode *Material Requirement Planning* (MRP). Metode yang tidak hanya menitik beratkan pada berapa banyak suatu komponen perlu dipesan (atau diproduksi), tetapi juga memperhatikan kapan komponen yang bersangkutan dipesan atau diproduksi. Metode ini cocok digunakan untuk perusahaan manufaktur, khususnya mengenai penjadwalan alur barang ke dan melalui proses pembuatan barang jadi. Untuk menentukan nilai Ekonomis pemesanan dapat menggunakan persamaan (3)

$$EOQ = \sqrt{\frac{2AD}{h}} \dots\dots\dots \text{Persamaan (3)}$$

Dimana :

D = Permintaan per periode waktu

A = Biaya pemesanan per pesanan

h = Biaya penyimpanan per unit per periode

Adapun penelitian yang menerapkan metode *Economic Order Quantity* (EOQ) yaitu Analisis Efisiensi Pengelolaan Persediaan Bahan Baku Kedelai Pada Perusahaan Kecap Pt. Lombok Gandaria Food Industry Palur Karanganyar(Sugiarto, 1996), dalam penelitian tersebut persediaan pengamanan (safety stock) lebih besar sebelum menerapkan metode *Economic Order Quantity*, selain itu juga persediaan bahan baku juga lebih besar daripada sebelum menggunakan metode EOQ, hal tersebut dapat mengganggu keuntungan bagi perusahaan. Penelitian lain yang dilakukan oleh Eyverson Ruauw yang berjudul pengendalian persediaan bahan baku juga menunjukkan penggunaan metode EOQ sangat efisien dalam mengelola



persediaan bahan baku disebuah usaha Grenda Bakery Lianly, Manado(Ruauw, PENGENDALIAN PERSEDIAAN BAHAN BAKU, 2011).

Dari hasil penelitian yang dilakukan oleh Tunjung dan Eyverson Ruauw maka perlunya diterapkan metode EOQ terhadap persediaan bahan baku di Roots Cafe, karena permasalahan utama di Roots cafe yaitu persediaan bahan baku yang tidak terkelola dengan baik. Permasalahan tersebut dapat merugikan pemilik cafe karena menyediakan bahan baku yang terlalu banyak dan bahkan sampai kadaluwarsa.

Dalam menerapkan metode *Economic Order Quantity*, peneliti memerlukan nilai pemesanan barang paling ekonomis pada periode selanjutnya, maka dari itu peneliti melakukan peramalan menggunakan metode *Single Moving Average* (rata-rata bergerak tunggal), metode ini sangat cocok untuk pola data horisontal yaitu data yang berfluktuasi di sekitar tingkatan atau rata-rata yang konstan, data yang terdapat di Roots Cafe termasuk pola data horisontal, karena data penjualan setiap menunya tidak mengalami kenaikan ataupun penurunan sekuler jangka panjang. Menurut penelitian yang berjudul penerapan metode *winter's exponential smoothing* dan *Single Moving Average* dalam sistem informasi pengadaan obat rumah sakit (Tanuwijaya, 2010) menyimpulkan bahwa 61,9% obat di rumah sakit memiliki pola data bersifat stationer(horisontal) maka peramalan yang cocok menggunakan metode *Single Moving Average* dan 38,1% bersifat musiman, sehingga lebih cocok menggunakan metode *winter exponential smoothing*. Dari penelitian tersebut dapat disimpulkan bahwa metode *Single Moving Average* sangat cocok diterapkan dalam sistem persediaan bahan baku di Roots Cafe.

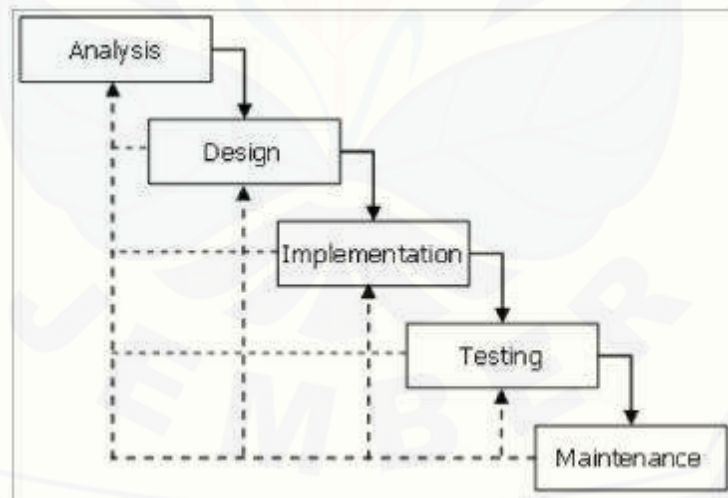


### BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode-metode yang digunakan selama penelitian, seperti pengumpulan data dan sumber data penelitian, serta tahapan analisis perancangan sistem dengan metode *single moving average* dan *economic order quantity*.

#### 3.1 Tahap Penelitian

Penelitian ini akan dilakukan dengan teknik pengembangan sistem yang digunakan yaitu model waterfall karena model ini memiliki alur yang urut mulai dari analisis hingga support, selain itu setiap proses memiliki spesifikasinya sendiri sehingga sebuah sistem dapat dikembangkan sesuai dengan apa yang dikehendaki. Alur SDLC waterfall dapat dilihat pada gambar 3.1.



Gambar 3. 1 System Development Life Cycle water fall  
(Sumber : Bassil 2012)

### 3.2 Metode Pengumpulan Data

Teknik pengumpulan data pada penelitian ini meliputi :

#### a. Studi Literatur

Studi literatur bertujuan untuk menyusun dasar teori yang digunakan dalam penelitian. Sumber yang dapat digunakan sebagai studi literatur studi literatur seperti buku, jurnal, internet dan dokumen-dokumen yang terkait dengan Peramalan, metode prsediaan bahan baku.

#### b. Wawancara

Salah satu cara untuk mengumpulkan data adalah melakukan wawancara dengan menanyakan langsung kepada narasumber. Dalam penelitian yang akan dilakukan ini penulis melakukan wawancara secara langsung pada pemilik Roots dan pegawai roots cafe.

#### c. Observasi

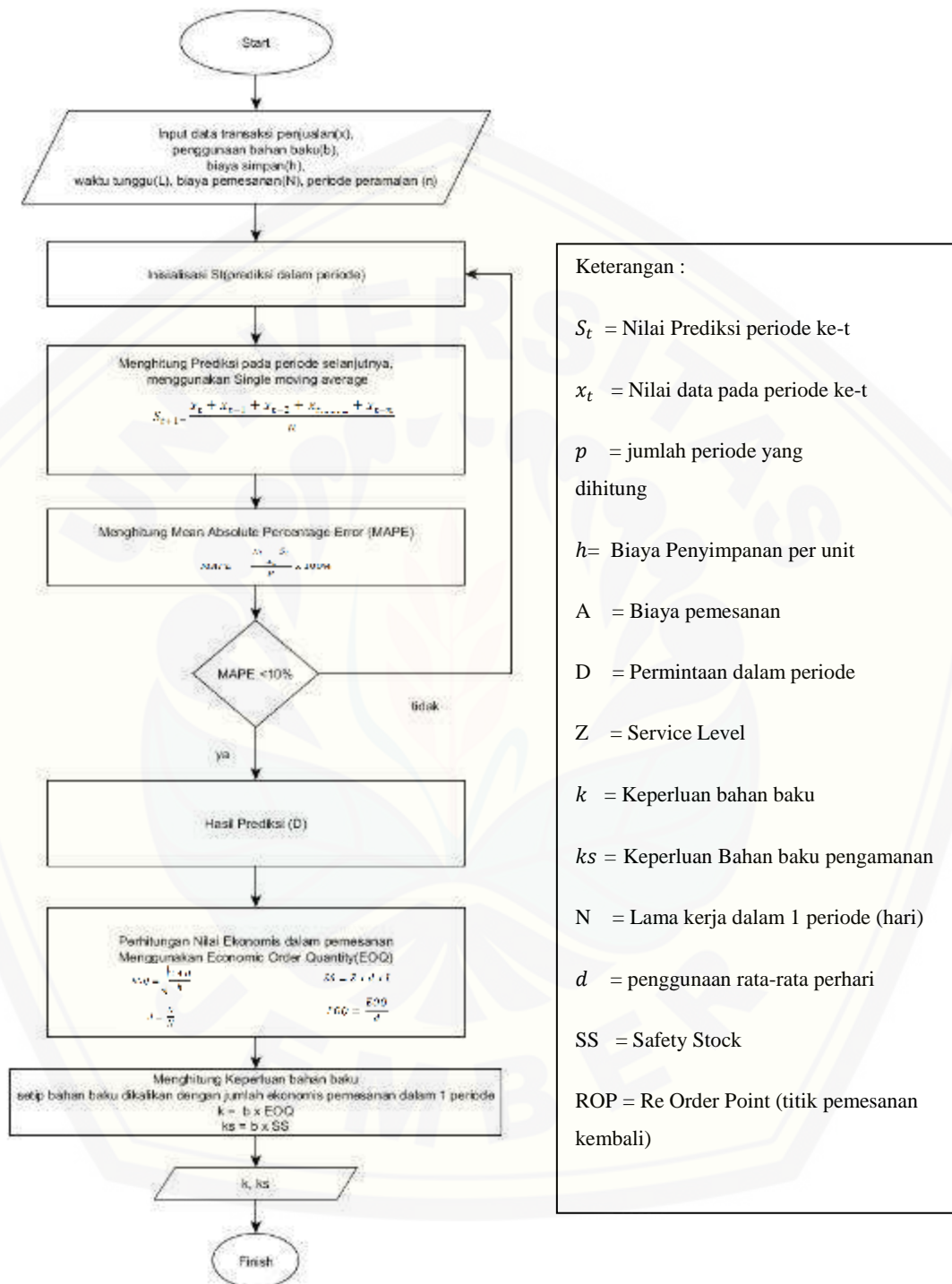
Pengumpulan data dengan cara observasi merupakan pengumpulan data dengan melihat secara langsung objek penelitian. Dalam penelitian ini, observasi dilakukan di Roots Cafe yang berada dijalan mastrip kabupaten Jember. Dalam observasi yang diamati merupakan proses penyimpanan bahan baku, transaksi penjualan dan pengadaan bahan baku.

### 3.3 Tahapan Analisis

Tahapan analisis dimulai dengan mengolah data hasil pengumpulan data yang menggunakan studi literatur, wawancara dan observasi. Dari data yang didapat kemudian diterapkan metode peramalan dengan menggunakan *Single Moving Average* selanjutnya diuji nilai rata-rata errornya, apabila kurang dari 10% maka dilanjutkan penerapan metode *Economic Order Quantity*(EOQ) lalu ditentukan kebutuhan bahan baku setiap menunya.

Penerapan metode *Single Moving Average* dan *Economic Order Quantity* dapat dilihat pada gambar 3.2, dimana data periode diambil dari data penjualan setiap bulannya. Penerapan *Single Moving Average* menggunakan 3 bulanan, apabila nilai rata-rata errornya lebih dari 10% maka akan diulangi lagi menggunakan 3 bulanan.





Gambar 3. 2Alur proses penerapan metode *Single Moving Average* dan EOQ

### 3.4 Tahapan Desain

Tahap desain sistem merupakan tahap perancangan arsitektur sistem penunjang keputusan yang akan dibangun. Desain sistem yang akan dibangun menggunakan model Object Oriented Desain (OOD), maka diagram yang harus dibuat antara lain :

1. Business Process

Business Process dalam perancangan sistem ini digunakan untuk menggambarkan seluruh proses yang dibutuhkan hingga proses goal yang diinginkan tercapai.

2. Use Case Diagram

Use Case Diagram dalam perancangan sistem ini digunakan untuk menggambarkan tugas user dalam sistem yang dirancang berdasarkan hak akses yang dimiliki user.

3. Scenario

Scenario dalam perancangan sistem ini digunakan untuk menjelaskan secara detail alur sistem dalam setiap fitur yang dirancang.

4. Sequence Diagram

Sequence Diagram dalam perancangan sistem ini digunakan untuk menunjukan rangkaian pesan yang dikirim antar object dan interaksi antar object.

5. Activity Diagram

Activity Diagram dalam perancangan sistem ini digunakan untuk menjelaskan secara detail alur sistem yang dirancang dengan menggunakan diagram.

6. Class Diagram

Class Diagram pada perancangan sistem ini digunakan untuk menggambarkan struktur class dalam sistem untuk mempermudah proses pengkodean.

7. Entity Relationship Diagram (ERD)

Entity Relationship Diagram pada perancangan sistem ini digunakan untuk menggambarkan relasi antar table dalam database.

### 3.5 Tahapan Implementasi

Setelah tahap desain sistem selesai dikerjakan, maka tahap selanjutnya adalah tahap implementasi sistem. Di tahap ini desain sistem yang telah dibuat diubah menjadi sistem nyata. Beberapa hal yang dilakukan dalam tahap implementasi sistem antara lain :

1. Penulisan kode program dengan menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP).
2. Basis data yang digunakan untuk media penyimpanan data menggunakan DBMS MySQL.

### 3.6 Tahapan Testing

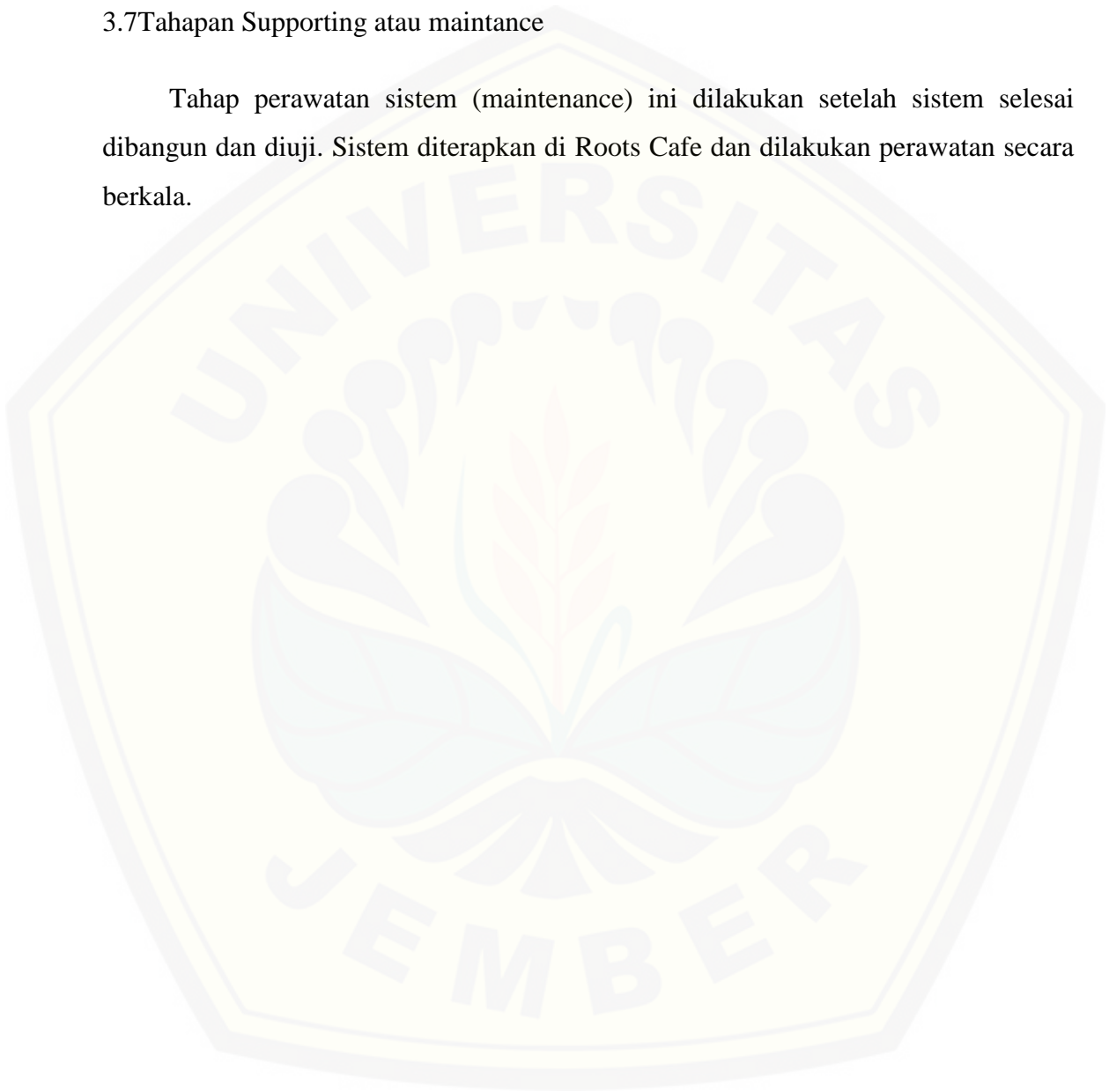
Tahap pengujian dilakukan dua kali yaitu pengujian perhitungan manual dan pengujian sistem. Pengujian perhitungan secara manual dilakukan dengan menghitung nilai *Mean Absolute Percentage Error* (MAPE) apakah peramalan yang dilakukan dengan menggunakan metode *single moving average* memiliki nilai MAPE rendah, jika nilai MAPE kurang dari 10% maka peramalan yang dilakukan sangat akurat, apabila dibawah 20% maka peramalan masih bisa dibilang akurat sedangkan apabila MAPE lebih dari 20% peramalan tidak akurat. Setelah itu dilakukan pengujian sistem, pengujian ini dilakukan setelah proses pembuatan sistem (implementasi) telah selesai dilakukan. Proses pengujian sistem dilakukan dengan 2 metode, yaitu metode white box yang akan dilakukan oleh peneliti dan metode black box yang akan dilakukan oleh pemilik cafe, pegawai cafe. Metode pengujian white box dilakukan untuk mengecek apakah ada kesalahan dalam penulisan kode program, teknik yang digunakan *cyclomatic complexity* yaitu dengan menggambarkan kode program kedalam *graph* yang berbentuk *node*(titik) dan *edge*(garis). Sedangkan untuk metode pengujian black box dilakukan untuk mengetahui apakah ada kesalahan



dalam fungsionalitas sistem yang telah dibangun, dalam hal ini akan langsung diuji oleh pemilik dan pegawai cafe.

### 3.7 Tahapan Supporting atau maintance

Tahap perawatan sistem (maintenance) ini dilakukan setelah sistem selesai dibangun dan diuji. Sistem diterapkan di Roots Cafe dan dilakukan perawatan secara berkala.



## BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan tentang tahapan-tahapan yang dilakukan dalam menganalisis kebutuhan sistem dan merancang sistem yang akan dibangun dengan menerapkan model pengembangan *waterfall*. Tahapan yang dilakukan meliputi analisa kebutuhan, desain, pengkodean, dan pengujian sistem.

### 4.1 Deskripsi Umum Sistem

Deskripsi umum mengenai Sistem manajemen persediaan bahan baku yang menerapkan metode *single moving average* dan *economic order quantity* yang dibangun dalam penelitian ini akan dijelaskan secara detail pada *Statement of Purpose* (SOP) dan fungsi sistem.

#### 4.1.1 *Statement of Purpose* (SOP)

Sistem persediaan bahan baku yang menerapkan metode *single moving average* dan *economic order quantity* yang diterapkan pada Roots cafe merupakan sistem yang dapat menampilkan hasil prediksi penjualan menu makanan dan minuman setiap menunya dan menampilkan persediaan bahan baku yang dibutuhkan pada periode selanjutnya. Data yang dibutuhkan oleh sistem ini antara lain data transaksi penjualan setiap menunya, data bahan baku, data menu makanan dan minuman, dan data penggunaan bahan baku. Dari data tersebut kemudian dilakukan perhitungan prediksi menggunakan metode *single moving average* hasil dari perhitungan tersebut digunakan untuk menentukan nilai ekonomis kebutuhan bahan baku pada periode selanjutnya.

Selain dapat mengelolah kebutuhan persediaan bahan baku, sistem juga dapat mengelola data pegawai, data transaksi, data persediaan bahan baku. Dengan menggunakan sistem persediaan bahan baku maka kebutuhan bahan baku untuk periode selanjutnya lebih akurat dan efisien.

#### 4.1.2 Fungsi Sistem

Fungsi utama dari sistem pengelola persediaan kebutuhan bahan baku di Roots cafe terletak pada perhitungan prediksi nilai penjualan pada periode selanjutnya dan perhitungan dalam menentukan kebutuhan bahan baku yang menerapkan metode EOQ untuk mengelola persediaan kebutuhan bahan baku. Fitur ini hanya dapat diakses oleh *owner cafe / admin* sistem. Sistem ini hanya memiliki 5 hak akses yaitu *admin/owner, Cheff, Barkeeper, Waiters* dan Kasir.

##### a. *Owner / Admin*

Owner atau Admin merupakan pengguna sistem yang memiliki hak akses untuk mengelola data user yang dapat mengakses sistem dan owner juga dapat me-*non aktif*-kan user sehingga tidak dapat mengakses kembali sistem. Owner juga dapat melihat total pendapatan dari transaksi penjualan setiap harinya, mengelolah data menu makanan, data menu minuman apabila ada penambahan menu baru ataupun penghapusan menu makanan maupun minuman, data prediksi penjualan pada periode selanjutnya dan data kebutuhan persediaan bahan baku pada periode berikutnya.

##### b. *Cheff*

*Cheff* merupakan pengguna sistem yang memiliki hak akses untuk mengelola data diri (*biodata*), penggunaan bahan baku dalam setiap menu makanan, dan persediaan bahan baku di cafe. Dalam penggunaan bahan baku owner menentukan nama menu makanan ataupun menu minuman yang dapat dilihat oleh semua user, selanjutnya *cheff* menentukan bahan baku yang digunakan untuk menu makanan tersebut. Dari penggunaan bahan baku untuk setiap menunya akan digunakan untuk menghitung kebutuhan persediaan bahan baku dengan menerapkan metode EOQ yang hasilnya hanya dapat diakses oleh Owner.

c. *Barkeeper*

*Barkeeper* merupakan pengguna sistem yang memiliki hak akses untuk mengelola data diri(*biodata*), penggunaan bahan baku dalam setiap menu minuman, dan persediaan bahan baku di cafe. Dalam penggunaan bahan baku owner menentukan nama menu makanan ataupun menu minuman yang dapat dilihat oleh semua user, selanjutnya *barkeeper* menentukan bahan baku yang digunakan untuk menu minuman tersebut. Dari penggunaan bahan baku untuk setiap menunya akan digunakan untuk menghitung kebutuhan persediaan bahan baku dengan menerapkan metode EOQ yang hasilnya hanya dapat diakses oleh Owner.

d. *Waiters*

*Waiters* merupakan pengguna sistem yang memiliki hak akses untuk mengelola data diri(*biodata*), dan mengelola transaksi penjualan di cafe. *Waiters* menambahkan transaksi ketika ada pelanggan ingin melakukan pemesanan menu, *waiters* juga mempunyai kewajiban memvalidasi apabila pesanan sudah diantar pada meja pemesan.

e. Kasir

Kasir merupakan pengguna sistem yang memiliki hak akses untuk mengelola data diri(*biodata*), dan mengelola data bayar total, yang dimaksud bayar total yaitu ketika pelanggan telah selesai dan akan melakukan pembayaran, pada fitur transaksi ada tombol bayar yang menampilkan total yang harus dibayar, dan nantinya akan merubah status meja menjadi kosong kembali.

#### 4.2 Analisis Kebutuhan Sistem

Berdasarkan teknik pengembangan sistem yang dibangun yaitu model *waterfall*, tahap awal yang dilakukan adalah tahap analisis kebutuhan. Analisis kebutuhan bertujuan untuk mengetahui apa saja yang dibutuhkan untuk proses pembangunan sistem. Kebutuhan yang dimaksud berupa kebutuhan fungsional dan kebutuhan non

fungsional. Hasil dari analisis kebutuhan yang dilakukan sangat berpengaruh pada fungsionalitas sistem yang dibangun.

#### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi fitur – fitur yang harus dipenuhi dalam sistem. Kebutuhan fungsional harus terpenuhi agar sistem berjalan sesuai dengan target yang diharapkan. Kebutuhan fungsional dari Sistem kebutuhan persediaan bahan baku di Roots Cafe yang menerapkan metode *single moving average* dan *economic order quantity* adalah sebagai berikut:

- a. Sistem memiliki proses login dan logout untuk membedakan hak akses dalam sistem.
- b. Sistem mampu mengelola data user yang mampu mengakses sistem sesuai dengan hak akses.
- c. Sistem mampu transaksi penjualan setiap harinya setiap menu makanan maupun minuman.
- d. Sistem mampu mengelola data penggunaan bahan baku setiap menu yang digunakan sebagai bahan perhitungan kebutuhan bahan baku dengan metode *Economic Order Quantity*.
- e. Sistem mampu memprediksi penjualan setiap menu pada periode selanjutnya dengan perhitungan *Single Moving Average*.
- f. Sistem mampu mengelola data bahan baku yang digunakan di Roots Cafe.

#### 4.2.2 Kebutuhan Non Fungsional

##### a. *Security*

Sistem memiliki keamanan berupa batasan hak akses yang digunakan untuk mengakses Sistem kebutuhan persediaan bahan baku di Roots Cafe yang menerapkan metode *single moving average* dan *economic order quantity*.

##### b. *Usability*



Sistem memiliki tampilan yang *user friendly* sehingga pengguna dapat dengan mudah menggunakan sistem.

c. *Reliability*

Sistem mampu bekerja sesuai dengan fungsinya dan dapat diakses dengan menggunakan computer dan *browser* yang berbeda.

d. *Availability*

Sistem bisa diakses kapan saja oleh pengguna dan mampu bekerja selama 24 jam.

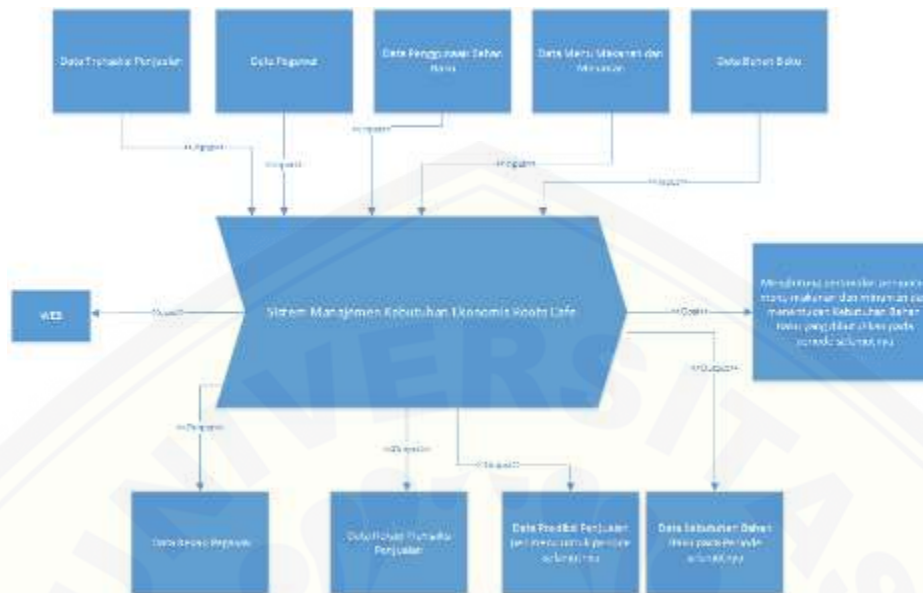
#### 4.3 Desain Sistem

Tahap yang dilakukan setelah analisis kebutuhan yaitu tahap desain sistem. Desain sistem yang dibuat pada Sistem pengelola kebutuhan persediaan bahan baku di Roots Cafe yang menerapkan metode *single moving average* dan *economic order quantity* meliputi *Business Process*, *Use Case Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class diagram*, dan *Entity Relation Diagram*.

##### 4.3.1 *Business Process*

*Business Process* merupakan gambaran dari proses yang terjadi di dalam sistem berupa *input*, *output*, serta *goal* sistem yang dibangun. *Business Process* menjelaskan *input* (masukan) yang dibutuhkan oleh sistem kemudian diolah menjadi *output* (keluaran) sesuai dengan *goal* (tujuan) dari sistem yang dibangun. Proses *input* dilakukan oleh pengguna, sedangkan proses *output* dilakukan oleh sistem. *Business Process* dari Sistem pengelola kebutuhan persediaan bahan baku di Roots Cafe yang menerapkan metode *single moving average* dan *economic order quantity* dijelaskan pada gambar 4.1

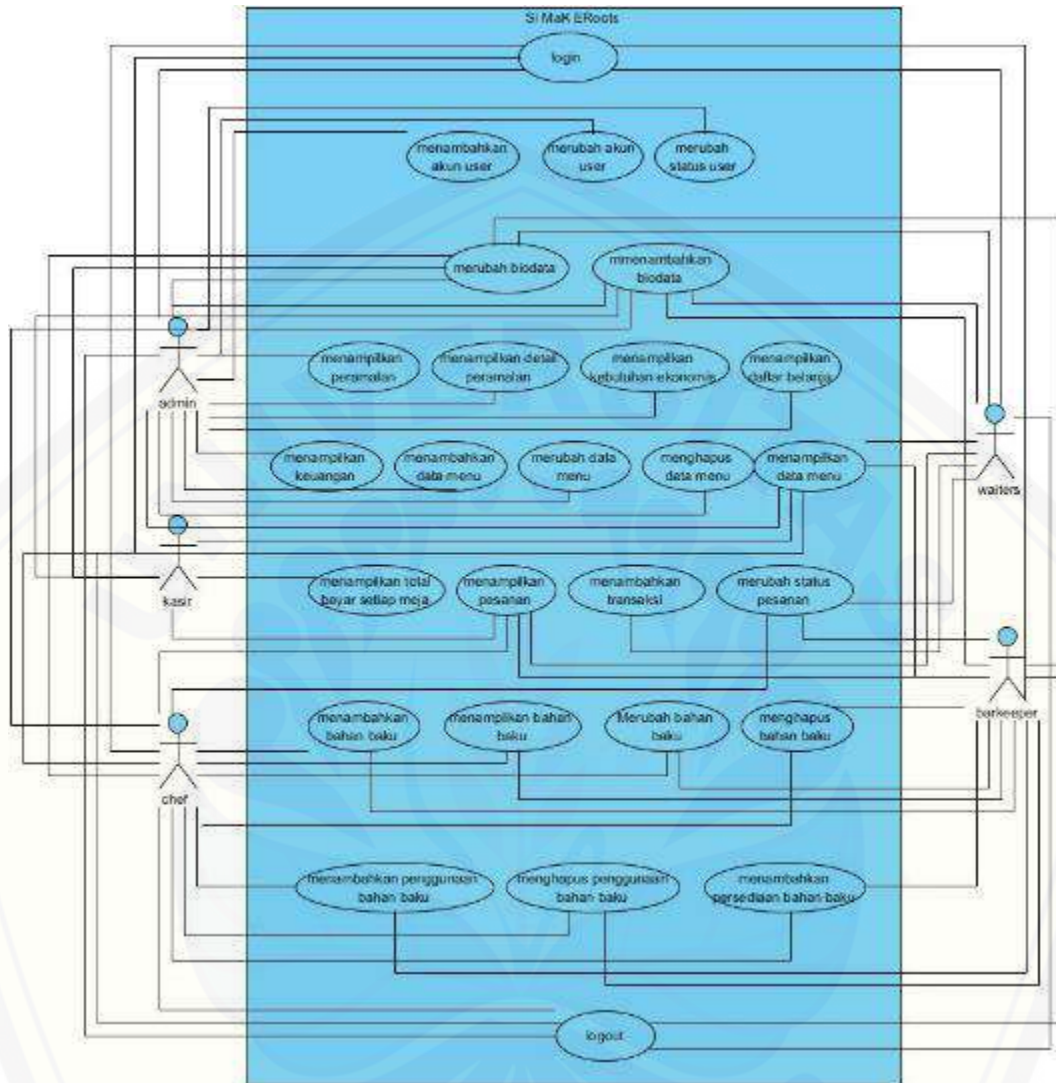




Gambar 4. 1 *Business Process* Sistem pengelola kebutuhan persediaan bahan baku

#### 4.3.2 Use Case Diagram

*Use Case Diagram* merupakan permodelan diagram yang menjelaskan interaksi antara aktor dengan sistem yang dibangun. Dari *Use Case Diagram* dapat diketahui fitur – fitur apa saja yang terdapat dalam sistem dan aktor siapa saja yang dapat mengakses sistem tersebut. *Use Case Diagram* dari Sistem kebutuhan persediaan bahan baku di Roots Cafe yang menerapkan metode *single moving average* dan *economic order quantity* dijelaskan pada gambar 4.2



Gambar 4. 2 Use Case Sistem pengelola kebutuhan persediaan bahan baku

Pada *Use Case Diagram* diatas dapat diketahui bahwa terdapat 5 aktor yang mampu mengakses sistem pengelola kebutuhan persediaan bahan baku. Aktor yang mengakses sistem tersebut yaitu *Owner/Admin*, *cheff*, *barkeeper*, *waiters* dan kasir. Penjelasan mengenai deskripsi aktor yang mengakses sistem tersebut dijelaskan pada Tabel 4.1

Tabel 4. 1 Deskripsi aktor sistem pengelola kebutuhan persediaan bahan baku

| No | Aktor              | Hak Akses  |
|----|--------------------|--|
| 1  | <i>Admin/Owner</i> | <i>Admin/Owner</i> merupakan aktor yang memiliki hak akses terhadap mengelola user yang mampu mengakses sistem, mengelola menu makanan dan minuman, data keuangan hasil transaksi harian ,perhitungan peramalan dengan metode <i>Single Moving Average</i> , dan perhitungan ekonomis kebutuhan persediaan bahan baku dengan metode <i>Economic Order Quantity</i> . |
| 2  | <i>Cheff</i>       | <i>Cheff</i> merupakan aktor yang memiliki hak akses terhadap data diri, mengelola penggunaan bahan baku pada menu makanan, memvalidasi pesanan menu makanan dan mengelola persediaan bahan baku yang terdapat di Roots Cafe.  |
| 3  | <i>Barkeeper</i>   | <i>Barkeeper</i> merupakan aktor yang memiliki hak akses terhadap data diri, mengelola penggunaan bahan baku pada menu minuman, memvalidasi pesanan menu minuman dan mengelola persediaan bahan baku yang terdapat di Roots Cafe.  |
| 4  | <i>Waiters</i>     | <i>Waiters</i> merupakan aktor yang memiliki hak akses terhadap data diri, dan menambahkan transaksi dan memvalidasi pesanan yang sudah terkirim kemeja pelanggan.   |
| 5  | Kasir              | Kasir merupakan aktor yang memiliki hak akses terhadap data diri, dan mengolah total bayar pesanan pelanggan.  |

Selain memiliki 5 aktor yang mampu mengakses sistem, sistem juga memiliki 27 *use case* yang merupakan fitur-fitur dalam sistem tersebut. Penjelasan mengenai *use case* dijelaskan pada table 4.2.

Tabel 4. 2 Deskripsi *use case* sistem pengelola kebutuhan persediaan bahan baku

| No | Use Case              | Deskripsi   |
|----|-----------------------|---|
| 1  | Login                 | Fitur ini digunakan untuk masuk ke dalam sistem sesuai dengan hak akses yang telah ditentukan yaitu Admin/Owner dan Pegawai.                    |
| 2  | Menambahkan akun user | Fitur ini dilakukan oleh aktor Admin/Owner untuk menambah <i>user</i> yang dapat mengakses sistem   |
| 3  | Merubah akun user     | Fitur ini dilakukan oleh Admin / Owner untuk merubah hak akses berupa username dan password   |
| 4  | Merubah status user   | Fitur ini dilakukan oleh Admin / Owner untuk menonaktifkan user, sehingga tidak dapat mengakses sistem kembali                                  |
| 5  | Merubah biodata       | Fitur ini dilakukan oleh semua aktor pada sistem untuk merubah biodata dan disimpan di database   |
| 6  | Menambahkan Biodata   | Fitur ini dilakukan oleh semua aktor pada sistem untuk menambahkan biodata yang nantinya akan tersimpan di database                             |
| 7  | Menampilkan Peramalan | Fitur ini dilakukan oleh Admin / Owner untuk menampilkan hasil peramalan pada transaksi 3 periode terakhir untuk peramalan periode selanjutnya. |

- 8 Menampilkan detail peramalan      Fitur ini dilakukan oleh Admin / Owner untuk menampilkan secara detail peramalan pada menu yang dipilih.
- 9 Menampilkan kebutuhan ekonomis      Fitur ini dilakukan oleh Admin / Owner untuk menampilkan hasil perhitungan EOQ, *Safety Stok* dan *ROP* untuk manajemen kebutuhan pada periode selanjutnya
- 10 MenampilkanDaftar Belanja      Fitur ini dilakukan oleh Admin / Owner dan pegawaiuntuk menampilkan daftar belanja hasil dari perhitungan kebutuhan ekonomis dikurangi dengan stok bahan baku yang tersedia. Apabila nilainya negative makan akan masuk pada daftar belanja.
- 11 Menampilkan Keuangan      Fitur ini terdapat pada fituowner untuk menampilkan pendapatan harian Roots cafe.
- 12 Menambah data menu      Fitur ini dilakukan oleh Admin / Owner untuk menambahkan menu makanan ataupun minuman pada sistem
- 13 Merubah data menu      Fitur ini dilakukan oleh Admin / Owner untuk merubah menu makanan ataupun minuman pada sistem
- 14 Menghapus data menu      Fitur ini dilakukan oleh Admin / Owner untuk menghapus menu makanan ataupun minuman pada sistem
- 15 Menampilkan menu      Fitur ini dilakukan oleh semua aktor untuk menampilkan menu makanan ataupun minuman pada



sistem. Sehingga semua aktor tahu menu makanan dan minuman yang tersedia di Roots Cafe

- 16 Menampilkan total bayar setiap meja      Fitur ini dilakukan oleh Kasir untuk menampilkan total transaksi setiap meja dan mengaktifkan kembali meja untuk pengunjung selanjutnya.
- 17 Menampilkan pesanan      Fitur ini dilakukan oleh *chef*, *barkeeper*, *waiters*, dan *kasir* untuk mengetahui proses pesanan pelanggan
- 18 Menambahkan data Transaksi      Fitur ini dilakukan oleh *Waiters* untuk menambahkan hasil transaksi setiap harinya
- 19 Merubah status pesanan      Fitur ini hanya dapat diakses oleh *Waiters*, *Barkeeper*, dan *chef*, untuk merubah status pesanan apabila tugas dari setiap aktor selesai.
- 20 Menambahkan data Bahan Baku      Fitur ini dilakukan oleh *Cheff* dan *Barkeeper* untuk menambahkan data bahan baku yang diperlukan pada sistem
- 21 Menampilkan Bahan Baku      Fitur ini dilakukan oleh *Admin/Owner* dan pegawai untuk menampilkan bahan baku yang digunakan pada sistem
- 22 Merubah data Bahan Baku      Fitur ini dilakukan oleh *Cheff* dan *Barkeeper* untuk merubah data bahan baku yang sudah tersimpan pada sistem
- 23 Menampilkan Daftar Belanja      Fitur ini hanya dapat diakses oleh *Admin/Owner* yang menampilkan daftar belanja dengan peramalan dan perhitungan *Economic Order Quantity* untuk periode



selanjutnya.

- |    |                                   |   |
|----|-----------------------------------|---|
| 24 | Menambahkan Penggunaan Bahan Baku | Fitur ini dilakukan oleh <i>Cheff</i> dan <i>Barkeeper</i> untuk menambahkan penggunaan bahan baku pada setiap menu makanan dan minuman |
| 25 | Menghapus Penggunaan Bahan Baku   | Fitur ini dilakukan oleh <i>Cheff</i> dan <i>Barkeeper</i> untuk menghapus penggunaan bahan baku pada setiap menu makanan dan minuman   |
| 26 | Menambahkan persediaan bahan baku | Fitur ini hanya dapat diakses oleh <i>Cheff</i> dan <i>Barkeeper</i> yang digunakan untuk menambahkan persediaan bahan baku.            |
| 27 | Logout                            | Fitur ini dilakukan oleh aktor Admin/Owner dan pegawai untuk keluar dari sistem   |

Berikut ini merupakan penjelasan mengenai desain sistem dari sistem pengelola kebutuhan persediaan bahan baku yang akan dibangun berdasarkan use case yang telah dibuat.

1) *Use Case Login*

*Use Case Login* merupakan fitur yang memungkinkan *user* untuk melakukan proses *login* agar *user* dapat menjalankan sistem sesuai hak akses yang dimiliki setiap *user*. Desain sistem yang lebih detail untuk *use case login* digambarkan melalui *scenario*. *Scenario login* menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur login seperti pada tabel 4.3

Tabel 4.3 *Scenario Login*

| ID Skenario | USC1 |
|-------------|------|
|-------------|------|

|                  |                              |
|------------------|------------------------------|
| Usecase          | Login                        |
| Partisipan Aktor | Semua Aktor                  |
| Pre-Kondisi      | Aktor sudah memasuki sistem. |
| Pra-Kondisi      | Aktor sudah melakukan login. |

### SKENARIO UTAMA

#### “Login”

| Aksi Aktor  | Reaksi Sistem   |
|---|---|
| 1. Membuka Sistem Manajemen Kebutuhan Ekonomis Roots Cafe |   |
|   | 2. Menampilkan halaman Sistem Manajemen Kebutuhan Ekonomis Roots Cafe |
| 3. Mengisi username dan password sesuai hak akses.        |   |
| 4. Klik tombol Login.                                     |   |
|   | 5. Menampilkan halaman Dashboard sesuai dengan hak akses.             |

### SKENARIO ALTERNATIF

#### “Username dan Password tidak diisi”

| Aksi Aktor            | Reaksi Sistem   |
|-----------------------|---|
| 4. Klik tombol Login. |   |
|                       | 5. Menampilkan notifikasi “Please fill out this field.” |

### SKENARIO ALTERNATIF

#### “Username dan Password salah”

| Aksi Aktor   | Reaksi Sistem |
|--|---------------|
| 3. Mengisi username dan password sesuai hak akses. |               |

---

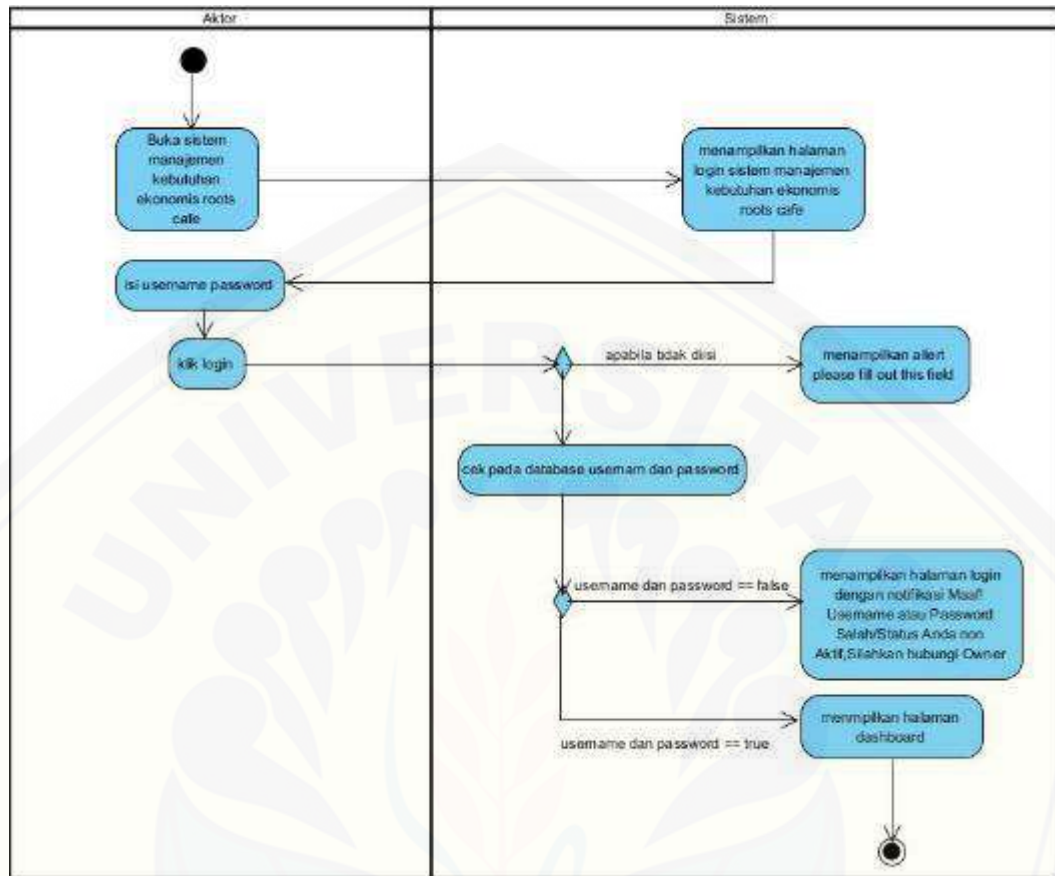
4. Klik tombol Login.

---

5. Menampilkan notifikasi “Maaf!  
Username atau Password  
Salah/Status Anda non  
Aktif,Silahkan hubungi Owner”

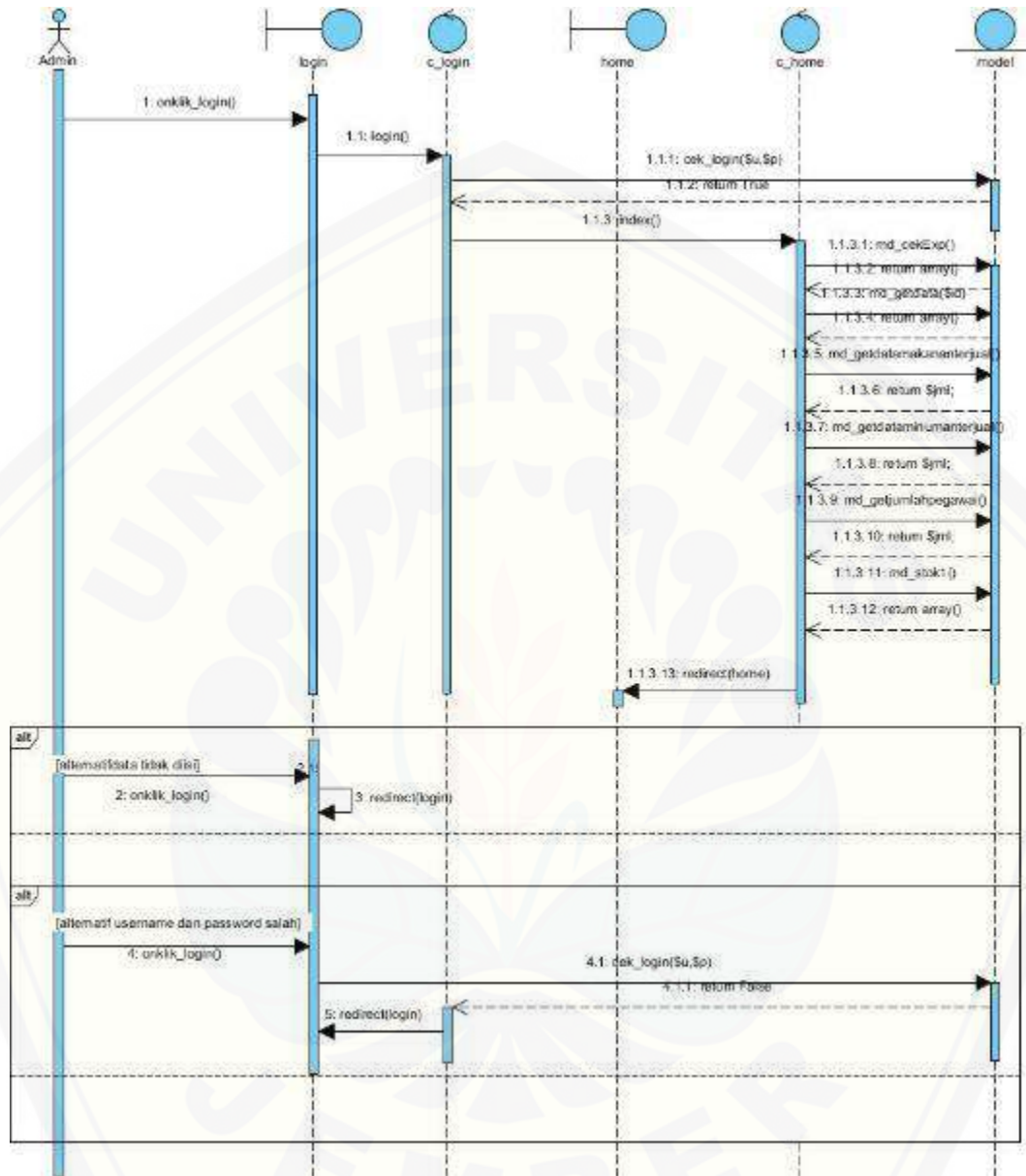
---

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur *login* merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses *login*. Penjelasan *activity diagram* fitur *login* dapat dilihat pada Gambar 4.3



Gambar 4.3 Activity Diagram Login

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur *login* menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses *login*. Penjelasan *sequence diagram* fitur *login* dapat dilihat pada Gambar 4.4 untuk aktor admin dan untuk aktor Cheff dapat dilihat pada lampiran C.1.1, untuk aktor Barkeeper pada lampiran C.1.2, untuk aktor Waiters pada lampiran C.1.3 dan untuk kasir pada lampiran C.1.4



Gambar 4. 4Sequence Diagram Login (Admin)

## 2) Menambahkan akun User

Menambahkan akun user merupakan fitur untuk menambahkan *user* yang dapat mengakses sistem, fitur ini berisi *form username*, *password* yang akan diisi dengan akun user yang akan mengakses sistem dan hanya dapat diakses oleh *owner/admin*. Desain sistem yang lebih detail untuk menambahkan data user digambarkan melalui

*scenario*. *Scenario* menambahkan data user menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan data user seperti pada tabel 4.4

Tabel 4. 4*Scenario* menambahkan data user

|  |   |
|--|---|
| <b>ID Skenario</b>                               | <b>USC2</b>   |
| Usecase  | MenambahkanUser   |
| Partisipan Aktor                                 | Owner   |
| Pre-Kondisi                                      | Owner sudah melakuakan login.   |
| Pra-Kondisi                                      | Owner sudah menambahkan user yang dapat mengakses sistem.                             |
| <b>SKENARIO UTAMA</b>                            |   |
| <b>“Menambah User”</b>                           |   |
| <b>Aksi Aktor</b>                                | <b>Reaksi Sistem</b>  |
| 1. Klik Menu Data Pegawai.                       |   |
|  | 2. Menampilkan Halaman Data Pegawai Roots Cafe  |
| 3. Klik tombol Tambah Data.                      |   |
|  | 4. Menampilkan Halaman Form Tambah User.  |
| 5. Mengisi Username, Password, dan jabatan user. |   |
| 6. Klik tombol Submit                            |   |
|  | 7. Menampilkan Halaman Data Pegawai dan menampilkan notifikasi Success “User Berhasil |



---

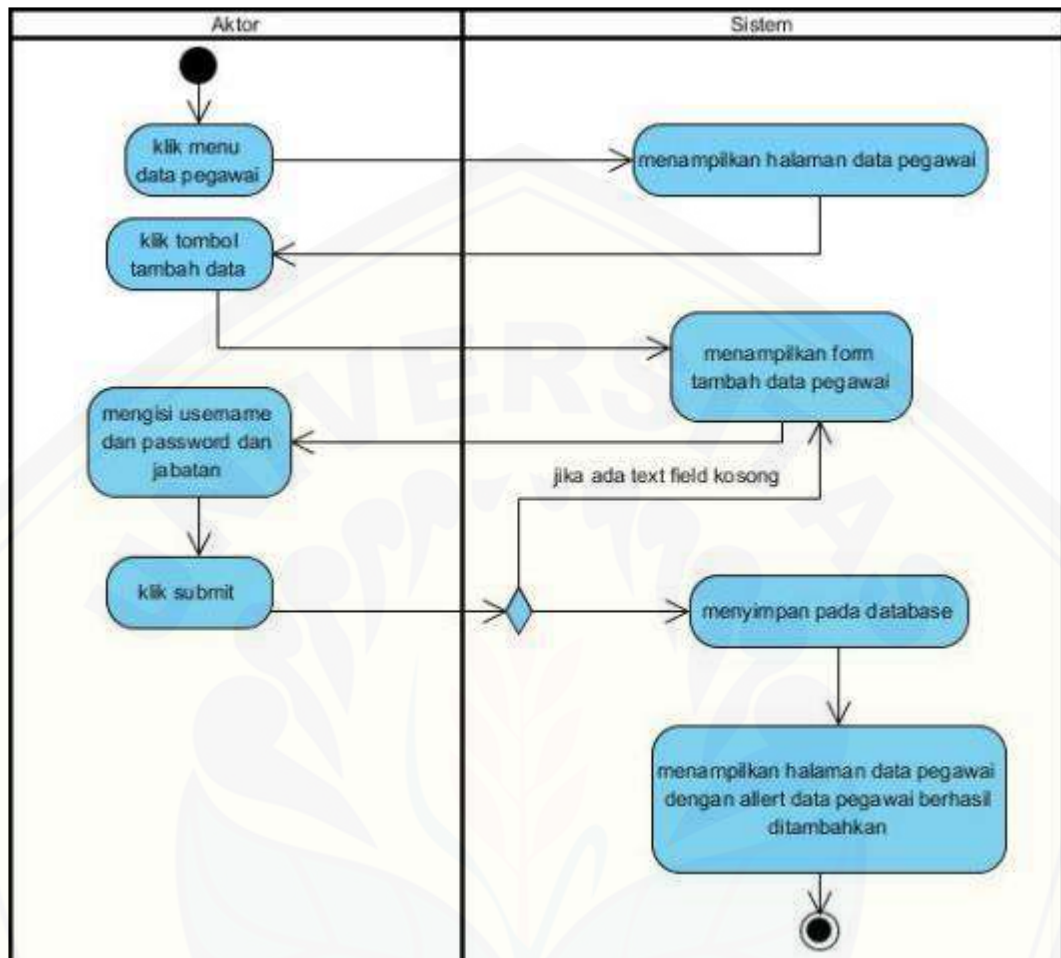
Ditambahkan”.

**SKENARIO ALTERNATIF**

**“Data belum diisi”**

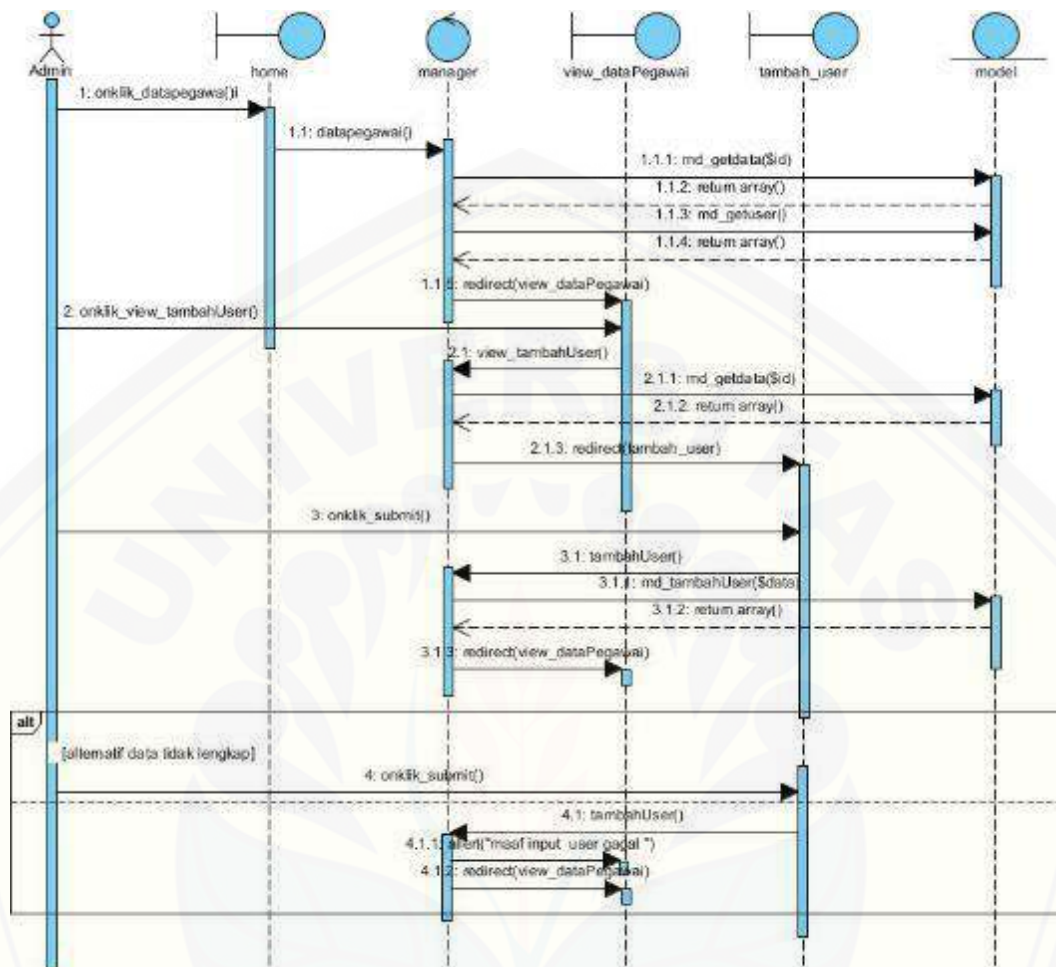
| Aksi Aktor   | Reaksi Sistem                                       |
|--|---|
| 5. Mengisi Username dan jabatan tanpa mengisi password |   |
| 6. Klik tombol Simpan                                  |   |
|  | 7. Menampilkan notifikasi “Maaf! input user gagal.” |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menambahkan data user merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan data user. Penjelasan *activity diagram* fitur menambahkan data user dapat dilihat pada gambar 4.5



Gambar 4.5 Activity Diagram Menambahkan akun user

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan data user menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data user. Penjelasan *sequence diagram* fitur menambahkan data user dapat dilihat pada gambar 4.6



Gambar 4. 6Sequence menambahkan akun user

### 3) Merubah akun user

Merubah akun user merupakan fitur untuk merubah akun user login yang tersimpan pada sistem, fitur ini berisi *form username*, *password* yang terisi dengan data user login yang ingin dirubah untuk user mengakses sistem dan fitur ini hanya dapat diakses oleh *owner/admin*. Desain sistem yang lebih detail untuk *merubah data user* digambarkan melalui *scenario*. *Scenario* merubah data user menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur merubah data user seperti pada tabel 4.5

Tabel 4. 5 *Scenariomerubah data user*

| <b>ID Skenario</b>   | <b>USC3</b>   |
|--|---|
| Usecase  | Merubah Akun User   |
| Partisipan Aktor   | Owner   |
| Pre-Kondisi  | Owner sudah melakukan login.  |
| Pra-Kondisi  | Owner sudah merubah username dan password user yang dapat mengakses sistem.                     |
| <b>SKENARIO UTAMA</b>  |   |
| <b>“Merubah Data Akses User”</b>   |   |
| <b>Aksi Aktor</b>  | <b>Reaksi Sistem</b>  |
| 1. Klik Menu Data Pegawai.   | 2. Menampilkan Halaman Data Pegawai Roots Cafe  |
| 3. Klik tombol berwarna biru pada data yang akan dirubah username dan passwordnya. | 4. Menampilkan Halaman Form Edit User.  |
| 5. MerubahUsername, Password, dan jabatan user.                                    |   |
| 6. Klik tombol Submit  | 7. Menampilkan Halaman Data Pegawai dan menampilkan notifikasi Success “User Berhasil Dirubah”. |
| <b>SKENARIO ALTERNATIF</b>   |   |
| <b>“Data belum diisi”</b>  |   |

| Aksi Aktor   | Reaksi Sistem  |
|--|--|
| 5. Mengisi Username dan jabatan tanpa mengisi password |  |
| 6. Klik tombol Submit                                  |  |
|  | 7. Menampilkan notifikasi “user gagal edit!! Username atau password tidak boleh kosong.” |

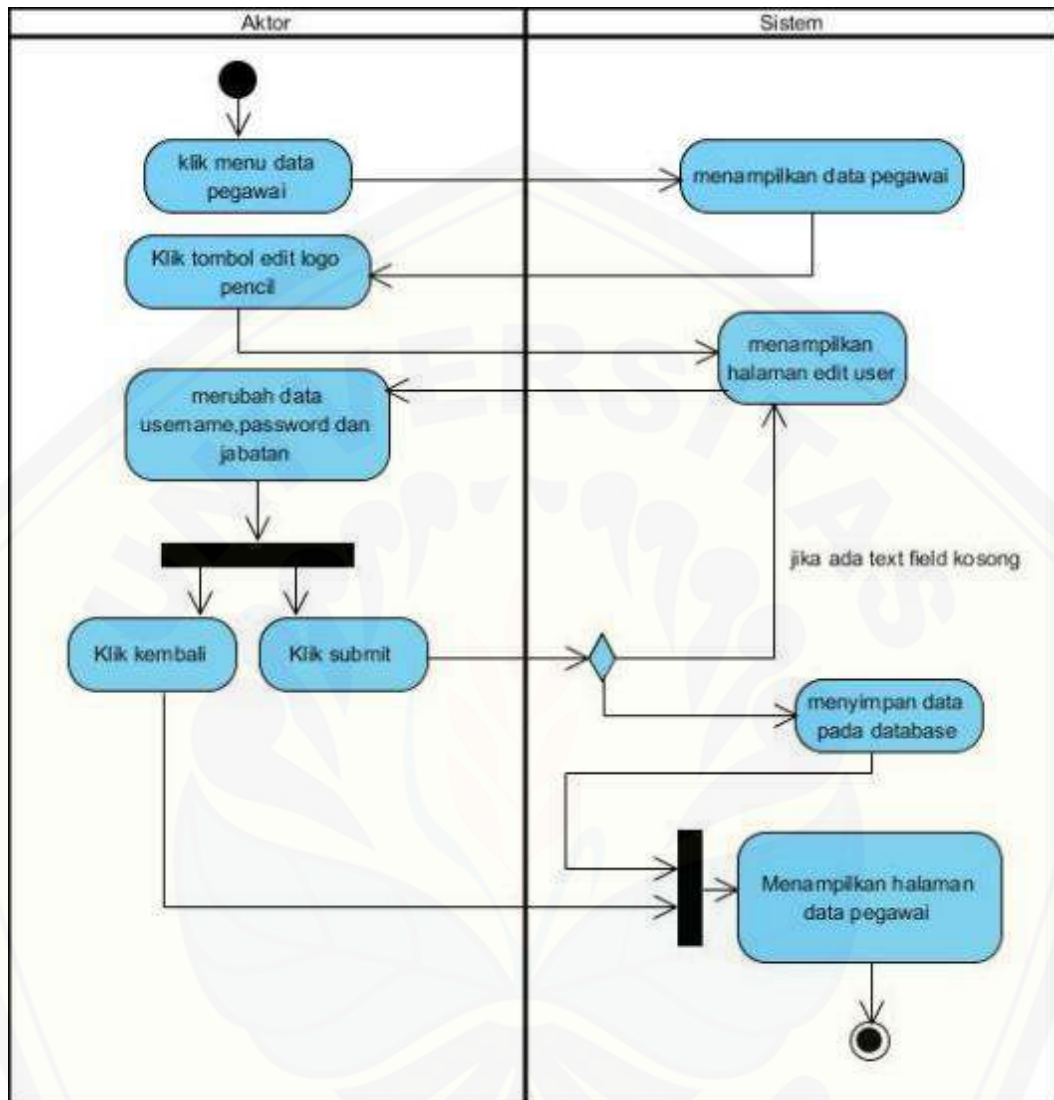
#### SKENARIO ALTERNATIF

##### “Batal Edit”

| Aksi Aktor             | Reaksi Sistem                  |
|------------------------|--------------------------------|
| 5. Klik tombol Kembali |                                |
|                        | 6. Menampilkan Halaman Pegawai |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur merubah akun user merupakan gambaran alur interaksi yang dilakukan aktor. Penjelasan *activity diagram* fitur merubah akun user dapat dilihat pada gambar 4.6

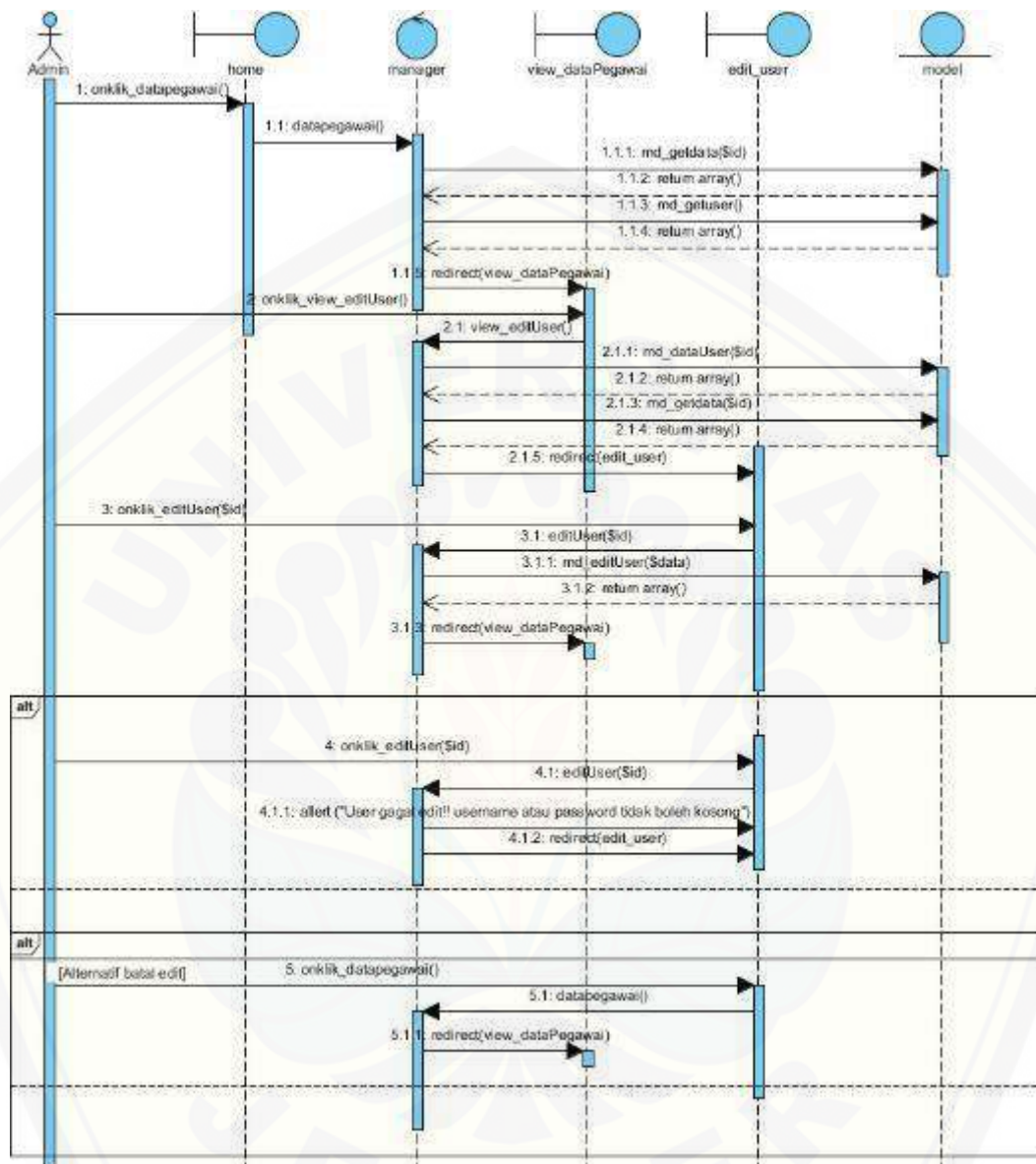




Gambar 4.7 Activity Diagram merubah aku user

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur merubah akun user menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah akun user. Penjelasan *sequence diagram* fitur merubah data user dapat dilihat pada gambar 4.8





Gambar 4. 8Sequence Diagram merubah akun user

4) Merubah status User

Merubah status user merupakan fitur untuk menon-aktifkan data *user* *loginyang* tersimpan pada sistem, fitur ini berfungsi menon-aktifkan username dan password sehingga username tersebut tidak dapat mengakses sistem dan fitur ini hanya dapat diakses oleh *owner/admin* Desain sistem yang lebih detail untuk

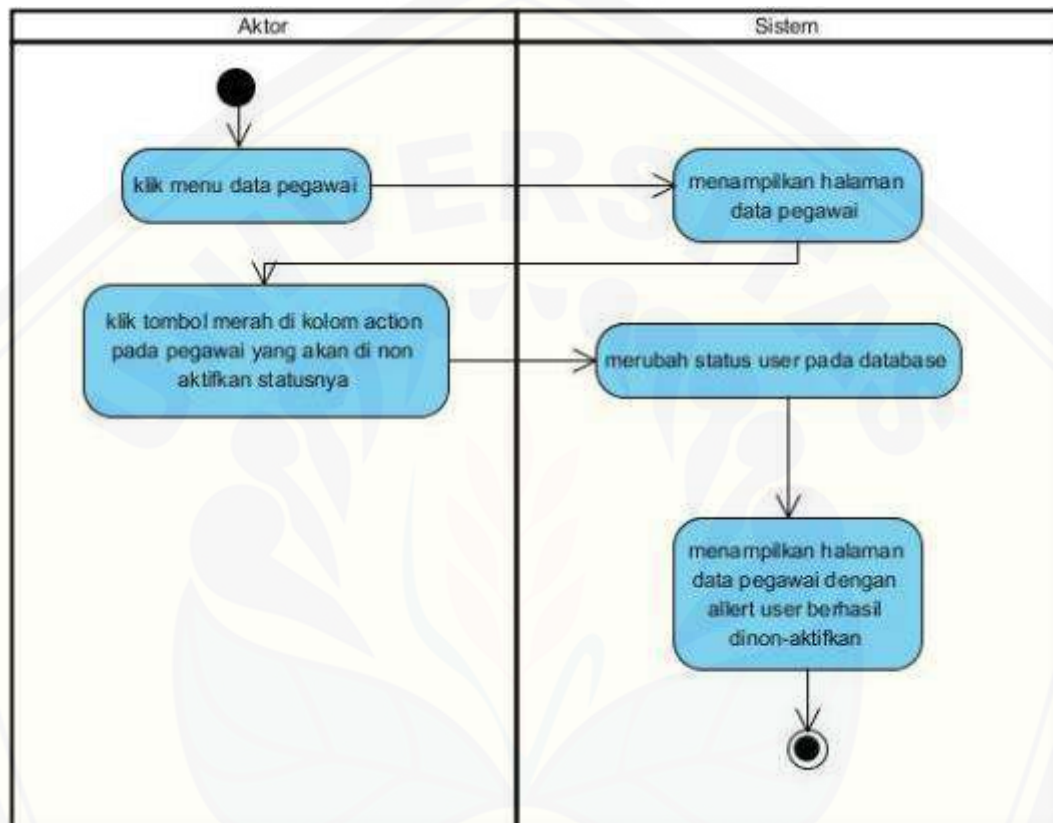
*menghapus data user* digambarkan melalui *scenario*. *Scenario* merubah status user menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menghapus data user seperti pada tabel 4.6

Tabel 4. 6 *Scenario* merubah status user

| ID Skenario  | USC4  |
|--|---|
| Usecase  | MerubahStatusUser   |
| Partisipan Aktor   | Owner   |
| Pre-Kondisi  | Owner sudah melakukan login.  |
| Pra-Kondisi  | Owner sudah menon-aktifkan user dan user tidak dapat mengakses sistem.  |
| <b>SKENARIO UTAMA</b>  |   |
| <b>“Menghapus User”</b>  |   |
| Aksi Aktor   | Reaksi Sistem   |
| 1. Klik Menu Data Pegawai.   | 2. Menampilkan Halaman Data Pegawai Roots Cafe  |
| 3. Klik tombol berwarna merah pada data user yang akan dinon-aktifkan. | 4. Menampilkan Halaman Data Pegawai Roots Cafe dengan notifikasi “user berhasil di non aktifkan, silahkan lihat pada user off.” |

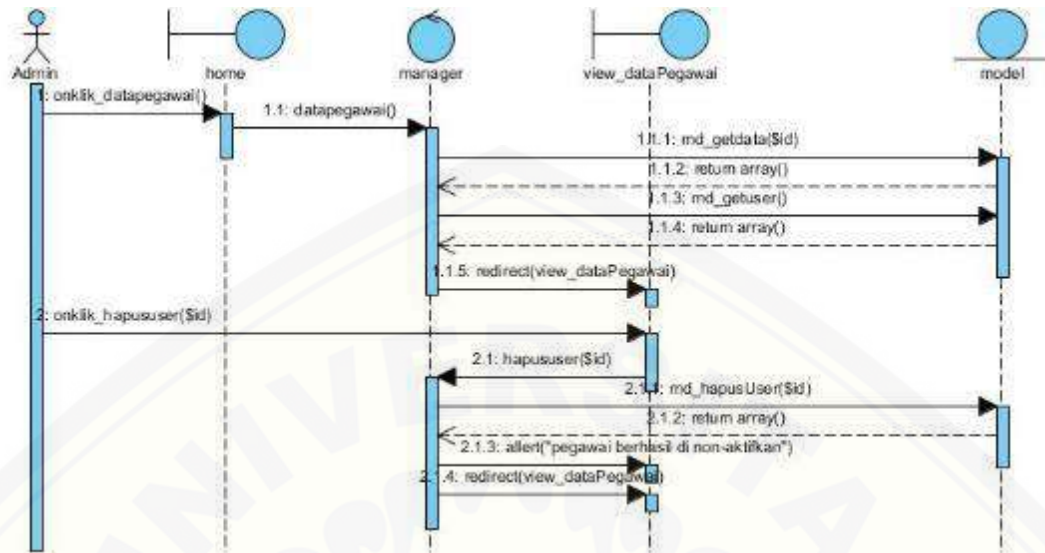
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menghapus data user merupakan

gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses *menghapus data user*. Penjelasan *activity diagram* fitur merubah status user dapat dilihat pada gambar 4.9



Gambar 4. 9Activity Diagram merubah status user

Tahap selanjutnyaitu pembuatan *sequence diagram*. *Squence diagram* fitur merubah status usermenggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah status user. Penjelasan *sequence diagram* fitur menghapus data *user* dapat dilihat pada gambar 4.10



Gambar 4. 10Sequence Diagram merubah status user

#### 5) Merubah Biodata

Merubah biodata merupakan fitur untuk merubah informasi biodata user yang telah tersimpan pada sistem, fitur ini berisi *formusername*, *password*, nama, tanggal lahir, jenis kelamin, alamat dan nomor telephone berisi data diri yang pernah tersimpan dan akan dirubah. Fitur ini dapat diakses oleh semua aktor pada sistem. Desain sistem yang lebih detail untuk merubah data biodata digambarkan melalui *scenario*. *Scenario* merubah data biodata menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur merubah data biodata seperti pada tabel 4.7

Tabel 4. 7Scenario merubah biodata

|                    |   |
|--------------------|---|
| <b>ID Skenario</b> | <b>USC5</b>                                       |
| Usecase            | Merubah Data diri                                 |
| Partisipan Aktor   | Semua Aktor                                       |
| Pre-Kondisi        | Owner sudah melakukan login.                      |
| Pra-Kondisi        | Aktor sudah mengubah Data diri dan tersimpan pada |

sistem.

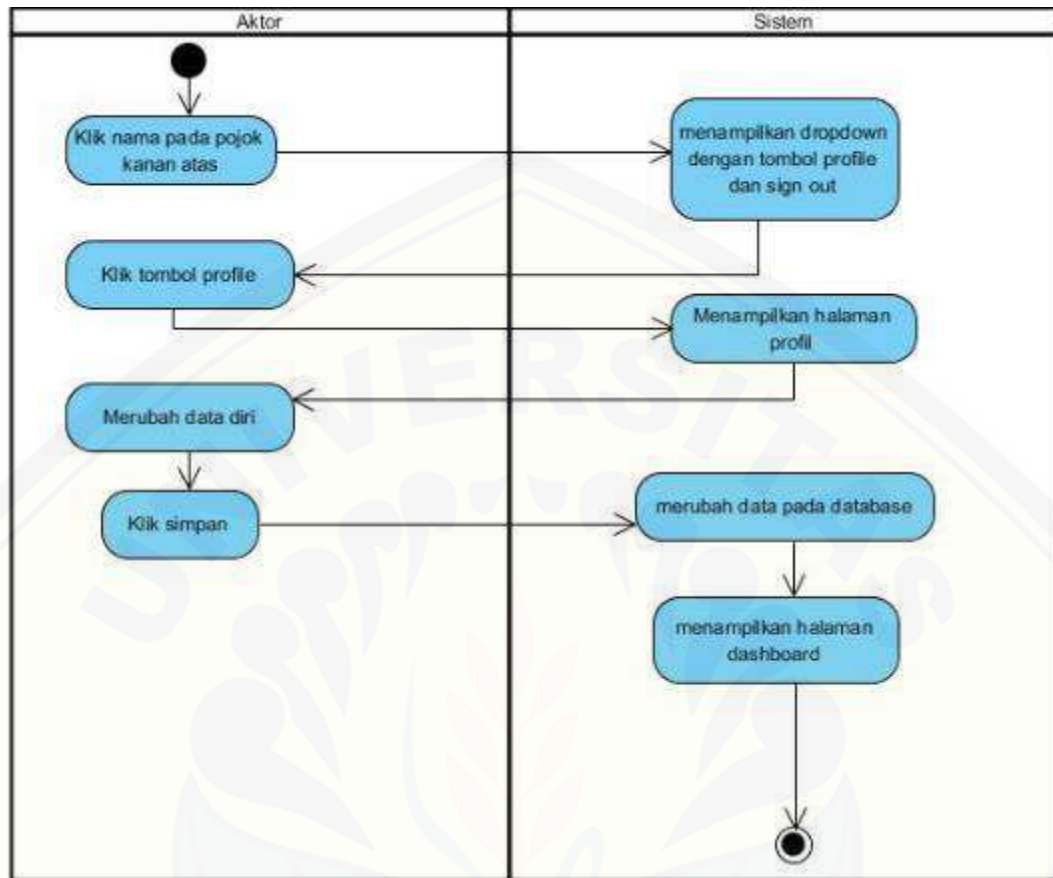
### SKENARIO UTAMA

#### “Merubah Data diri”

| Aksi Aktor  | Reaksi Sistem   |
|---|---|
| 1. Klik nama user yang berada pada pojok kanan tampilan   | 2. Menampilkan Dropdown dengan tombol profile dan sign out. |
| 3. Klik tombol profile.   | 4. Menampilkan Halaman Form Profil manager/pegawai          |
| 5. Mengisi ulang Username Password , nama, tanggal lahir, pilih jenis kelamin, alamat dan nomer telpon (apabila ingin dirubah), |   |
| 6. Klik tombol Simpan   | 7. Menampilkan Halaman Home kembali.                        |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur merubah biodatamerupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses merubah data biodata. Penjelasan *activity diagram* fitur merubah biodata dapat dilihat pada gambar 4.11

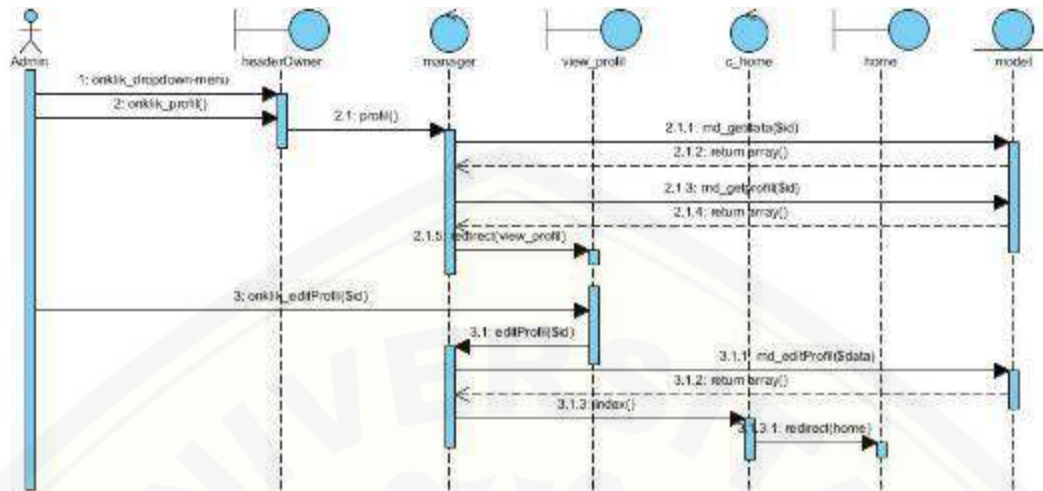




Gambar 4. 11 Activity Diagram merubah biodata

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Sequence diagram* fitur merubah biodatamenggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah biodata. Penjelasan *sequence diagram* fitur merubah biodatauntuk aktor admin dapat dilihat pada gambar 4.12 dan untuk aktor *Cheff* dapat dilihat pada lampiran C.5.1, untuk aktor *Barkeeper* dapat dilihat pada lampiran C.5.2, untuk aktor *Waiters* dapat dilihat pada lampiran C.5.3 dan untuk aktor kasir dapat dilihat pada C.5.4





Gambar 4. 12Sequence Diagram Merubah biodata

6) Menambahkan Biodata

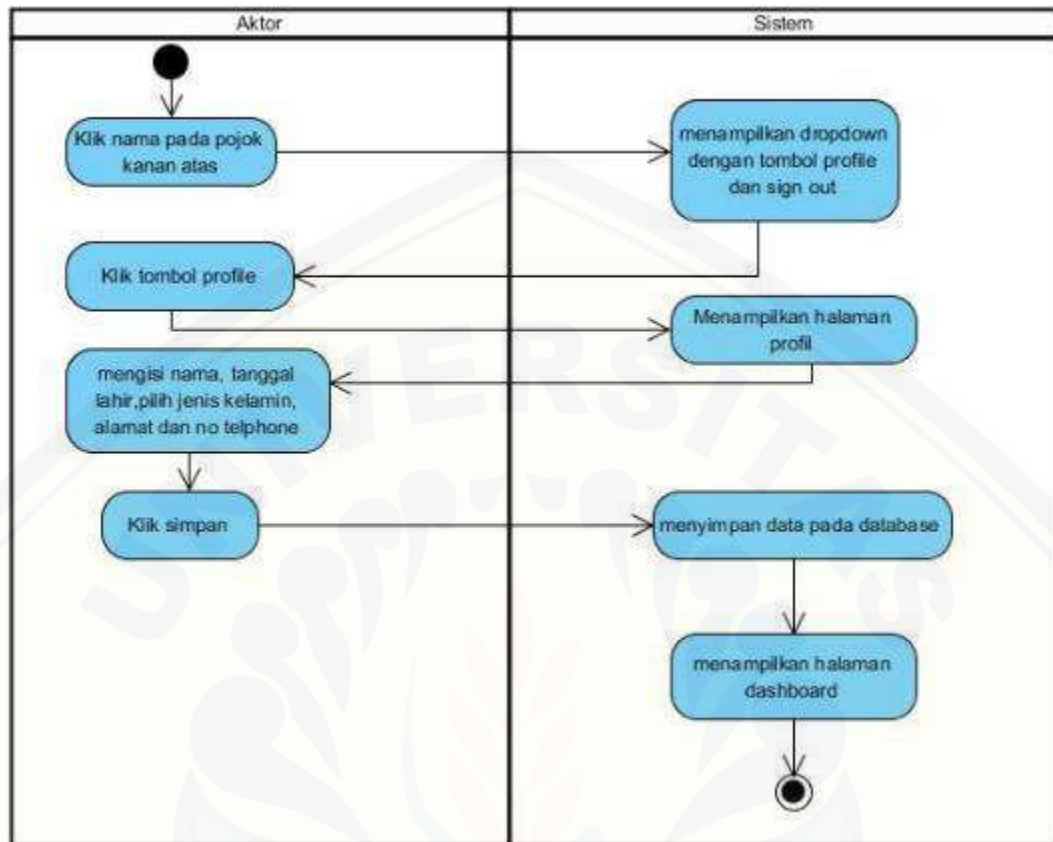
Menambahkan Biodata merupakan fitur untuk menambahkan informasi biodata user yang akan tersimpan pada sistem, fitur ini berisi *formusername*, *password*, nama, tanggal lahir, jenis kelamin, alamat dan nomor telephone yang nantinya setiap aktor (*user*) diperkenankan mengakses fitur ini untuk mengisi data diri *user*. Fitur ini dapat diakses oleh semua aktor pada sistem. Desain sistem yang lebih detail untuk menambahkan data biodata digambarkan melalui *scenario*. *Scenario* menambahkan data biodata menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan data biodata seperti pada tabel.4.8

Tabel 4. 8Scenario menambahkan biodata

| ID Skenario      | USC6   |
|------------------|--|
| Usecase          | Menambahkan Data diri                                    |
| Partisipan Aktor | Semua Aktor  |
| Pre-Kondisi      | Aktor sudah melakuakan login.                            |
| Pra-Kondisi      | Aktorsudah mengisi Data diri dan tersimpan pada sistem.. |

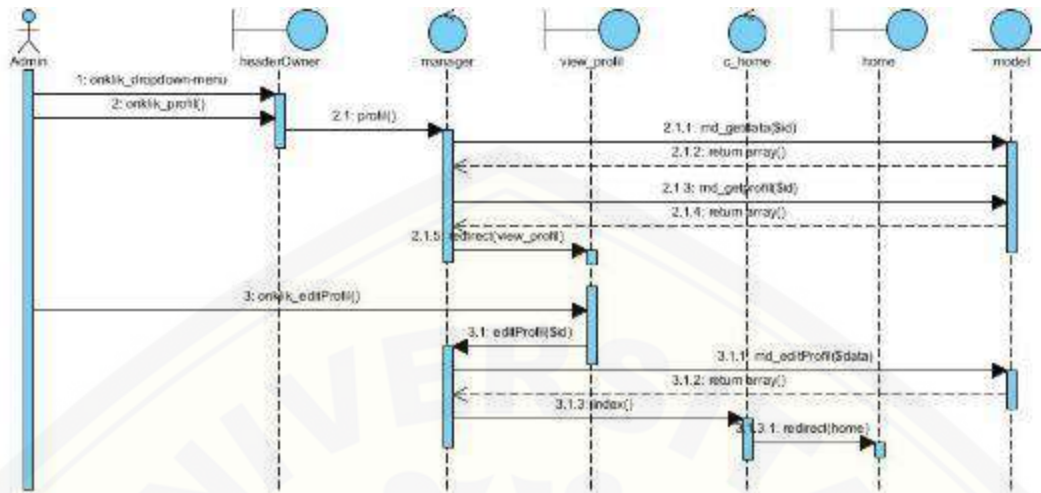
| <b>SKENARIO UTAMA</b>  |   |
|--|---|
| <b>“Menambahkan Data diri”</b>   |   |
| <b>Aksi Aktor</b>  | <b>Reaksi Sistem</b>  |
| 1. Klik nama user yang berada pada pojok kanan tampilan  |   |
|  | 2. Menampilkan Dropdown dengan tombol profile dan sign out. |
| 3. Klik tombol profile.  |   |
|  | 4. Menampilkan Halaman Form Profil manager/pegawai          |
| 5. Mengisi ulang Username dan Password (apabila ingin dirubah), dan mengisi nama, tanggal lahir, pilih jenis kelamin, alamat dan nomer telpon. |   |
| 6. Klik tombol Simpan  |   |
|  | 7. Menampilkan Halaman Home kembali.                        |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menambahkan data biodata merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan data biodata. Penjelasan *activity diagram* fitur menambahkan data biodata dapat dilihat pada gambar 4.13



Gambar 4. 13Activity menambahkan biodata

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Sequence diagram* fitur *menambahkan data biodata* menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data *biodata*. Penjelasan *sequence diagram* fitur *menambahkan data biodata* dapat dilihat pada gambar 4.14 dan untuk aktor *Cheff* dapat dilihat pada lampiran C.6.1, untuk aktor *Barkeeper* dapat dilihat pada lampiran C.6.2, untuk aktor *Waiters* dapat dilihat pada lampiran C.6.3 dan untuk aktor kasir dapat dilihat pada C.6.4



Gambar 4. 14Sequence Diagram menambahkan biodata

7) Menampilkan peramalan

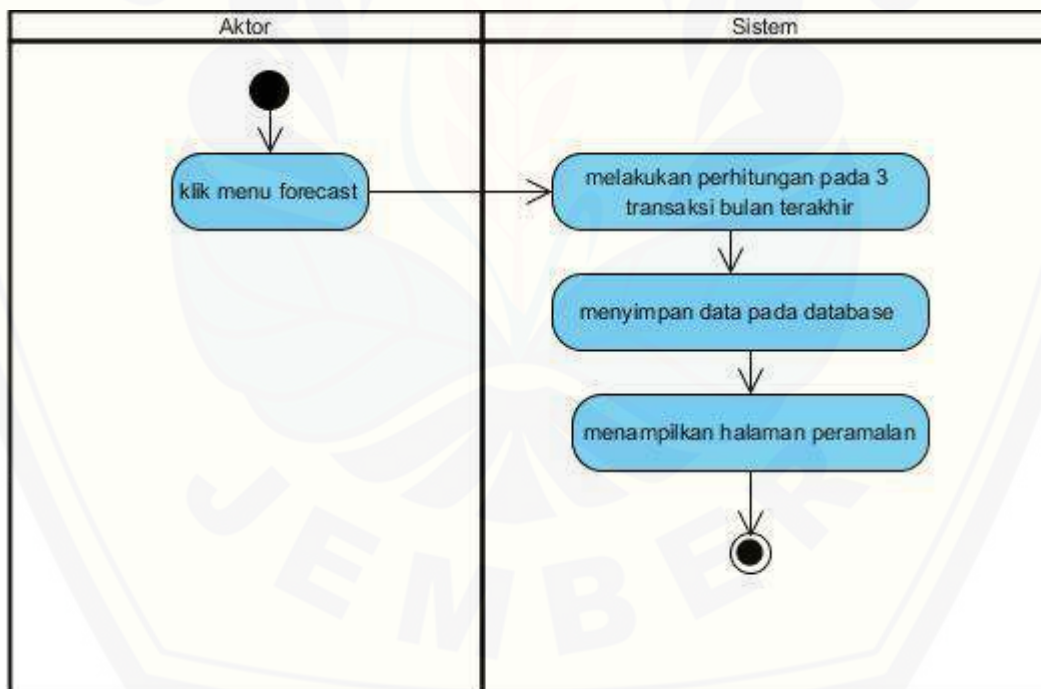
Menampilkan peramalan merupakan fitur untuk menampilkan data transaksi 3 bulan terakhir dan peramalannya untuk periode selanjutnya. Fitur ini menyimpan data secara otomatis dan merubah data secara otomatis apabila terdapat data transaksi baru yang ditambahkan. Fitur ini hanya dapat diakses oleh Owner/Admin pada sistem. Desain sistem yang lebih detail untuk menampilkan peramalan digambarkan melalui *scenario*. *Scenario* menampilkan peramalan menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan peramalan seperti pada tabel 4.9

Tabel 4. 9 Menampilkan peramalan

|                                |   |
|--------------------------------|---|
| <b>ID Skenario</b>             | <b>USC7</b>                                     |
| Usecase                        | Menampilkan Peramalan                           |
| Partisipan Aktor               | Owner   |
| Pre-Kondisi                    | Aktor sudah melakukan login.                    |
| Pra-Kondisi                    | Owner selesai menampilkan peramalan pada sistem |
| <b>SKENARIO UTAMA</b>          |   |
| <b>“Menambahkan Peramalan”</b> |   |

| Aksi Aktor              | Reaksi Sistem   |
|-------------------------|---|
| 1. Klik menu “forecast” | 2. Menampilkan halaman peramalan dan hasil peramalan pada bulan selanjutnya |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan peramalan merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan peramalan. Penjelasan *activity diagram* fitur menampilkan peramalan dapat dilihat pada gambar 4.15

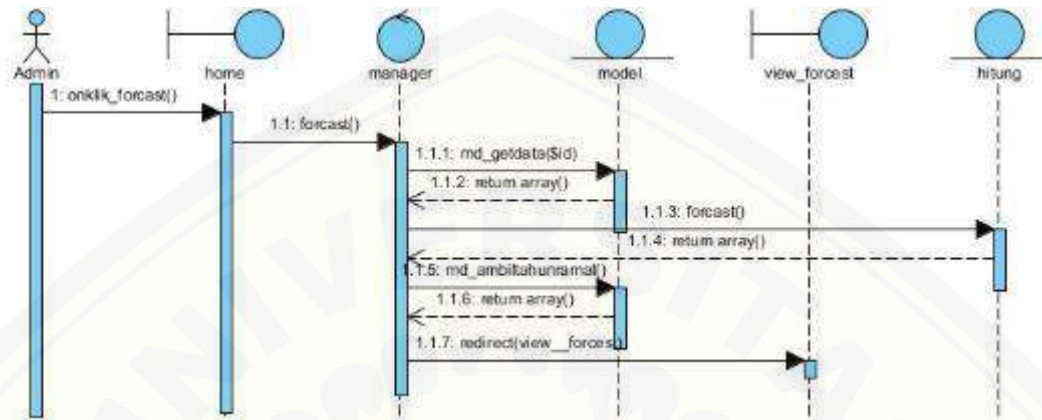


Gambar 4. 15 Activity Diagram menampilkan peramalan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan peramalan menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan



peramalan. Penjelasan *sequence diagram* fitur menampilkan peramalandapat dilihat pada gambar 4.16.



Gambar 4. 16Sequence Diagram menampilkan peramalan

#### 8) Menampilkan Detail Peramalan

Menampilkan detail peramalan merupakan fitur untuk menampilkan secara detail peramalan untuk periode selanjutnya pada setiap menu makanan ataupun minuman. Fitur ini menampilkan hasil peramalan beserta nilai MAPE (*Mean Absolute Percentage Error*), apabila nilai MAPE berada dikisaran 10% maka peramalan tersebut sangat akurat, apabila dibawah 20% maka peramalan itu masih tergolong bagus dan apabila lebih dari 20% maka peramalan tersebut tidak akurat. Fitur ini hanya dapat diakses oleh Owner/Admin pada sistem. Desain sistem yang lebih detail untuk menampilkan detail peramalan digambarkan melalui *scenario*. *Scenario* menampilkan detail peramalan menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan detail peramalan seperti pada tabel 4.10

Tabel 4. 10Scenario menampilkan detail peramalan

| ID Skenario | USC8                         |
|-------------|------------------------------|
| Usecase     | Menampilkan Detail Peramalan |



|                  |  |
|------------------|--|
| Partisipan Aktor | Owner  |
| Pre-Kondisi      | Owner sudah melakukan login.                                   |
| Pra-Kondisi      | Owner selesai melihat detail peramalan Setiap Menu pada system |

### SKENARIO UTAMA

#### “Menampilkan Detail Peramalan”

| Aksi Aktor  | Reaksi Sistem   |
|---|---|
| 1. Klik menu “forecast”   |   |
|   | 2. Menampilkan halaman peramalan dan hasil peramalan pada bulan selanjutnya |
| 3. Klik tombol “detail forecast”  |   |
|   | 4. Menampilkan pop up daftar menu makanan dan minuman.                      |
| 5. Pilih menu makanan atau minuman yang akan dilihat detail peramalannya. |   |
| 6. Klik prediksi  |   |
|   | 7. Menampilkan detail prediksi untuk menu yang dipilih.                     |

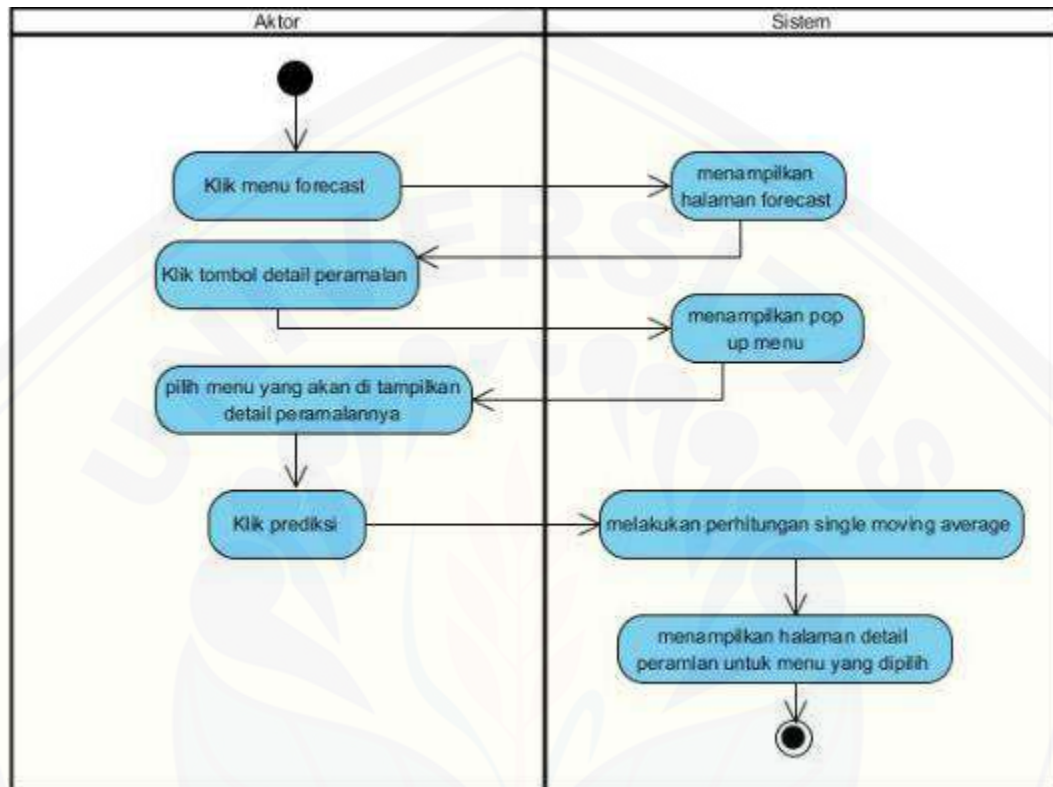
### SKENARIO ALTERNATIF

#### “Batal MenampilkanDetail Peramalan”

|                     |                                |
|---------------------|--------------------------------|
| 6 Klik tombol close |                                |
|                     | 7 kembali ke halaman peramalan |

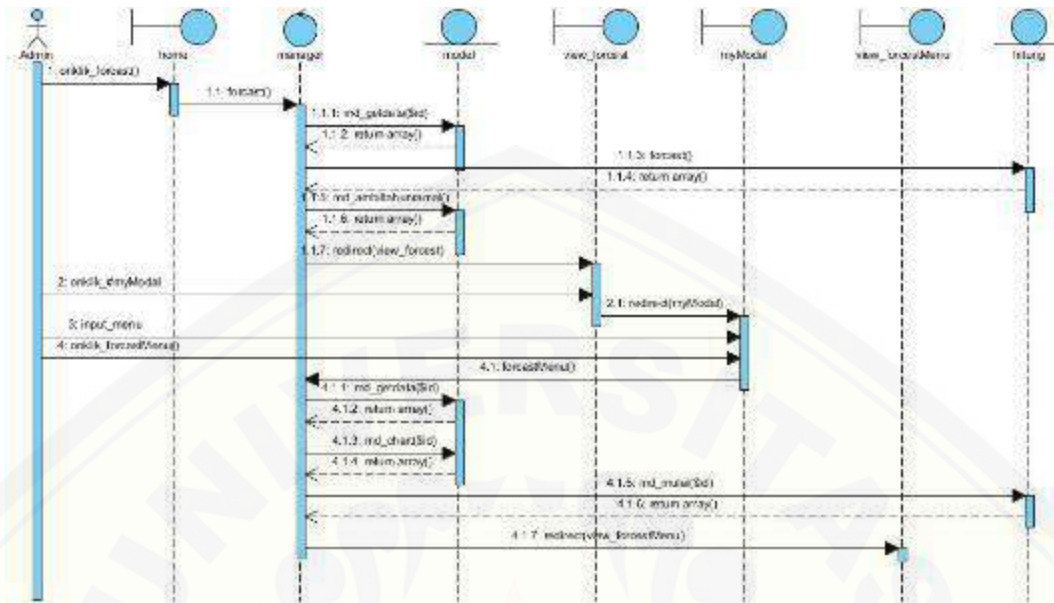
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan detail peramalan merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan detail

peramalan. Penjelasan *activity diagram* fitur menampilkan detail peramalan dapat dilihat pada gambar 4.17



Gambar 4. 17Activity Diagram menampilkan detail peramalan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan detail peramalan menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan detail peramalan. Penjelasan *sequence diagram* fitur menampilkan detail peramalan dapat dilihat pada gambar 4.18



Gambar 4. 18Sequence Diagram menampilkan detail peramalan

9) MenampilkanKebutuhan Ekonomis

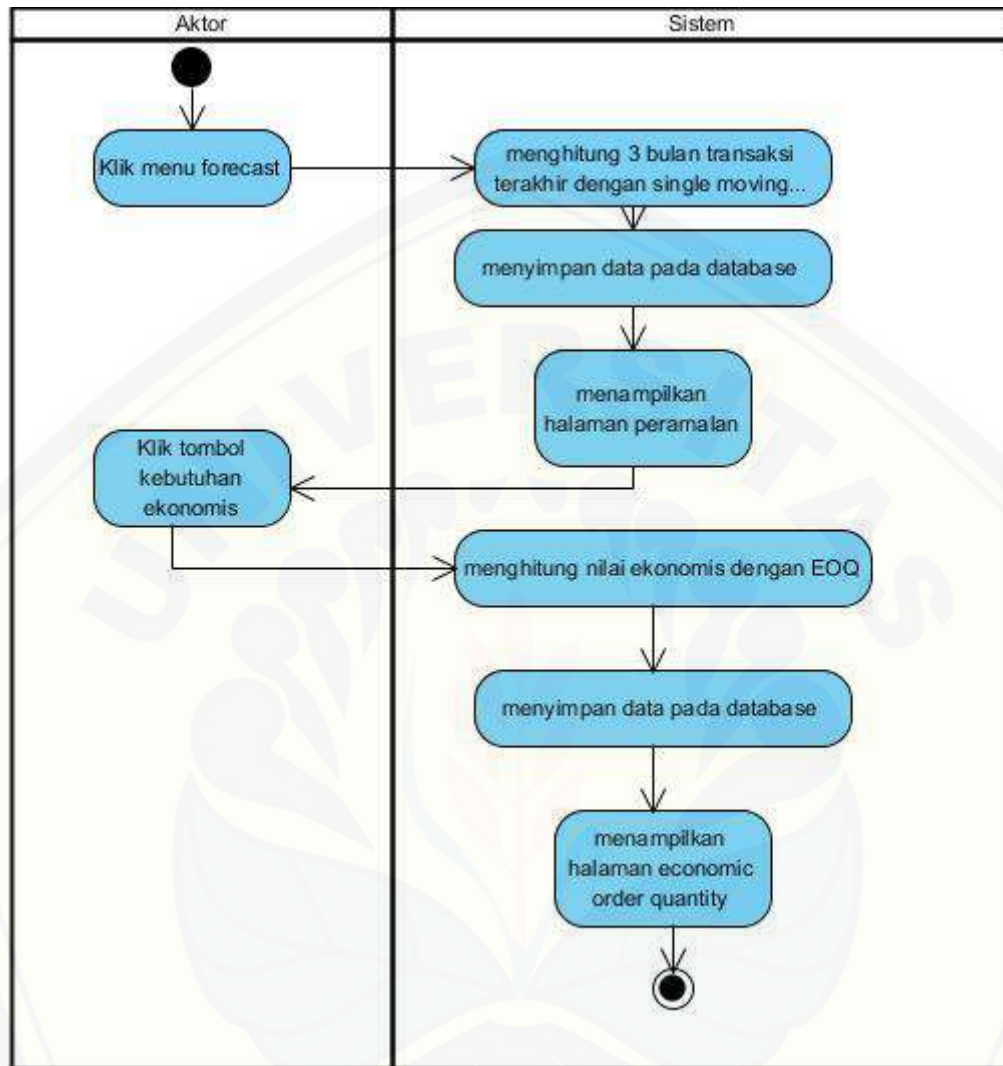
Menampilkankebutuhan Ekonomis merupakan fitur untuk menampilkan kebutuhan bahan baku untuk periode selanjutnya. Fitur ini menampilkan hasil perhitungan menggunakan metode *Economic Order Quantity* (EOQ) dan nilai yang dihasilkan yaitu nilai EOQ, *safety stok*(SS), dan *reorder point*(ROP) dan tersusun pada sebuah tabel dengandaftar bahan baku yang digunakan.Fitur ini hanya dapat diakses oleh admin/owner pada sistem. Desain sistem yang lebih detail untuk menampilkankebutuhan ekonomisdigambarkan melalui *scenario*. *Scenario* menampilkankebutuhan ekonomismenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkankebutuhan ekonomisseperti pada tabel 4.11

Tabel 4. 11Scenario menampilkan kebutuhan ekonomis

|                    |                                |
|--------------------|--------------------------------|
| <b>ID Skenario</b> | <b>USC9</b>                    |
| Usecase            | Menampilkan Kebutuhan Ekonomis |

|   |   |
|---|---|
| Partisipan Aktor                        | Owner   |
| Pre-Kondisi                             | Owner sudah melakukan login.  |
| Pra-Kondisi                             | Owner selesai melihat detail Kebutuhan Ekonomis untuk Periode Selanjutnya   |
| <b>SKENARIO UTAMA</b>                   |   |
| <b>“Menampilkan Kebutuhan Ekonomis”</b> |   |
| <b>Aksi Aktor</b>                       | <b>Reaksi Sistem</b>  |
| 1. Klik menu “Forecast”                 | 2. Menampilkan halaman peramalan dan hasil peramalan pada bulan selanjutnya |
| 3. Klik tombol “Kebutuhan Ekonomis”     | 4. Menampilkan Halaman Kebutuhan Ekonomis untuk periode bulan selanjutnya.  |

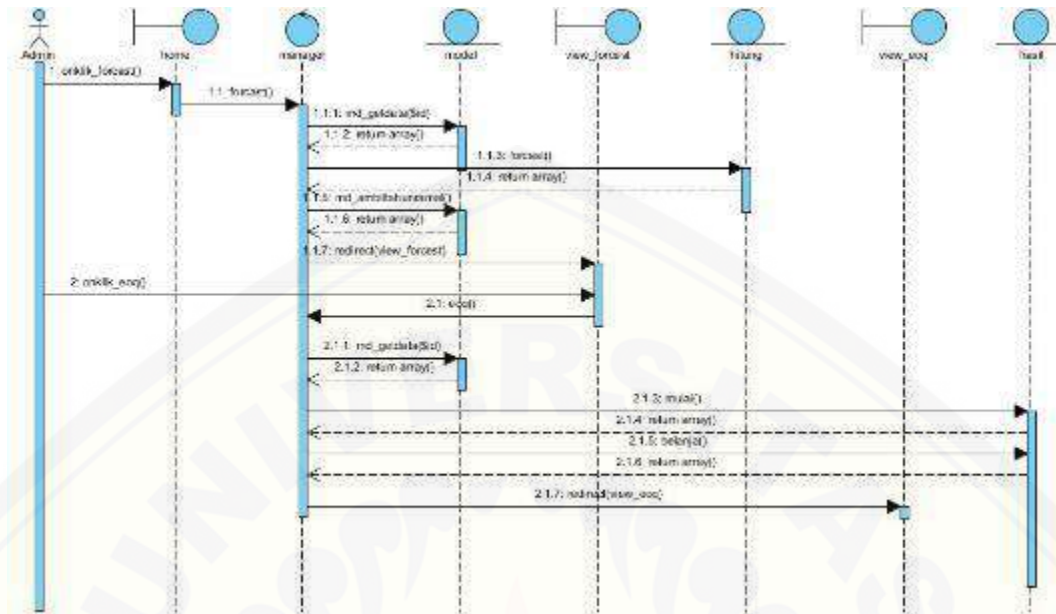
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan kebutuhan ekonomis merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan kebutuhan ekonomis. Penjelasan *activity diagram* fitur menampilkan kebutuhan ekonomis dapat dilihat pada gambar 4.19



Gambar 4. 19Activity Diagram menampilkan kebutuhan ekonomis

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan kebutuhan ekonomis menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan kebutuhan ekonomis. Penjelasan *sequence diagram* fitur menampilkan kebutuhan ekonomis dapat dilihat pada gambar 4.20





Gambar 4. 20Sequence Diagram menampilkan kebutuhan ekonomis

10) Menampilkan Daftar Belanja

Menampilkandaftar belanja merupakan fitur untuk menampilkan daftar belanja untuk periode selanjutnya. Fitur ini menampilkan hasil olah perhitungan dengan metode *Economic Order Quantity* (EOQ) dan diimplementasikan pada bahan baku yang tersedia di cafe. Fitur ini hanya dapat diakses oleh admin/owner pada sistem. Desain sistem yang lebih detail untuk menampilkandaftar belanjadigambarkan melalui *scenario*. *Scenario* menampilkandaftar belanjamenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkandaftar belanjaseperti pada tabel 4.12

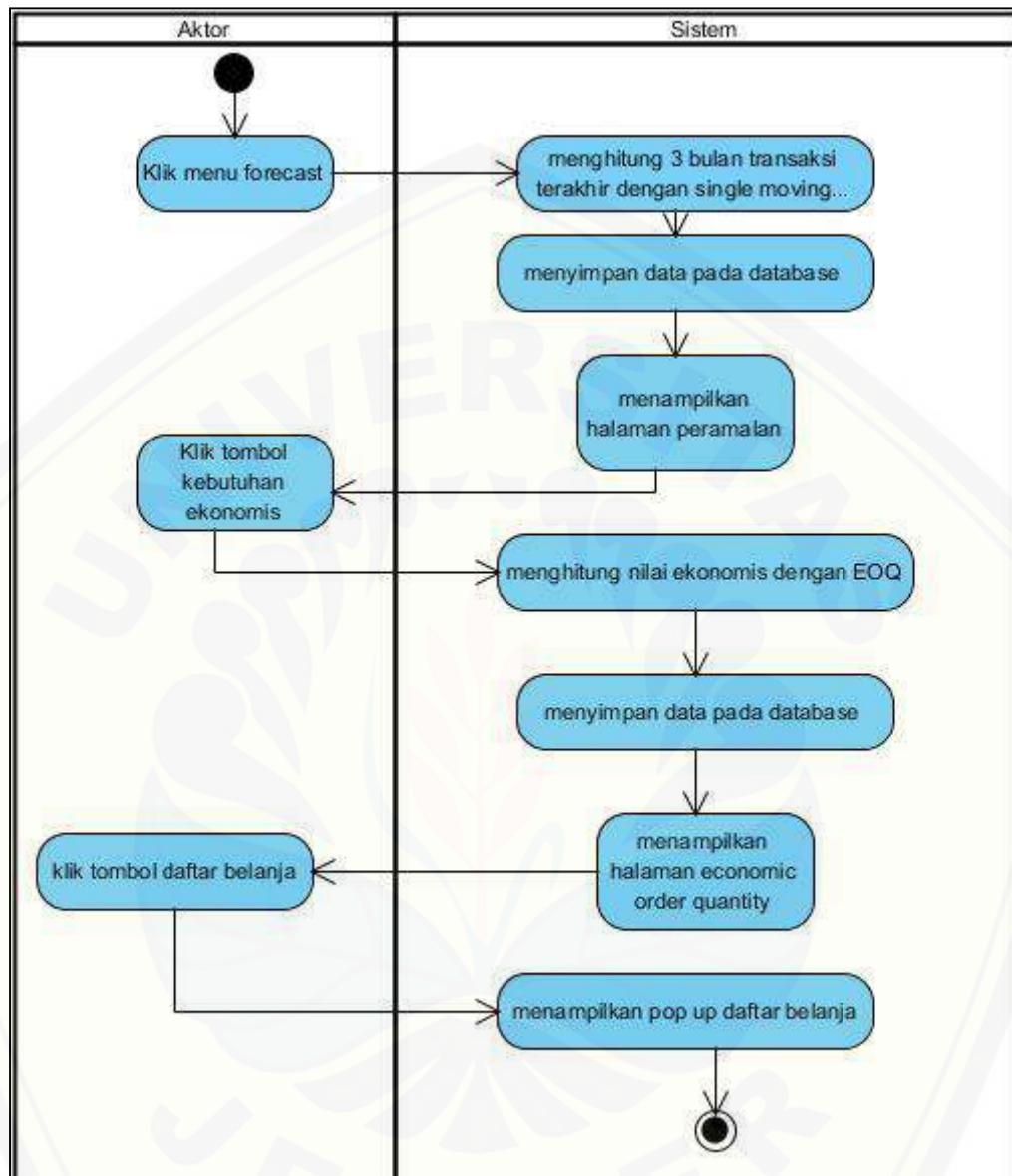
Tabel 4. 12Scenario menampilkan daftar belanja

|                    |  |
|--------------------|--|
| <b>ID Skenario</b> | <b>USC10</b>                                       |
| Usecase            | Menampilkan Daftar Belanja                         |
| Partisipan Aktor   | Owner  |
| Pre-Kondisi        | Owner sudah melakuakan login.                      |
| Pra-Kondisi        | Owner selesai melihat daftar belanja untuk Periode |



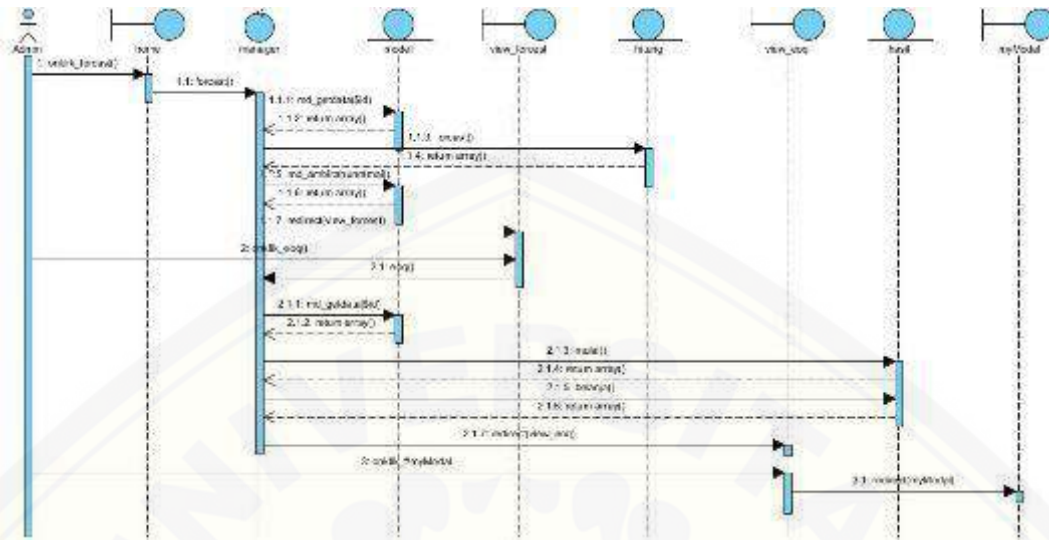
| Selanjutnya                         |   |
|-------------------------------------|---|
| SKENARIO UTAMA                      |   |
| “Menampilkan Daftar Belanja”        |   |
| Aksi Aktor                          | Reaksi Sistem   |
| 1. Klik menu “Forecast”             | 2. Menampilkan halaman peramalan dan hasil peramalan pada bulan selanjutnya |
| 3. Klik tombol “Kebutuhan Ekonomis” | 4. Menampilkan Halaman Kebutuhan Ekonomis untuk periode bulan selanjutnya.  |
| 5. Klik tombol Daftar belanja       | 6. Menampilkan <i>pop up</i> daftar belanja untuk periode selanjutnya.      |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan daftar belanja merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan daftar belanja. Penjelasan *activity diagram* fitur menampilkan daftar belanja dapat dilihat pada gambar 4.21



Gambar 4. 21 Activity Diagram menampilkan daftar belanja

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan daftar belanja menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan daftar belanja. Penjelasan *sequence diagram* fitur menampilkan daftar belanja dapat dilihat pada gambar 4.22



Gambar 4. 22Sequence Diagram menampilkan daftar belanja

11) Menampilkan Keuangan

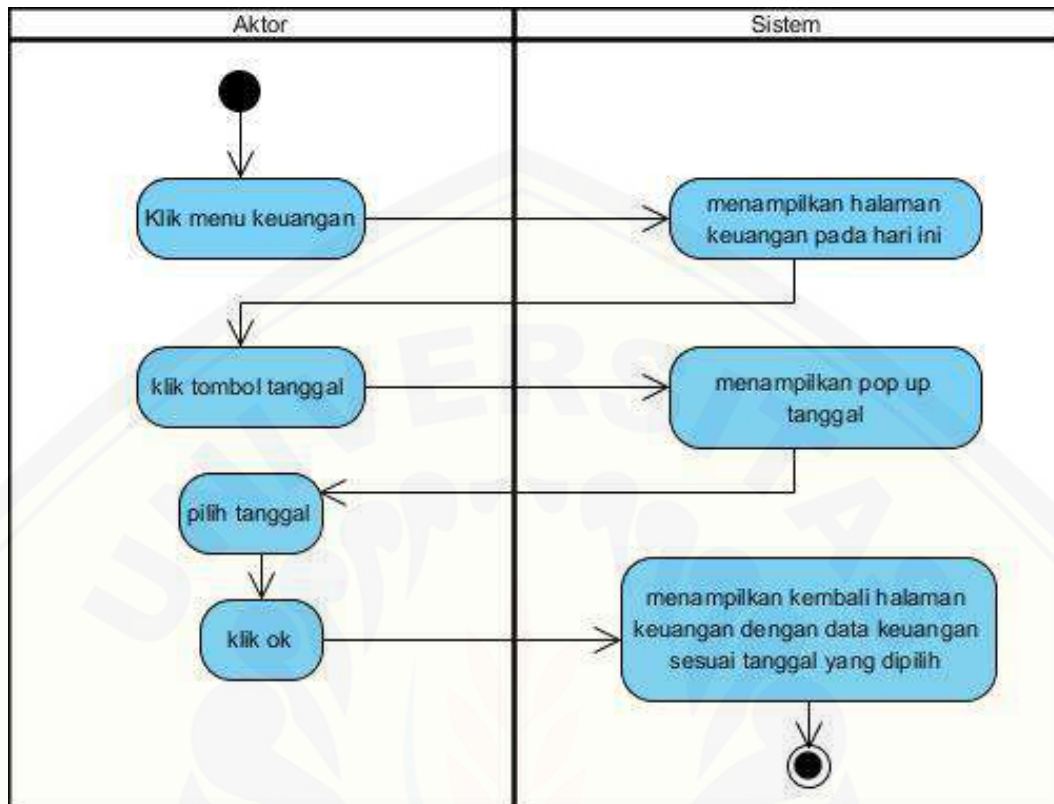
Menampilkan keuangan merupakan fitur untuk menampilkan nilai keuangan pada hari itu dan tanggal yang diinginkan, fitur ini hanya dapat diakses oleh owner.admin. Desain sistem yang lebih detail untuk menampilkan keuangandigambarkan melalui *scenario*. *Scenario* menampilkan keuangan menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkankeuangan seperti pada tabel 4.13

Tabel 4. 13Scenario menampilkan keuangan

| ID Skenario           | USC11   |
|-----------------------|---|
| Usecase               | Menampilkan Keuangan                                      |
| Partisipan Aktor      | Owner   |
| Pre-Kondisi           | Owner sudah melakuakan login.                             |
| Pra-Kondisi           | Owner telah melihat hasil penjualan pada tanggal tertentu |
| <b>SKENARIO UTAMA</b> |   |
| <b>“Keuangan”</b>     |   |

| Aksi Aktor   | Reaksi Sistem  |
|--|--|
| 1. Klik menu keuangan                                  |  |
|  | 2. Menampilkan halaman keuangan pada hari itu                                    |
| 3. Klik tombol pilih tanggal                           |  |
|  | 4. Menampilkan pop up tanggal  |
| 5. Pilih tanggal yang akan ditampilkan hasil penjualan |  |
| 6. Klik ok   |  |
|  | 7. Menampilkan halaman keuangan dengan data keuangan sesuai tanggal yang dipilih |

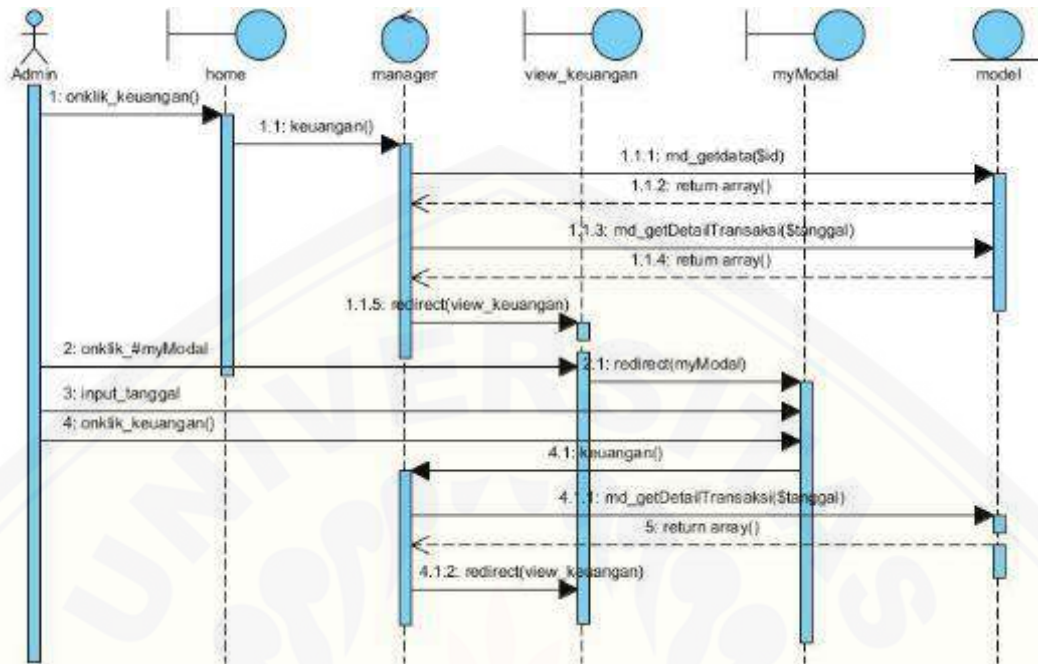
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan keuangan merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan keuangan. Penjelasan *activity diagram* fitur menampilkan keuangan dapat dilihat pada gambar 4.23



Gambar 4. 23Activity Diagram menampilkan keuangan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan keuangan menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan keuangan. Penjelasan *sequence diagram* fitur menampilkan keuangan dapat dilihat pada gambar 4.24





Gambar 4. 24Sequence Diagram menampilkan keuangan

12) Menambahkan data menu

Menambahkan data Menu merupakan fitur untuk menambahkan menu yang akan dijual pada perusahaan. Dalam fitur menambahkan data menu dibagi menjadi 2 yaitu menambahkan data menu makanan dan menambahkan data minuman, fitur ini hanya dapat diakses oleh *Admin/owner* dalam menambahkan menu yang akan dijual oleh perusahaan. Menambahkan data menu mengisi form nama menu makanan/minuman dan harga jual. Desain sistem yang lebih detail untuk *menambahkan data menu* digambarkan melalui *scenario*. *Scenario* menambahkan data biodata menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan data menu seperti pada tabel 4.14

Tabel 4. 14Tabel menambahkan data menu

| ID Skenario | USC12            |
|-------------|------------------|
| Usecase     | Menambahkan Menu |

|                  |   |
|------------------|---|
| Partisipan Aktor | Owner   |
| Pre-Kondisi      | Owner sudah melakukan login.  |
| Pra-Kondisi      | Owner menambah Menu makanan ataupun minuman pada sistem menambah Menu makanan ataupun minuman dan tersimpan pada sistem.. |

**SKENARIO UTAMA**  
**“Menambahkan Menu”**

| Aksi Aktor   | Reaksi Sistem   |
|--|---|
| 1. Klik menu “Menu” makanan ataupun minuman.   | 2. Menampilkan halaman menu makanan apabila memilih makanan dan halaman menu minuman apabila memilih menu minuman.                  |
| 3. Klik tombol tambah data.  | 4. Menampilkan Halaman tambah menu makanan atau minuman roots cafe.   |
| 5. Mengisi nama menu makanan atau minuman yang akan ditambahkan dan harga yang diperjualkan. |   |
| 6. Klik tombol Submit  | 7. Menampilkan halaman menu makanan apabila memilih makanan dan halaman menu minuman apabila memilih menu minuman dengan notifikasi |

---

“menu baru berhasil  
ditambahkan”

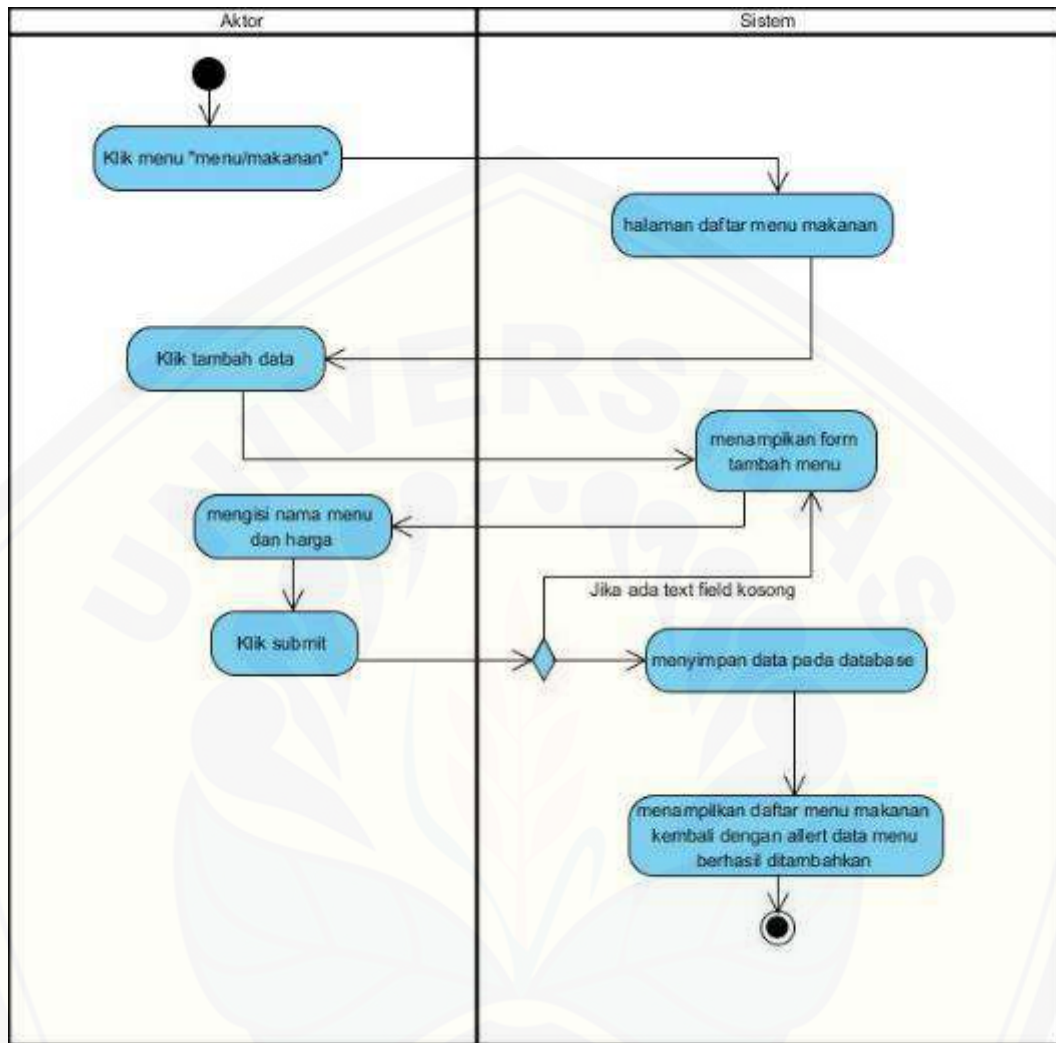
---

### SKENARIO ALTERNATIF

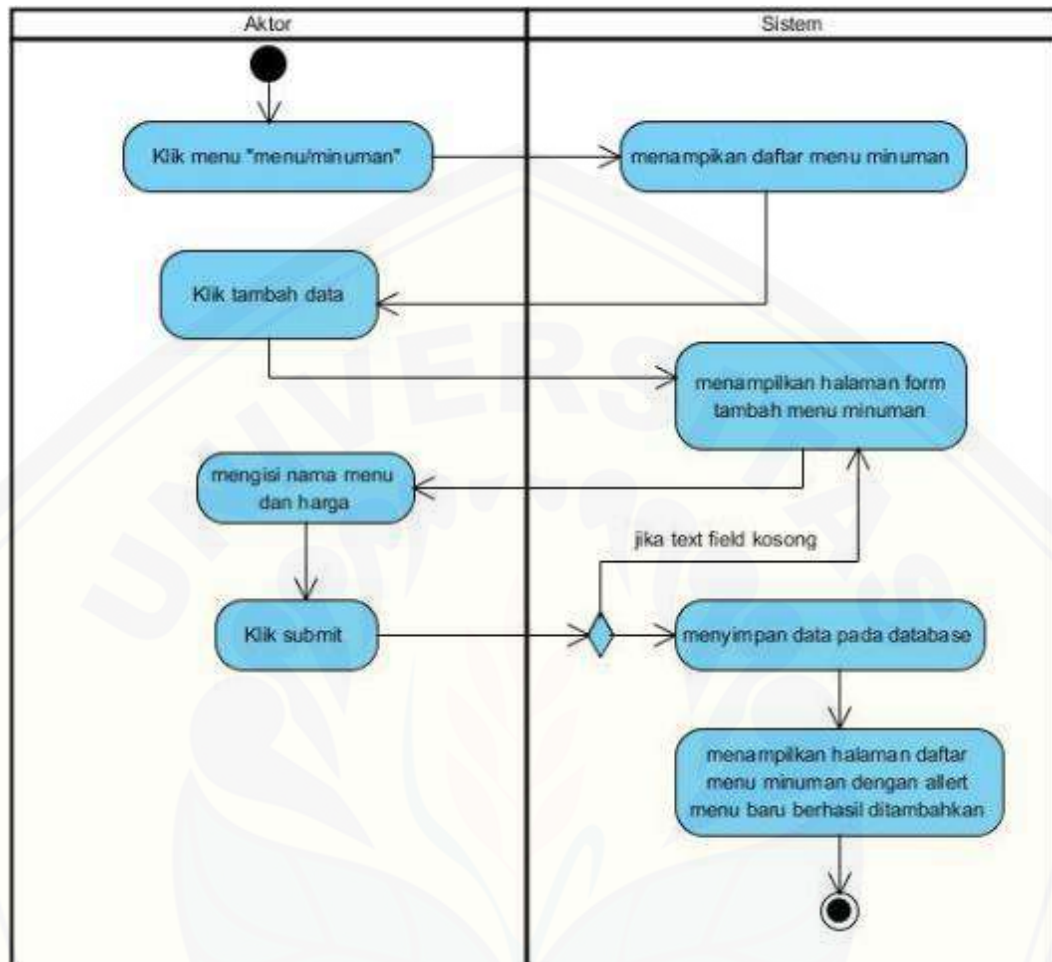
#### “Menambahkan Menu Dengan Field Kosong”

| Aksi Aktor  | Reaksi Sistem   |
|---|---|
| 5. Mengisi nama menu makanan atau minuman yang akan ditambahkan, dan form harga di kosongi. |   |
|   | 7. Menampilkan Halaman tambah menu makanan atau minuman roots cafe dengan alert “please fill out this field.” |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menambahkan data menumerupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan data biodata. Penjelasan *activity diagram* fitur menambahkan data menu dapat dilihat pada gambar 4.25 untuk data menu makanan gambar 4.26 untuk data menu minuman.



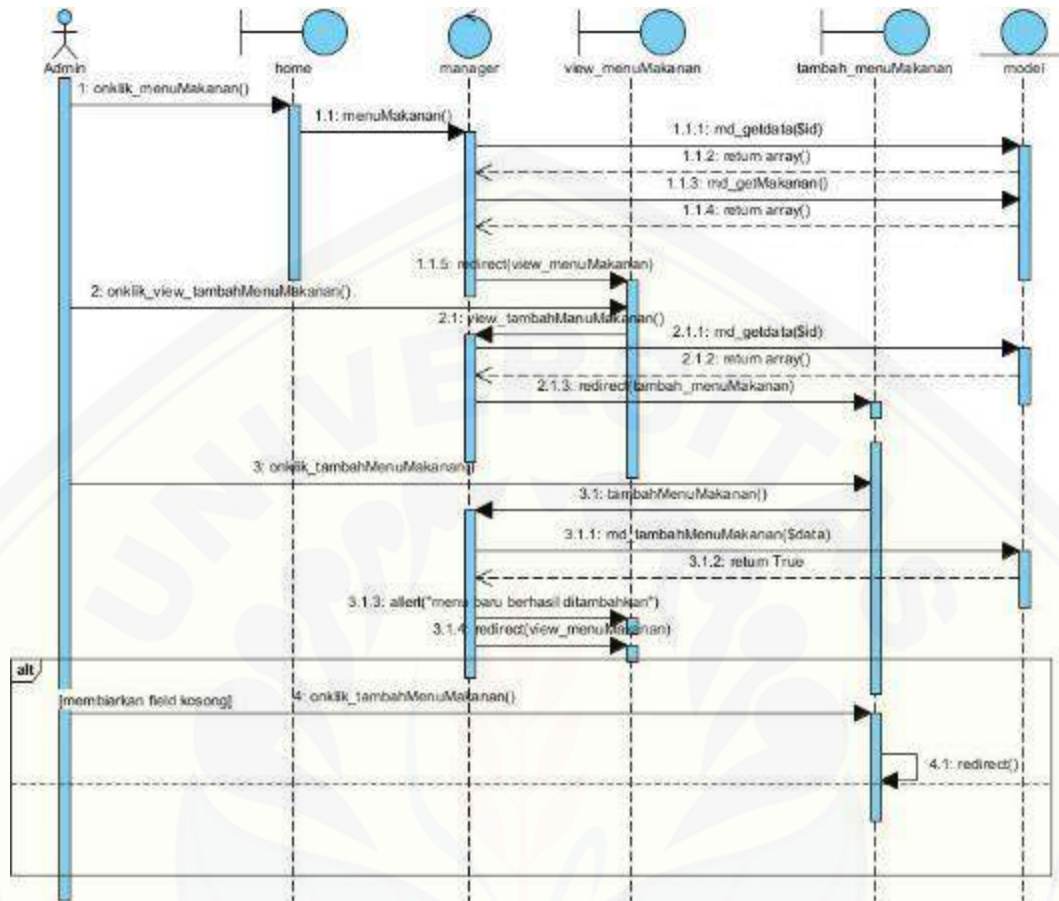
Gambar 4. 25 Activity Diagram menambahkan menu makanan



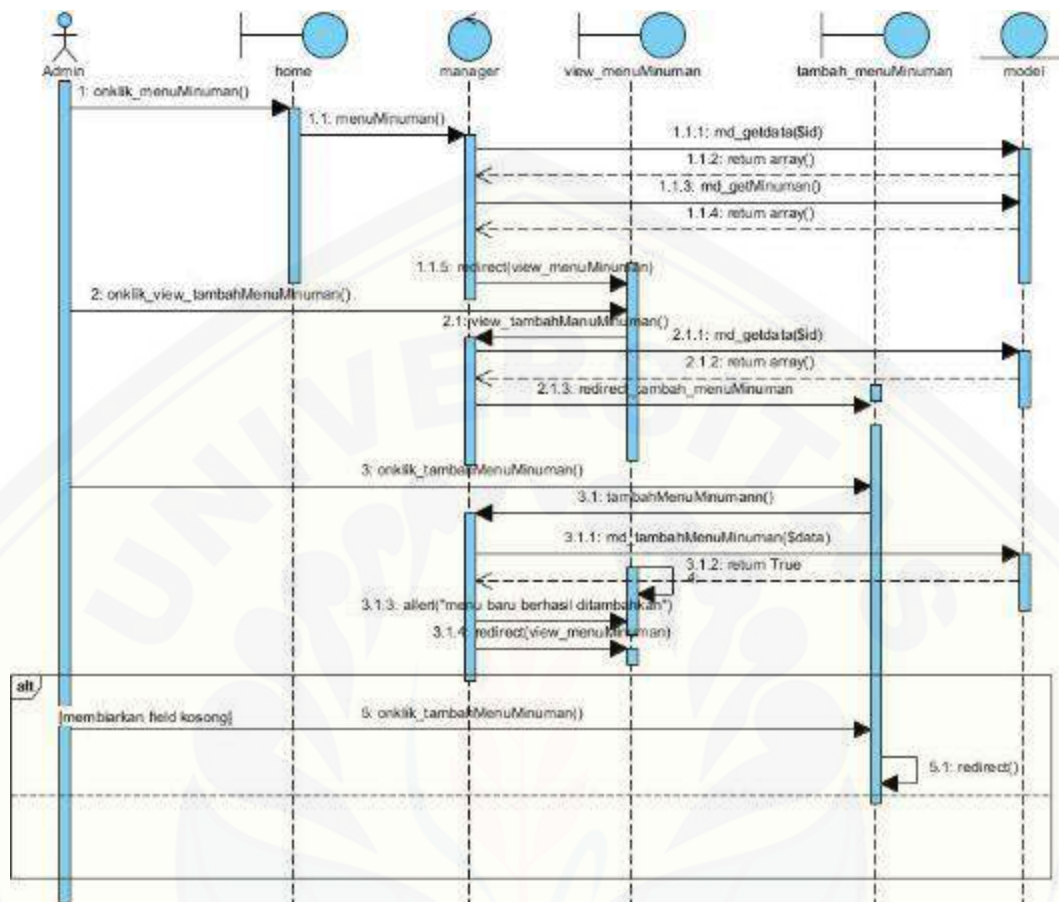
Gambar 4. 26Activity Diagram menambahkan menu minuman

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan data menu menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data menu. Penjelasan *sequence diagram* fitur menambahkan data menu dapat dilihat pada gambar 4.21 untuk menu makanan dan gambar 4.22 untuk menu minuman.





Gambar 4. 27Sequence Diagram menambahkan menu makanan



Gambar 4. 28Sequence Diagram menambahkan menu minuman

### 13) Merubah data Menu

Merubah data menu merupakan fitur untuk merubah informasi menu makanan/minuman yang dijual diperusahaan yang telah tersimpan pada sistem, fitur ini berisi *form* nama menu dan harga dengan data yang tersimpan sebelumnya yang akan dirubah sesuai keinginan aktor. Fitur ini hanya dapat diakses oleh *Admin/Owner* pada sistem. Desain sistem yang lebih detail untuk merubah data menu digambarkan melalui *scenario*. *Scenario* merubah data menu menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur merubah data menu seperti pada tabel 4.15

Tabel 4. 15Scenario merubah data menu

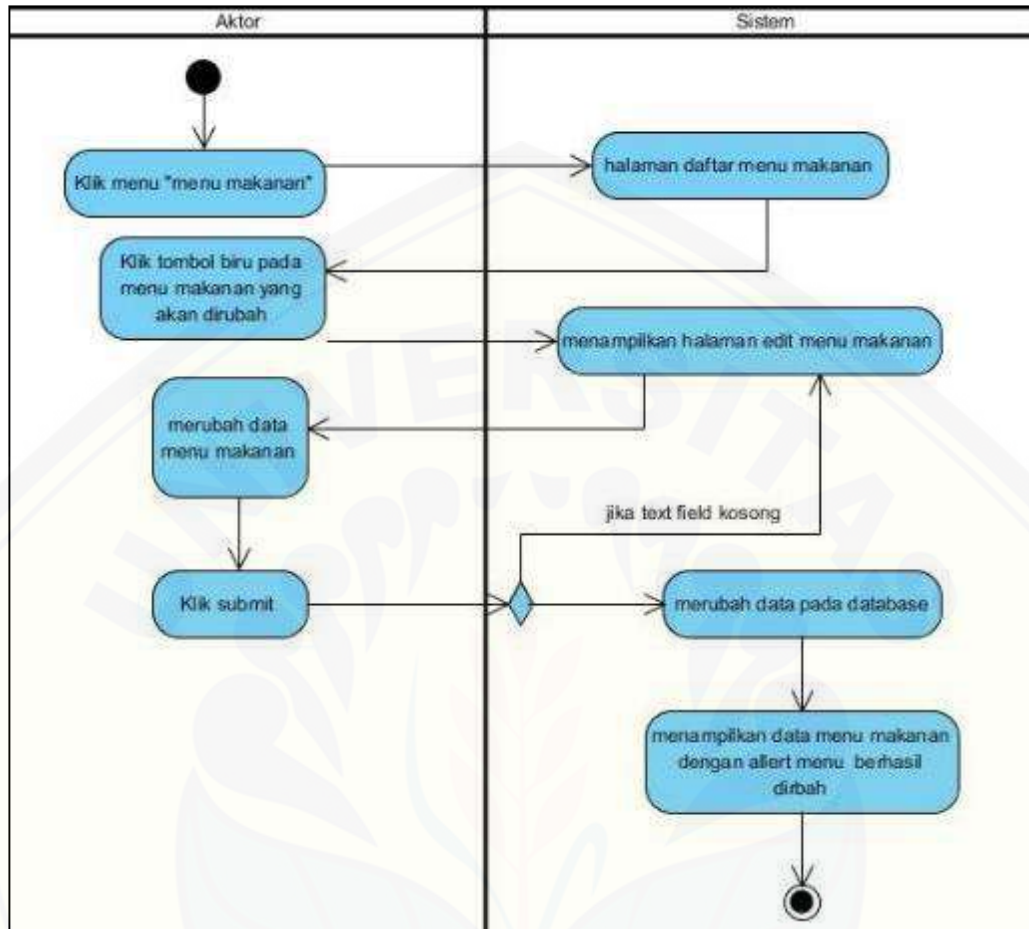
| <b>ID Skenario</b>  | <b>USC13</b>   |
|---|--|
| Usecase   | Merubah Menu   |
| Partisipan Aktor  | Owner  |
| Pre-Kondisi   | Owner sudah melakukan login.   |
| Pra-Kondisi   | Owner merubah Menu makanan ataupun minuman pada sistem menambah Menu makanan ataupun minuman dan tersimpan pada sistem.. |
| <b>SKENARIO UTAMA</b>   |  |
| <b>“Merubah Menu”</b>   |  |
| <b>Aksi Aktor</b>   | <b>Reaksi Sistem</b>   |
| 1. Klik menu “Menu” makanan ataupun minuman.                          | 2. Menampilkan halaman menu makanan apabila memilih makanan dan halaman menu minuman apabila memilih menu minuman.       |
| 3. Klik tombol berwarna biru pada menu yang akan dirubah              | 4. Menampilkan Halaman edit menu makanan atau minuman roots cafe.  |
| 5. Merubahnama menu makanan atau minuman dan harga yang diperjualkan. |  |
| 6. Klik tombol Submit   | 7. Menampilkan halaman menu  |

makanan apabila memilih makanan dan halaman menu minuman apabila memilih menu minuman dengan notifikasi “menu berhasil dirubah”

**SKENARIO ALTERNATIF**  
**“Merubah Menu Field Kosong”**

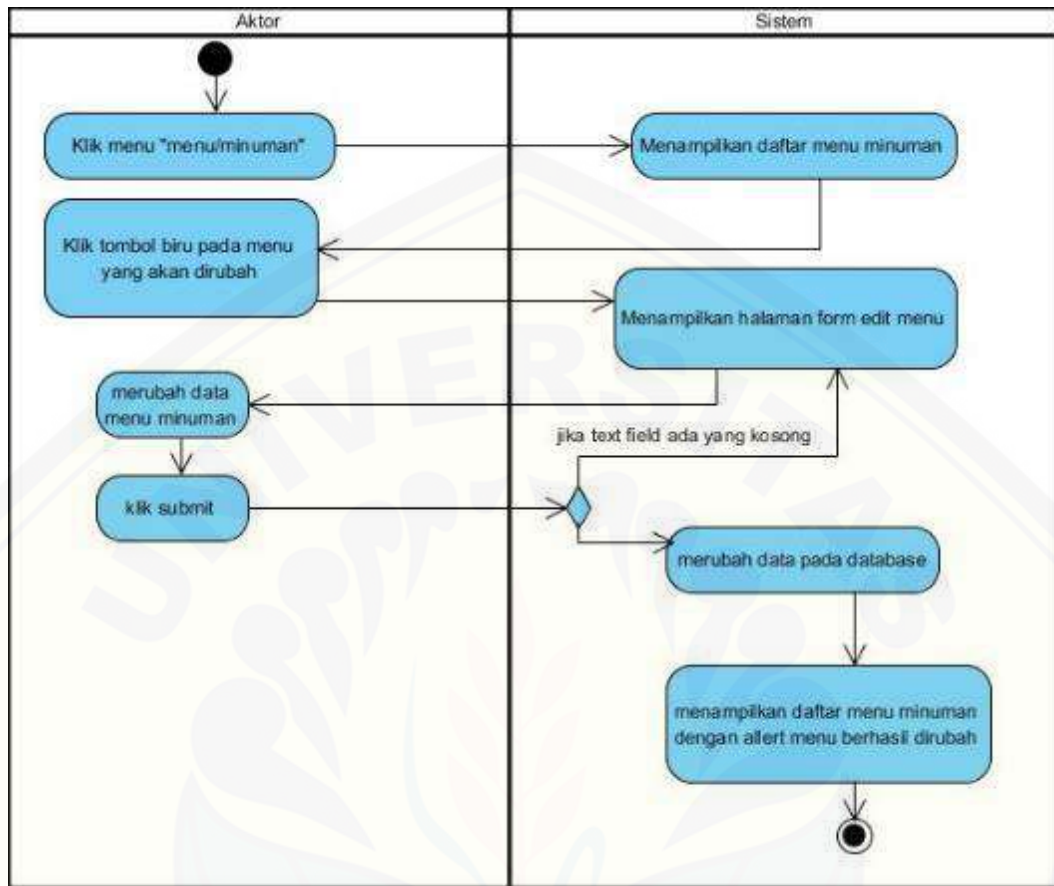
| Aksi Aktor  | Reaksi Sistem   |
|---|---|
| 5. Mengisi nama menu makanan atau minuman yang akan ditambahkan, dan form harga di kosongi. |   |
| 6. Klik tombol Submit   | 7. Menampilkan Halaman tambah menu makanan atau minuman roots cafe dengan alert “please fill out this field.” |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur merubah data menu merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses merubah data menu. Penjelasan *activity diagram* fitur merubah data menu dapat dilihat pada gambar 4.29 untuk menu makanan dan gambar 4.30 untuk menu minuman.



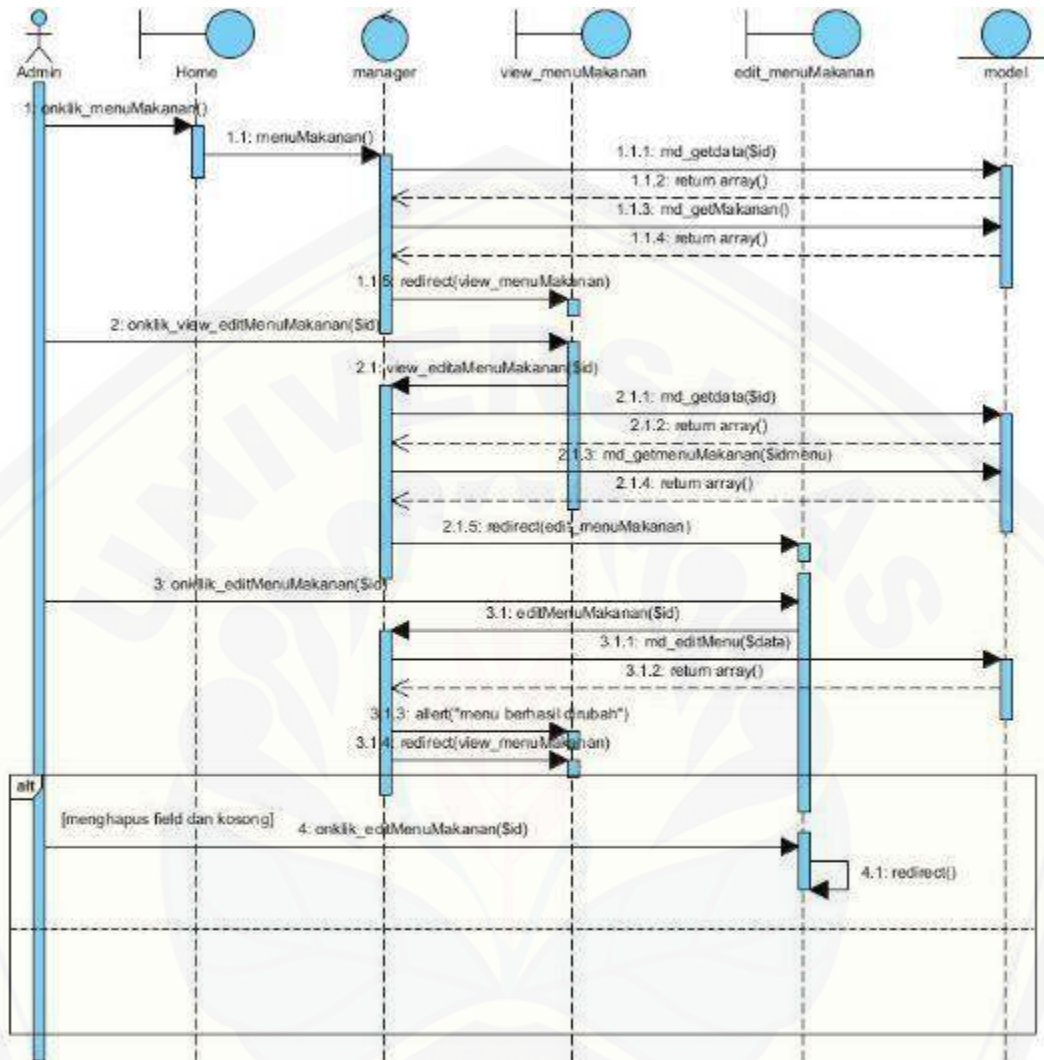
Gambar 4. 29Activity Diagram merubah data menu makanan



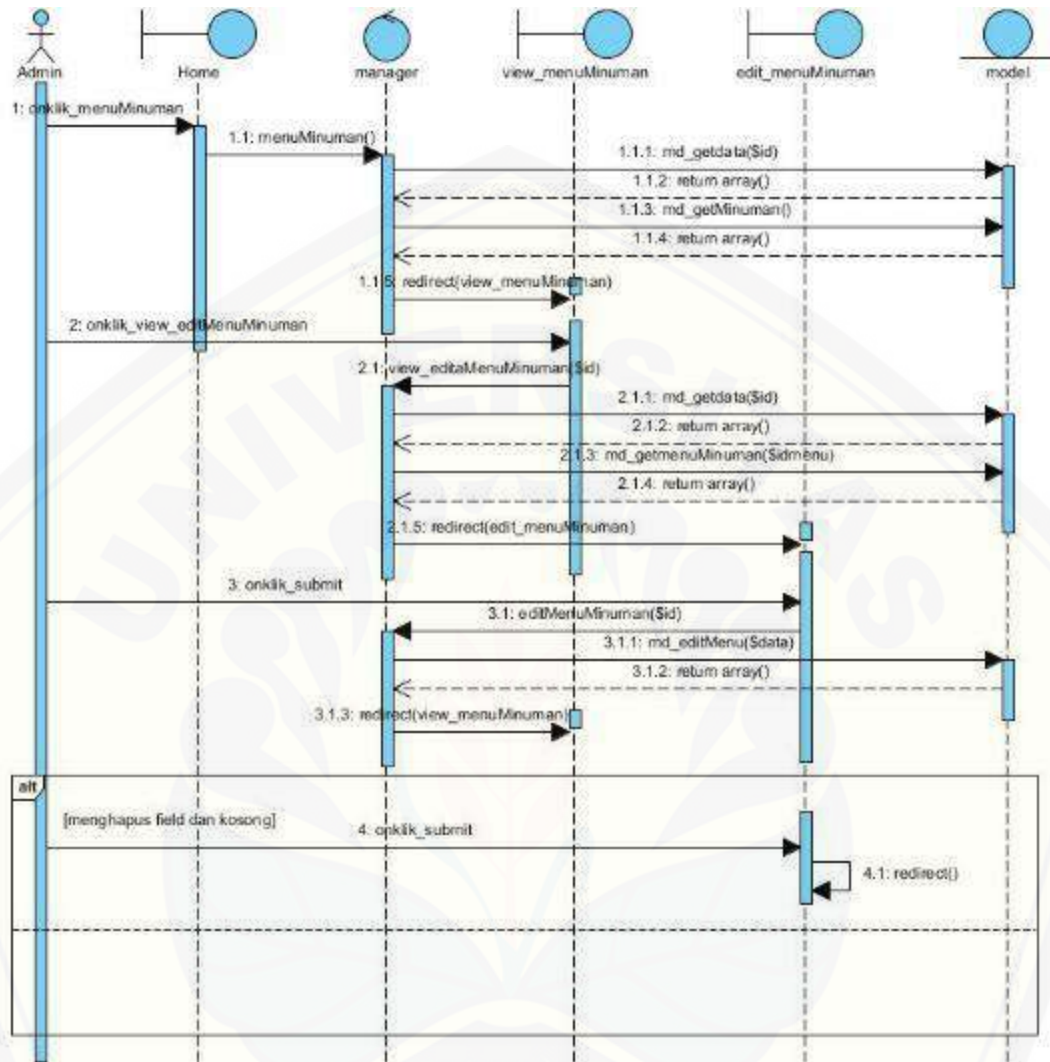


Gambar 4. 30Activity Diagram merubah data menu minuman

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Sequence diagram* fitur merubah data menumenggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah data menu. Penjelasan *sequence diagram* fitur merubah data menudapat dilihat pada gambar 4.31 untuk menu makanan dan gambar 4.32 untuk menu minuman



Gambar 4. 31Sequence Diagram merubah data menu makanan



Gambar 4. 32Sequence Diagram merubah data menu minuman

#### 14) Menghapus data Menu

Menghapus data menu merupakan fitur untuk menghapus menu makanan/minuman yang dijual perusahaan yang telah tersimpan pada sistem, fitur ini berupa tombol menghapus data (warna merah) yang terdapat pada tabel menu makanan/minuman pada hak akses *admin*. Fitur ini hanya dapat diakses oleh *Admin/Owner* pada sistem. Desain sistem yang lebih detail untuk menghapus data menu digambarkan melalui *scenario*. *Scenario* menghapus data menu menjelaskan

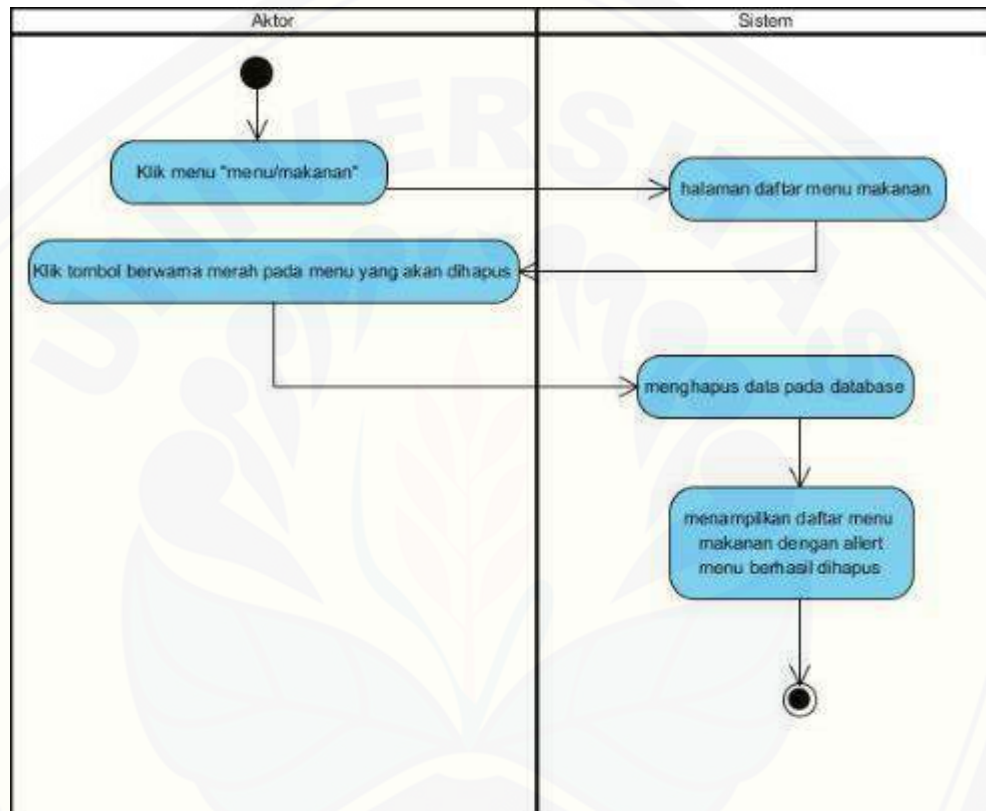
tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menghapus data menu seperti pada tabel 4.16

Tabel 4. 16Scenario menghapus data menu

|  |   |
|--|---|
| <b>ID Skenario</b>   | <b>USC14</b>  |
| Usecase  | Menghapus Menu  |
| Partisipan Aktor   | Owner   |
| Pre-Kondisi  | Owner sudah melakukan login.  |
| Pra-Kondisi  | Ownersudah menghapus menu makanan atau minuman.   |
| <b>SKENARIO UTAMA</b>  |   |
| <b>“Menghapus Menu”</b>  |   |
| <b>Aksi Aktor</b>  | <b>Reaksi Sistem</b>  |
| 1. Klik Menu “menu” dan pilih menu makanan atau minuman yang akan dihapus. | 2. Menampilkan Halaman menu makanan atau menu minuman   |
| 3. Klik tombol berwarna merah pada menu yang akan dihapus.                 | 4. Menampilkan Halaman menu makanan atau menu minuman dengan notifikasi “menu berhasil dihapus” |

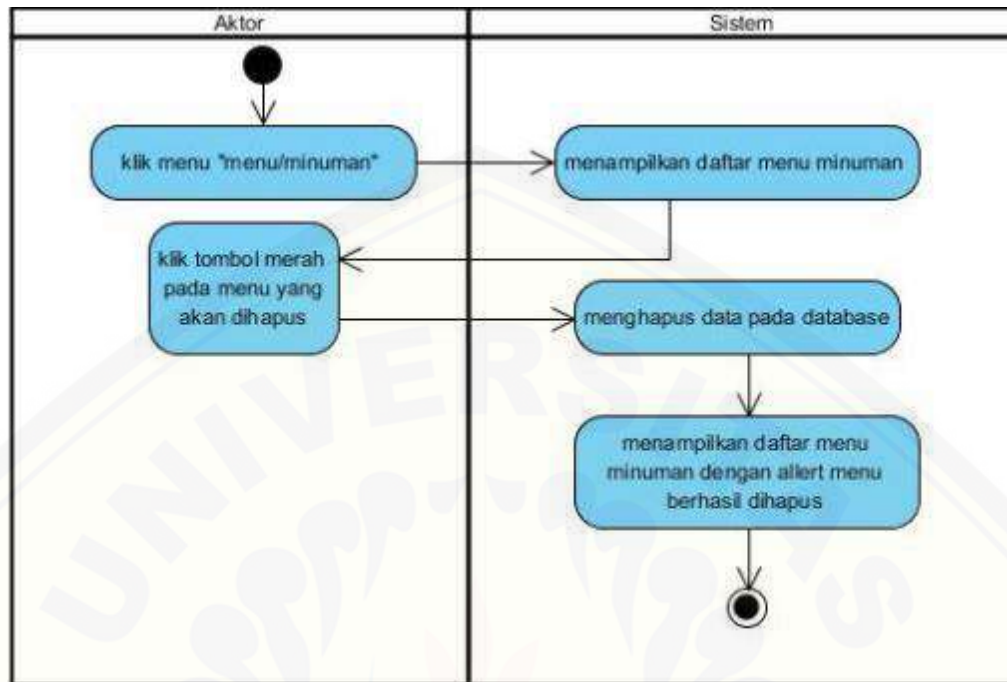
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menghapus data menumerupakan

gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menghapus data menu. Penjelasan *activity diagram* fitur menghapus data menu dapat dilihat pada gambar 4.33 untuk menu makanan dan gambar 4.34 untuk menu minuman.



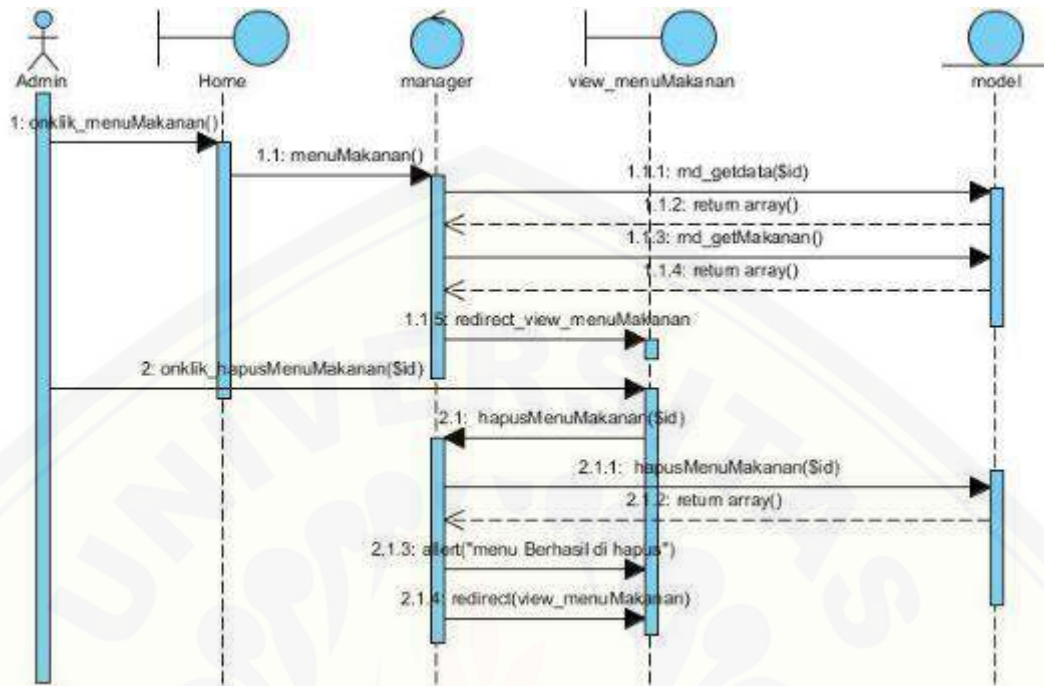
Gambar 4. 33Activity Diagram menghapus data menu makanan



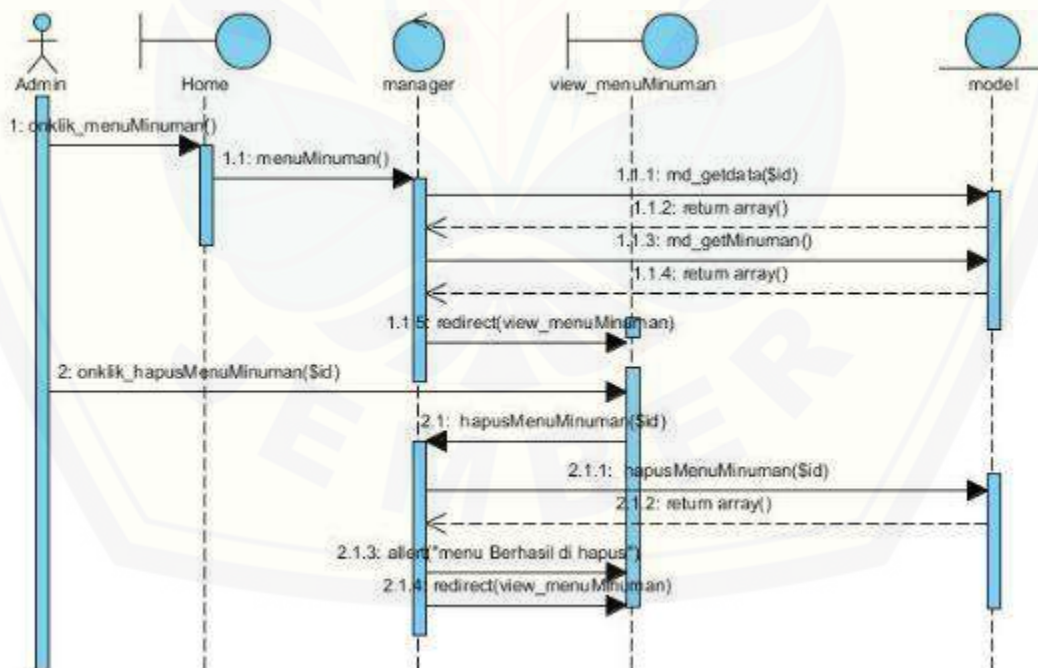


Gambar 4. 34Activity Diagram menghapus data menu minuman

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menghapus data menu menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menghapus data menu. Penjelasan *sequence diagram* fitur menghapus data menu dapat dilihat pada gambar 4.35 untuk menu makanan dan gambar 4.36 untuk menu minuman.



Gambar 4. 35Sequence Diagram menghapus data menu makanan



Gambar 4. 36Sequence Diagram menghapus data menu minuman

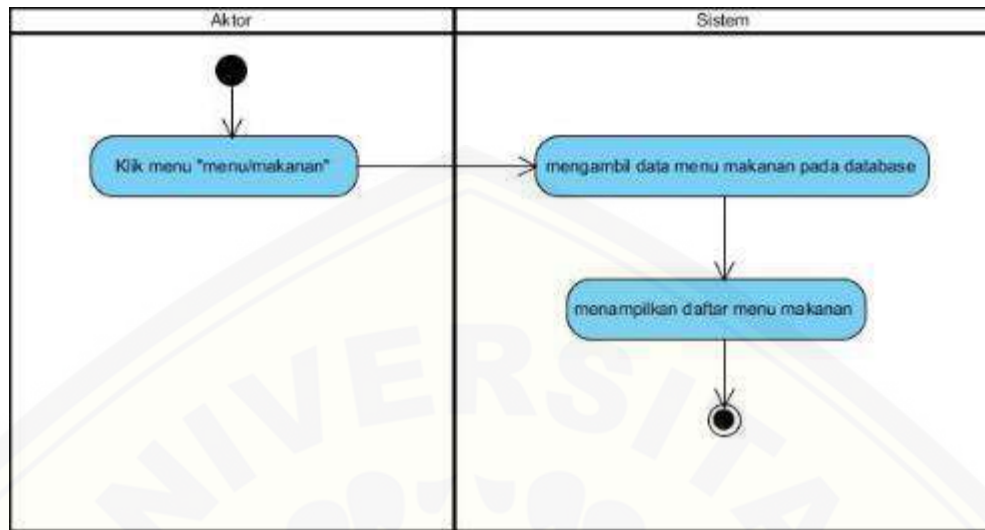
## 15) Menampilkan Menu

Menampilkan menu merupakan fitur untuk menampilkan menu makanan/minuman yang dijual diperusahaan yang telah tersimpan pada sistem, fitur ini berupa tabel menu makanan/minuman.. Fitur ini dapat diakses oleh *Admin/Owner* dan pegawai pada sistem. Desain sistem yang lebih detail untuk *Menampilkan menu* digambarkan melalui *scenario*. *Scenario* menampilkan menu menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan menu seperti pada tabel 4.17

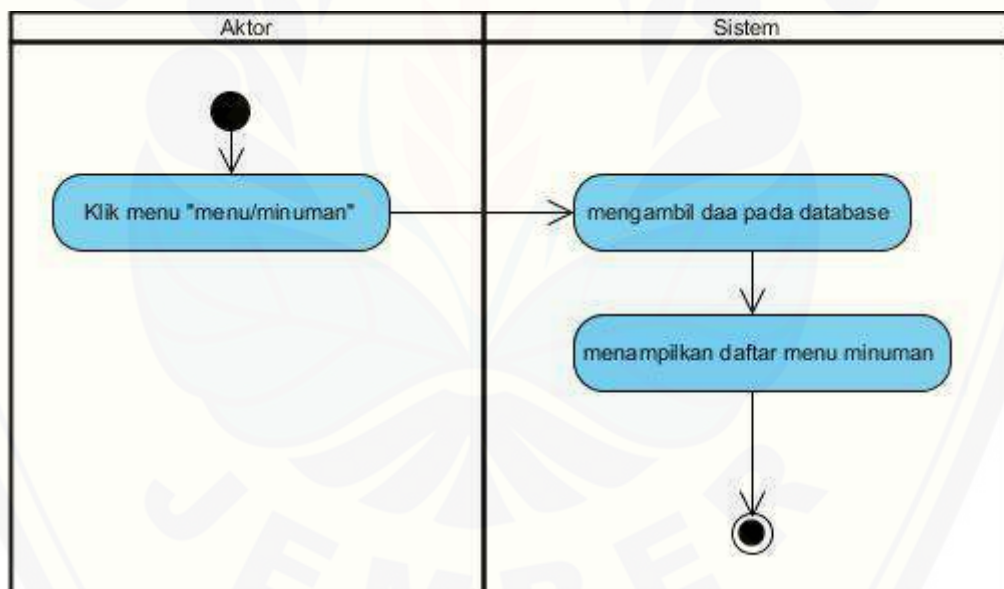
Tabel 4. 17 Scenario menampilkan menu

| ID Skenario  | USC15   |
|--|---|
| Usecase  | Menampilkan Menu                                      |
| Partisipan Aktor   | Owner   |
| Pre-Kondisi  | Owner sudah melakukan login.                          |
| Pra-Kondisi  | Owner sudah menampilkan menu makanan atau minuman.    |
| <b>SKENARIO UTAMA</b>                                    |   |
| <b>“Menampilkan Menu”</b>                                |   |
| Aksi Aktor   | Reaksi Sistem   |
| 1. Klik Menu “menu” dan pilih menu makanan atau minuman. | 2. Menampilkan Halaman menu makanan atau menu minuman |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan menu merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan menu. Penjelasan *activity diagram* fitur menampilkan menu dapat dilihat pada gambar 4.37 untuk menu makanan dan gambar 4.38 untuk menu minuman.



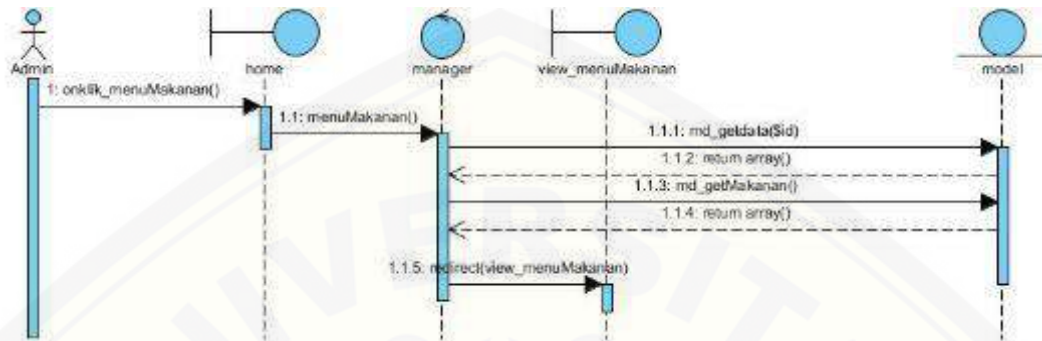
Gambar 4. 37 Activity Diagram menampilkan menu makanan



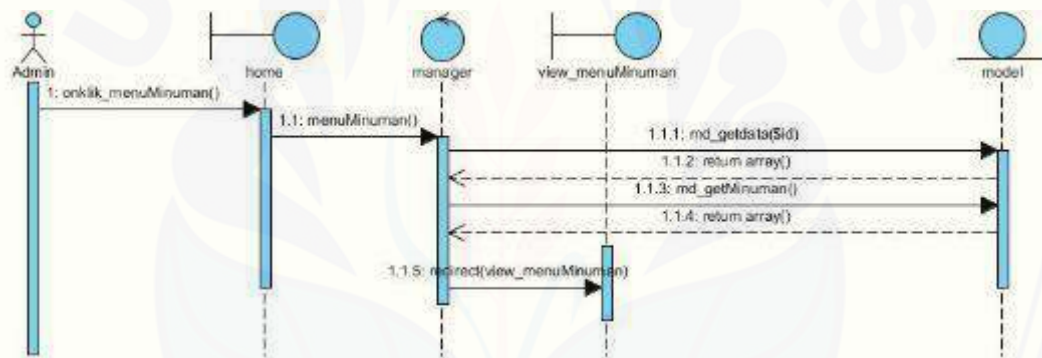
Gambar 4. 38 Activity Diagram menampilkan menu minuman

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan menu menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan

menu. Penjelasan *sequence diagram* fitur menampilkan menudapat dilihat pada gambar 4.39 untuk menu makanan dan gambar 4.40 untuk menu minuman



Gambar 4. 39Sequence Diagram menampilkan menu makanan



Gambar 4. 40Sequence Diagram menampilkan menu minuman

#### 16) Menampilkan total bayar setiap meja

Menampilkan total bayar setiap mejamerupakan fitur untuk menampilkan total yang harus dibayar oleh pemesan pada nomer meja tertentu, fitur ini sekalian mengaktifkan kembali nomer meja sehingga dapat dipesan oleh pengunjung selanjutnya. Fitur ini hanya dapat diakses oleh kasir pada sistem. Desain sistem yang lebih detail untuk menampilkan total bayar setiap mejadigambarkan melalui *scenario*. *Scenario* menampilkan total bayar setiap mejamenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan total bayar setiap meja seperti pada tabel 4.18

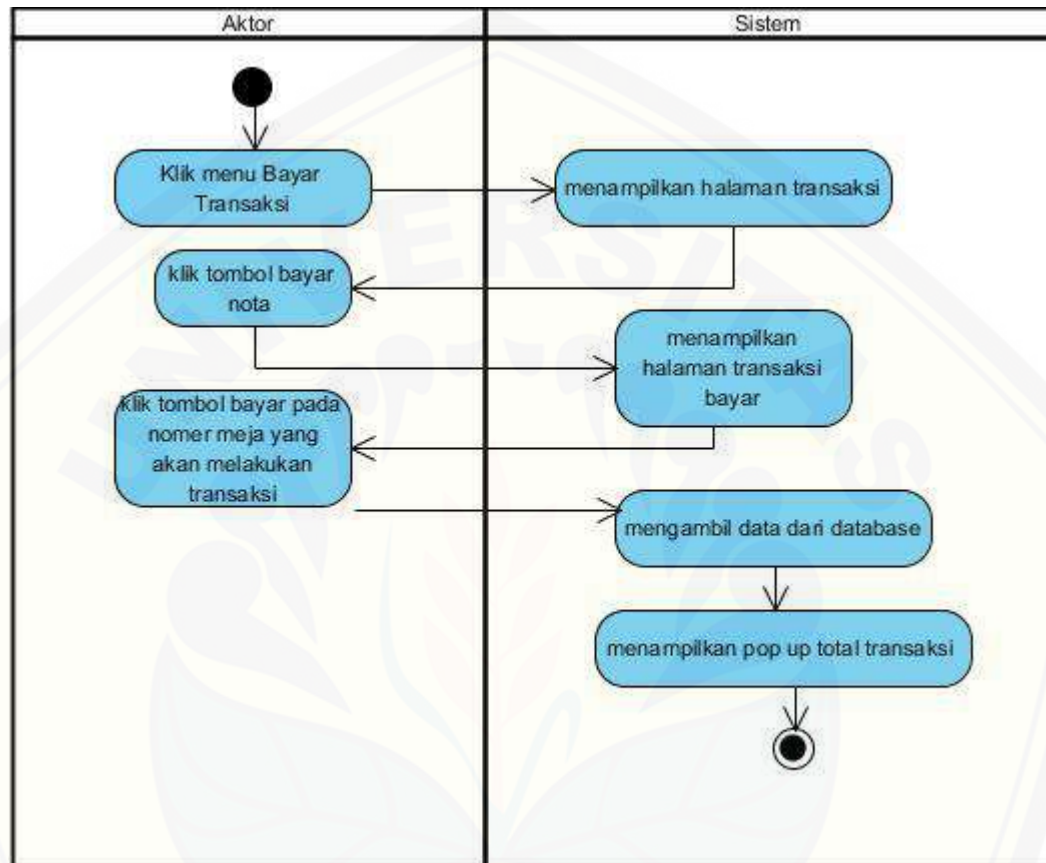


Tabel 4. 18Scenario menampilkan total bayar setiap meja

| <b>ID Skenario</b>                                     | <b>USC16</b>   |
|--|--|
| Usecase  | Menampilkan Total Bayar Setiap Meja                                      |
| Partisipan Aktor                                       | Kasir  |
| Pre-Kondisi  | Kasir sudah melakukan login.   |
| Pra-Kondisi  | Kasir selesai mengembalikan status meja                                  |
| <b>SKENARIO UTAMA</b>                                  |  |
| <b>“Menampilkan total bayar setiap meja”</b>           |  |
| <b>Aksi Aktor</b>                                      | <b>Reaksi Sistem</b>   |
| 1. Klik menu “Bayar Transaksi”                         | 2. Menampilkan halaman Transaksi   |
| 3. Klik tombol “Bayar nota”                            | 4. Menampilkan Halaman Transaksi Bayar.                                  |
| 5. Klik tombol Bayar pada data meja yang telah selesai | 6. Menampilkan <i>pop up</i> rincian biaya dan total yang harus dibayar. |
| 7. Klik selesai  |  |

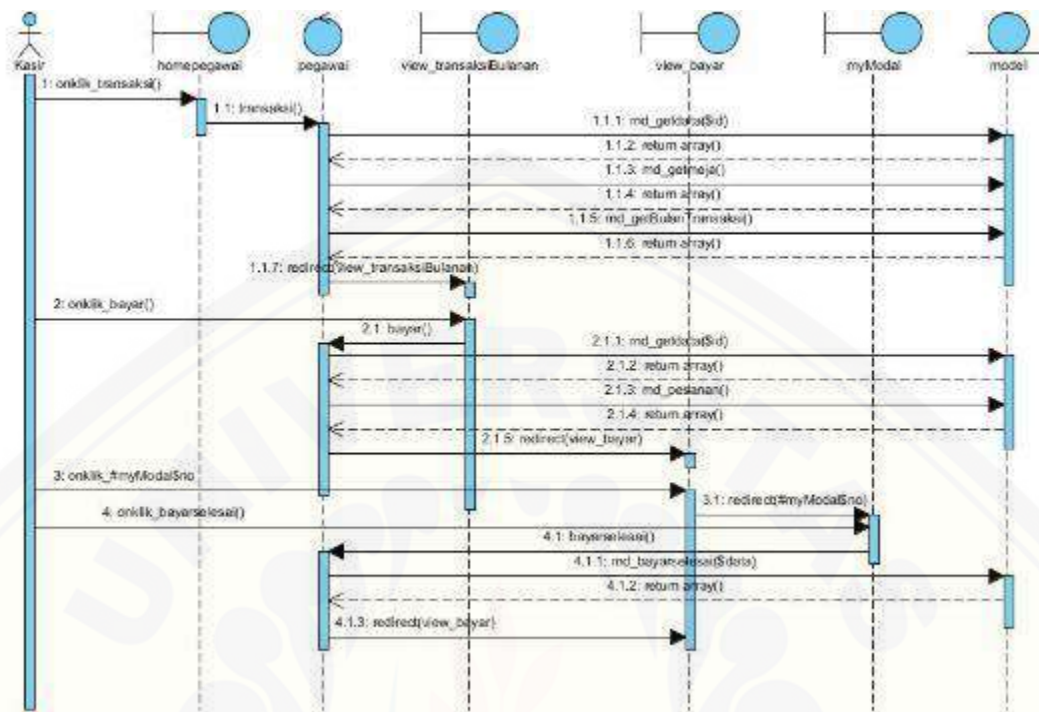
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menampilkan total bayar setiap mejamerupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan total

bayar setiap meja. Penjelasan *activity diagram* fitur menampilkan total bayar setiap mejadapat dilihat pada gambar 4.41



Gambar 4. 41Activity Diagram menampilkan total bayar setiap meja

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan total bayar setiap mejamenggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan total bayar setiap meja. Penjelasan *sequence diagram* fitur menampilkan total bayar setiap meja dapat dilihat pada gambar 4.42



Gambar 4. 42Sequence Diagram menampilkan total bayar setiap meja

17) Menampilkan Pesanan

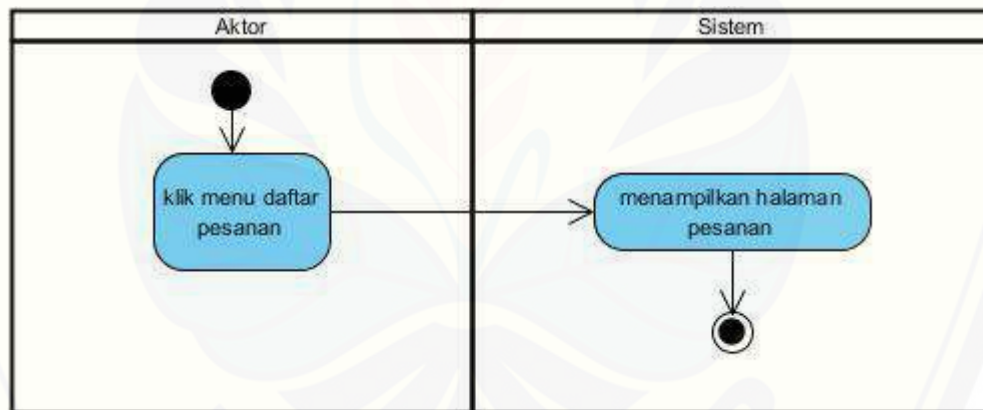
Menampilkan pesanan merupakan fitur untuk menampilkan dan proses pesanan yang pelanggan pesan, fitur ini dapat diakses oleh cheff dan barkeeper. Fitur ini bertujuan untuk menampilkan tugas yang harus dilakukan oleh cheff dan barkeeper untuk mengelola pesanan. Desain sistem yang lebih detail untuk fitur menampilkan pesanandigambarkan melalui *scenario*. *Scenario*menampilkan pesanandenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan pesanan seperti pada tabel 4.19

Tabel 4. 19Scenario menampilkan pesanan

|                    |                              |
|--------------------|------------------------------|
| <b>ID Skenario</b> | <b>USC17</b>                 |
| Usecase            | Menampilkan pesanan          |
| Partisipan Aktor   | Chef dan Barkeeper           |
| Pre-Kondisi        | Aktor sudah melakukan login. |

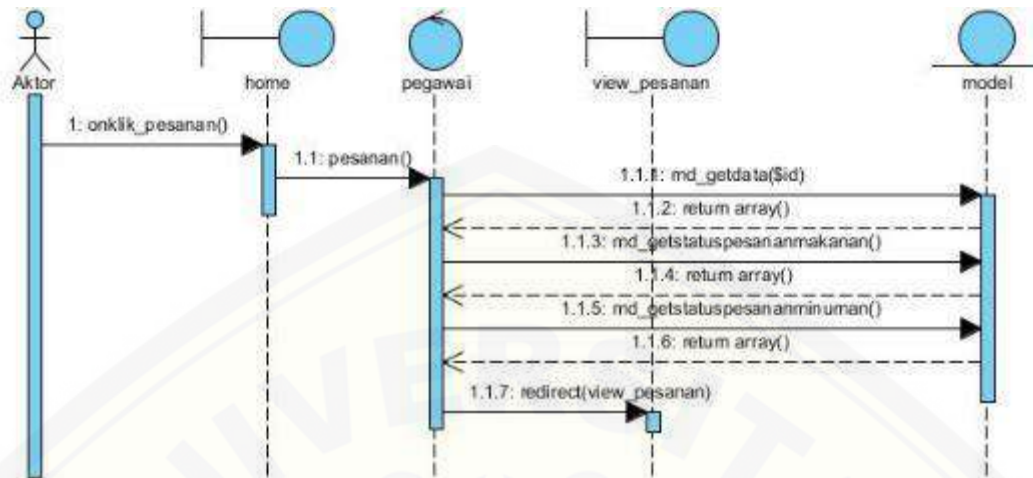
|                               |                                   |
|-------------------------------|-----------------------------------|
| Pra-Kondisi                   | Aktor telah menyelesaikan pesanan |
| <b>SKENARIO UTAMA</b>         |                                   |
| <b>“Menampilkan pesanan”</b>  |                                   |
| <b>Aksi Aktor</b>             | <b>Reaksi Sistem</b>              |
| 1. Klik menu “Daftar Pesanan” | 2. Menampilkan halaman pesanan    |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* menampilkan pesan merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan pesanan. Penjelasan *activity diagram* fitur menampilkan pesan dapat dilihat pada gambar 4.43



Gambar 4. 43Activity Diagram menampilkan pesanan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan pesan menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan pesanan. Penjelasan *sequence diagram* fitur menampilkan pesan dapat dilihat pada gambar 4.44



Gambar 4. 44Sequence Diagram menampilkan pesanan

## 18) Menambahkan dataTransaksi

Menambahkan datatransaksimerupakan fitur untuk menambahkan transaksi harian berdasarkan penjualan menu makanan dan minuman pada cafe. Fitur ini hanya dapat diakses oleh Waiters pada sistem. Desain sistem yang lebih detail untuk menambahkan dataTransaksidigambarkan melalui *scenario*. *Scenario* menambahkan dataTransaksi menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan dataTransaksi seperti pada tabel 4.20

Tabel 4. 20Scenario menambahkan data transaksi

| ID Skenario                    | USC18   |
|--------------------------------|---|
| Usecase                        | Menambahkan Transaksi                                       |
| Partisipan Aktor               | Waiters   |
| Pre-Kondisi                    | Waiterssudah melakuakan login.                              |
| Pra-Kondisi                    | Kasir selesai menambah transaksi dan tersimpan pada sistem. |
| <b>SKENARIO UTAMA</b>          |   |
| <b>“Menambahkan Transaksi”</b> |   |



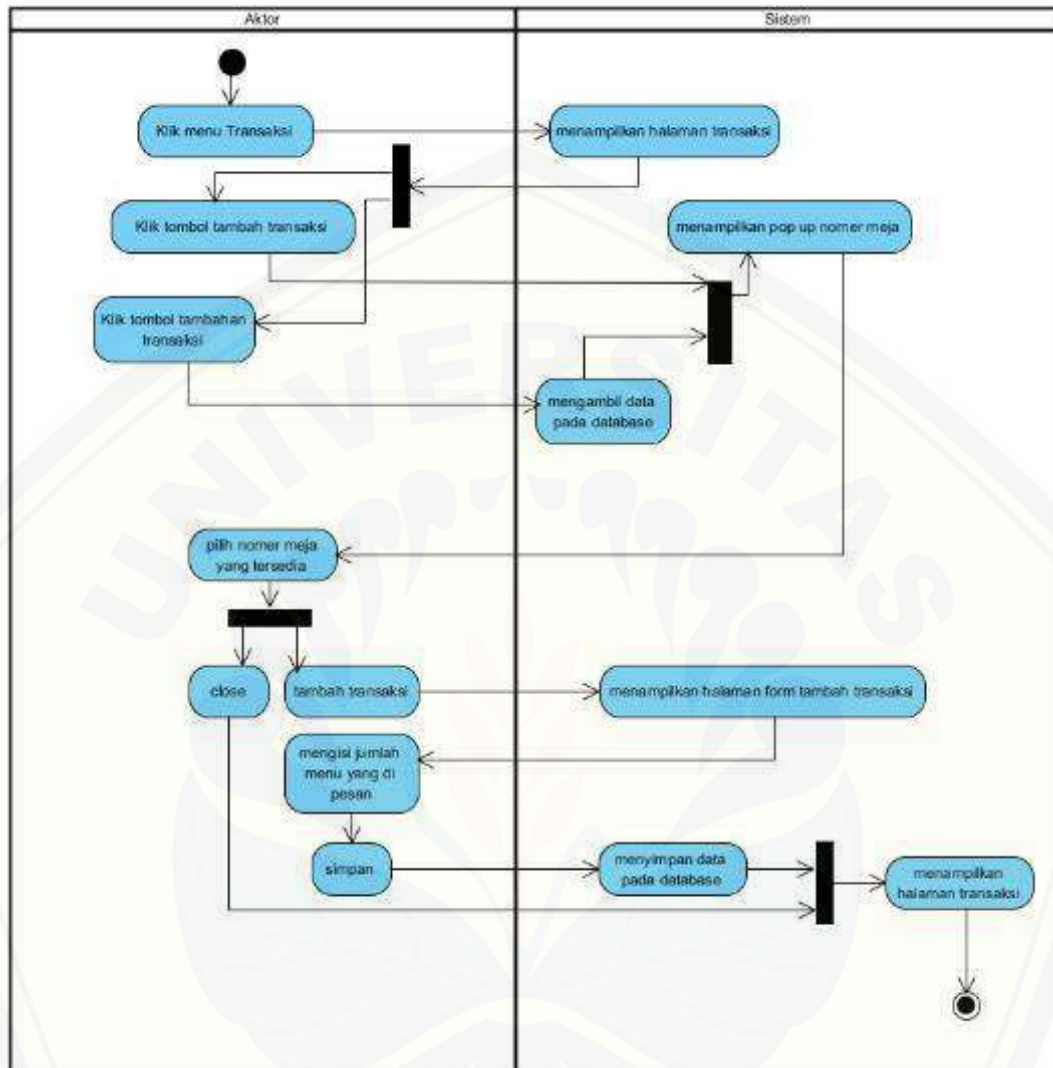
| Aksi Aktor                                  | Reaksi Sistem  |
|---|--|
| 1. Klik menu “Transaksi”                    | 2. Menampilkan halaman Transaksi   |
| 3. Klik tombol tambah data.                 | 4. Menampilkan pop up no. meja   |
| 5. Pilih no. meja yang di tempati pelanggan | 7. Menampilkan Form transaksi  |
| 6. Klik Tambah Transaksi                    | 8. Isi jumlah pada menu yang dipesan                                       |
| 8. Isi jumlah pada menu yang dipesan        | 9. Klik simpan atau enter  |
| 9. Klik simpan atau enter                   | 10. Menampilkan halaman Transaksi dengan alert “tambah transaksi berhasil” |

#### SKENARIO ALTERNATIF

##### “Menambahkan Transaksi Field Kosong”

- |   |  |
|---|--|
| 5. Mengisi beberapa Fild dan menghapus field jadi kosong. | 6. Menampilkan allert “please fill out this field” |
|---|--|

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menambahkan dataTransaksimerupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan dataTransaksi. Penjelasan *activity diagram* fitur menambahkan dataTransaksidapat dilihat pada gambar 4.45



Gambar 4. 45Activity Diagram menambahkan data transaksi

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan data Transaksi menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data Transaksi. Penjelasan *sequence diagram* fitur menambahkan data Transaksi dapat dilihat pada gambar 4.46



## 19) Merubah Status Pesanan

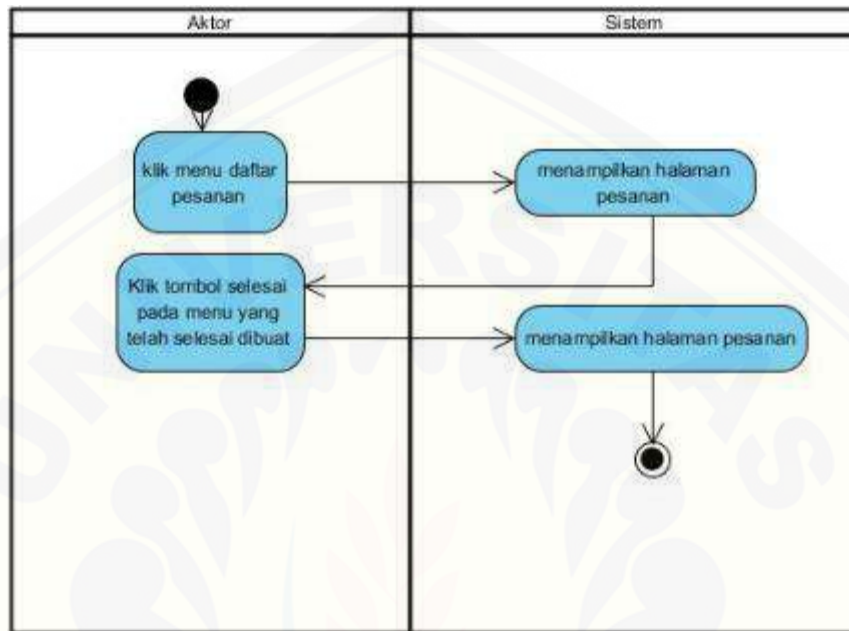
Merubah status pesanan merupakan fitur untuk merubah status pesanan, fitur ini bisa diakses oleh cheff, barkeeper dan waiters, barkeeper dan cheff merubah status pesanan menjadi selesai apabila pesanan sudah selesai dibuat dan waiters merubah status pesanan menjadi selesai ketika pesanan sudah terkirim di atas meja pesanan. Desain sistem yang lebih detail untuk merubah status pesanandigambarkan melalui *scenario*. *Scenario* merubah status pesanan menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur merubah status pesanan seperti pada tabel 4.21

Tabel 4. 21Scenario merubah status pesanan

| ID Skenario   | USC19   |
|---|---|
| Usecase   | Menampilkan pesanan                           |
| Partisipan Aktor  | Chef dan Barkeeper                            |
| Pre-Kondisi   | Aktor sudah melakukan login.                  |
| Pra-Kondisi   | Chef dan Barkeper telah menyelesaikan pesanan |
| <b>SKENARIO UTAMA</b>   |   |
| <b>“Menampilkan pesanan”</b>                                  |   |
| Aksi Aktor  | Reaksi Sistem                                 |
| 1. Klik menu “Daftar Pesanan”                                 | 2. Menampilkan halaman pesanan                |
| 3. Klik tombol “selesai” pada menu yang telah selesai di buat | 4. Menampilkan halaman pesanan                |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur merubah status pesanan merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses merubah status

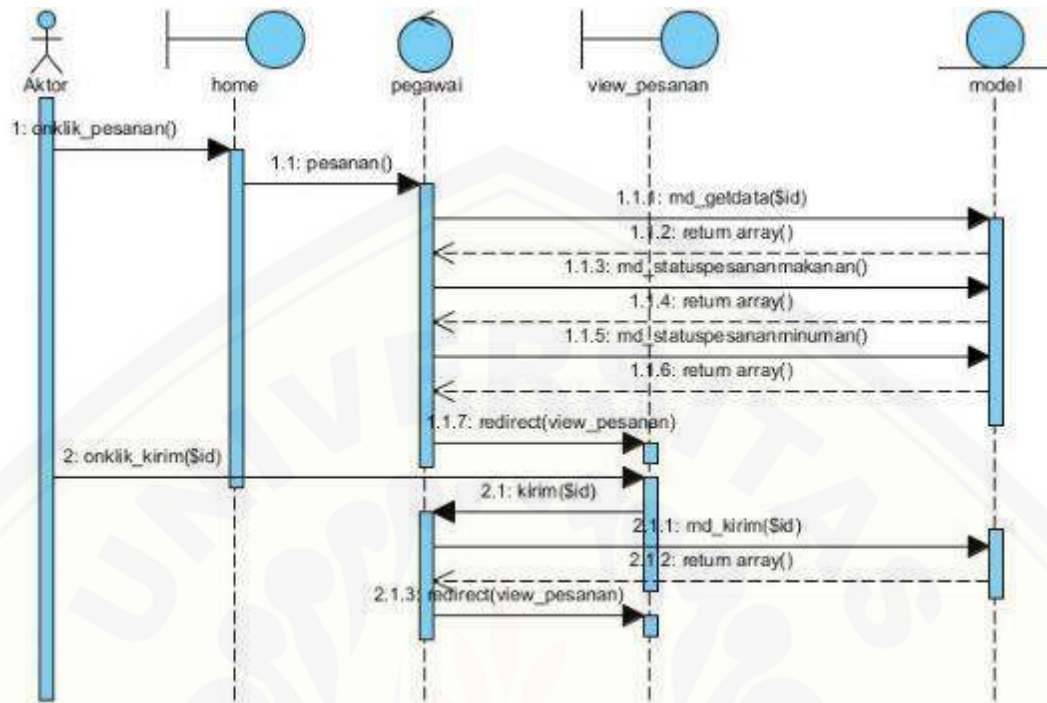
pesanan. Penjelasan *activity diagram* fitur merubah status pesanan dapat dilihat pada gambar 4.47



Gambar 4. 47Activity Diagram merubah status pesanan

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Squence diagram* fitur merubah status pesananmenggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah status pesanan. Penjelasan *sequence diagram* fitur merubah status pesanandapat dilihat pada gambar 4.48





Gambar 4. 48Sequence Diagram merubah status pesanan

## 20) Menambahkan Bahan Baku

Menambahkan bahan baku merupakan fitur untuk menambahkan bahan baku yang digunakan pada cafe dalam membuat menu makanan/minuman dan akan tersimpan pada sistem, fitur ini berupa *form* nama bahan baku, lama pengiriman dan jumlah ketersediaan.. Fitur ini hanya dapat diakses oleh Cheff dan Barkeeper pada sistem. Desain sistem yang lebih detail untuk menambahkan databahan bakudigambarkan melalui *scenario*. *Scenario* menambahkan databahan baku menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan databahan baku seperti pada tabel 4.22

Tabel 4. 22Scenario menambahkan bahan baku

| ID Skenario      | USC20                  |
|------------------|------------------------|
| Usecase          | Menambahkan Bahan Baku |
| Partisipan Aktor | Cheff dan Barkeeper    |

|             |  |
|-------------|--|
| Pre-Kondisi | Aktor sudah melakukan login.                                       |
| Pra-Kondisi | Cheff dan Barkeeper menambah bahan baku dan tersimpan pada sistem. |

**SKENARIO UTAMA**  
**“Menambahkan Bahan Baku”**

| Aksi Aktor   | Reaksi Sistem   |
|--|---|
| 1. Klik menu “Stok Bahan Baku”   |   |
|  | 2. Menampilkan halaman Stok bahan baku Roots Cafe   |
| 3. Klik tombol tambah data.  |   |
|  | 4. Menampilkan Halaman tambah Bahan Baku roots cafe.  |
| 5. Mengisi nama bahan baku, lama pengiriman bahan baku dan jumlah penggunaan bahan baku setiap satuannya, dan satuannya. |   |
| 6. Klik tombol Submit  |   |
|  | 7. Menampilkan halaman stok bahan baku dengan notifikasi “Bahan Baku baru berhasil ditambahkan” |

**SKENARIO ALTERNATIF**

**“Menambahkan Bahan Baku Dengan Field Kosong”**

| Aksi Aktor   | Reaksi Sistem |
|--|---------------|
| 5. Mengisi nama bahan baku, lama pengiriman bahan baku dan jumlah penggunaan bahan baku setiap |               |

---

satuannya, dan membiarkan salah satu *field* kosong.

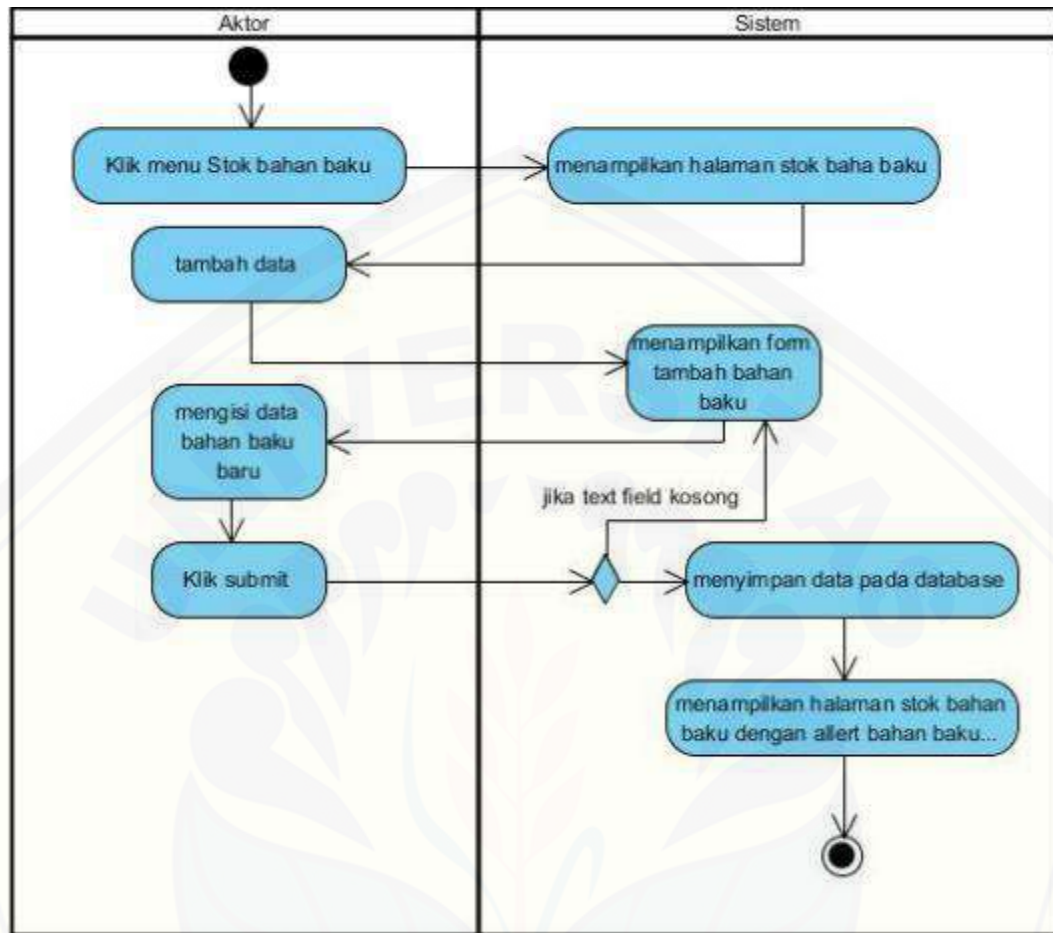
---

**6. Klik Submit**

---

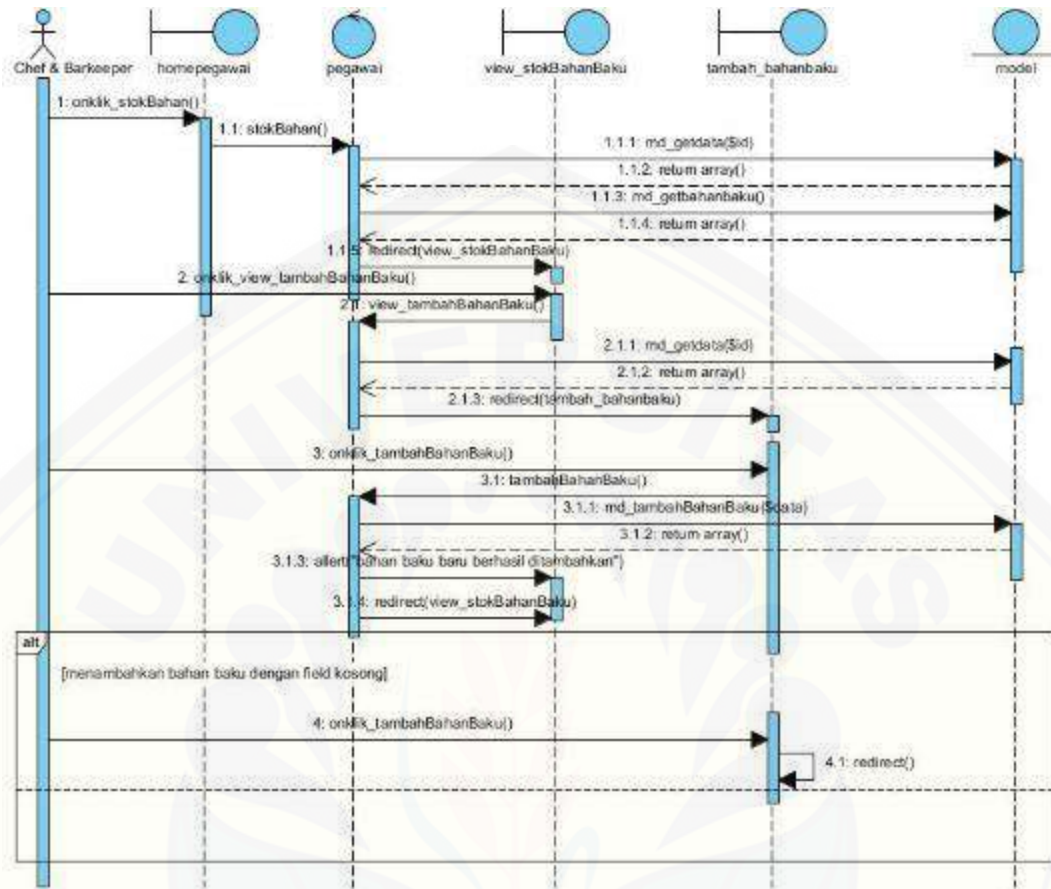
- 7. Menampilkan Halaman tambah Bahan Baku roots cafe dengan alert “please fill out this field.” pada field yang kosong.**
- 

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur Menambahkan databahan baku merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses Menambahkan databahan baku. Penjelasan *activity diagram* fitur menambahkan databahan baku dapat dilihat pada gambar 4.49



Gambar 4. 49Activity Diagram menambahkan data bahan baku

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan databahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses Menambahkan databahan baku. Penjelasan *sequence diagram* fitur Menambahkan databahan bakudapat dilihat pada gambar 4.50



Gambar 4. 50Sequence Diagram menambahkan data bahan baku

## 21) Menampilkan Bahan Baku

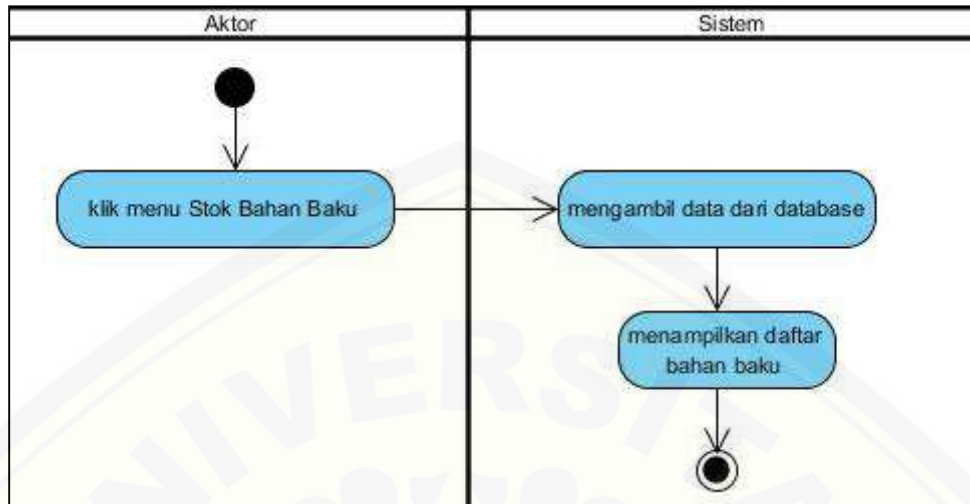
Menampilkan bahan baku merupakan fitur untuk menampilkan bahan baku yang digunakan pada cafe dalam membuat menu makanan/minuman yang sudah tersimpan pada sistem, fitur ini berupa tabel nama bahan baku, jumlah ketersediaannya. Fitur ini dapat diakses oleh Cheff dan Barkeeper pada sistem. Desain sistem yang lebih detail untuk Menampilkan bahan baku digambarkan melalui *scenario*. *Scenario* menampilkan bahan baku menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menampilkan bahan baku seperti pada tabel 4.23



Tabel 4. 23Scenario menampilkan bahan baku

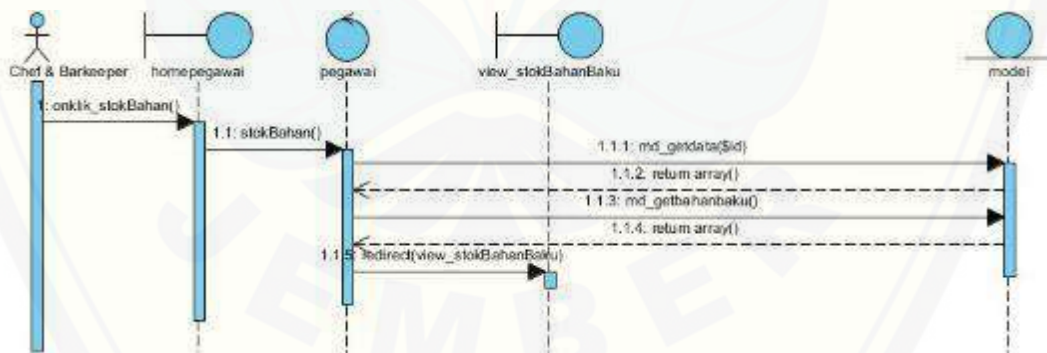
| <b>ID Skenario</b>              | <b>USC21</b>  |
|---------------------------------|---|
| Usecase                         | Menampilkan Bahan Baku                              |
| Partisipan Aktor                | Chef & Barkeeper                                    |
| Pre-Kondisi                     | Aktor sudah melakukan login.                        |
| Pra-Kondisi                     | Chef & Barkeepersudah menampilkan daftar Bahan Baku |
| <b>SKENARIO UTAMA</b>           |   |
| <b>“Menampilkan Bahan Baku”</b> |   |
| <b>Aksi Aktor</b>               | <b>Reaksi Sistem</b>                                |
| 1. Klik Menu “Stok Bahan Baku”  |   |
|                                 | 2. Menampilkan Halaman Stok bahan baku roots cafe   |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur *menampilkan bahan baku* merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menampilkan bahan baku. Penjelasan *activity diagram* fitur *menampilkan bahan baku* dapat dilihat pada gambar 4.51



Gambar 4. 51Activity Diagram menampilkan bahan baku

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menampilkan bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menampilkan bahan baku. Penjelasan *sequence diagram* fitur menampilkan bahan baku dapat dilihat pada gambar 4.52



Gambar 4. 52Sequence Diagram menampilkan bahan baku

22) Merubah Bahan Baku

Merubah data bahan baku merupakan fitur untuk merubah bahan baku yang sudah tersimpan pada sistem, fitur ini berupa *form* nama bahan baku, lama pengiriman dan jumlah ketersediaan yang sudah terisi dari database dan akan dirubah.

Fitur ini hanya dapat diakses oleh Cheff dan Baarkeeper pada sistem. Desain sistem yang lebih detail untuk fitur merubah databahan bakudigambarkan melalui *scenario*. *Scenario* merubah databahan baku menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur merubah databahan baku seperti pada tabel 4.24

Tabel 4. 24Scenario merubah bahan baku

|   |  |
|---|--|
| <b>ID Skenario</b>  | <b>USC22</b>   |
| Usecase   | Merubah Bahan Baku   |
| Partisipan Aktor  | Chef dan Barkeeper   |
| Pre-Kondisi   | Aktor sudah melakuakan login.                                    |
| Pra-Kondisi   | Chef dan Barkeeper merubah Bahan Baku dan tersimpan pada sistem. |
| <b>SKENARIO UTAMA</b>   |  |
| <b>“Merubah Bahan Baku”</b>   |  |
| <b>Aksi Aktor</b>   | <b>Reaksi Sistem</b>   |
| 1. Klik menu “Stok Bahan Baku”  | 2. Menampilkan halaman Bahan Baku                                |
| 3. Klik tombol bewarna biru dengan tulisan edit   | 4. Menampilkan pop up merubah jumlah stok bahan baku             |
| 5. Merubahnama,lama pengiriman, jumlah penggunaan bahan baku dan satuannya setiap bahan baku. |  |
| 6. Klik tombol Simpan   |  |

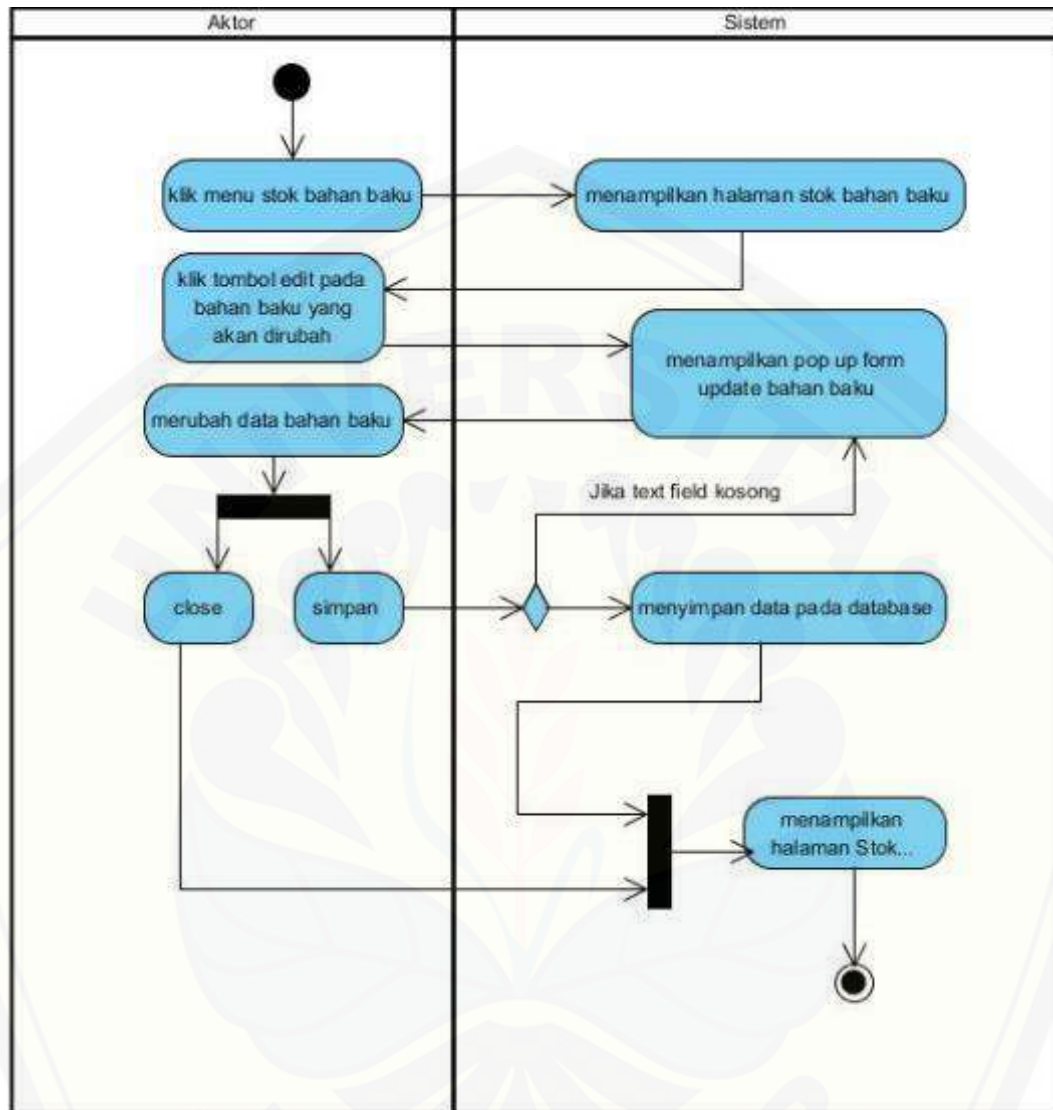
- 
7. Menampilkan halaman Stok bahan baku roots cafe dengan notifikasi “bahan baku berhasil dirubah”

### SKENARIO ALTERNATIF

#### “Merubah Bahan Baku Field Kosong”

| Aksi Aktor   | Reaksi Sistem  |
|--|--|
| <p>5. Merubahnama,lama pengiriman dan jumlah penggunaan bahan baku dan membiarkan satu field kosong.</p> |  |
| <p>6. Klik tombol Submit</p>   | <p>7. Menampilkan pop up merubah jumlah stok bahan baku roots cafe pada field yang kosong terdapat notification dengan alert “please fill out this field.”</p> |

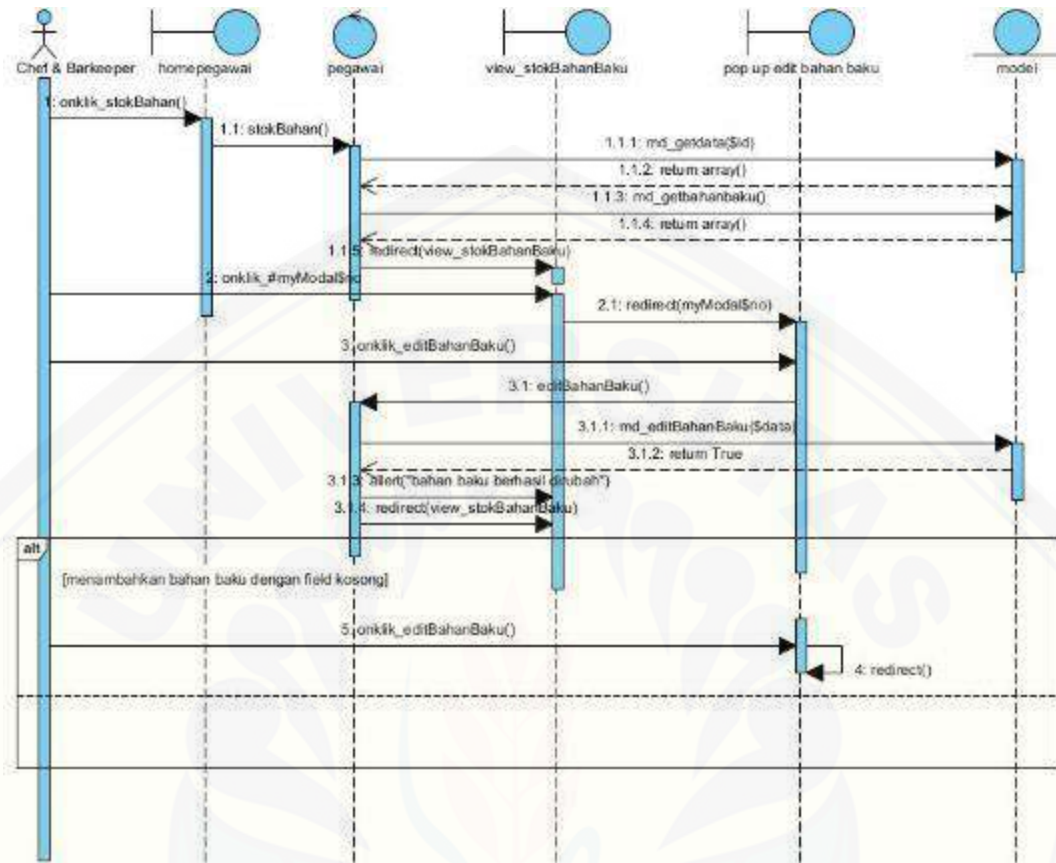
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur merubah databahan bakumerupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses merubah databahan baku. Penjelasan *activity diagram* fitur merubah databahan baku dapat dilihat pada gambar 4.53



Gambar 4. 53Activity Diagram merubah bahan baku

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur merubah data bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses merubah data bahan baku. Penjelasan *sequence diagram* fitur merubah data bahan baku dapat dilihat pada gambar 4.54





Gambar 4. 54Sequence Diagram merubah bahan baku

### 23) Menghapus Bahan Baku

Menghapus bahan baku merupakan fitur untuk menghapus bahan baku yang sudah tidak digunakan lagi oleh cafe dalam membuat menu makanan/minuman, fitur ini berupa tombol hapus dan apabila menghapus bahan baku, otomatis menghapus penggunaan bahan baku pada menu makanan/minuman.. Fitur ini hanya dapat diakses oleh pegawai pada sistem. Desain sistem yang lebih detail untuk Menghapus data bahan baku digambarkan melalui *scenario*. *Scenario* Menghapus data bahan baku menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur Menghapus data bahan baku seperti pada tabel 4.25

Tabel 4. 25Scenario menghapus bahan baku

|   |  |
|---|--|
| <b>ID Skenario</b>  | <b>USC23</b>   |
| Usecase   | Menghapus Bahan Baku   |
| Partisipan Aktor  | Chef dan Barkeeper   |
| Pre-Kondisi   | Aktor sudah melakukan login.   |
| Pra-Kondisi   | Chef dan Barkeepersudah menghapus Bahan Baku   |
| <b>SKENARIO UTAMA</b>   |  |
| <b>“Menghapus Bahan Baku”</b>   |  |
| <b>Aksi Aktor</b>   | <b>Reaksi Sistem</b>   |
| <b>1.</b> Klik Menu “Stok Bahan Baku”                                   |  |
|   | <b>2.</b> Menampilkan Halaman Stok bahan baku roots cafe   |
| <b>3.</b> Klik tombol berwarna merah pada bahan baku yang akan dihapus. |  |
|   | <b>4.</b> Menampilkan pop up menghapus bahan baku dengan notification “Apakah anda ingin menghapus bahan baku (nama bahan baku)?? apabila anda menghapus bahan baku ini, akan menghapus penggunaan bahan baku yang menggunakan bahan baku (nama bahan baku)” |
| <b>5.</b> Klik tombol menghapus   |  |
|   | <b>6.</b> Menampilkan halaman stok   |

---

bahan baku dengan alert “bahan baku berhasil dihapus”

**SKENARIO ALTERNATIF**  
**“Batal Menghapus Bahan Baku”**

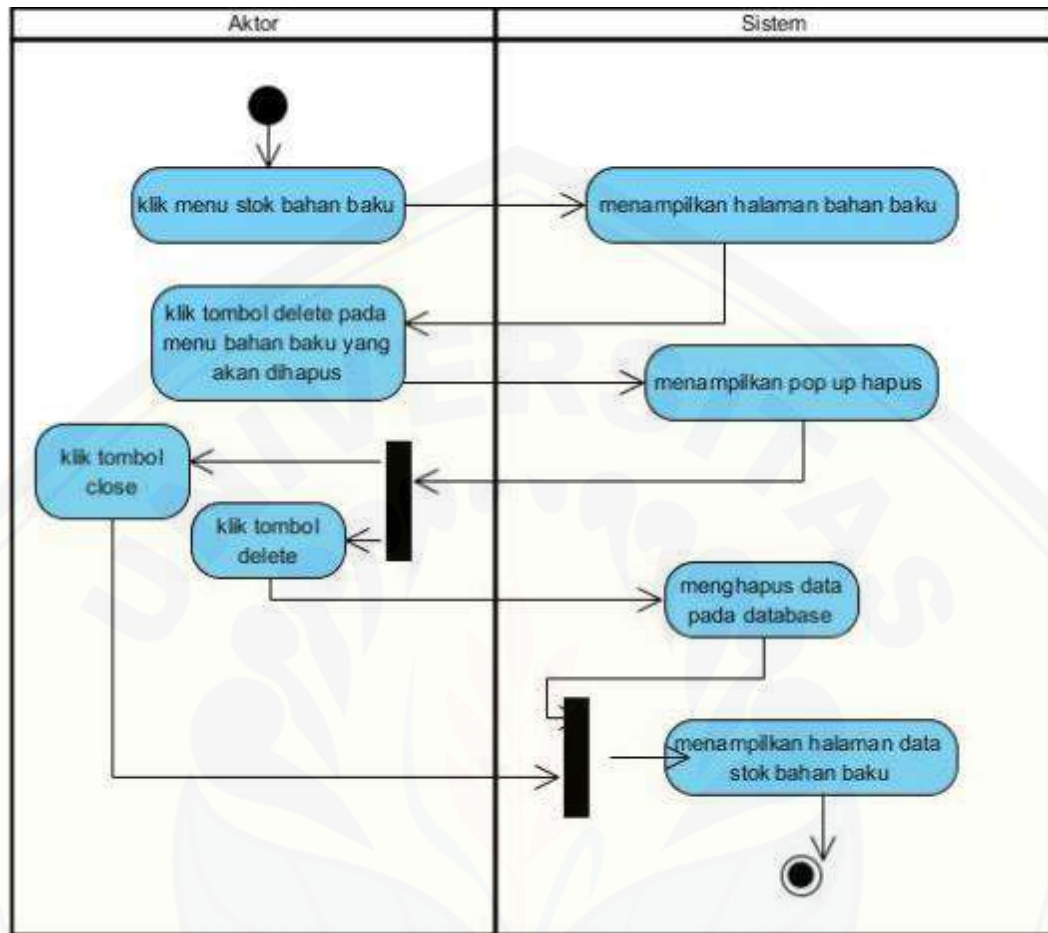
**5. Klik Tombol Batal**

---

**6. Menampilkan Kembali halaman stok bahan baku**

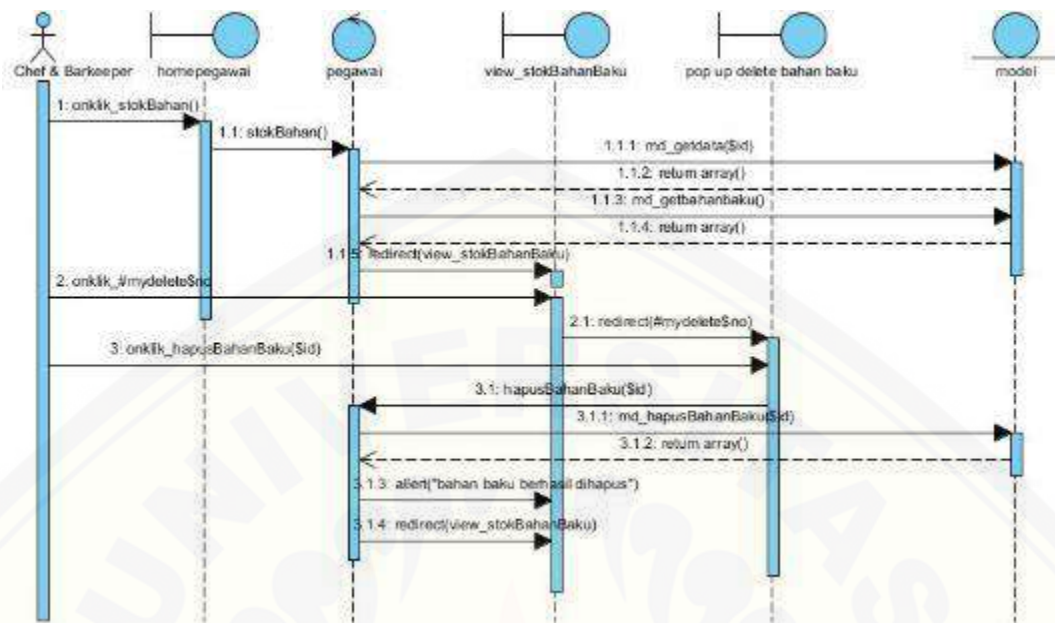
---

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur *Menghapus databahan baku* merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses *Menghapus databahan baku*. Penjelasan *activity diagram* fitur *Menghapus databahan baku* dapat dilihat pada gambar 4.55



Gambar 4. 55Activity Diagram menghapus bahan baku

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menghapus data bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menghapus data bahan baku. Penjelasan *sequence diagram* fitur Menambahkan data bahan baku dapat dilihat pada gambar 4.56



Gambar 4. 56Sequence Diagram menghapus bahan baku

## 24) Menambahkan Penggunaan Bahan Baku

Menambahkan dataPenggunaan Bahan Baku merupakan fitur untuk menambahkan penggunaan bahan baku pada setiap menu makanan dan minuman yang terjual di Roots Cafe. Fitur ini menampilkan *pop up* daftar bahan baku dan aktor tinggal memilih bahan baku mana saja yang digunakan pada menu tersebut. Fitur ini hanya dapat diakses oleh Cheff dan Barkeeper pada sistem. Desain sistem yang lebih detail untuk menambahkan data penggunaan bahan bakudigambarkan melalui *scenario*. *Scenario* menambahkan data penggunaan bahan bakumenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan data penggunaan bahan bakuseperti pada tabel 4.26

Tabel 4. 26Scenario menambahkan bahan baku

| ID Skenario | USC24 |
|-------------|-------|
|-------------|-------|



|                  |  |
|------------------|--|
| Usecase          | Menambahkan Penggunaan Bahan Baku  |
| Partisipan Aktor | Chef dan Barkeeper   |
| Pre-Kondisi      | Aktor sudah melakukan login.   |
| Pra-Kondisi      | Chef dan Barkeeper menambah penggunaan bahan baku untuk makanan ataupun minuman dan tersimpan pada sistem. |

### SKENARIO UTAMA

#### “Menambahkan Penggunaan Bahan Baku”

| Aksi Aktor  | Reaksi Sistem  |
|---|--|
| 1. Klik menu “Olah Bahan Baku” pilih makanan ataupun minuman. |  |
|   | 2. Menampilkan halaman olah bahan baku dengan tabel daftar menu makanan apabila memilih makanan dan menu minuman apabila memilih menu minuman. |
| 3. Klik tombol Olah pada setiap menu                          |  |
|   | 4. Menampilkan pop up daftar bahan baku yang terdaftar pada sistem   |
| 5. Check bahan baku yang digunakan pada menu tersebut.        |  |
| 6. Klik tombol Simpan   |  |
|   | 7. Menampilkan halaman olah bahan baku menu makanan apabila memilih makanan dan halaman olah bahan baku menu minuman apabila memilih menu      |

---

minuman dengan notifikasi  
 “menu sudah ditambah  
 penggunaan bahan baku”

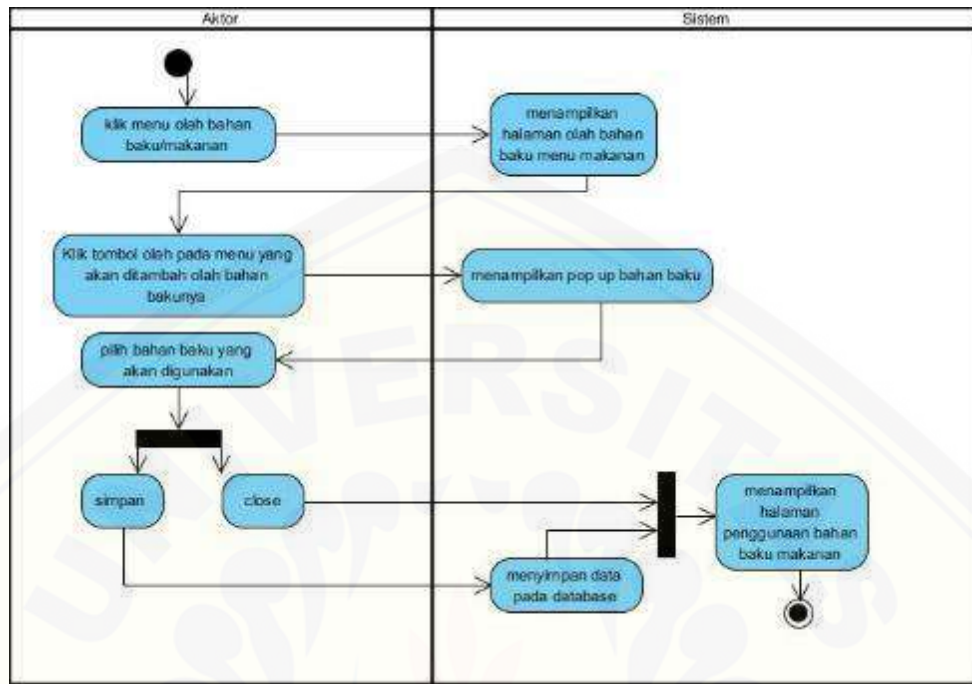
---

### SKENARIO ALTERNATIF

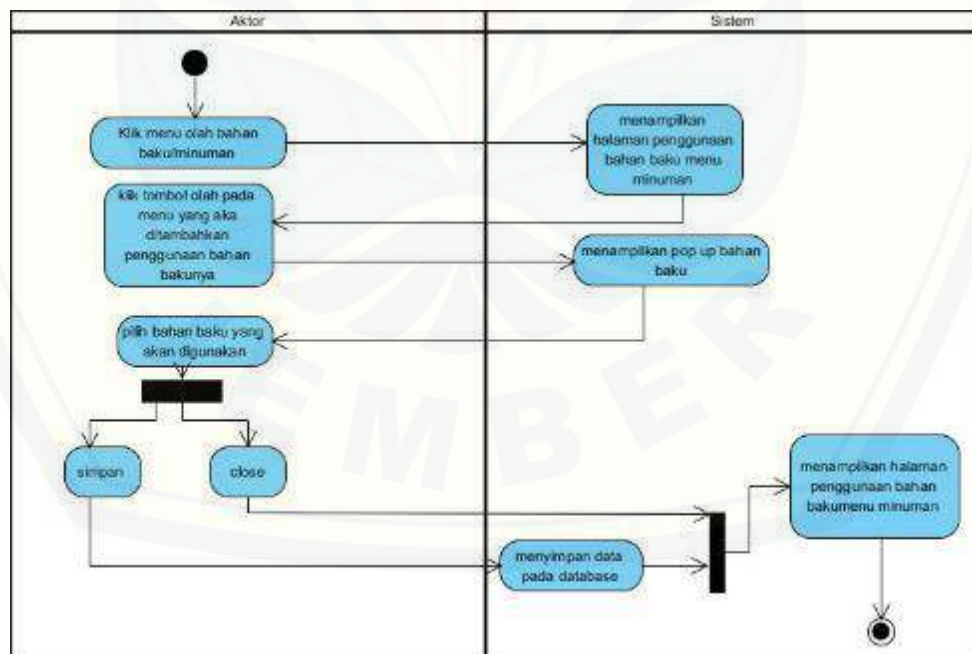
#### “Batal Menambahkan Penggunaan Bahan Baku”

| Aksi Aktor          | Reaksi Sistem   |
|---------------------|---|
| 6 Klik tombol close | 7 Menampilkan kembali Halaman<br>olah bahan baku menu makanan<br>atau minuman roots cafe. |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menambahkan data penggunaan bahan baku merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan data penggunaan bahan baku. Penjelasan *activity diagram* fitur menambahkan data penggunaan bahan baku dapat dilihat pada gambar 4.57 untuk menu makanan dan gambar 4.58 untuk menu minuman

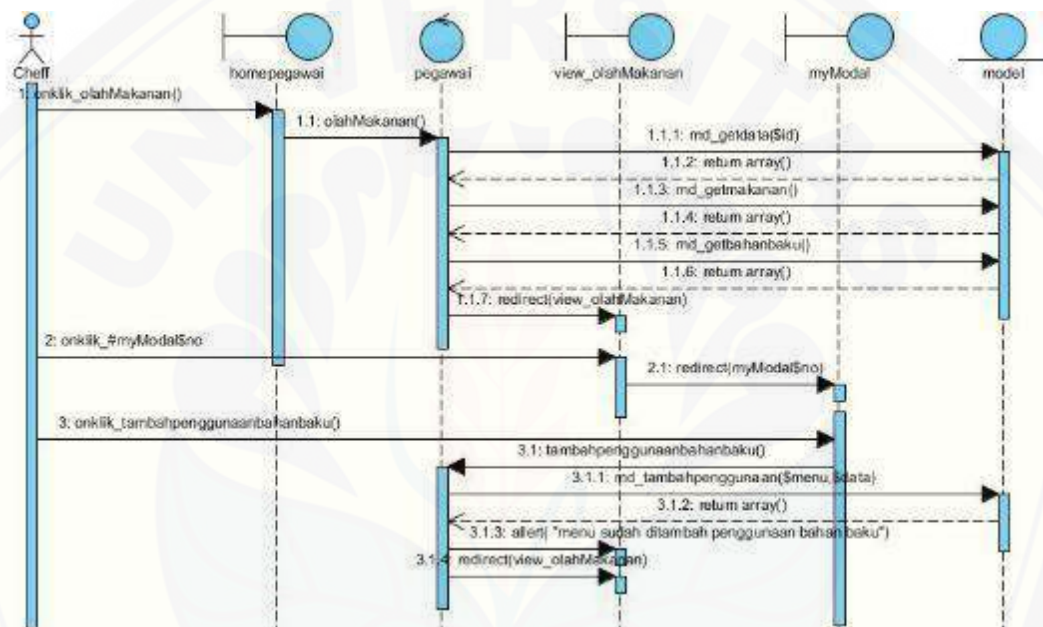


Gambar 4. 57Activity Diagram menambahkan penggunaan

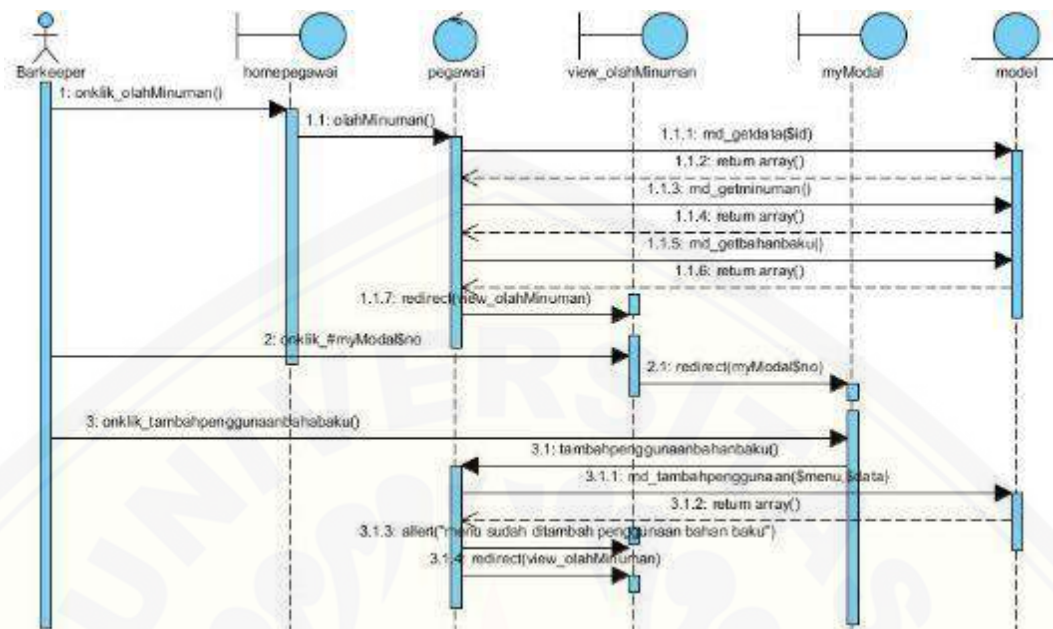


Gambar 4. 58Activity Diagram menambahkan penggunaan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan data penggunaan bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data penggunaan bahan baku. Penjelasan *sequence diagram* fitur menambahkan data penggunaan bahan baku dapat dilihat pada gambar 4.59 untuk data bahan baku makanan dan gambar 4.60 untuk data bahan baku minuman.



Gambar 4. 59Sequence Diagram menambahkan penggunaan



Gambar 4. 60Sequence Diagram menambahkan penggunaan

## 25) Menghapus penggunaan bahan baku

Menghapus dataPenggunaan Bahan Baku merupakan fitur untuk menghapus penggunaan bahan baku pada setiap menu makanan dan minuman yang terjual di Roots Cafe apabila ketika proses menambahkan data penggunaan bahan baku terjadi *human error* atau bahkan menu makanan dan minuman tersebut tidak menggunakan bahan baku itu lagi. Fitur ini menampilkan *pop up* daftar bahan baku yg sudah tersimpan pada sistem untuk setiap menyanya dan *aktor* tinggal memilih bahan baku mana saja yang dihapus pada menu tersebut. Fitur ini hanya dapat diakses oleh Cheff dan Barkeeper pada sistem. Desain sistem yang lebih detail untuk menghapus datapenggunaan bahan bakudigambarkan melalui *scenario*. *Scenario* menghapusdatapenggunaan bahan bakumenjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menghapus datapenggunaan bahan bakuseperti pada tabel 4.27



Tabel 4. 27Scenario menghapus penggunaan bahan baku

| <b>ID Skenario</b>  | <b>USC25</b>   |
|---|--|
| Usecase   | Menghapus Penggunaan Bahan Baku  |
| Partisipan Aktor  | Chef dan Barkeeper   |
| Pre-Kondisi   | Aktor sudah melakukan login.   |
| Pra-Kondisi   | Chef dan Barkeeper selesai menghapus beberapa penggunaan bahan baku untuk makanan ataupun minuman pada sistem.                                 |
| <b>SKENARIO UTAMA</b>   |  |
| <b>“Menambahkan Penggunaan Bahan Baku”</b>                    |  |
| <b>Aksi Aktor</b>   | <b>Reaksi Sistem</b>   |
| 1. Klik menu “Olah Bahan Baku” pilih makanan ataupun minuman. | 2. Menampilkan halaman olah bahan baku dengan tabel daftar menu makanan apabila memilih makanan dan menu minuman apabila memilih menu minuman. |
| 3. Klik tombol Menghapus pada setiap menu                     | 4. Menampilkan pop up daftar bahan baku yang digunakan pada menu tersebut.   |
| 5. Check bahan baku yang akan dihapus.                        |  |
| 6. Klik tombol Menghapus                                      | 7. Menampilkan halaman olah bahan baku menu makanan  |

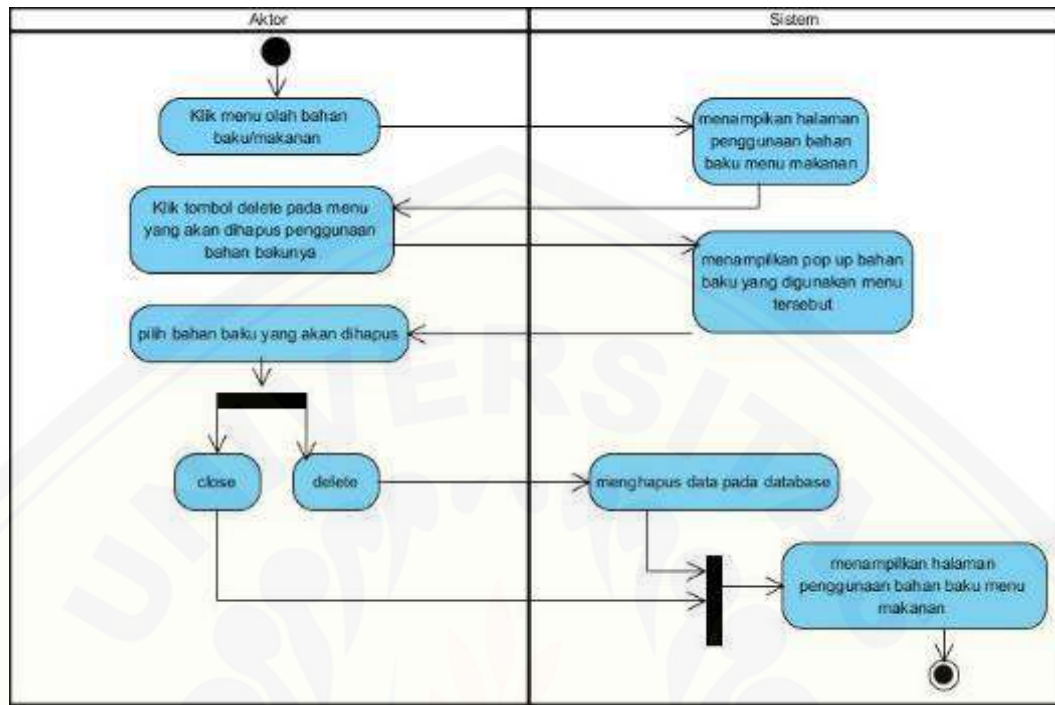
apabila memilih makanan dan halaman olah bahan baku menu minuman apabila memilih menu minuman dengan notifikasi “Penggunaan Bahan Baku pada Menu Berhasil Dihapus”

### SKENARIO ALTERNATIF

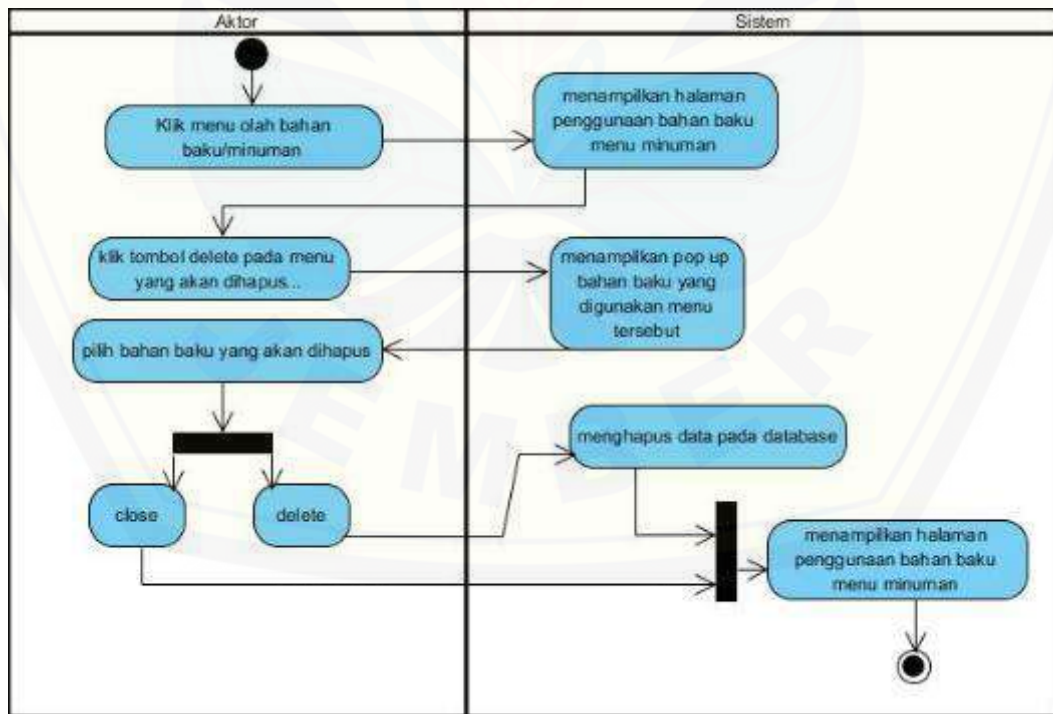
#### “Batal Hapus Penggunaan Bahan Baku”

| Aksi Aktor          | Reaksi Sistem   |
|---------------------|---|
| 8 Klik tombol close |   |
|                     | 9 Menampilkan kembali Halaman olah bahan baku menu makanan atau minuman roots cafe. |

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur menghapus data penggunaan bahan baku merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menghapus data penggunaan bahan baku. Penjelasan *activity diagram* fitur menghapus data penggunaan bahan baku dapat dilihat pada gambar 4.61 untuk menu makanan dan gambar 4.62 untuk menu minuman.

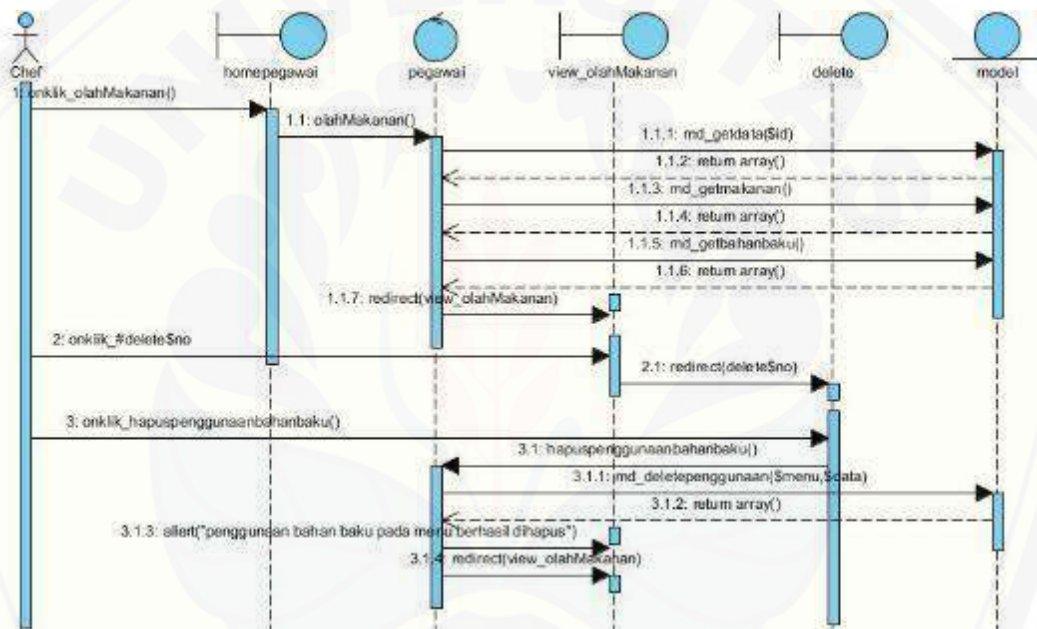


Gambar 4. 61 Activity Diagram menghapus penggunaan

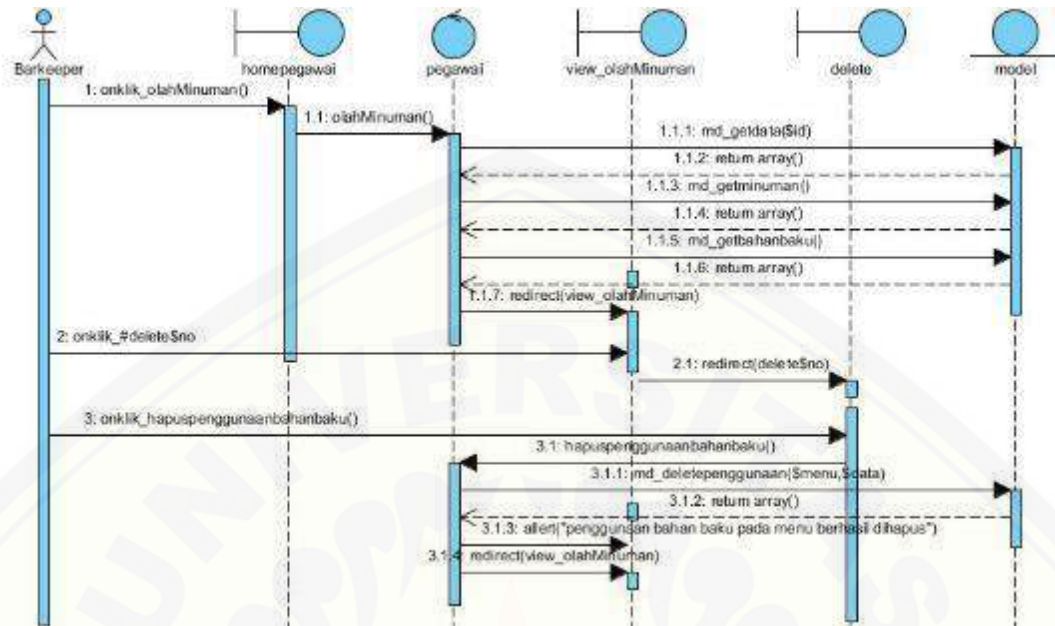


Gambar 4. 62 Activity Diagram menghapus penggunaan

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menghapus data penggunaan bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menghapus data penggunaan bahan baku. Penjelasan *sequence diagram* fitur menghapus data penggunaan bahan baku dapat dilihat pada gambar 4.63 untuk penggunaan bahan baku menu makanan dan gambar 4.64 untuk penggunaan bahan baku menu minuman.



Gambar 4. 63 Sequence Diagram menghapus penggunaan



Gambar 4. 64Sequence Diagram menghapus penggunaan

## 26) Menambahkan persediaanBahan Baku

Menambahkan persediaan bahan baku merupakan fitur untuk menambahkan persediaan bahan baku pada sistem. Fitur ini menampilkan form tanggal kadaluwarsa dan jumlah bahan baku berdasarkan paket bahan baku dan akan tersimpan pada database. Fitur ini hanya dapat diakses oleh Cheff dan Barkeeper pada sistem. Desain sistem yang lebih detail untuk menambahkan data bahan baku digambarkan melalui *scenario*. *Scenario* menambahkan data bahan baku menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur menambahkan data bahan baku seperti pada tabel 4.28

Tabel 4. 28Scenario menambahkan penggunaan bahan baku

| ID Skenario      | USC26                             |
|------------------|-----------------------------------|
| Usecase          | Menambahkan Persediaan Bahan Baku |
| Partisipan Aktor | Chef dan Barkeeper                |



|             |   |
|-------------|---|
| Pre-Kondisi | Aktor sudah melakukan login.                                  |
| Pra-Kondisi | Chef dan Barkeeper selesai menambahkan persediaan bahan baku. |

### SKENARIO UTAMA

#### “Menambahkan Persediaan Bahan Baku”

| Aksi Aktor   | Reaksi Sistem   |
|--|---|
| 1. Klik menu “Stok Bahan Baku”   |   |
|  | 2. Menampilkan halaman stok bahan baku roots cafe   |
| 3. Klik tombol “stok” pada bahan baku yang akan ditambahkan persediaannya. |   |
|  | 4. Menampilkan <i>pop up</i> menambahkan stok bahan baku dengan <i>field</i> tanggal kadaluwarsa dan jumlah persediaan yang akan disimpan dalam sistem. |
| 5. Mengisi <i>text field</i> yang tersedia                                 |   |
| 6. Klik simpan.  |   |
|  | 7. Menampilkan halaman <i>stok</i> bahan baku dengan <i>notification</i> “Bahan Baku Berhasil ditambah persediaannya”                                   |

### SKENARIO ALTERNATIF

#### “Text Field persediaan bahan baku Kosong”

---

5. Membiarkan salahsatu field  
kosong

---

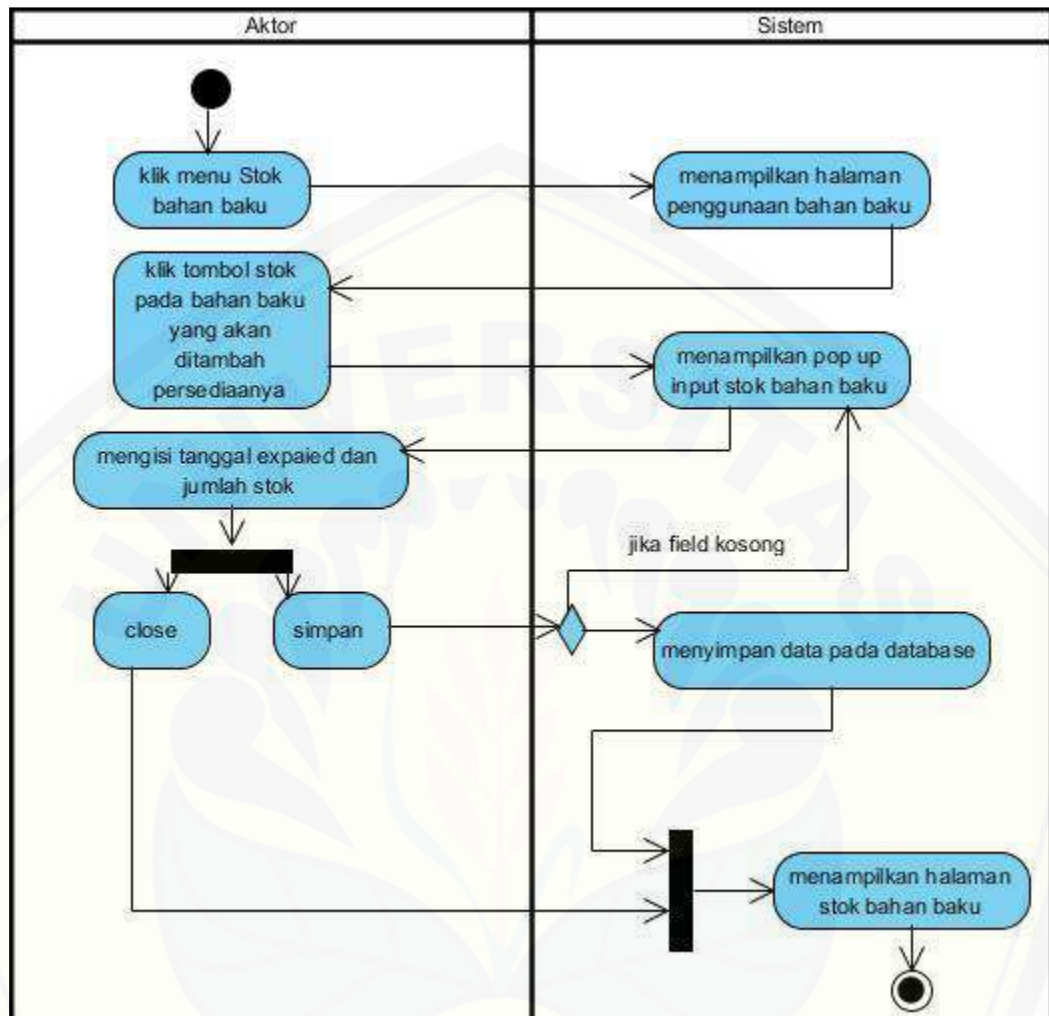
6. Klik simpan

---

7. Menampilkan *notification*  
“*please fill out this field*” pada  
*field* yang kosong.

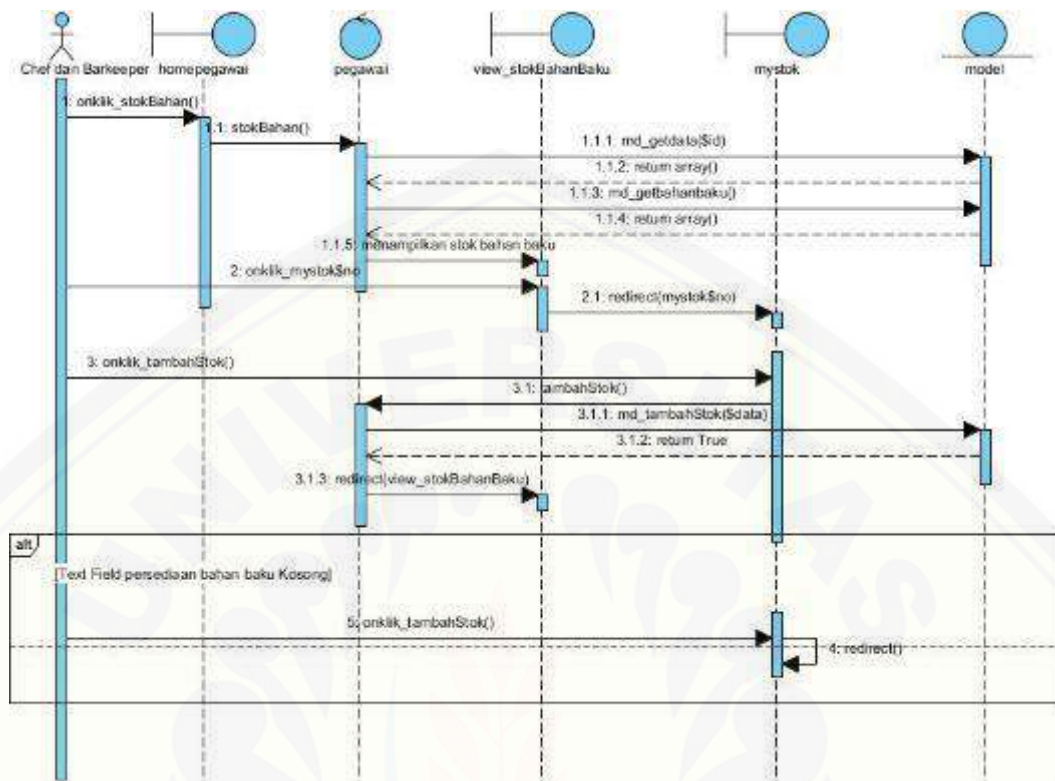
---

Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fiturmenambahkan data bahan baku merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses menambahkan data bahan baku. Penjelasan *activity diagram* fitur menambahkan data bahan baku dapat dilihat pada gambar 4.65



Gambar 4. 65 Activity Diagram menambahkan persediaan bahan baku

Tahap selanjutnya yaitu pembuatan *sequence diagram*. *Sequence diagram* fitur menambahkan data bahan baku menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses menambahkan data bahan baku. Penjelasan *sequence diagram* fitur menambahkan data bahan baku dapat dilihat pada gambar 4.66



Gambar 4. 66Sequence Diagram menambahkan persediaan baha baku

27) *Logout*

*Logout* merupakan fitur yang digunakan apabila ingin meninggalkan sistem. Fitur ini menjamin keamanan data diri karena ketika keluar dari sistem, data *session* yang berjalan bersamaan dengan sistem akan ikut terhapus. Fitur ini dapat diakses oleh semua aktor pada sistem. Desain sistem yang lebih detail untuk *logout* digambarkan melalui *scenario*. *Scenario logout* menjelaskan tentang interaksi yang dilakukan oleh aktor dan respon yang dihasilkan oleh sistem pada fitur *logout* seperti pada tabel 4.29

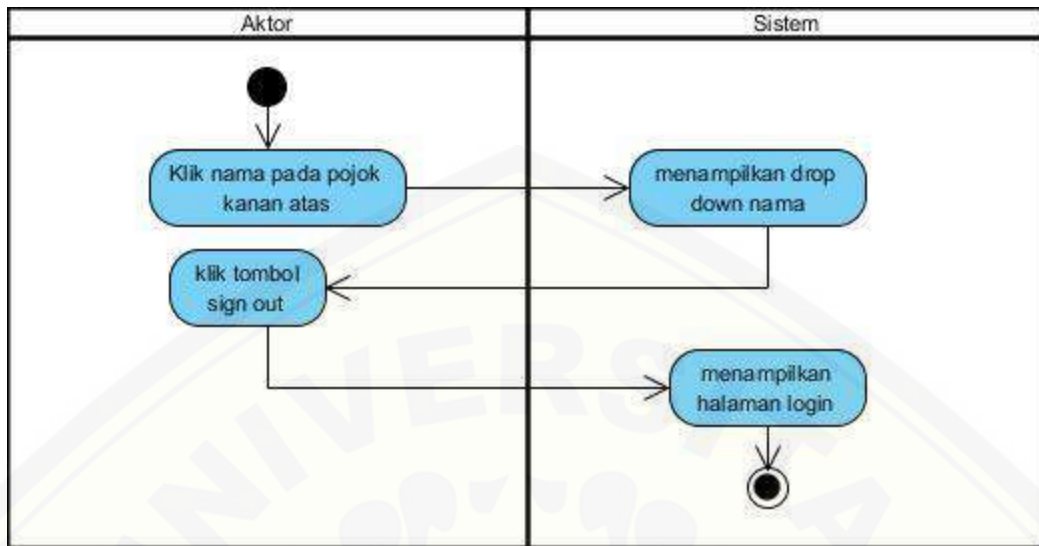
Tabel 4. 29Scenario logout

|                    |              |
|--------------------|--------------|
| <b>ID Skenario</b> | <b>USC27</b> |
| Usecase            | Log Out      |
| Partisipan Aktor   | Semua Aktor  |

|   |   |
|---|---|
| Pre-Kondisi   | Aktor sudah melakukan login.                                |
| Pra-Kondisi   | Aktor sudah keluar dari sistem                              |
| <b>SKENARIO UTAMA</b>                                   |   |
| <b>“Log Out”</b>  |   |
| <b>Aksi Aktor</b>                                       | <b>Reaksi Sistem</b>  |
| 1. Klik nama user yang berada pada pojok kanan tampilan | 2. Menampilkan Dropdown dengan tombol profile dan sign out. |
| 3. Klik tombol sign out.                                | 4. Menampilkan Halaman login                                |

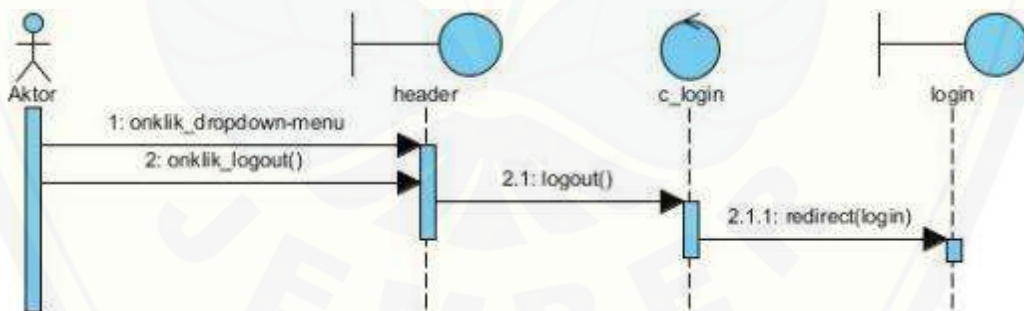
Dari *scenario* yang telah dibuat, maka desain sistem dilanjutkan pada tahap pembuatan *activity diagram*. *Activity diagram* fitur *logout* merupakan gambaran alur interaksi yang dilakukan aktor dan respon sistem yang dihasilkan dalam bentuk diagram pada saat melakukan proses *logout*. Penjelasan *activity diagram* fitur *logout* dapat dilihat pada gambar 4.67





Gambar 4. 67Activity Diagram Logout

Tahap selanjutnyayaitu pembuatan *sequence diagram*. *Squence diagram* fitur *logout* menggambarkan alur yang dilakukan dalam sistem untuk menghasilkan *output* yang diharapkan oleh *user* saat melakukan proses *logout*. Penjelasan *sequence diagram* fitur *logout* dapat dilihat pada gambar 4.68



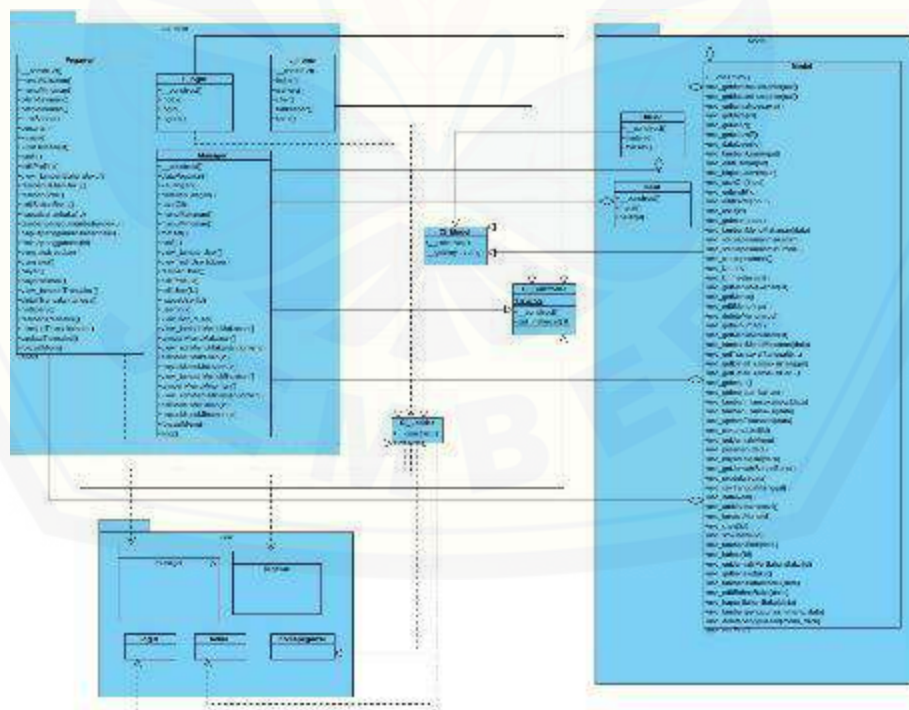
Gambar 4. 68Sequence Diagram Logout

### 4.3.3 Class Diagram

Setelah melalui tahap desain dengan *sequence diagram*, tahap selanjutnya adalah tahap pembuatan *class diagram*. *Class diagram* menggambarkan hubungan antar *class* dalam membangun sistem. *Class* yang terdapat dalam sistem Manajemen

Kebutuhan Ekonomis Roots Cafe terdiri dari kelas *view*, *controller*, dan *model*. Dalam *class view* terdapat beberapa file seperti *view login*, *view home*, *view homepegawai*, *Headerowner* dan *Hederpegawai*, *view* yang berhubungan dengan manager terdapat dalam *folder manager*, dan *view* yang berhubungan dengan pegawai terdapat pada *folderpegawai*. Dalam kelas *controller* juga terdapat beberapa tabel yaitu *controller c\_login*, *controller c\_home*, *controller manager*, dan *controllerpegawai*. Sedangkan pada kelas *model* juga terdapat beberapa tabel yaitu *table model*, tabel hitung, tabel hasil, khusus tabel hitung untuk perhitungan metode *single moving average* dan tabel hasil untuk perhitungan metode *Economic Order Quantity* sedangkan untuk tabel model berisi semua *query* yang digunakan dalam sistem.

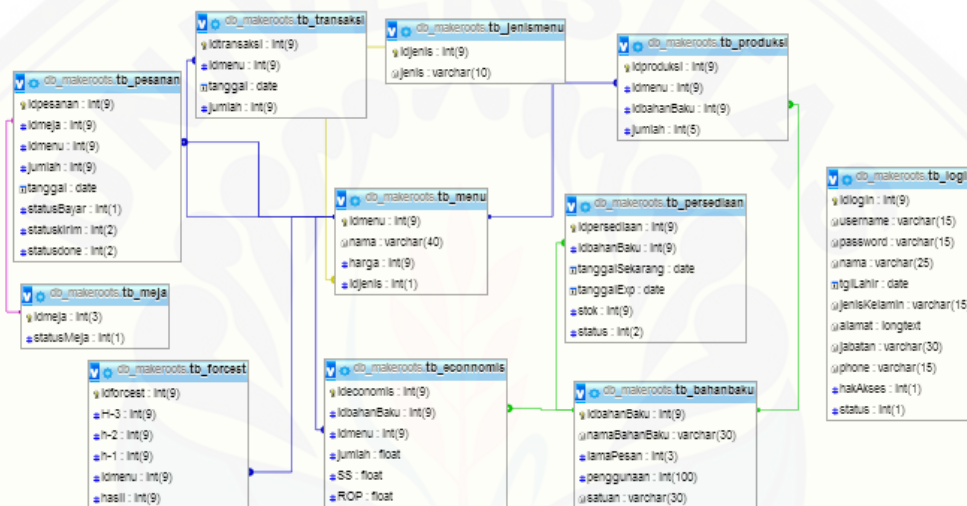
*Class diagram* manajemen kebutuhan ekonomis Roots Cafe yang mengimplementasikan metode *single moving average* dan *economic order quantity* pada Roots Cafe digambarkan lebih jelas pada Gambar 4.69



Gambar 4. 69Class Diagram Sistem manajemen kebutuhan

#### 4.3.4 Entity Relationship Diagram(ERD)

*Entity Relationship Diagram(ERD)* merupakan gambaran dari komponen-komponen yang terdapat dalam database sistem manajemen kebuuhan ekonomis Roots Cafe. *Entity Relationship Diagram* juga menggambarkan relasi yang terbentuk antar table sehingga setiap table dapat saling berhubungan satu sama lain dalam database. Gambaran *Entity Relationship Diagram* dijelaskan pada Gambar 4.70.



Gambar 4. 70Entity Relationship Diagram SiMaKERoots

#### 4.4 Implementasi

Setelah tahap desain perancangan sistem selesai dilakukan, tahap selanjutnya adalah tahap implementasi desain perancangan sistem ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi sistem antara lain :

1. Penulisan kode program menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter(CI)*.
2. Basis data yang digunakan untuk media penyimpanan data menggunakan *DBMS MySQL*.

Kode program perhitungan metode *single moving average* terdapat pada class *hitungfunction* forecast dan perhitungan detail terdapat pada *function* mulai, sedangkan untuk perhitungan metode *Economic Order Quantity*(EOQ) terdapat pada class *hasilfunction* mulai dan untuk daftar belanja terdapat pada class *hasil function* *belanja*.

*Function* mulai pada class *hitung* berfungsi untuk menghitung peramalan dengan metode *single moving average* setiap 3 periode sekaligus untuk menghitung *Mean Absolute Percentage Error*(MAPE) secara detail. Kode program *function* *forecast* pada class *hitung* dijelaskan pada Listing Program 4.1.

```
public function mulai($id) {
    $query = $this->db->query
    ('
        SELECT
        a.tanggal
        ,DATE(a.tanggal) AS tanggal
        , MONTH(a.tanggal) AS bulan
        , YEAR(a.tanggal) AS tahun
        , sum(a.jumlah) as jumlah
        FROM
        tb_transaksi a
        WHERE
        a.idmenu = ".$id."
        group by Month(a.tanggal), year(a.tanggal)
        order by a.tanggal asc
    ');
    $list = $query->result_array();
    if (count($list) <=2) {
        return "";
    } else {
        $table = array();
        for ($i = 0, $ln = count($list); $i < $ln; $i++) {
            $table[0][$i] = $list[$i]['tahun'];
            $table[1][$i] = $list[$i]['bulan'];
            $table[2][$i] = $list[$i]['jumlah'];
        }
    }
}
//hitung average 1
```

```

    $saksen = array();
    for ($i = 0, $ln = count($list); $i <= $ln; $i++) {
        if ($i >= 3) {
            $saksen[$i]=(($table[2][$i-3]+$table[2][$i-2]+$table[2][$i-1])/3);
        } else {
            $saksen[$i] = "";
        }
    }
    $table[3] = $saksen;
}
//hitung mape
$mape = array();
for ($i = 0, $ln = count($list); $i <=$ln; $i++) {
    if ($i >= 3 && $i < $ln) {
        if($table[2][$i]==0){
            $mape[$i] = 0;
        }else{
            $hasil = (abs($table[2][$i] - $table[3][$i])/($table[2][$i]))*100;
            $mape[$i] = $hasil;
        }
    } else if ($i == $ln ) {
        $tmp = 0;
        $n = 0;
        for ($j = 0, $ln2 = count($mape); $j < $ln2; $j++) {
            if ($j >= 3 && $j < $ln) {
                $tmp += $mape[$j];
                $n++;
            }
        }
        if($n>0){
            $mape[$i] = ($tmp / $n);
        }
        break;
    } else {
        $mape[$i] = "";
    }
}
$table[4] = $mape;
return $table;
} //tutup function

```

Listing Program 4.1 *Function* mulai dalam *class* hitung

*Function* forecast pada class hitung berfungsi untuk menghitung peramalan dengan metode single moving average pada 3 periode terakhir untuk mengetahui



peramalan pada periode selanjutnya dan tersimpan dalam database. Kode program *function* forecast pada class hitung dijelaskan pada Listing Program 4.2

```
public function forecast() {
    $ambil = $this->db->query("select idmenu from tb_menu");
    foreach ($ambil->result_array() as $row) {
        $sid = $row['idmenu'];
        $query = $this->db->query
        (
            SELECT
            a.tanggal
            ,b.nama
            ,DATE(a.tanggal) AS tanggal
            , MONTH(a.tanggal) AS bulan
            , YEAR(a.tanggal) AS tahun
            , sum(a.jumlah) as jumlah

        FROM
        tb_transaksi a join tb_menu b on a.idmenu = b.idmenu

        WHERE
        a.idmenu = ".$sid."
        group by Month(a.tanggal), year(a.tanggal)

        order by a.tanggal desc
        limit 0,3

        ');
        $list = $query->result_array();
    }
}
```

```

        if (count($list) >= 3) {
            $table = array();
            for ($i = 0, $ln = count($list); $i < $ln; $i++) {
                $table[1][$i] = $list[0]['jumlah'];
                $table[2][$i] = $list[1]['jumlah'];
                $table[3][$i] = $list[2]['jumlah'];
            }
        }
        //hitung average 1
        for ($i = 0, $ln = 3; $i < $ln; $i++) {
            $saksen = (($table[1][$i]+$table[2][$i]+$table[3][$i])/3);
            //rumus SMA
        }
        $tablex [4] = $saksen;
    }
    $cek = $this->db->query("select count(idmenu) as kolom from tb_forcast");
    $jmldata = $this->db->query("select count(idmenu) as jumlah from tb_menu");

    $hasil = '0';
    foreach ($jmldata ->result_array() as $a) {
        $hasil = $a['jumlah'];
    }
    foreach ($cek ->result_array() as $row) {
        if ($row['kolom']<$hasil){
            $hapus = $this->db->query("Truncate tb_forcast");
            $peramalan = $this->db->query("INSERT INTO `tb_forcast`
            (`idforcast`, `h-3`, `h-2`, `h-1`, `idmenu`, `hasil`)
            VALUES ('
            , ".$table[3][0]."
            , ".$table[2][0]."
            , ".$table[1][0]."
            , '$id'
            , '$saksen')");
        }
        }else{
            $peramalan = $this->db->query("UPDATE `tb_forcast` set
            `h-3` = ".$table[3][0]."
            , `h-2` = ".$table[2][0]."
            , `h-1` = ".$table[1][0]."
            , `hasil` = '$saksen' where idmenu = '$id'");
            }
        }
    }
}

```

Listing Program 4.2 *Function forecast class* hitung

*Function* mulai pada class hasil berfungsi untuk menghitung nilai ekonomis dengan menggunakan metode Economic Order Quantity dari hasil peramalan

penjualan setiap menunya dilakukan perhitungan berdasarkan penggunaan bahan baku sehingga ditemukan nilai kebutuhan bahan baku, nilai persediaan pengamanan(safety stok) dan nilai pemesanan kembali(Reorder point). Kode program *function* forecast pada class hitung dijelaskan pada Listing Program 4.3.

```
public function mulai() {
    $biaya = '10000';
    $simpan = '500';
    $query = $this->db->query
    ('SELECT * FROM tb_forecast ');

    foreach ($query->result_array() as $row) {
        $d = $row['hasil'];
        $idmenu = $row['idmenu'];
        $ambil = $this->db->query
        ('
        SELECT * FROM tb_produksi a join tb_bahanbaku b
        on a.idbahanBaku = b.idbahanBaku where
        idmenu = '.$idmenu.'');
        $idbahanbaku = array();
        $penggunaan = array();
        $eoq = array();
        $Z = '1.645';
        $rata = array();
        $L = array();
        $SS = array();
        $am = $ambil->result_array();
        for ($i = 0, $ln = count($am); $i < $ln; $i++) {
            $idbahanbaku[0][$i] = $am[$i]['idbahanBaku'];
            $penggunaan[1][$i] = $am[$i]['jumlah'];
        } //tutup for ambil
    } //rumus EOQ
}
```

```

        for ($j = 0, $ln = count($am); $j < $ln; $j++){
$eoq[1][$j] = sqrt((2*$biaya*$d*$penggunaan[1][$j])/$simpan);
        }

        //rumus SS dan ROP
        for ($b = 0, $ln = count($am); $b < $ln; $b++){
$rata[2][$b]=$eoq[1][$b]/29;
$cek = $this->db->query("select lamaPesan from tb_bahanbaku
where idbahanBaku = '". $idbahanbaku[0][$b]."' ");
foreach ($cek->result_array() as $key) {
$SL[3][$b]=$key['lamaPesan'];
        } //tutup foreach
    } //tutup for
    for ($c = 0, $ln = count($am); $c < $ln; $c++){
$SS[4][$c] = $Z*$rata[2][$c]*$L[3][$c];
$ROP[5][$c] = ($rata[2][$c]*$L[3][$c])+$SS[4][$c];
    }
    for ($l = 0, $ln = count($am); $l < $ln; $l++){
$cek = $this->db->query("select count(idbahanBaku)
as kolom from tb_economis where idmenu='". $idmenu."'
and idbahanBaku = '". $idbahanbaku[0][$l]."'");
$hasil = '0';
foreach ($cek->result_array() as $ini) {
$hasil = $ini['kolom'];
    }
    if ($hasil == 0){
$this->db->query("INSERT INTO `tb_economis`
(`ideconomis`,`idbahanBaku`,`idmenu`,`jumlah`,`SS`,`ROP`)
VALUES (''
, '". $idbahanbaku[0][$l]."'
, '". $idmenu."'
, "$eoq[1][$l]."'
, "$SS[4][$l]."'
, "$ROP[5][$l]."'");
    }else{
$this->db->query("UPDATE `tb_economis` set
`jumlah`= "$eoq[1][$l].", `SS`="$SS[4][$l].",
`ROP`="$ROP[5][$l]."'
where idmenu = '". $idmenu."' and
idbahanBaku = '". $idbahanbaku[0][$l]."'");
    } //tutup else if
} //tutup for 57
}
$economis = $this->db->query("SELECT SS,ROP,b.idbahanBaku,
namaBahanBaku,a.idbahanBaku,sum(jumlah) as total
FROM `tb_economis` a join tb_bahanBaku b
on a.idbahanBaku = b.idbahanBaku group by a.idbahanBaku");
return $economis;
} //tutup function

```

Listing Program 4.3 *Function* mulai dalam class hasil

*Function* belanja pada class hasil berfungsi untuk menghitung dan memanggil bahan baku yang persediaannya kurang dari kebutuhan ekonomis. Perhitungannya mengurangi persediaan bahan baku dengan hasil perhitungan metode Economic Order Quantity ditambah *Safety stok*. Kode program *function* forcast pada class hitung dijelaskan pada Listing Program 4.4.

```

public function belanja() {
    $economy = $this->db->query("SELECT * FROM `tb_economy` a join
    tb_bahanbaku b on a.idbahanBaku = b.idbahanBaku
    group by a.idbahanBaku ");
    $nilai = array();
    $buang = array();
    $p=0;
    foreach($economy->result_array() as $row) {
        $jumlah = $row['jumlah'];
        $id = $row['idmenu'];
        $idb = $row['idbahanBaku'];
        $nama = $row['namaBahanBaku'];
        $idmenu = $row['idmenu'];
        $ROP = $row['ROP'];
        $SS = $row['SS'];
        $stok = 0;
        $data= $this->db->query("SELECT sum(stok) as stok
        FROM `tb_persediaan`
        where idbahanBaku = '". $idb. "' and status = 1");
        foreach($data->row() as $key) {
            $stok = $key;
        }
        $hitung = $stok-($jumlah+$SS);
        $nilai[0][$p] = $nama;
        $nilai[1][$p] = $idb;
        $nilai[2][$p] = $hitung;
        $nilai[3][$p] = $SS;

        $nilai[4][$p] = $jumlah;
        $nilai[5][$p] = $stok;
        $nilai[6][$p] = $ROP;
        $p++; //tutup if
    }
    return $nilai;
} //tutup function

```

Listing Program 4.4 *Function* belanja class hasil

#### 4.5 Pengujian Sistem

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan



sebuah metode pengujian sistem. Hal tersebut dilakukan dengan tujuan untuk mengevaluasi apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Metode yang digunakan untuk menguji Sistem Manajemen Kebutuhan Bahan Baku Roots Cafe yaitu metode *White box* dan *Black box*.

#### 4.5.1 Metode *White Box*

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *White Box* dilakukan dengan cara menghitung independent path yaitu dengan menggunakan pengukuran kuantitatif *cyclomatic complexity*, *listing* program, penentuan jalur independen, dan *test case*. Tahapan pengujian *White Box* meliputi:

##### a. *Listing* Program

*Listing* program merupakan baris-baris kode program yang nantinya akan diuji. Setiap langkah dari kode-kode yang terdapat pada program akan diberi nomor baik menjalankan statement biasa atau penggunaan kondisi dalam program. *Listing* program yang digunakan untuk pengujian *White box* yaitu *functionmulai()* pada class hitung dapat dilihat pada Gambar 4.71, *functionforcest()* pada class hitung dapat dilihat pada Gambar 4.72, *functionmulai()* pada class hasil dapat dilihat pada Gambar 4.73, *functionbelanja()* pada class hasil dapat dilihat pada Gambar 4.74.

```
8 public function mulai($id) {
9     $query = $this->db->query
10     (
11         SELECT
12         a.tanggal
13         ,DATE(a.tanggal) AS tanggal
14         , MONTH(a.tanggal) AS bulan
15         , YEAR(a.tanggal) AS tahun
16         , sum(a.jumlah) as jumlah
17         FROM
18         tb_transaksi a
19         WHERE
20         a.idmenu = ".$id."
21         group by Month(a.tanggal), year(a.tanggal)
22         order by a.tanggal asc
23     ');
24     $list = $query->result_array();
25     if (count($list) <=2) {
26         return "";
27     } else {
28         $table = array();
29         for ($i = 0, $ln = count($list); $i < $ln; $i++) {
30             $table[0][$i] = $list[$i]['tahun'];
31             $table[1][$i] = $list[$i]['bulan'];
32             $table[2][$i] = $list[$i]['jumlah'];
33         }
34         //hitung average 1
35
36         $saksen = array();
37         for ($i = 0, $ln = count($list); $i <= $ln; $i++) {
38             if ($i >= 3) {
39                 $saksen[$i] = ($table[2][$i-3] + $table[2][$i-2] + $table[2][$i-1]) / 3;
40             } else {
41                 $saksen[$i] = "";
42             }
43             $table[3] = $saksen;
44         }
45         //hitung mape
46         $mape = array();
47         for ($i = 0, $ln = count($list); $i <= $ln; $i++) {
48             if ($i >= 3 && $i < $ln) {
49                 if ($table[2][$i] == 0) {
50                     $mape[$i] = 0;
51                 } else {
52                     $hasil = (abs($table[2][$i] - $table[3][$i]) / $table[2][$i]) * 100;
53                     $mape[$i] = $hasil;
54                 }
55             } else if ($i == $ln) {
56                 $tmp = 0;
57                 $n = 0;
58                 for ($j = 0, $ln2 = count($mape); $j < $ln2; $j++) {
59                     if ($j >= 3 && $j < $ln) {
60                         $tmp += $mape[$j];
61                     }
62                 }
63                 $n++;
64             }
65         }
66     }
67 }
```

```
62     }
63   }
64   if($n>0){
65
66       $mape[$i] = ($tmp / $n);
67
68
69       break;
70   }
71   } else {
72       $mape[$i] = "";
73   }
74 }
75 $table[4] = $mape;
76 return $table;
77 } //tutup function
```

Gambar 4. 71 Function mulai() pada class hitung

```
80 public function forecast() {
81     $ambil = $this->db->query("select idmenu from tb_menu");
82     foreach ($ambil->result_array() as $row) {
83         $id = $row['idmenu'];
84         $query = $this->db->query
85         (
86             SELECT
87             a.tanggal
88             ,b.nama
89             ,DATE(a.tanggal) AS tanggal
90             , MONTH(a.tanggal) AS bulan
91             , YEAR(a.tanggal) AS tahun
92             , sum(a.jumlah) as jumlah
93         FROM
94         tb_transaksi a join tb_menu b on a.idmenu = b.idmenu
95         WHERE
96         a.idmenu = "'. $id.'"
97         group by Month(a.tanggal), year(a.tanggal)
98         order by a.tanggal desc
99         limit 0,3
100         ');
101     $list = $query->result_array();
102     if (count($list) >= 3) {
103         $table = array();
104         for ($i = 0, $ln = count($list); $i < $ln; $i++) {
105             $table[1][$i] = $list[0]['jumlah'];
106             $table[2][$i] = $list[1]['jumlah'];
```

```

107         $table[3][$i] = $list[2]['jumlah'];
108     }
109     //hitung average 1
110     for ($i = 0, $ln = 3; $i < $ln; $i++) {
111         $saksen = (($table[1][$i]+$table[2][$i]+$table[3][$i])/3);
112         //rumus SMA
113     }
114     $tablex [4] = $saksen;
115 }
116 $cek = $this->db->query("select count(idmenu) as kolom from tb_forcest");
117 $jmldata = $this->db->query("select count(idmenu) as jumlah from tb_menu");
118
119 $hasil = '0';
120 foreach ($jmldata ->result_array() as $a) {
121     $hasil = $a['jumlah'];
122 }
123 foreach ($cek ->result_array() as $row) {
124     if ($row['kolom']<$hasil){
125         $hapus = $this->db->query("Truncate tb_forcest");
126         $peramalan = $this->db->query("INSERT INTO `tb_forcest`
127 ('idforcest', `h-3`, `h-2`, `h-1`, `idmenu`, `hasil`)
128 VALUES ('
129     ,". $table[3][0]. "
130     ,". $table[2][0]. "
131     ,". $table[1][0]. "
132     , '$id'
133     , '$saksen')");
134
135     }else{
136     $peramalan = $this->db->query("UPDATE `tb_forcest` set
137     `h-3`= ". $table[3][0]. "
138     , `h-2`= ". $table[2][0]. "
139     , `h-1`= ". $table[1][0]. "
140     , `hasil`= '$saksen' where idmenu = '$id'");
141         //tutup else if
142         //tutup foreach cek
143     }//tutup forach
144     $queryakhir = $this ->db->query("select * from tb_forcest a
145     join tb_menu b on a.idmenu = b.idmenu ");
146     return $queryakhir;
147 }

```

Gambar 4. 72Function forcest() pada class hitung

```

8 public function mulai() {
9     $biaya = '10000';
10    $simpan = '500';
11    $query = $this->db->query
12    ('SELECT * FROM tb_forcast ');
13
14    foreach ($query->result_array() as $row) {
15        $d = $row['hasil'];
16        $idmenu = $row['idmenu'];
17        $ambil = $this->db->query
18        (
19            SELECT * FROM tb_produksi a join tb_bahanbaku b
20            on a.idbahanBaku = b.idbahanBaku where
21            idmenu = ".$idmenu.";
22        $idbahanbaku = array();
23        $penggunaan = array();
24        $eoq = array();
25        $Z = '1.645';
26        $rata = array();
27        $L = array();
28        $SS = array();
29        $am = $ambil->result_array();
30        for ($i = 0, $ln = count($am); $i < $ln; $i++) {
31            $idbahanbaku[0][$i] = $am[$i]['idbahanBaku'];
32            $penggunaan[1][$i] = $am[$i]['jumlah'];
33        } //tutup for ambil
34        //rumus EOQ
35
36        for ($j = 0, $ln = count($am); $j < $ln; $j++){
37            $eoq[1][$j] = sqrt((2*$biaya*$d*$penggunaan[1][$j])/$simpan);
38        }
39        //rumus SS dan ROP
40        for ($b = 0, $ln = count($am); $b < $ln; $b++){
41            $rata[2][$b] = $eoq[1][$b]/29;
42            $cek = $this->db->query("select lamaPesan from tb_bahanbaku
43            where idbahanBaku = ".$idbahanbaku[0][$b]." ");
44            foreach ($cek->result_array() as $key) {
45                $L[3][$b] = $key['lamaPesan'];
46            } //tutup foreach
47        } //tutup for
48        for ($c = 0, $ln = count($am); $c < $ln; $c++){
49            $SS[4][$c] = $Z*$rata[2][$c]*$L[3][$c];
50            $ROP[5][$c] = ($rata[2][$c]*$L[3][$c])+$SS[4][$c];
51        }

```



```
52 for ($l = 0, $ln = count($am); $l < $ln; $l++){
53     $cek = $this->db->query("select count(idbahanBaku)
54     as kolom from tb_economis where idmenu='".$sidmenu.'"
55     and idbahanBaku='".$idbahanbaku[0][$l]."'");
56     $hasil = '0';
57     foreach ($cek ->result_array() as $ini) {
58         $hasil = $ini['kolom'];
59     }
60     if ($hasil == 0){
61         $this->db->query("INSERT INTO `tb_economis`
62         (`ideconomis`,`idbahanBaku`,`idmenu`,`jumlah`,`SS`,`ROP`)
63         VALUES ('
64             ,".$idbahanbaku[0][$l].".
65             ,".$sidmenu.".
66             ,$.Seq[1][$l].
67             ,$.SS[4][$l].
68             ,$.ROP[5][$l].")");
69     }
70     }else{
71         $this->db->query("UPDATE `tb_economis` set
72         `jumlah`= ".$Seq[1][$l].", `SS`=".$SS[4][$l].",
73         `ROP`=".$ROP[5][$l].
74         where idmenu = ".$sidmenu.'" and
75         idbahanBaku = ".$idbahanbaku[0][$l]."'");
76     }
77     } //tutup else if
78     } //tutup for 57
79 }
80 $seconomis = $this->db->query("SELECT SS,ROP,b.idbahanBaku,
81 namaBahanBaku,a.idbahanBaku,sum(jumlah) as total
82 FROM `tb_economis` a join tb_bahanBaku b
83 on a.idbahanBaku = b.idbahanBaku group by a.idbahanBaku");
84 return $seconomis;
85 } //tutup function
```

Gambar 4. 73Function mulai() pada class hasil

```

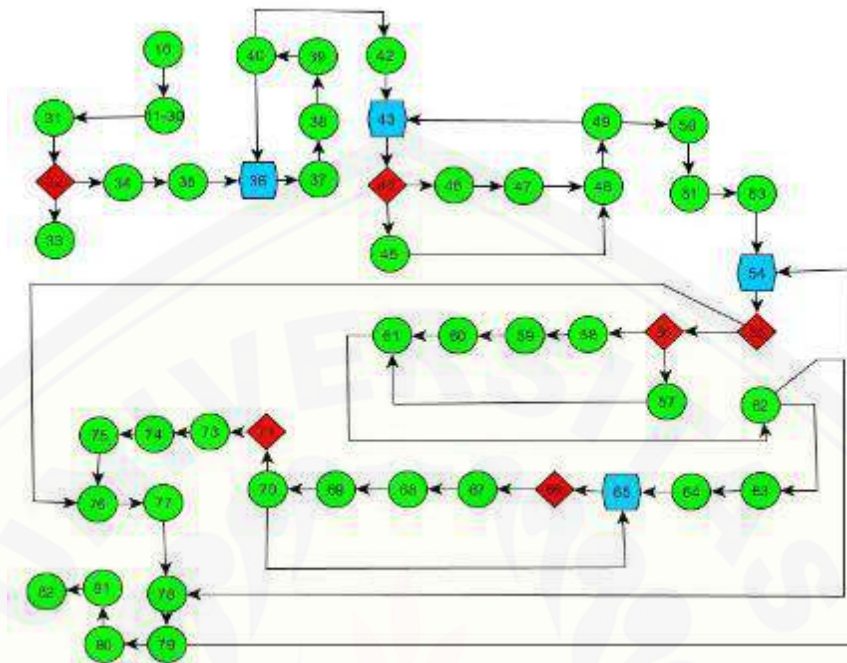
87 public function belanja() {
88     $economy = $this->db->query("SELECT * FROM `tb_economy` a join
89     tb_bahanbaku b on a.idbahanBaku = b.idbahanBaku
90     group by a.idbahanBaku ");
91     $nilai = array();
92     $buang = array();
93     $p=0;
94     foreach($economy->result_array() as $row) {
95         $jumlah = $row['jumlah'];
96         $id = $row['idmenu'];
97         $idb = $row['idbahanBaku'];
98         $nama = $row['namaBahanBaku'];
99         $idmenu = $row['idmenu'];
100        $ROP = $row['ROP'];
101        $SS = $row['SS'];
102        $stok = 0;
103        $data= $this->db->query("SELECT sum(stok) as stok
104        FROM `tb_persediaan`
105        where idbahanBaku = '". $idb. "' and status = 1");
106        foreach($data->row() as $key) {
107            $stok = $key;
108        }
109        $hitung = $stok-($jumlah+$SS);
110        $nilai[0][$p] = $nama;
111        $nilai[1][$p] = $idb;
112        $nilai[2][$p] = $hitung;
113        $nilai[3][$p] = $SS;
114
115        $nilai[4][$p] = $jumlah;
116        $nilai[5][$p] = $stok;
117        $nilai[6][$p] = $ROP;
118        $p++; //tutup if
119    }
120    return $nilai;
121 } //tutup function

```

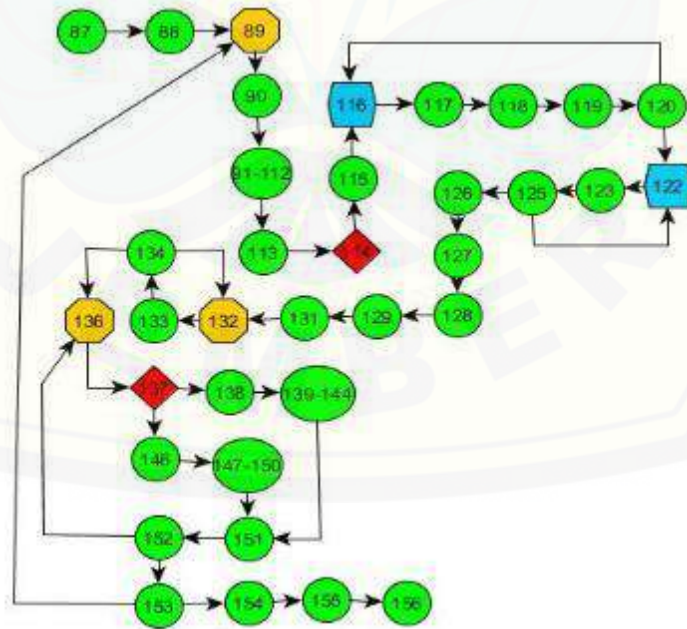
Gambar 4. 74Function belanja pada class hasil

#### b. Data Alir

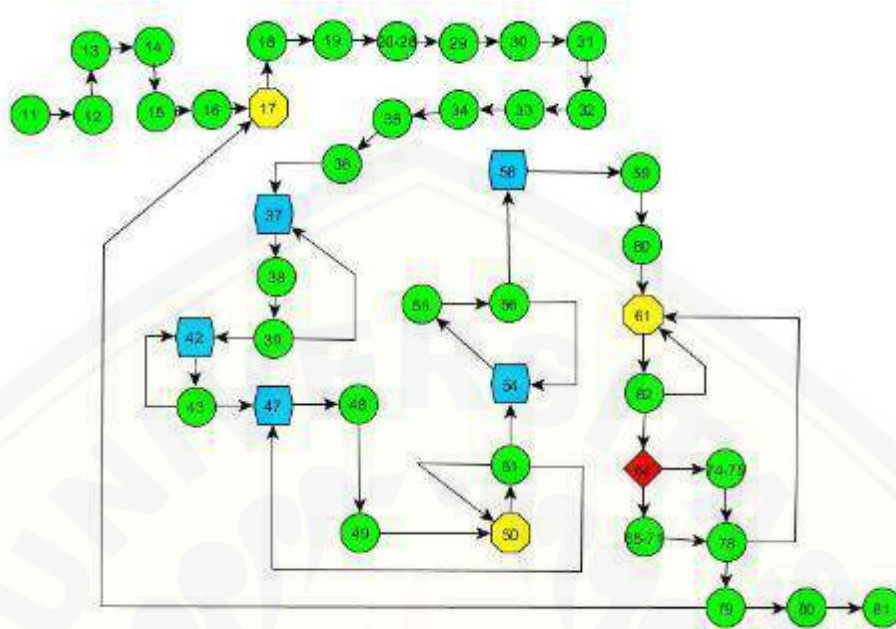
Diagram alir merupakan notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Diagram alir yang digunakan untuk pengujian *White box* yaitu *function* mulai() pada class hitung dapat dilihat pada Gambar 4.75, *function* forcest() pada class hitung dapat dilihat pada Gambar 4.76, *function* mulai() pada class hasil dapat dilihat pada Gambar 4.77, *function* belanja() pada class hasil dapat dilihat pada Gambar 4.78.



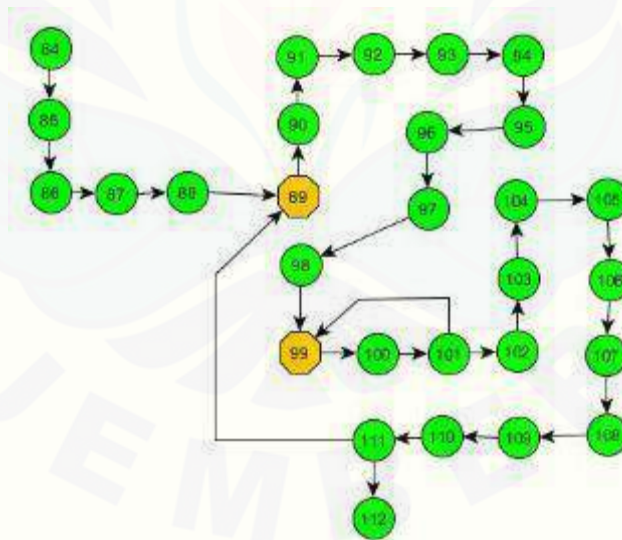
Gambar 4. 75 Data alir function mulai() pada class hitung



Gambar 4. 76 Data alir function forcest() pada class hitung



Gambar 4. 77Data alir function mulai() pada class hasil



Gambar 4. 78Data alir function belanja() pada class hasil

c. *Cyclomatic complexity*

*Cyclomatic complexity* merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam

konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jumlah jalur independen dalam basis set suatu program. *Cyclomatic complexity* dapat dihitung dengan menggunakan persamaan (4.1).

$$V(G) = E - N + 2 \dots \dots \dots \text{Persamaan(4.1)}$$

Keterangan :

$V(G)$  : *Cyclomatic complexity*

$E$  : Jumlah *edge*

$N$  : Jumlah *node*

Perhitungan *Cyclomatic complexity* dari listing program dan diagram alir yang telah dibuat adalah sebagai berikut :

1. Function mulai() class hitung

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 58 - 51 + 2 \\ &= 9 \end{aligned}$$

2. Function forcest() class hitung

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 41 - 36 + 2 \\ &= 7 \end{aligned}$$

3. Function mulai() class hasil

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 51 - 43 + 2 \\ &= 10 \end{aligned}$$

4. Function belanja() class hasil

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 30 - 29 + 2 \\ &= 3 \end{aligned}$$

- d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan



atau kondisi baru. Bila dinyatakan dalam grafikalir, jalur independen harus bergerak setidaknya sepanjang satu edge yang belum dilintasi sebelum jalur tersebut didefinisi. Jalur independen dari *function* yang digunakan untuk pengujian adalah sebagai berikut :

1) Function mulai() class hitung

Jalur 1 = 10-11-31-32-33

Jalur 2 = 10-11-31-32-34-35-36-37-38-39-40-41-42-43-44-45-48-49-50-51-53-54-55-76-77-78-79-80-81-82

Jalur 3 = 10-11-31-32-34-35-36-37-38-39-40-42-43-44-45-48-49-50-51-53-54-55-56-57-61-62-63-64-65-66-67-68-69-70-71-73-74-75-76-77-78-79-80-81-82

Jalur 4 = 10-11-31-32-34-35-36-37-38-39-40-42-43-44-45-48-49-50-51-53-54-55-56-58-59-60-61-62-63-64-65-66-67-68-69-70-71-73-74-75-76-77-78-79-80-81-82

2) Function forcest() class hitung

Jalur 1 = 87-88-89-90-91-113-114-115-116-117-118-119-120-122-123-125-126-127-128-129-131-132-133-134-136-137-138-139-151-152-153-154-155-156

Jalur 2 = 87-88-89-90-91-113-114-115-116-117-118-119-120-122-123-125-126-127-128-129-131-132-133-134-136-137-146-151-152-153-154-155-156

3) Function mulai() class hasil

Jalur 1 = 10-11-12-13-14-15-16-17-18-19-20-29-30-31-32-33-34-35-36-38-39-42-43-47-48-49-50-51-54-55-56-58-59-60-61-62-64-65-78-79-80-81

Jalur 2 = 10-11-12-13-14-15-16-17-18-19-20-29-30-31-32-33-34-35-36-38-39-42-43-47-48-49-50-51-54-55-56-58-59-60-61-62-64-74-78-79-80-81

4) Function belanja() class hasil

Jalur 1 = 84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112

e. Pengujian Basis Set

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Pengujian basis set untuk *function* mulai() class hitung dapat dilihat pada Tabel 4.30, *function* forecast() class hitung dapat dilihat pada Tabel 4.31, *function* mulai() class hasil dapat dilihat pada Tabel 4.32, *function* belanja() class hasil dapat dilihat pada Tabel 4.33.

Tabel 4. 30 Test case function mulai() class hitung

| <b><i>Test case function mulai() class hitung</i></b> |   |
|---|---|
| <b>Jalur 1</b>  |   |
| <b><i>Test Case</i></b>                               | Melakukan peramalan apabila jumlah data tidak sesuai dengan peramalan yaitu minimal 3 periode |
| <b>Target yang diharapkan</b>                         | Menampilkan data kosong   |
| <b>Hasil pengujian</b>                                | Benar   |
| <b>Path/Jalur</b>                                     | 10-11-31-32-33  |
| <b>Jalur 2</b>  |   |
| <b><i>Test Case</i></b>                               | Sistem memberikan nilai peramalan dan mape kosong pada data kurang dari sama dengan 3         |
| <b>Target yang diharapkan</b>                         | Mengirim data peramalan dan mape kosong pada periode kurang dari sama dengan 3                |
| <b>Hasil pengujian</b>                                | Benar   |
| <b>Path/Jalur</b>                                     | 10-11-31-32-34-35-36-37-38-39-40-41-42-43-44-45-48-49-50-51-53-54-55-76-77-78-79-80-81-82     |
| <b>Jalur 3</b>  |   |
| <b><i>Test Case</i></b>                               | Sistem menampilkan nilai mape pada data real sama dengan 0                                    |
| <b>Target yang diharapkan</b>                         | Mengirim data dengan mape sama dengan 0 ketika data real sama dengan 0                        |
| <b>Hasil pengujian</b>                                | Benar   |

|                               |  |
|-------------------------------|--|
| <b>Path/Jalur</b>             | 10-11-31-32-34-35-36-37-38-39-40-42-43-44-45-48-49-50-51-53-54-55-56-57-61-62-63-64-65-66-67-68-69-70-71-73-74-75-76-77-78-79-80-81-82       |
| <b>Jalur 4</b>                |  |
| <b>Test Case</b>              | Sistem melakukan perhitungan dan menampilkan peramalan dengan nilai peramalan dan nilai mape sesuai perhitungan                              |
| <b>Target yang diharapkan</b> | Menampilkan peramalan dan nilai mape sesuai perhitungan  |
| <b>Hasil pengujian</b>        | Benar  |
| <b>Path/Jalur</b>             | 10-11-31-32-34-35-36-37-38-39-40-42-43-44-45-48-49-50-51-53-54-55-56-58-59-60-61-62-63-64-65-66-67-68-69-70-71-73-74-75-76-77-78-79-80-81-82 |

Tabel 4. 31 Test case function forcest() class hitung

| <b>Test case function forcest() class hitung</b> |  |
|--|--|
| <b>Jalur 1</b>                                   |  |
| <b>Test Case</b>                                 | Sistem menyimpan hasil perhitungan peramalan pada data 3 periode terakhir apabila jumlah data pada tb_forcest lebih kecil daripada jumlah menu |
| <b>Target yang diharapkan</b>                    | Data tersimpan pada tabel forcest  |
| <b>Hasil pengujian</b>                           | Benar  |
| <b>Path/Jalur</b>                                | 87-88-89-90-91-113-114-115-116-117-118-119-120-122-123-125-126-127-128-129-131-132-133-134-136-137-138-139-151-152-153-154-155-156             |
| <b>Jalur 2</b>                                   |  |
| <b>Test Case</b>                                 | Sistem merubah data hasil peramalan pada data 3 periode terakhir   |
| <b>Target yang diharapkan</b>                    | Data tersimpan pada tabel forcest  |
| <b>Hasil pengujian</b>                           | Benar  |
| <b>Path/Jalur</b>                                | 87-88-89-90-91-113-114-115-116-117-118-119-120-122-123-125-126-127-128-129-131-132-133-134-136-137-146-151-152-153-154-155-156                 |

Tabel 4. 32 Test case function mulai() class hasil

| <b>Test case function mulai() class hasil</b> |   |
|---|---|
| Jalur 1                                       |   |
| <b>Test Case</b>                              | Sistem menyimpan hasil perhitungan kebutuhan ekonomis apabilanama bahan baku dan penggunaan pada maenu belum pernah disimpan                            |
| <b>Target yang diharapkan</b>                 | Data tersimpan pada tabel economis  |
| <b>Hasil pengujian</b>                        | Benar   |
| <b>Path/Jalur</b>                             | 10-11-12-13-14-15-16-17-18-19-20-29-30-31-32-33-34-35-36-38-39-42-43-47-48-49-50-51-54-55-56-58-59-60-61-62-64-65-78-79-80-81                           |
| Jalur 2                                       |   |
| <b>Test Case</b>                              | Sistem merubah hasil perhitungan kebutuhan ekonomis data perhitungan ekonomis pada data bahan baku dan penggunaan bahan baku pernah disimpan sebelumnya |
| <b>Target yang diharapkan</b>                 | Data dirubah otomatis pada tabel economis   |
| <b>Hasil pengujian</b>                        | Benar   |
| <b>Path/Jalur</b>                             | 87-88-89-90-91-113-114-115-116-117-118-119-120-122-123-125-126-127-128-129-131-132-133-134-136-137-146-151-152-153-154-155-156                          |

Tabel 4. 33Test case function belanja() class hasil

| <b>Test case function belanja() class hasil</b> |   |
|---|---|
| Jalur 1   |   |
| <b>Test Case</b>                                | Sistem mengirim data pada controller hasil perhitungan antara stok bahan baku dikurangi jumlah dan ditambahkan safety stok, |
| <b>Target yang diharapkan</b>                   | Data ditampilkan dalam pop up di sistem   |
| <b>Hasil pengujian</b>                          | Benar   |
| <b>Path/Jalur</b>                               | 84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112                         |

#### 4.5.2 Metode *Black Box*

Metode *Black Box* digunakan untuk menguji fungsionalitas sistem yang telah dibangun. Pengujian dilakukan oleh manager dan pegawai Roots cafe. Pengujian

dilakukan pada setiap fitur pada sistem manajemen kebutuhan ekonomis Roots cafe. Hasil dari pengujian ini juga dapat digunakan untuk melihat apakah sistem sudah sesuai denganyang diharapkan. Pengujian *Black Box* dipaparkan lebih jelas pada lampiran D.





## BAB 6. PENUTUP

Bab ini berisi kesimpulan dan saran mengenai penelitian yang telah dilakukan. Kesimpulan dan saran yang dipaparkan pada bab ini diharapkan dapat menjadi acuan untuk penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan yang didapat dari penelitian yang telah dilakukan adalah sebagai berikut :

1. Penerapan metode *Single Moving Average* terhadap prediksi penjualan pada Roots Cafe sangat akurat dengan nilai MAPE yang kecil kurang dari 10% sehingga prediksi menggunakan metode tersebut sangat tepat. Metode *Economic Order Quantity* yang diterapkan dapat mengatasi masalah pengelolaan persediaan bahan baku dan mengurangi nilai kerugian cafe.
2. Sistem informasi Manajemen Roots cafe sangat membantu pegawai dan pemilik cafe dengan memberikan informasi sesuai tugas masing-masing pegawai secara termanajemen.
3. Data aktual berupa data penjualan di Roots cafe berpengaruh terhadap hasil akurasi peramalan dalam menentukan pola data. Semakin banyak data aktual semakin baik akurasi yang dihasilkan.
4. Daftar belanja yang diperoleh dari hasil perhitungan sistem sangat membantu dalam menentukan kebutuhan bahan baku yang harus dibelanjakan untuk periode selanjutnya.
5. Penggunaan dua metode *Economic Order Quantity* dan *Single Moving Average* memiliki hubungan yang berketergantungan antara metode satu dengan metode lainnya karena nilai hasil peramalan SMA digunakan pada persamaan EOQ untuk menentukan permintaan pada periode selanjutnya.

## 6.2 Saran

Adapun saran yang ditujukan untuk membantu menyelesaikan masalah yang diteliti adalah sebagai berikut :

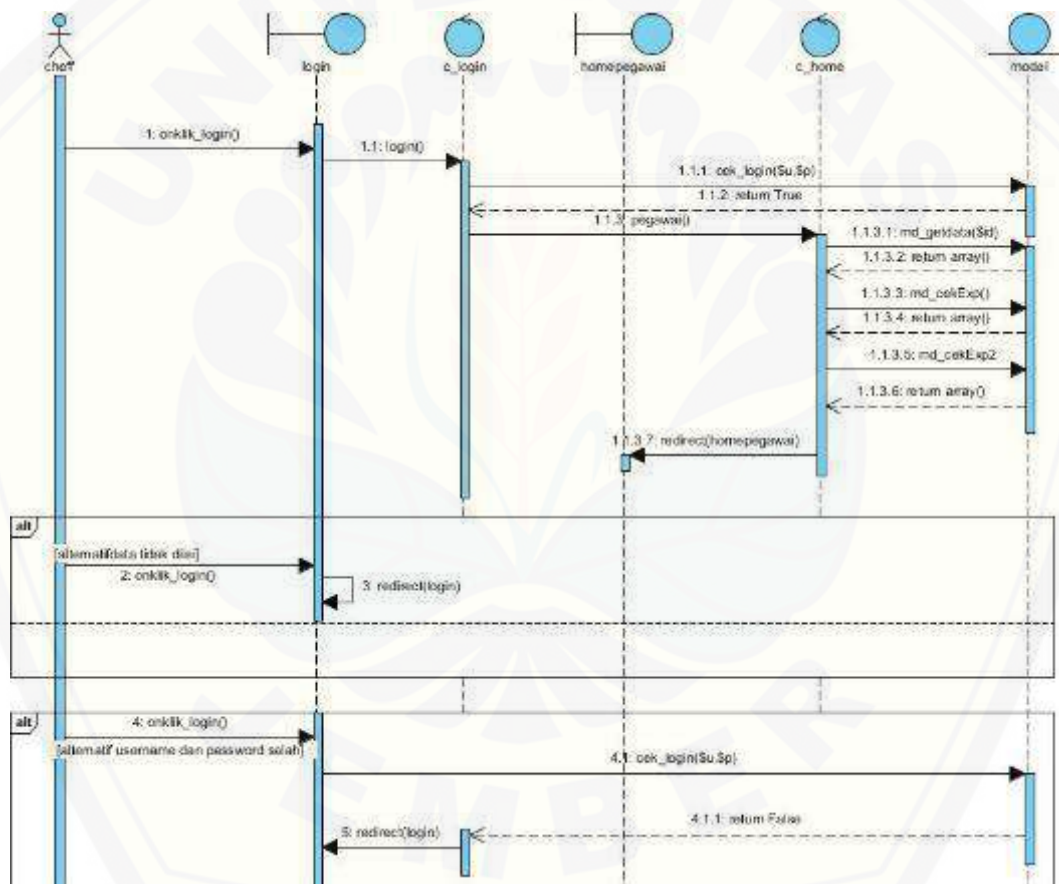
1. Sistem manajemen kebutuhan ekonomis yang dibangun harusnya terdapat fitur cetak, sehingga rekap data transaksi maupun penjualan lebih terjamin keamanannya.
2. Sistem manajemen kebutuhan ekonomis roots cafe yang dibangun sudah baik, untuk tampilannya sudah *user friendly*, akan lebih baik lagi apabila dapat mencantumkan foto diri atau data pegawai dilengkapi dengan foto, untuk menu makanan ataupun minuman harusnya ditunjukkan gambar juga agar bagian waiters mudah menjelaskan menu yang ditawarkan kepada pelanggan.
3. Dalam pengembangan sistem ditambahkan metode lain untuk memperoleh hasil peramalan yang lebih akurat dan lebih optimal apabila pola data transaksi penjualan tidak menggambarkan pola data *statistic* atau *horisontal*.

**DAFTAR PUSTAKA**

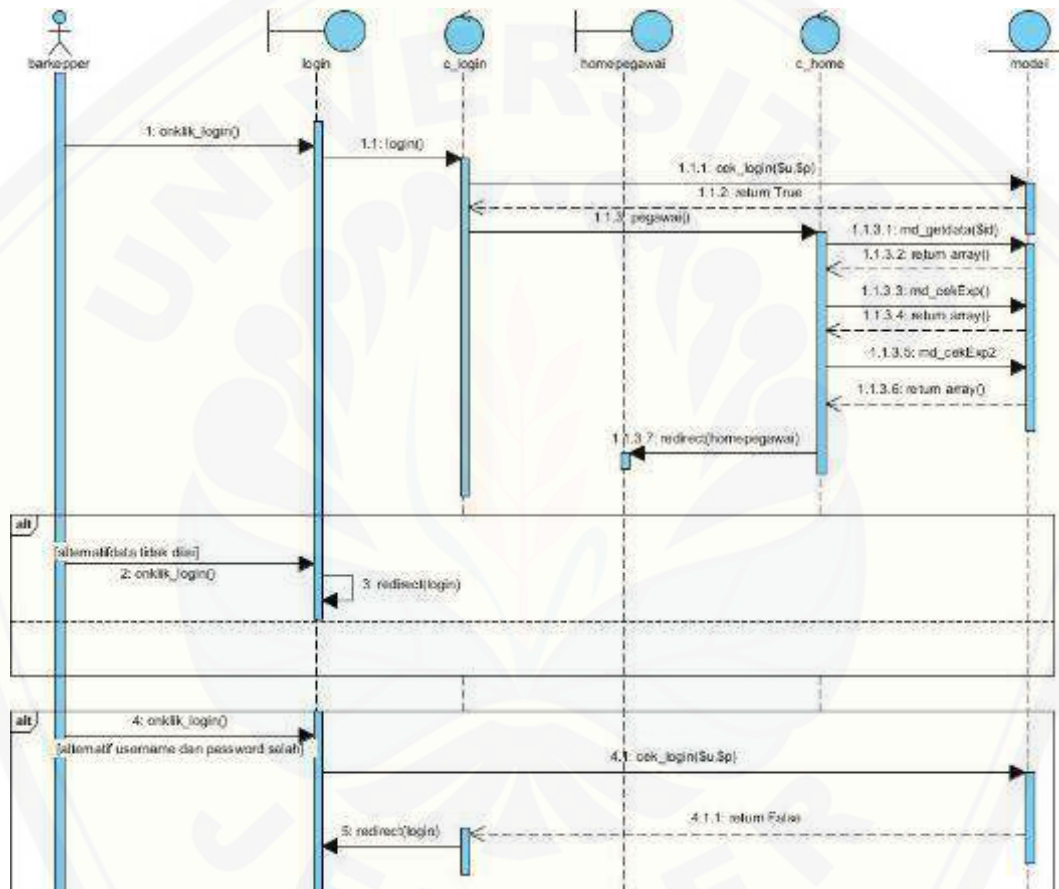
- Bassil, Y. (2012). A Simulation Model for the Waterfall. *International Journal of Engineering & Technology (iJET)* , vol 2, No 5.
- Fransisco, I. (2013). penerapan metode winters dan EOQ dalam sistem peramalan Ekspor Technical Specified Rubber untuk perencanaan kebutuhan bahan baku(Studi Kasus : PT.Bumi Asri Pasaman Buntok) . 9-31.
- Freddy, R. (2007). *Manajemen Persediaan*. Jakarta: Raja Grafindo.
- Jay, H., & Barry, R. (2005). *Operation Management*. Prentice Hall: New Jersey.
- Makridakis. (1992.). *Metode Dan Sistem Peramalan*. Jakarta: Erlangga.
- Ruauw, E. (2011). PENGENDALIAN PERSEDIAAN BAHAN BAKU. *PENGENDALIAN PERSEDIAAN BAHAN BAKU* , Volume 7 Nomor 1, 1 - 11.
- Sugiarto, I. (1996). *Pengantar Akomodasi dan Restorant* . Jakarta: PT.Gramedia Pustaka Utama.
- Tanuwijaya, H. (2010). PENERAPAN METODE WINTER'S EXPONENTIAL SMOOTHING. vol :1, 1-10.
- Tunjung. (2010). Analisis Efisiensi Pengelolaan Persediaan Bahan Baku Kedelai Pada Perusahaan Kecap Pt. Lombok Gandaria Food Industry Palur Karanganyar. *Analisis Efisiensi Pengelolaan Persediaan Bahan Baku Kedelai Pada Perusahaan Kecap Pt. Lombok Gandaria Food Industry Palur Karanganyar* , 1-3.

LAMPIRAN

C.1.1 Sequence Login aktor *cheff*

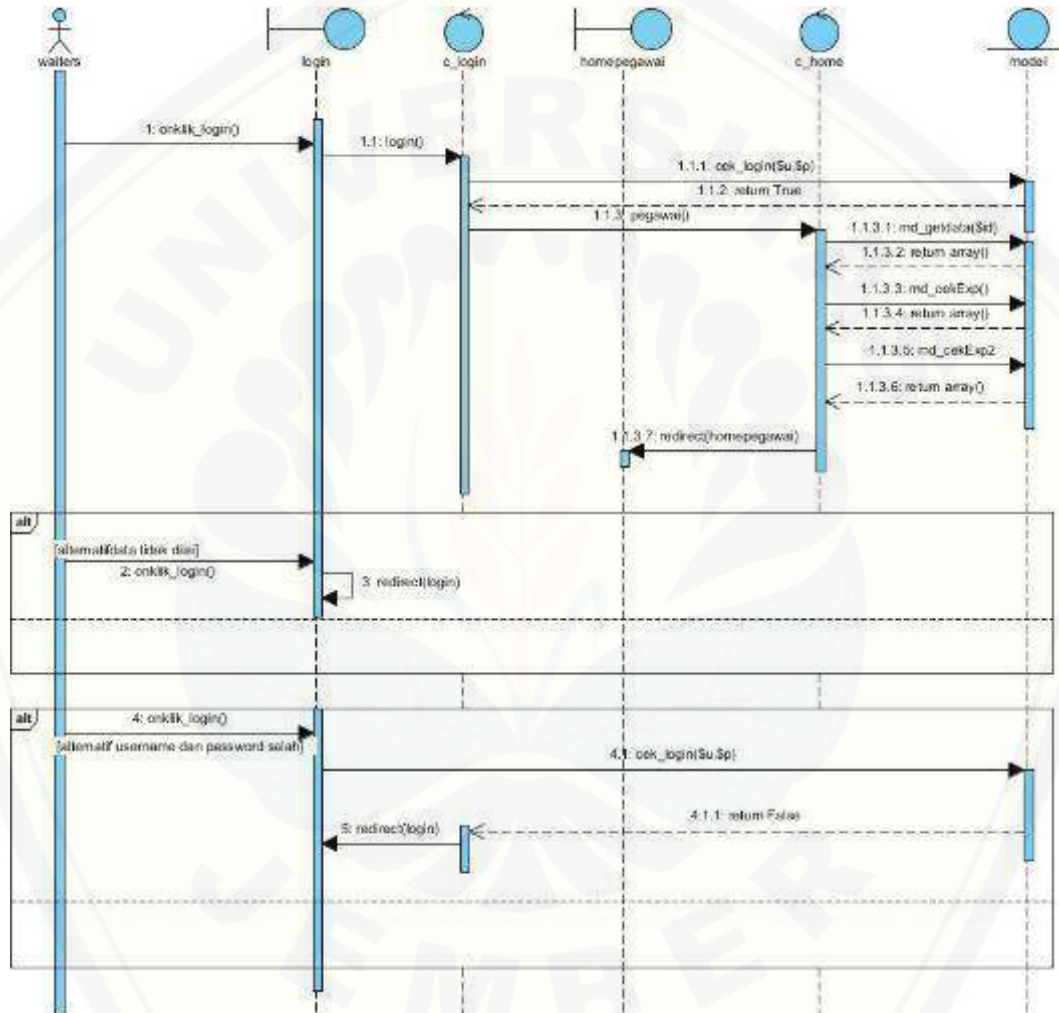


C.1.2 Sequence 1 Login aktor *Bakeeper*

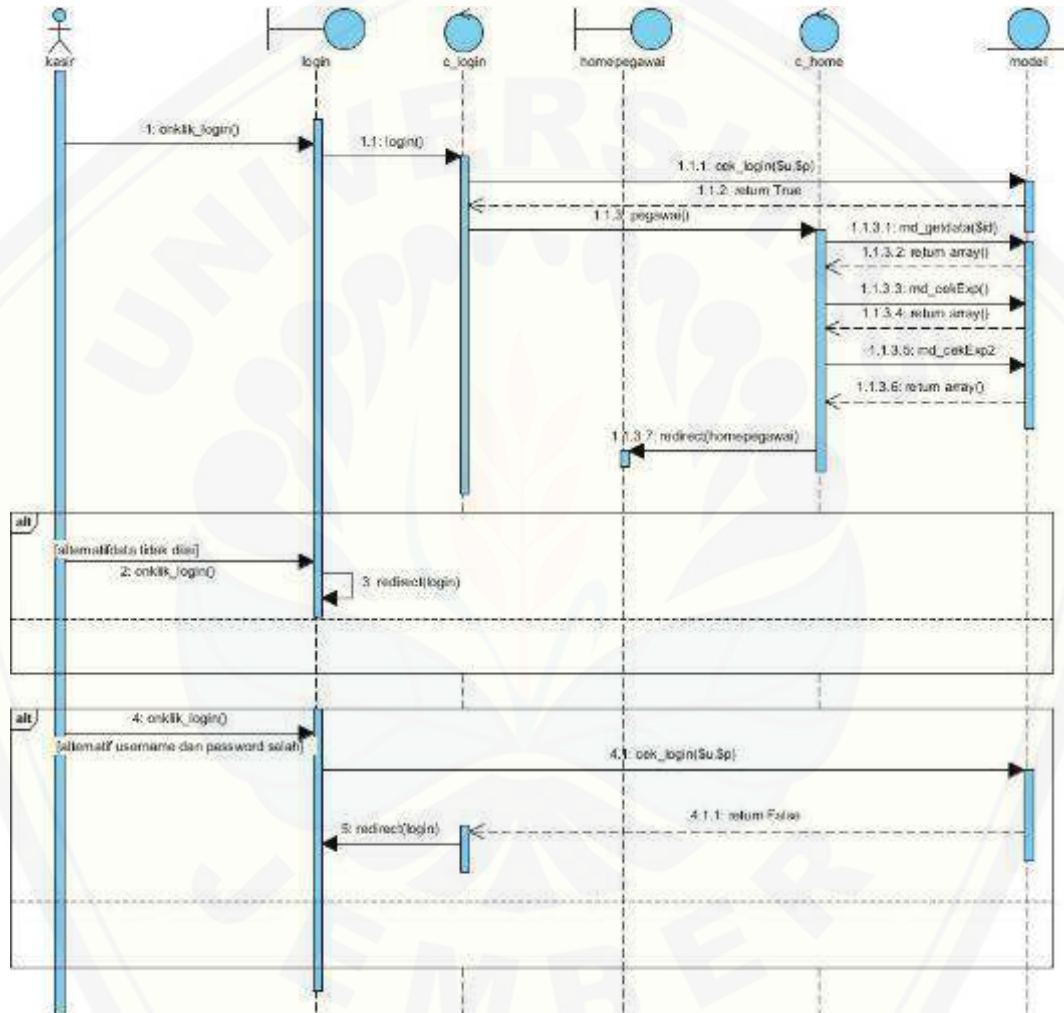




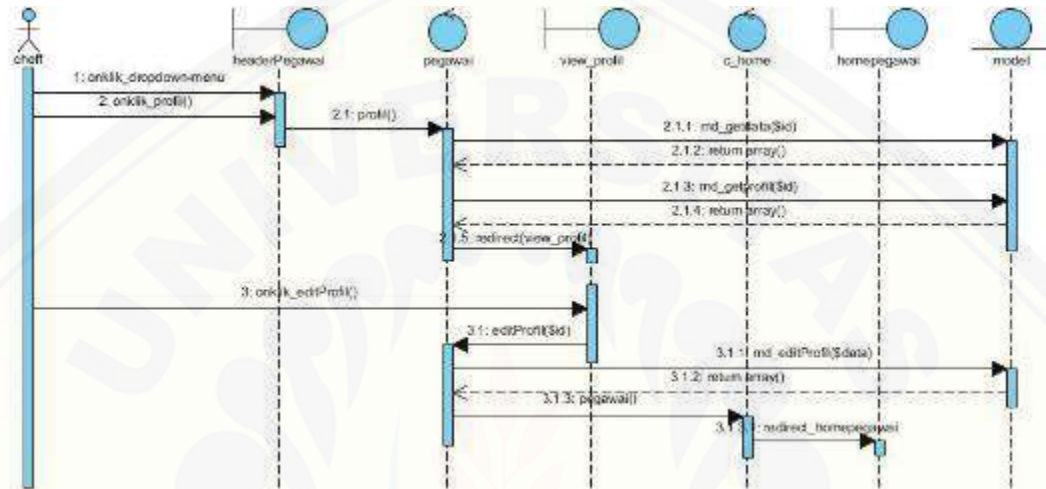
C.1.3 Sequence Login aktor *waiters*



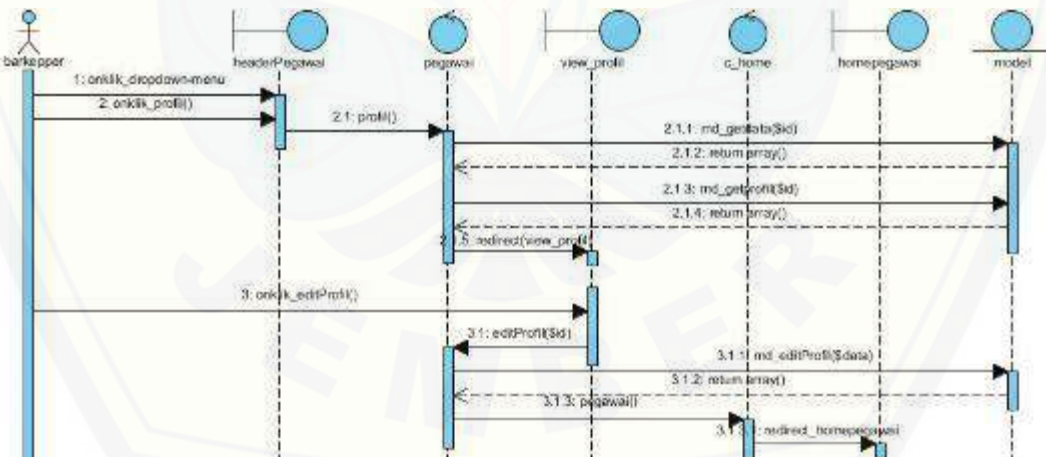
C.1.4 Sequence Login aktor kasir



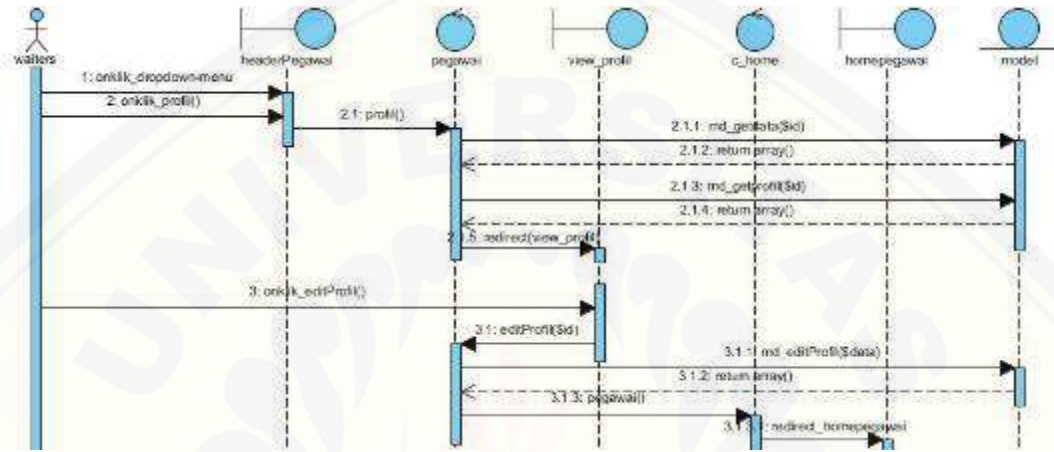
C.5.1 Sequence merubah *biodata* aktor *cheff*



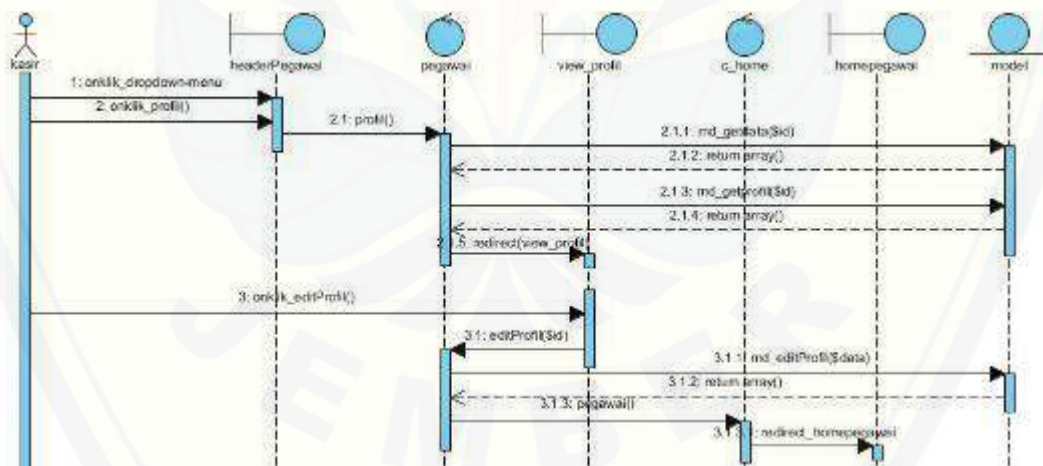
C.5.2 Sequence merubah *biodata* aktor *barkeeper*



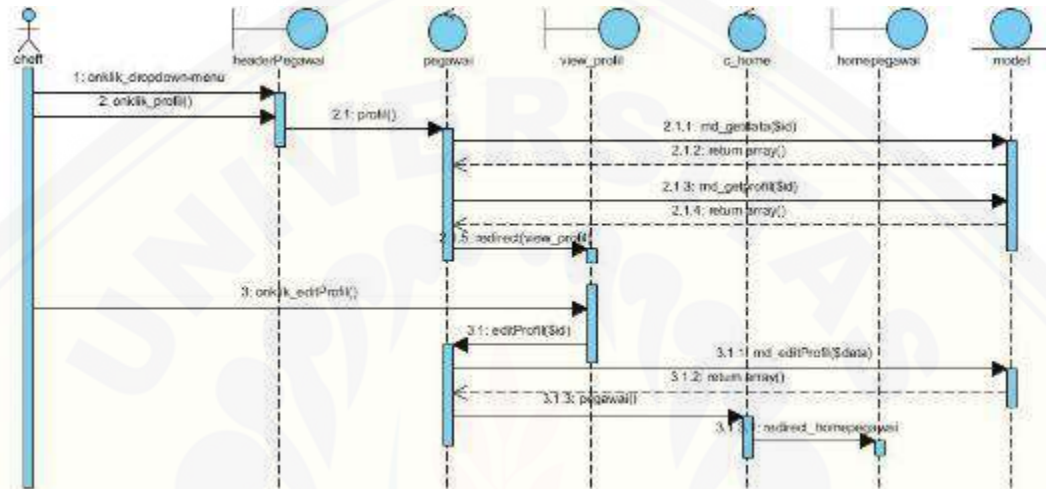
C.5.3 Sequence merubah *biodata* aktor *waiters*



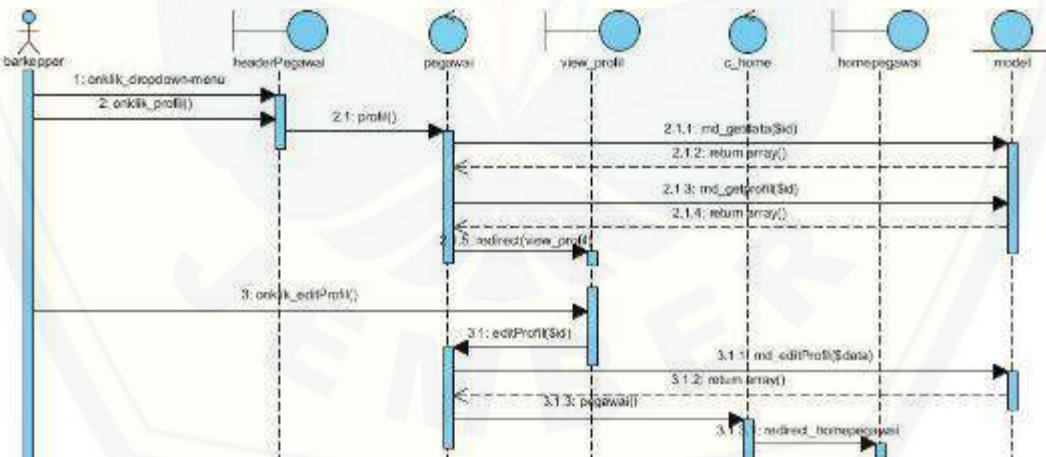
C.5.4 Sequence merubah *biodata* aktor *kasir*



C.6.1 Sequence menambahkan *biodata* aktor *cheff*

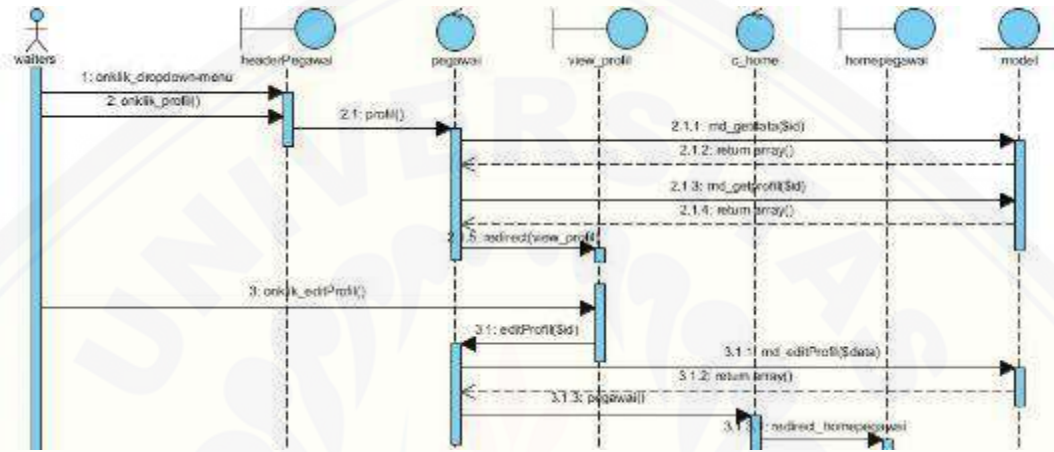


C.6.2 Sequence menambahkan *biodata* aktor *barkeeper*

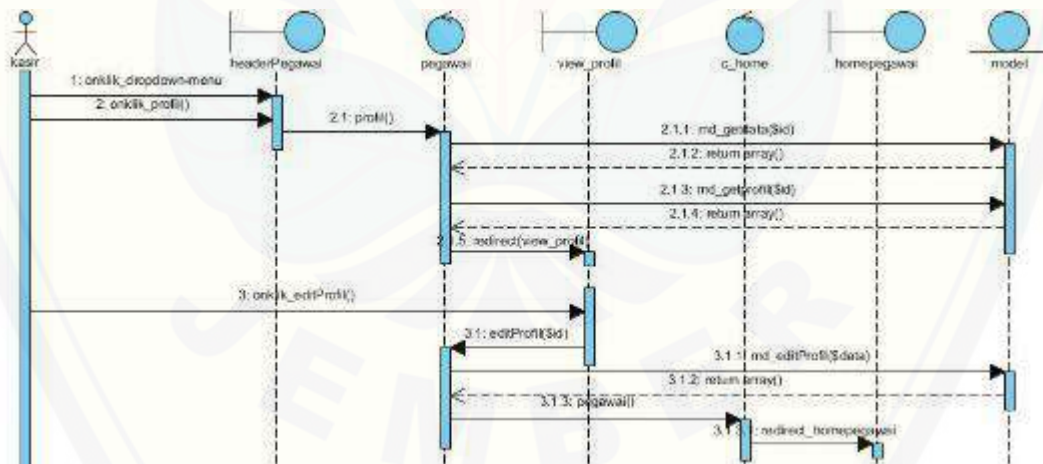




C.6.3 Sequence menambahkan *biodata* aktor *waiters*



C.6.4 Sequence menambahkan *biodata* aktor *kasir*



D. Tabel pengujian *Black Box*

| No | Menu               | Fungsi   | Hasil   | Keterangan   |
|----|--------------------|--|---|--|
| 1  | Login              | Mengisi <i>username</i> dan <i>password</i> klik tombol <i>login</i>           | Login Berhasil dan menampilkan halaman dashboard  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|    |                    | <i>Text field</i> belum diisi, klik tombol <i>login</i>                        | Menampilkan pesan "Please fill out this field." dengan menunjuk pada kolom yang belum diisi   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|    |                    | <i>Username</i> dan <i>password</i> salah, klik <i>login</i>                   | Menampilkan allert <i>username</i> dan <i>password</i> salah  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|    |                    | <i>Username</i> dan <i>password</i> dalam setatus non aktif, klik <i>login</i> | Menampilkan halaman <i>login</i> dengan allert <i>Username</i> atau <i>Password</i> Salah/ <i>Status</i> Anda non Aktif, Silahkan hubungi Owner | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 2  | Menambah data user | Mengisi <i>username</i> , <i>password</i> dan pilih jabatan klik submit        | Menyimpan data akun pegawai dan menampilkan allert user berhasil ditambah   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|   |                                       |  |   |  |
|---|---------------------------------------|--|---|--|
|   |                                       | Mengisi <i>username</i> dan pilih jabatan klik submit                      | Menampilkan halaman data pegawai dengan alert maaf, input user gagal  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 3 | Merubah data user                     | Merubah <i>username,password</i> dan jabatan, klik submit                  | Menyimpan data user yang dirubah dan menampilkan alert user berhasil dirubah  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|   |                                       | <i>Text field</i> belum diisi, klik tombol submit                          | Menampilkan halaman data pegawai dengan alert maaf, input user gagal  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 4 | Merubah status user menjadi non-aktif | Klik tombol delete berwarna merah pada user yang akan dinonaktifkan        | Merubah status pegawai menjadi 0 dan menampilkan halaman data pegawai dengan alert user berhasil di non aktifkan, silahkan lihat ada user off | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 5 | Menampilkan detail data pegawai       | Klik tombol detail berwarna hijau pada data pegawai yang ingin ditampilkan | Menampilkan halaman detail pegawai berupa tabel   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 6 | Tambah Menu                           | Mengisi <i>form</i> tambah menu, klik submit                               | Menyimpan databaru dan menampilkan halaman menu makanan/minuman dengan  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|   |                             |  |  |  |
|---|-----------------------------|--|--|--|
|   |                             |  | allert menu baru berhasil ditambahkan  |  |
|   |                             | <i>Text field</i> belum diisi, klik tombol submit      | Menampilkan pesan “Please fill out this field.” dengan menunjuk pada kolom yang belum diisi                      | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 7 | Merubah Menu                | Merubah data menu yang dipilih klik submit             | Merubah data menu pada database dan menampilkan halaman menu makanan/minuman dengan allert menu berhasil dirubah | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|   |                             | <i>Text field</i> belum diisi, klik tombol submit      | Menampilkan pesan “Please fill out this field.” dengan menunjuk pada kolom yang belum diisi                      | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 8 | Menghapus menu              | Klik tombol berwarna merah pada menu yang akan dihapus | Menghapus data didatabase dan menampilkan halaman menu dengan allert menu berhasil dihapus                       | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 9 | Menampilkan hasil peramalan | Klik menu <i>forecast</i>                              | Menampilkan peramalan <i>single moving average</i> pada transaksi 3 bulan terakhir                               | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|    |   |  |   |  |
|----|---|--|---|--|
| 10 | Menampilkan detail peramalan setiap menu  | Klik tombol <i>Detail Peramalan</i> , pilih menu yang akan ditampilkan secara detail | Menampilkan halaman detail peramalan untuk menu yang dipilih                | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 11 | Menampilkan kebutuhan ekonomis bahan baku | Klik tombol kebutuhan ekonomis   | Menampilkan halaman kebutuhan ekonomis berupa tabel                         | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 12 | Menampilkan Daftar Belanja                | Klik tombol daftar belanja   | Menampilkan pop up daftar belanja berupa tabel                              | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 13 | Menampilkan data keuangan                 | Klik menu keuangan   | Menampilkan halaman keuangan pada hari ini                                  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 14 | Menampilkan data keuangan                 | Klik tombol pilih tanggal pada halaman menu keuangan dan pilih tanggal               | Menampilkan halaman keuangan dengan data keuangan pada tanggal yang dipilih | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |



|    |                                 |   |  |  |
|----|---------------------------------|---|--|--|
|    | pada tanggal tertentu           |   |  |  |
| 15 | Menambah penggunaan bahan baku  | Klik tombol olah pada menu dan pilih bahan baku yang digunakan untuk menu yang dipilih            | Menyimpan penggunaan bahan baku untuk menu yang dipilih  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 16 | Menghapus penggunaan bahan baku | Klik tombol olah pada menu dan pilih bahan baku yang tidak lagi digunakan untuk menu yang dipilih | Menghapus penggunaan bahan baku untuk menu yang dipilih  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 17 | Tambah bahan baku               | <i>Input</i> nama bahan baku, lama pengiriman, jumlah penggunaan/satuan dan satuan, klik submit   | Menyimpan data bahan baku dan menampilkan halaman bahan baku dengan alert bahan baku baru berhasil ditambahkan | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|    |                                 | <i>Text field</i> belum diisi, klik tombol simpan   | Menampilkan pesan "Please fill out this field." dengan menunjuk pada kolom yang belum diisi                    | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 18 | Edit Bahan                      | Merubah data bahan baku klik simpan   | Merubah data bahan baku dan  | <input checked="" type="checkbox"/> Berhasil                                   |

|    |                         |   |   |  |
|----|-------------------------|---|---|--|
|    | Baku                    |   | menampilkan halaman stok bahan baku dengan allert bahan baku berhasil dirubah   | <input type="checkbox"/> Gagal   |
|    |                         | <i>Text field</i> belum diisi, klik tombol simpan                           | Menampilkan pesan “Please fill out this field.” dengan menunjuk pada kolom yang belum diisi                                       | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 19 | Menghapus bahan baku    | Klik tombol delete dan klik tombol delete lagi pada pop up                  | Menghapus penggunaan bahan baku dan bahan baku, dan menampilkan halaman stok bahan baku dengan allert bahan baku berhasil dihapus | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 20 | Tambah stok             | Mengisi <i>form</i> pada pop up tambah stok klik simpan                     | Menyimpan data persediaan bahan baku dan menampilkan halaman stok bahan baku dengan allert stok berhasil ditambah                 | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 21 | Menampilkan pesanan     | Pilih menu pesanan  | Menampilkan halaman pesanan   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 22 | Merubah setatus pesanan | Klik tombol selesai pada halaman pesanan, pada menu yang sudah diselesaikan | Merubah setatus pesanan pada database dan menampilkan halaman pesanan   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|    |                             |   |   |  |
|----|-----------------------------|---|---|--|
| 23 | Tambah transaksi            | Pilin no meja, <i>insert</i> jumlah pesanan pada menu yang dipesan dan klik simpan                  | Menyimpan pesanan dan menampilkan halaman transaksi                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 24 | Tambah tambahan transaksi   | Pilih nomer meja yang tersedia, <i>insert</i> jumlah pesanan pada menu yang dipesan dan klik simpan | Menyimpan pesanan dan menampilkan halaman transaksi                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 25 | Menampilkan Bayar Transaksi | Klik bayar pada meja yang telah selesai melakukan kunjungan   | Menampilkan pop up total transaksi yang harus dibayar <i>customer</i> | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 26 | Tambah biodata              | Mengisi data diri klik simpan   | Menyimpan data diri pada database dan menampilkan halaman dashboard   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 27 | Logout                      | Klik tombol sign out  | Menampilkan halaman login   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |