

**PENGEMBANGAN APLIKASI ANDROID UNTUK KEANGGOTAAN
PELANGGAN RUMAH MAKAN MENGGUNAKAN *NEAR FIELD
COMMUNICATION (NFC) READER* PADA E-KTP**

SKRIPSI

**Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana
Komputer**



Disusun oleh:

**Randyka Aurora Y
105090600111002**

**PROGRAM STUDI INFORMATIKA/ ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2014

LEMBAR PERSETUJUAN

PENGEMBANGAN APLIKASI ANDROID UNTUK KEANGGOTAAN PELANGGAN RUMAH MAKAN MENGGUNAKAN *NEAR FIELD COMMUNICATION (NFC) READER* PADA E-KTP

SKRIPSI

LABORATORIUM MOBILE

Untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer



Disusun Oleh :

Randyka Aurora Yudhistira

105090600111002

Telah diperiksa dan disetujui oleh dosen pembimbing

Pada tanggal 20 Juni 2014:

Dosen Pembimbing I

Dosen Pembimbing II

Aryo Pinandito, S.T., M.MT.
NIK. 83051916110374

Dr. Eng Herman Tolle, ST., MT
NIP. 197408232000121 001

LEMBAR PENGESAHAN

**PENGEMBANGAN APLIKASI ANDROID UNTUK KEANGGOTAAN
PELANGGAN RUMAH MAKAN MENGGUNAKAN *NEAR FIELD
COMMUNICATION (NFC) READER* PADA E-KTP**

SKRIPSI

LABORATORIUM MOBILE

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun Oleh :

Randyka Aurora Yudhistira

NIM. 105090600111002

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 10 Juli 2014
dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana dalam
bidang Ilmu Komputer

Penguji I

Fajar Pradana, S.ST.,M.Eng
NIK. 87112116110371

Penguji II

Agi Putra Kharisma, S.T.,M.T.
NIK.-

Penguji III

Eko Sakti Pramukantoro, S.Kom.,M.Kom
NIK. 86080506110252

Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT.
NIP. 196708011992031001

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Randyka Aurora Yudhistira
NIM : 105090600111002
Program Studi : Informatika / Ilmu Komputer
Jurusan : Ilmu Komputer
Fakultas : Program Teknologi Informasi Ilmu Komputer
Penulis skripsi berjudul : Pengembangan Aplikasi Android untuk Keanggotaan Pelanggan Rumah Makan Menggunakan *Near Field Communication (NFC) Reader* pada e-KTP

Dengan ini menyatakan bahwa :

1. Isi dari skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka skripsi ini.
2. Apabila di kemudian hari ternyata skripsi yang saya tulis terbukti hasil jiplakan, maka saya bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran dan penuh tanggung jawab dan digunakan sebagaimana mestinya.

Malang, 20 Juni 2014

Yang menyatakan,

Randyka Aurora Y
NIM. 105090600111002

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Aplikasi Android untuk Keanggotaan Pelanggan Rumah Makan Menggunakan *Near Field Communication (NFC) Reader* pada e-KTP”. Skripsi ini merupakan salah satu syarat untuk memenuhi persyaratan akademis menyelesaikan studi di program Sarjana Ilmu Komputer Universitas Brawijaya. Selama melaksanakan skripsi ini, penulis mendapat bantuan dan dukungan dari banyak pihak. Untuk itu, penulis ingin mengucapkan terima kasih kepada :

1. Papa Bambang Kuntoro, Mama Ariyati Susilorini atas kasih sayang, perhatian, dan didikannya serta doa dan semangat kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Aryo Pinandito, ST., M.MT selaku dosen pembimbing I dan Dr. Eng Herman Tolle, ST., MT sebagai dosen pembimbing II yang telah bijaksana dan sabar dalam membimbing dan menyalurkan ilmu kepada penulis serta semua waktu dan nasehat yang telah diberikan dalam proses penyelesaian skripsi ini.
3. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Informatika/Ilmu Komputer Universitas Brawijaya.
5. Seluruh Dosen Informatika/Ilmu Komputer Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Seluruh Civitas Akademika Informatika/Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di PTIIK Universitas Brawijaya dan selama penyelesaian skripsi ini.

7. Kakak dan adik – adikku; Aurora, Riansyah, dan Rika; atas bantuan kepada penulis dalam menyelesaikan skripsi ini.
8. Teman terdekat, Anis Eka Sukmadadari, S.Ked, atas kasih sayang, kesabaran, bantuan, dan semangat yang tidak ada hentinya kepada penulis dalam menyelesaikan skripsi ini.
9. Teman – teman seperjuangan kontrakan E718; Andrias, Ivan, Beta, Fitri, Wahyu, dan Toni; atas bantuannya selama 1 tahun penuh kepada penulis dalam menyelesaikan skripsi ini.
10. Teman-teman seperjuangan Angkatan 2010 Ilmu Komputer dan Informatika, terimakasih atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 12 Juli 2014

Penulis

ABSTRAK

Yudhistira, Randyka Aurora. 2014. Pengembangan Aplikasi Android untuk Keanggotaan Pelanggan Rumah Makan Menggunakan *Near Field Communication* (NFC) Reader pada e-KTP. Skripsi Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Pembimbing : AryoPinandito, ST, M.MT dan Dr. Eng Herman Tolle, ST., MT.

Perusahaan ritel makanan atau rumah makan saat ini banyak yang menerapkan program kartu keanggotaan sesuai dengan salah satu konsep dalam *Customer Relationship Management* dengan memberi keuntungan berupa potongan harga pada anggotanya. Konsep tersebut dapat menyulitkan pelanggan apabila setiap rumah makan memiliki program kartu anggota, sehingga pelanggan akan banyak memiliki kartu keanggotaan rumah makan yang berbeda-beda. Resiko kehilangan kartu anggota akan lebih besar dan pelanggan tidak mempunyai kewajiban untuk menggantinya, apalagi ditambah dengan padatnya pelanggan yang mengantri untuk mendaftar sebagai anggota pada bagian *customer service* atau kasir.

Solusi yang dapat ditawarkan adalah menggunakan e-KTP yang sudah dilengkapi dengan teknologi RFID sebagai identitas anggota rumah makan dan smartphone berbasis Android yang dilengkapi teknologi NFC untuk membaca identitas e-KTP tersebut sehingga pelayan dapat melakukan pendaftaran anggota dan transaksi rumah makan secara *mobile*. Apabila tiap rumah makan menerapkan solusi yang sama, maka pelanggan hanya membutuhkan satu kartu anggota untuk semua rumah makan di Indonesia.

Aplikasi Rumah Makan (ARuMa), aplikasi keanggotaan rumah makan yang menggunakan e-KTP Indonesia sebagai media identifikasi anggota dan melakukan transaksi-transaksi rumah makan. Hasil pengujian identitas e-KTP menunjukkan bahwa yang diterima oleh server sebagai hasil proses konversi dari client menunjukkan tidak adanya identitas kembar. Hasil pengujian validasi menunjukkan bahwa fitur-fitur yang sudah diimplementasikan dapat memenuhi kebutuhan dan sesuai dengan yang diharapkan. Berdasarkan hasil pengujian dapat disimpulkan bahwa identitas yang terbaca pada e-KTP dapat diimplementasikan sebagai identitas anggota, serta kebutuhan yang sudah diimplementasikan memberikan hasil yang sesuai dengan tahap perancangan dan layak untuk digunakan.

Kata kunci : e-KTP, *Radio Frequency Identification* (RFID), *Near Field Communication* (NFC).

ABSTRACT

***Yudhistira, Randyka Aurora. 2014. Android Application Development for Membership of Restaurant Using NFC Reader on Electronic Civil Identification Card (e-KTP) of Indonesia. Final Assignment Major in Informatics/Computer Science, Program Teknologi Informasi dan Ilmu Komputer, Brawijaya University.
Advisors : AryoPinandito, ST, M.MT and Dr. Eng Herman Tolle, ST., MT.***

In this day, almost every restaurant use membership card program that is written in Customer Relationship Management's concepts and give profit to members as a discount. This concept can be a negative effect to the costumer if every restaurants have the same program to make membership card, the customer will have different membership card for different restaurant. Risk for losing member card will be bigger and the customer desn't have the responsibility to administer new member card, moreover the customer queue on costumer service or cashier is long enough to wait.

Solution that can be offered is to make electronic civil identification of Indonesia as a member card combinig with Android mobile application with NFC technology to do membership task and restaurant's transaction by the waiters. And if every restaurants implement the same solution as above, then customer will only have one member card for every restaurants in Indonesia.

Aplikasi Rumah Makan (ARuMa), a restaurants membership application that use of an Indonesian electronic citizen identification card as a media for member identification in a restaurant and do restaurant's transactions. Identification testing results on electronic civil identification card shows that the identity sent from client to server with conversion method does not give twin value, only unique value. Validation testing result shows that implemented features fulfill the requirements and give expected results. Based on the results of the testing phase, stated that electronic civil identification card of Indonesia can be used as member identification and the requirements have been implemented and gives results that compatible with the design phase and ready for use.

Keywords: *electronic citizen identification card, Radio Frequency Identification (RFID), Near Field Communication (NFC).*

DAFTAR ISI

Kata Pengantar.....	i
Abstrak.....	iii
Abstract.....	iv
Daftar Isi.....	v
Daftar Tabel.....	viii
Daftar Gambar.....	x
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Sistematika Penulisan.....	3
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	5
2.1. Kartu Tanda Penduduk Elektronik.....	5
2.2. <i>Radio Frequency Identification (RFID)</i>	8
2.2.1. Pengertian RFID.....	8
2.2.2. Jenis- Jenis RFID.....	9
2.3. Android.....	10
2.3.1. NFC Reader.....	10
2.3.2. Intent Filter.....	12
2.3.3. Asyncrounus Task.....	13
BAB III METODOLOGI PENELITIAN.....	15
3.1. Studi Literatur.....	16



3.2. Analisis Sistem.....	16
3.3. Perancangan Sistem.....	17
3.4. Implementasi Sistem.....	17
3.5. Pengujian Sistem.....	18
3.6. Analisis Hasil Pengujian.....	19
3.7. Pengambilan Kesimpulan dan Saran.....	19
BAB IV ANALISIS DAN PERANCANGAN.....	20
4.1. Analisis Sistem.....	20
4.1.1. Gambaran Umum Sistem.....	20
4.1.2. Analisis Kebutuhan.....	21
4.1.3. Diagram <i>Use Case</i>	22
4.1.4. <i>Activity Diagram</i>	25
4.1.5. <i>Sequence Diagram</i>	27
4.2. Perancangan Aplikasi <i>Client</i>	30
4.2.1. Perancangan <i>Class Diagram</i>	30
4.2.2. Perancangan Antarmuka Pengguna.....	35
4.3. Perancangan Aplikasi <i>Server</i>	36
4.3.1. Perancangan <i>Class Diagram</i>	36
4.3.2. Perancangan Basis Data.....	37
BAB V IMPLEMENTASI.....	40
5.1. Spesifikasi Sistem.....	40
5.1.1. Spesifikasi Sistem <i>Client</i>	40
5.1.2. Spesifikasi Sistem <i>Server</i>	41
5.2. Batasan Implementasi.....	41
5.3. Implementasi <i>Client</i>	41



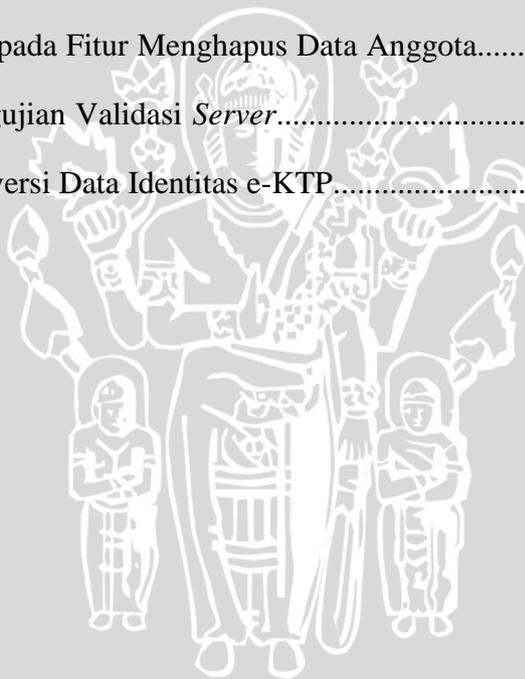
5.3.1. Implementasi Kebutuhan.....	42
5.3.2. Implementasi Kelas.....	44
5.3.3. Implementasi <i>Layout</i>	72
5.4. Implementasi <i>Server</i>	77
5.4.1. Implimentasi Basis Data.....	77
5.4.2. Implementasi Kelas.....	82
BAB VI PENGUJIAN DAN ANALISIS HASIL PENGUJIAN.....	87
6.1. Pengujian Validasi.....	87
6.1.1. Pengujian Validasi Aplikasi <i>Client</i>	87
6.1.2. Pengujian Validasi Aplikasi <i>Server</i>	95
6.2. Pengujian Identitas e-KTP dengan NFC Reader.....	98
6.3. Analisis Hasil Pengujian.....	99
BAB VII PENUTUP.....	100
7.1. Kesimpulan.....	100
7.2. Saran.....	101
DAFTAR PUSTAKA.....	102
LAMPIRAN.....	103

DAFTAR TABEL

Tabel 4.1 Daftar Kebutuhan Fungsional.....	21
Tabel 4.2. Skenario <i>Use Case</i> Mendaftarkan Pelanggan Menjadi Anggota.....	23
Tabel 4.3. Skenario <i>Use Case</i> Validasi Pembayaran Menggunakan Saldo.....	24
Tabel 4.4. Deskripsi Kelas Halaman Awal.....	31
Tabel 4.5. Deskripsi Kelas Halaman Menu.....	31
Tabel 4.6. Deskripsi Kelas Pendaftaran.....	32
Tabel 4.7. Deskripsi Kelas Pemesanan.....	32
Tabel 4.8. Deskripsi Kelas Pembayaran Saldo.....	33
Tabel 4.9. Deskripsi Kelas Pembayaran Tunai.....	34
Tabel 4.10. Deskripsi Kelas Deposit.....	34
Tabel 4.11. Deskripsi Kelas Pengaturan.....	35
Tabel 4.12. Deskripsi Kelas Server.....	36
Tabel 5.1. Spesifikasi Perangkat Keras <i>Client</i>	40
Table 5.2. Spesifikasi Perangkat Keras <i>Server</i>	41
Tabel 5.3. Spesifikasi Perangkat Lunak <i>Server</i>	41
Tabel 5.4. Implementasi Tabel Anggota.....	77
Tabel 5.5. Implementasi Tabel Detail Transaksi.....	78
Tabel 5.6. Implementasi Tabel Karyawan.....	78
Tabel 5.7. Implementasi Tabel Menu.....	79
Tabel 5.8. Implementasi Tabel Transaksi.....	79
Tabel 6.1. Kasus Uji pada Fitur Mengecek Keanggotaan Pelanggan.....	87
Tabel 6.2. Kasus Uji pada Fitur Mendaftarkan Pelanggan Menjadi Anggota.....	88
Tabel 6.3. Kasus Uji pada Fitur Mengatur Alamat Server.....	89



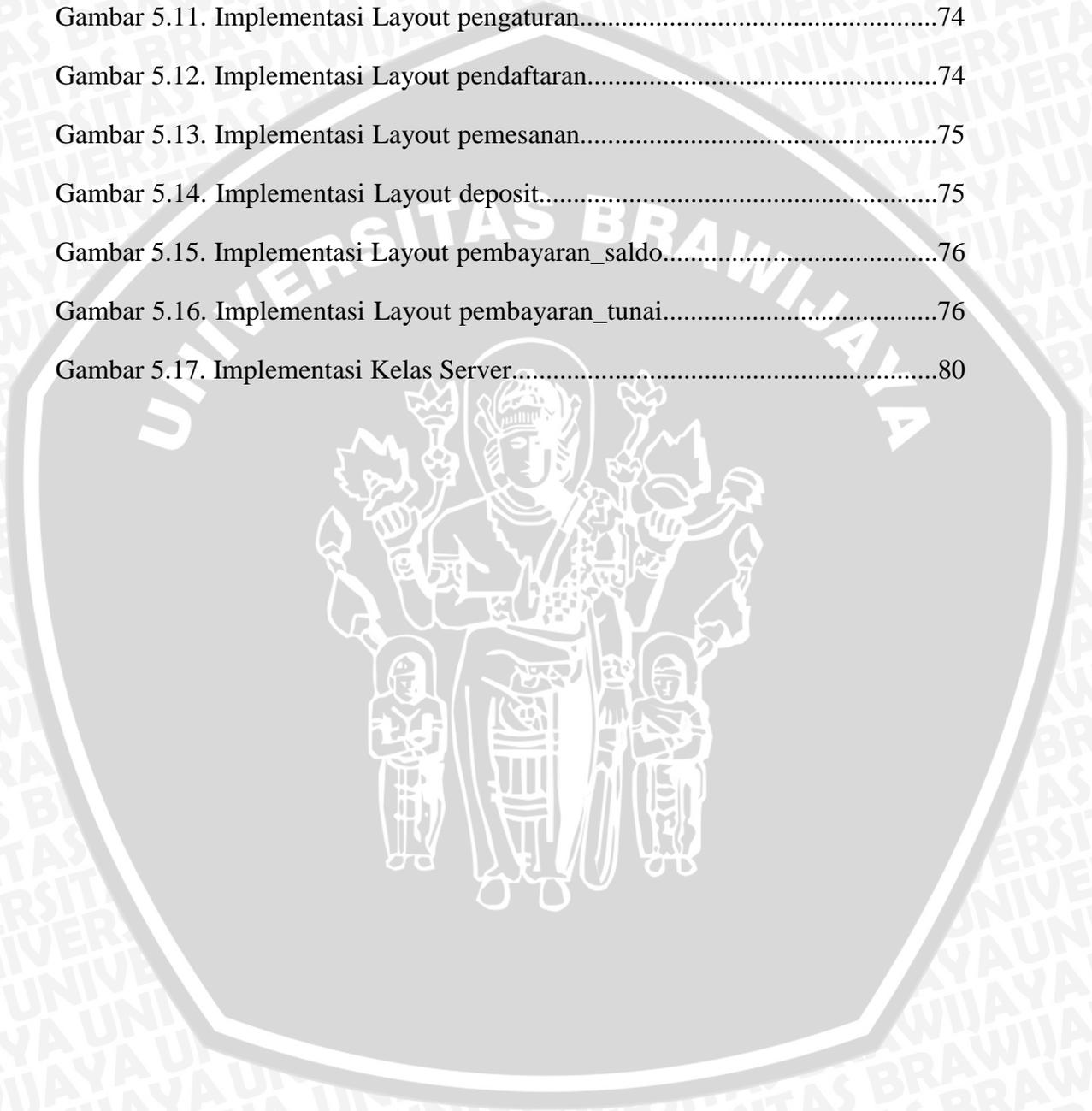
Tabel 6.4. Kasus Uji pada Fitur Mencatat Pesanan Pelanggan.....	89
Tabel 6.5. Kasus Uji pada Fitur Validasi Pembayaran Menggunakan Saldo.....	90
Tabel 6.6. Kasus Uji pada Fitur Validasi Pembayaran Tunai.....	91
Tabel 6.7. Kasus Uji pada Fitur Menambah Saldo Anggota.....	92
Tabel 6.8. Kasus Uji pada Fitur Melakukan Login.....	92
Tabel 6.9. Hasil Pengujian Validasi <i>Client</i>	93
Tabel 6.10. Kasus Uji pada Fitur Mencetak Bukti Pembayaran.....	95
Tabel 6.11. Kasus Uji pada Fitur Validasi Data Anggota Baru.....	95
Tabel 6.12. Kasus Uji pada Fitur Mengubah Data Anggota.....	96
Tabel 6.13. Kasus Uji pada Fitur Menghapus Data Anggota.....	97
Tabel 6.14. Hasil Pengujian Validasi <i>Server</i>	97
Tabel 6.15. Hasil Konversi Data Identitas e-KTP.....	99



DAFTAR GAMBAR

Gambar 2.1. Gambar Distribusi <i>Intent</i> pada <i>Tag Dispatch System</i>	12
Gambar 3.1. Diagram Alir Metode Penelitian.....	15
Gambar 3.2. Diagram Alir Implementasi Sistem.....	18
Gamabr 4.1. Gambaran Umum Sistem.....	21
Gambar 4.2. Diagram <i>Use Case</i> Sistem.....	23
Gambar 4.3. <i>Activity Diagram</i> pada Skenario Mendaftarkan Pelanggan Menjadi Anggota.....	26
Gambar 4.4. <i>Activity Diagram</i> pada Skenario Validasi Pembayaran Menggunakan Saldo.....	27
Gambar 4.5. <i>Sequence Diagram</i> pada Skenario Mendaftarkan Pelanggan Menjadi Anggota.....	28
Gambar 4.6. <i>Sequence Diagram</i> pada Skenario Validasi Pembayaran Menggunakan Saldo.....	29
Gambar 4.7. <i>Class Diagram</i> pada Aplikasi <i>Client</i>	30
Gambar 4.8. Tampilan Antarmuka Pengguna pada Pendaftaran Anggota.....	35
Gambar 4.9. <i>Class Diagram</i> pada Kelas <i>Server</i>	36
Gambar 4.10. <i>Entity Relationship Diagram</i> Basis Data Rumah Makan.....	38
Gambar 4.11. <i>Physical Data Model</i> Basis Data Rumah Makan.....	39
Gambar 5.1. Implementasi Kelas Halaman Awal.....	44
Gambar 5.2. Implementasi Kelas Halaman Menu.....	50
Gambar 5.3. Implementasi Kelas Pendaftaran.....	57
Gambar 5.4. Implementasi Kelas Pengaturan.....	60
Gambar 5.5. Implementasi Kelas Pemesanan.....	61
Gambar 5.6. Implementasi Kelas Deposit.....	64
Gambar 5.7. Implementasi Kelas Pembayaran Saldo.....	67

Gambar 5.8. Implementasi Kelas Pembayaran Tunai.....	70
Gambar 5.9. Implementasi Layout halaman_awal.....	73
Gambar 5.10. Implementasi Layout halaman_menu.....	73
Gambar 5.11. Implementasi Layout pengaturan.....	74
Gambar 5.12. Implementasi Layout pendaftaran.....	74
Gambar 5.13. Implementasi Layout pemesanan.....	75
Gambar 5.14. Implementasi Layout deposit.....	75
Gambar 5.15. Implementasi Layout pembayaran_saldo.....	76
Gambar 5.16. Implementasi Layout pembayaran_tunai.....	76
Gambar 5.17. Implementasi Kelas Server.....	80



BAB I

PENDAHULUAN

1.1 Latar Belakang

Customer Relationship Management merupakan salah satu strategi dalam bidang pemasaran yang saat ini semakin banyak diimplementasikan oleh perusahaan-perusahaan di Indonesia. Implementasi *Customer Relationship Management* banyak diterapkan pada perusahaan maskapai penerbangan, hotel, sampai perusahaan yang bergerak dalam bisnis ritel dengan program kepemilikan *membership card*. Berdasarkan observasi awal diketahui bahwa strategi *customer relationship management* yang dijalankan oleh penyedia bisnis jasa makanan bagi pelanggan restoran sebagian besar masih didominasi oleh penyediaan manfaat secara finansial. Lebih spesifik, manfaat finansial yang ditawarkan pada pelanggan adalah berupa program kartu keanggotaan (*membership card program*) pada restoran tertentu. Contoh seperti yang dilakukan oleh rumah makan Ikan Bakar Cianjur dimana pemegang kartu *membership* mendapatkan diskon dalam prosentase tertentu setiap kali mereka makan dan menggunakan kartu tersebut di semua outlet dari restoran yang bersangkutan. [WIJ - 08]

Uraian tersebut dapat diartikan bahwa salah satu metode pemasaran adalah melalui proses *membership* atau keanggotaan untuk menjaga loyalitas pelanggannya. Keuntungan yang didapat dari mempertahankan loyalitas pelanggan lama akan lebih besar daripada mencari pelanggan baru karena biaya untuk memperoleh pelanggan baru dapat lima kali lebih mahal dibandingkan dengan biaya untuk memelihara pelanggan lama. [SHO - 99]

Permasalahan yang timbul adalah apabila setiap rumah makan menerapkan program kartu anggota, sehingga pelanggan akan memiliki banyak kartu anggota pada rumah makan yang berbeda-beda. Resiko kehilangan akan lebih besar dan pelanggan tidak mempunyai kewajiban untuk mengganti kartu tersebut. Ditambah lagi dengan sulitnya mengurus kartu anggota dan antrinya pelanggan pada bagian *customer service* atau kasir yang ingin mendaftar sebagai anggota.

Dengan memanfaatkan perkembangan teknologi saat ini, pemilik rumah makan, khususnya di Indonesia, tidak harus membuat kartu anggota lagi karena

dengan adanya e-KTP, pelanggan sudah dapat menjadi anggota dari rumah makan tersebut. Kartu Tanda Penduduk elektronik atau electronic-KTP (e-KTP) adalah Kartu Tanda Penduduk (KTP) yang dibuat secara elektronik, dalam artian baik dari segi fisik maupun penggunaannya berfungsi secara komputerisasi [KUR – 13]. Saat ini jenis KTP tersebut sudah dilengkapi dengan *Radio Frequency Identification* (RFID) untuk proses identifikasi. Berdasarkan M. Mustafa Sarinanto, teknologi yang digunakan pada e-KTP berbasis smart card bertipe contactless card, yaitu chip smart card yang mampu berkomunikasi dengan pembaca (*reader*) tanpa kontak langsung secara fisik, melainkan menggunakan gelombang radio dengan frekuensi 13,56MHz [SAR – 13]. Hal tersebut memungkinkan perangkat bergerak yang memiliki *NFC reader*, dimana salah satunya adalah *smartphone*, dapat memanfaatkan teknologi tersebut untuk melakukan proses identifikasi pemilik kartu.

Penelitian ini dilakukan untuk memanfaatkan e-KTP yang sudah dilengkapi dengan teknologi RFID sebagai *membership card* atau kartu anggota pada ritel makanan atau rumah makan dengan *smartphone* berbasis Android sebagai media untuk mengenali pelanggan dan melakukan transaksi rumah makan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yang ada yaitu bagaimana merancang dan membangun serta melakukan pengujian terhadap aplikasi Android yang dilengkapi dengan *Near Field Communication* (NFC) *Reader* untuk mengenali identitas pelanggan menggunakan e-KTP sebagai media kartu anggota dan melakukan proses transaksi rumah makan secara *mobile*.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini antara lain :

1. Perancangan dan pengembangan aplikasi untuk membaca identitas keanggotaan pelanggan menggunakan *Near Field Communication* (NFC) *Reader* difokuskan pada *smartphone* berbasis Android.

2. Konsep *Customer Relationship Management* yang digunakan hanya pada keuntungan finansial berupa potongan harga yang didapatkan oleh anggota pada program kartu anggota.
3. Perusahaan yang digunakan pada penelitian adalah perusahaan ritel makanan atau rumah makan.
4. e-KTP yang digunakan adalah e-KTP Indonesia.

1.4 Tujuan

Tujuan penelitian ini adalah merancang dan membangun serta melakukan pengujian terhadap aplikasi Android yang dilengkapi dengan *Near Field Communication* (NFC) *Reader* untuk mengenali identitas pelanggan menggunakan e-KTP sebagai media kartu anggota dan melakukan proses transaksi rumah makan secara *mobile*.

1.5 Manfaat

1. Bagi Pemilik Rumah Makan

Manfaat yang dapat diambil dari penelitian ini bagi pemilik rumah makan yaitu dengan adanya aplikasi ini, pemilik rumah makan tidak lagi mengeluarkan *membership card* sebagai identitas pelanggan dan keuntungan yang diambil dari sistem keanggotaan semakin optimal.

2. Bagi Pelanggan Rumah Makan

Manfaat yang dapat diambil dari penelitian ini bagi pelanggan rumah makan yaitu dengan adanya aplikasi ini, pelanggan tidak perlu lagi menyimpan berbagai macam kartu anggota rumah makan sehingga mengurangi resiko kehilangan.

1.6 Sistematika Penulisan

Untuk mempermudah pemahaman, penelitian ini dibagi menjadi beberapa tahapan, antara lain :

1. BAB I PENDAHULUAN

Bab I berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

2. BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Bab II berisi kajian pustaka yang didalamnya terdapat rangkuman singkat yang komprehensif mengenai materi terkait yang ada dalam berbagai referensi dan berisi dasar teori yang berisi teori, temuan, dan bahan penelitian sebelumnya yang didapat dari berbagai sumber yang bisa dijadikan dasar dalam melakukan penelitian.

3. BAB III METODE PENELITIAN

Bab III berisi metode atau langkah-langkah yang digunakan dalam penelitian ini.

4. BAB IV ANALISIS DAN PERANCANGAN

Bab IV berisi analisis dan perancangan aplikasi Android untuk keanggotaan rumah makan menggunakan *Near Field Communication* (NFC) *Reader* pada e-KTP.

5. BAB V IMPLEMENTASI

Bab V berisi implementasi dari hasil perancangan aplikasi Android untuk keanggotaan rumah makan menggunakan *Near Field Communication* (NFC) *Reader* pada e-KTP pada BAB IV.

6. BAB VI PENGUJIAN DAN ANALISIS HASIL PENGUJIAN

Bab VI berisi penjelasan mengenai strategi pengujian dan teknik pengujian yang dilakukan terhadap rancangan dan implementasi aplikasi Android untuk keanggotaan rumah makan menggunakan *Near Field Communication* (NFC) *Reader* pada e-KTP, serta analisis hasil pengujian yang telah dilakukan.

7. BAB VII PENUTUP

Bab VII berisi kesimpulan dan saran dari penelitian ini. Kesimpulan sendiri didasarkan atas pengujian dan analisis yang telah dilakukan selama proses penelitian yang harus mempunyai korelasi dengan rumusan masalah yang telah ditetapkan di Bab I. Saran berisi masukan kepada peneliti lain yang ingin mengembangkan topik penelitian ini.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian pustaka yang didalamnya terdapat rangkuman singkat yang komprehensif mengenai materi terkait yang ada dalam berbagai referensi dan berisi dasar teori yang berisi teori, temuan, dan bahan penelitian sebelumnya yang didapat dari berbagai sumber yang bisa dijadikan dasar dalam melakukan penelitian.

2.1 Kartu Tanda Penduduk Elektronik (e-KTP)

Kartu Tanda Penduduk elektronik atau *electronic*-KTP (e-KTP) adalah Kartu Tanda Penduduk (KTP) yang dibuat secara elektronik, dalam artian baik dari segi fisik maupun penggunaannya berfungsi secara komputerisasi. Program e-KTP diluncurkan oleh Kementerian Dalam Negeri Republik Indonesia pada bulan Februari 2011 dimana pelaksanaannya terbagi dalam dua tahap. Tahap pertama dimulai pada tahun 2011 dan berakhir pada 30 April 2012 yang mencakup 67 juta penduduk di 2348 kecamatan dan 197 kabupaten/kota. Sedangkan tahap kedua mencakup 105 juta penduduk yang tersebar di 300 kabupaten/kota lainnya di Indonesia. Secara keseluruhan, pada akhir 2012, ditargetkan setidaknya 172 juta penduduk sudah memiliki e-KTP. [KUR – 13]

Teknologi dalam e-KTP setidaknya ada empat, yaitu *chip*, blangko atau *smart card*, biometrik, dan perangkat pembaca e-KTP. Berikut adalah penjelasan masing-masing teknologi e-ktp.

A. *Chip*

Kepala Program Penelitian dan Perakayasa e-KTP Gembong S Wibowanto, mengungkapkan, salah satu teknologi inti dalam e-KTP adalah *chip*. *Chip* e-KTP adalah kartu pintar berbasis mikroprosesor dengan memori 8 KB.

Chip dalam e-KTP berfungsi menyimpan data biodata pemilik, tanda tangan, pas foto, dan dua data sidik jari. *Default* sidik jari yang dipakai adalah telunjuk tangan kanan serta telunjuk tangan kiri.

Chip e-KTP tidak tampak dari luar. *Chip* ini sudah memenuhi standar ISO 14443 A dan 14443 B, mendukung kerahasiaan data pemilik e-KTP. *Chip* juga hanya bisa dibaca oleh perangkat pembaca tertentu untuk menjamin keamanan data.

Ke depan, *chip* e-KTP masih mungkin dikembangkan. Salah satunya, dalam e-KTP generasi kedua, kapasitas penyimpanan dalam *chip* akan ditambah. Penambahan akan mendukung penggunaan e-KTP selain sebagai kartu identitas, misalnya kartu jaminan sosial dan sebagainya. [UTO – 13]

B. Smart Card

Blangko atau *smart card* mungkin terlihat biasa, tetapi dibuat dengan bahan khusus *polyethylene terephthalate glycol* (PET-G). Blangko ini telah lulus uji Sentra Teknologi Polimer BPPT di Puspiptek Serpong.

Kartu terdiri dari tujuh lapisan. Cip dan antena yang menjadi komponen utama e-KTP terletak di lapisan tengah yang memiliki ketebalan 330 mikron. Lapisan lain berfungsi sebagai lapisan pelindung, lapisan gambar, dan *free layer* dengan ketebalan sekitar 95 mikron.

Material blangko telah didesain sehingga tahan faktor lingkungan tertentu. Misalnya, untuk suhu, material blangko dan cip tahan temperatur antara -25 hingga 70 derajat celsius. Blangko juga tahan tekanan dan bahan kimia tertentu. [UTO – 13]

C. Biometrik

Biometrik secara sederhana adalah identifikasi individu berdasarkan ciri-ciri yang melekat padanya. Ciri bisa berupa ciri fisiologis seperti sidik jari dan mata ataupun ciri perilaku seperti suara.

Dalam e-KTP, ciri individu yang digunakan dalam biometrik adalah 10 sidik jari, iris pada dua mata dan foto wajah. Ciri tersebut secara otomatis akan

disimpan ke pusat data di Kementerian Dalam Negeri saat pemilik mengurus e-KTP.

Teknologi biometrik dalam e-KTP punya dua fungsi. Pertama, untuk memastikan ketunggalan identitas penduduk. Konsekuensinya, dengan teknologi ini, penduduk tidak bisa memiliki dua e-KTP baik dengan biodata sama ataupun berbeda.

Saat seseorang mengurus e-KTP, akan dilakukan uji ketunggalan. Sepuluh sidik jari, dua iris mata, dan wajah akan diambil dan dipadankan. Jika terbukti belum terdaftar, seluruh data akan dimasukkan ke pusat data dan cip e-KTP. Jika sudah terdaftar, proses tak bisa dilanjutkan.

Fungsi kedua biometrik adalah proses verifikasi. Proses ini memastikan pemegang kartu adalah benar-benar pemiliknya. Dalam proses ini, hanya data sidik jari yang dibaca dengan bantuan perangkat pembaca e-KTP. [UTO – 13]

D. Perangkat pembaca e-KTP

Perangkat pembaca e-KTP sebenarnya adalah perangkat yang terpisah. Namun, teknologi ini menjadi satu kesatuan. Tanpa pembaca e-KTP, e-KTP tak berfungsi karena data-data penting di dalamnya tak bisa diketahui.

Perangkat pembaca e-KTP dilengkapi dengan modul biometrik sidik jari. Modul tersebut berfungsi memadankan sidik jari pemegang KTP dengan sidik jari yang tersimpan dalam e-KTP. Jika cocok, terkonfirmasi bahwa pemegang e-KTP benar-benar pemiliknya.

Untuk menggunakan perangkat pembaca, e-KTP terlebih dahulu diletakkan pada perangkat tersebut. Keterangan e-KTP bisa dibaca akan muncul dalam waktu kurang lebih 10 detik. Bila tak bisa dibaca, besar kemungkinan e-KTP tersebut palsu.

Proses selanjutnya, pemegang e-KTP diminta meletakkan sidik jari pada modul biometrik sidik jari. Apabila sidik jari cocok dengan sidik jari yang tersimpan dalam e-KTP, data yang ada dalam cip e-KTP akan muncul. Bila tidak, akses pada data akan ditolak.

Dengan mengenal teknologi dalam e-KTP, pemilik bisa memiliki pengetahuan yang benar. Dengan teknologi yang ada, e-KTP tidak akan rusak jika difotokopi, tetapi akan rusak bila di-stapler. E-KTP juga takkan rusak bila terkena air kecuali air dengan suhu di atas 75 derajat celsius. [UTO – 13]

2.2 *Radio Frequency Identification (RFID)*

2.2.1 *Pengertian RFID*

Identifikasi frekuensi radio, atau RFID, adalah istilah umum untuk teknologi yang menggunakan gelombang radio untuk secara otomatis mengidentifikasi orang atau benda. Ada beberapa metode identifikasi, tetapi yang paling umum adalah untuk menyimpan nomor seri yang mengidentifikasi orang atau benda, dan mungkin informasi lainnya, pada microchip yang terpasang pada antena (chip dan antena bersama-sama disebut transponder RFID atau tag RFID). Antena memungkinkan chip untuk mengirimkan informasi identifikasi untuk pembaca. Pembaca mengubah gelombang radio dipantulkan kembali dari tag RFID menjadi informasi digital yang kemudian dapat diteruskan ke komputer yang dapat memanfaatkannya [YAN – 09]. RFID mempunyai beberapa keuntungan yang utama melebihi sistem *barcode* yaitu kemungkinan data dapat dibaca secara otomatis tanpa memperhatikan garis arah pembacaan, melewati bahan non-konduktor seperti kartun kertas dengan kecepatan akses beberapa ratus *tag* per detik. *Tag* RFID terbuat dari *microchip* dengan dasar bahan dari silikon yang mempunyai kemampuan fungsi identitas sederhana yang disatukan dalam satu desain. Kemampuan *tag* RFID yaitu untuk membaca dan menulis (*read/write*), menyimpan data *storage* untuk mendukung enkripsi dan *control access*. [PRA – 12]

Karena setiap RFID *tag* adalah unik, maka dua buah produk dengan jenis yang sama akan memiliki kode yang berbeda, dimana sebaliknya jika menggunakan *barcode* semua produk sejenis akan menggunakan kode yang sama. Perbedaan lain antara *barcode* dan RFID adalah RFID *tag* memerlukan sumber tenaga listrik untuk menggerakkan sirkuit rangkaian terpadu di dalam tag tersebut, dan biasanya dengan cara mengirimkan energi listrik melalui medan elektromagnet dari *reader* ke RFID *tag*. Sebaliknya *reader* dapat membaca

banyak RFID *tag* dalam waktu bersamaan dalam jarak antara beberapa cm sampai 10 meter atau lebih. [PRA – 12]

2.2.1 Jenis- Jenis RFID

Metode pengiriman data kartu RFID pasif ke piranti pembaca dapat dibagi menjadi 2 macam, yaitu:

1. *Inductive Coupling*

Gulungan tembaga pada piranti pembaca membangkitkan medan elektromagnetik, kemudian gulungan yang ada di kartu RFID terinduksi oleh medan ini, hasil induksi inilah yang menjadi sumber tenaga bagi kartu RFID untuk mengirimkan kembali sinyal yang berisi data ke piranti pembaca. Karena menggunakan prinsip induksi ini, maka jarak antara kartu RFID dengan piranti pembaca juga harus pendek agar induksi dapat ditangkap. *Inductive coupling* ini digunakan pada kartu RFID dengan *low frequency* dan *high frequency*.

2. *Propagation Coupling*

Pada sistem ini, energi yang digunakan berasal dari energi elektromagnetik (gelombang radio) yang dipancarkan oleh piranti pembaca. Kartu RFID kemudian akan mengumpulkan energi elektromagnetik ini untuk digunakan sebagai sumber daya mengirimkan data yang dimilikinya ke piranti pembaca. Mekanisme ini disebut dengan backscatter. Modulasi bit data ke frekuensi bisa menggunakan amplitude shift keying, phase shift keying, atau frequency shift keying. [PRA – 12]

Berdasarkan besarnya gelombang yang dipancarkan, RFID dapat dibagi menjadi 4 jenis, yaitu *Low Frequency* (LF) atau frekuensi rendah yang besar frekuensinya berkisar antara 120–150 kHz dan digunakan pada umumnya untuk identifikasi hewan atau produk pabrik, *High Frequency* (HF) atau frekuensi tinggi yang besar frekuensinya mencapai 13.56 MHz dan digunakan pada umumnya sebagai *smart card*, *Ultra High Frequency* (UHF) atau frekuensi sangat tinggi yang besar frekuensinya mencapai 433 MHz dan digunakan sebagai aplikasi keamanan, dan jenis terakhir adalah *Microwave* atau gelombang mikro yang besar

frekuensinya mencapai 5800 MHz, digunakan pada 802.11 WLAN Bluetooth. [YAN – 09]

Tag RFID adalah perangkat yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari *tag* RFID umumnya memiliki memori sehingga *tag* ini mempunyai kemampuan untuk menyimpan data. Memori pada *tag* dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read-Only*, misalnya *serial number* yang unik yang disimpan pada saat *tag* tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan sumber daya *tag*, *tag* RFID dapat digolongkan menjadi:

1. *Tag* Aktif: yaitu *tag* yang sumber dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh RFID *reader* dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh.

2. *Tag* Pasif: yaitu *tag* yang sumber dayanya diperoleh dari medan yang dihasilkan oleh RFID *reader*. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. [PRA – 12]

2.3 Android

2.3.1 NFC Reader

Ada dua penggunaan dasar ketika bekerja dengan NDEF data pada Android:

1. Membaca data NDEF dari tag NFC.
2. Mengirimkan pesan NDEF dari satu perangkat ke perangkat lainnya dengan Android Beam™

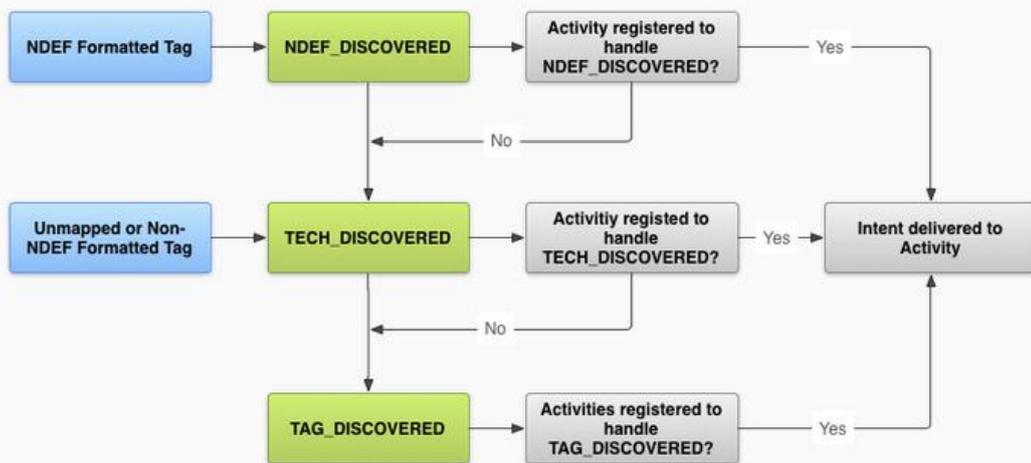
Membaca data NDEF dari tag NFC ditangani oleh *tag dispatch system*, yang menganalisis tag NFC saat ditemukan, mengkategorikan data secara tepat, dan mulai aplikasi yang tertarik pada data yang dikategorikan. Sebuah aplikasi yang ingin menangani NFC tag dapat mendeklarasikan sebuah *intent filter* dan permintaan untuk menangani data.

Android Beam™ fitur memungkinkan perangkat untuk mengirim pesan NDEF ke perangkat lain dengan menempelkan perangkat satu ke yang lainnya. Interaksi ini memberikan cara yang lebih mudah untuk mengirim data dari

teknologi nirkabel lainnya seperti Bluetooth, karena dengan NFC, tidak perlu mencari perangkat secara manual dan melakukan *pairing*. Koneksi secara otomatis dimulai ketika dua perangkat berada dalam jangkauan. Android Beam tersedia melalui satu set API NFC, sehingga aplikasi apapun dapat mengirimkan informasi antar perangkat. Sebagai contoh, Contacts, Browser, dan aplikasi YouTube menggunakan Android Beam untuk berbagi kontak, halaman web, dan video dengan perangkat lain.

Ketika *tag dispatch system* selesai membuat *intent* yang menggunakan tag NFC dan informasi didalamnya, sistem tersebut akan mengirimkan *intent* untuk aplikasi yang mendukung *intent* tersebut. Jika lebih dari satu aplikasi dapat menangani *intent*, *Activity Chooser* ditampilkan sehingga pengguna dapat memilih sistem *activity* yang ingin dijalankan. Berdasarkan prioritasnya, *tag dispatch system* mendefinisikan tag yang ditemukan menjadi tiga bagian, yaitu:

1. **ACTION_NDEF_DISCOVERED:** *intent* ini digunakan untuk memulai kegiatan ketika tag yang berisi data NDEF ditemukan dan merupakan jenis yang diakui. Ini adalah *intent* dengan prioritas tertinggi, dan jika memungkinkan, *tag dispatch system* akan mencoba untuk memulai *activity* dengan *intent* ini terlebih dahulu sebelum *intent* yang lainnya.
2. **ACTION_TECH_DISCOVERED:** Jika tidak ada *activity* yang terdaftar untuk menangani *intent* ACTION_NDEF_DISCOVERED, *tag dispatch system* mencoba untuk memulai aplikasi dengan *intent* ini. *Intent* ini juga langsung dimulai (tanpa memulai ACTION_NDEF_DISCOVERED terlebih dahulu) jika tag yang ditemukan berisi data NDEF yang tidak dapat dipetakan ke tipe MIME atau URI, atau jika tag tidak berisi data NDEF tetapi dari teknologi tag yang dikenal.
3. **ACTION_TAG_DISCOVERED:** *intent* ini dimulai jika tidak ada *activity* yang menangani ACTION_NDEF_DISCOVERED atau *intent* ACTION_TECH_DISCOVERED.



Gambar 2.1 Gambar Distribusi *Intent* pada *Tag Dispatch System*

2.3.2 Intent Filter

Komponen inti dari sebuah aplikasi (*activity*, *service*, dan *broadcast receiver*) yang diaktifkan oleh *intent*. Sebuah *intent* adalah sekumpulan informasi yang menjelaskan tindakan yang akan dilakukan oleh aplikasi - termasuk data yang akan diproses, kategori komponen yang harus melakukan tindakan tersebut, dan instruksi lainnya yang berhubungan dengan aplikasi tersebut. Android menempatkan komponen yang tepat untuk menanggapi *intent*, membuat komponen baru yang akan mengolah *intent* tersebut jika diperlukan, dan dikirimkan ke objek *Intent*.

Komponen menginformasikan kemampuan mereka - jenis *intent* apa yang dapat mereka respon - melalui *intent-filter*. Karena sistem Android harus mengetahui *intent* dari suatu komponen sebelum menjalankan komponen tersebut, *intent-filter* yang ditentukan dalam *Android manifest* sebagai elemen `<intent-filter>`. Sebuah komponen mungkin memiliki sejumlah *filter*, masing-masing menggambarkan kemampuan yang berbeda.

Sebuah *intent* yang secara eksplisit memberikan nama komponen akan mengaktifkan komponen tersebut saat *intent* ditemukan, tidak perlu adanya *intent-filter*. Selain itu *intent* yang tidak menentukan target dengan nama komponen dapat mengaktifkan komponen hanya jika *intent* tersebut dapat melewati salah satu *intent-filter*. [ANO – 14]

2.3.3 Asyncrounus Task

AsyncTask memungkinkan penggunaan untuk mengatur *thread User Interface* secara tepat dan mudah. Kelas ini memungkinkan untuk melakukan *background process* dan mempublikasikan hasilnya di *thread UI* tanpa harus memanipulasi *thread* inti.

AsyncTask dirancang untuk menjadi *helper class* pada *Thread* dan *Handler* dan bukan merupakan kerangka utama dari proses *threading*. AsyncTasks idealnya harus digunakan untuk operasi singkat (beberapa detik). Jika pengguna perlu untuk menjaga *thread* berjalan untuk jangka waktu yang lama, sangat disarankan pengguna menggunakan berbagai API yang disediakan oleh package `java.util.concurrent` seperti `Executor`, `ThreadPoolExecutor` dan `FutureTask`.

Sebuah AsyncTask didefinisikan dengan komputasi yang berjalan pada *background process* dan yang hasilnya dikirimkan ke *thread UI*. Sebuah AsyncTask didefinisikan oleh 3 bagian utama, yang disebut Params, Progress, dan Result, dan 4 tahap yang disebut `onPreExecute`, `doInBackground`, `onProgressUpdate` dan `onPostExecute`. Params adalah parameter yang dikirimkan oleh *thread* utama ke dalam AsyncTask, Progress adalah hasil yang diberikan pada *thread* utama saat komputasi sedang berjalan pada AsyncTask, dan Result adalah hasil dari proses tersebut yang dikirimkan ke *thread* utama.

Ketika AsyncTask dijalankan, proses data berjalan melalui 4 tahap, yaitu:

1. `onPreExecute ()`, dipanggil di *thread UI* sebelum proses dijalankan. Tahap ini biasanya digunakan untuk pengaturan proses, misalnya dengan menunjukkan progress bar di antarmuka pengguna.
2. `doInBackground (params. ..)`, dipanggil di *background process* setelah `onPreExecute ()` selesai mengeksekusi. Langkah ini digunakan untuk melakukan komputasi yang dapat memakan waktu yang lama. Parameter AsyncTask diterima pada tahap ini. Hasil perhitungan tersebut harus dikembalikan oleh langkah ini dan akan diteruskan kembali ke langkah terakhir. Langkah ini juga dapat menggunakan `publishProgress (Progress. ..)` untuk menerbitkan satu atau lebih unit kemajuan. Nilai-nilai ini diterbitkan di *thread UI*, di `onProgressUpdate (Progress. ..)` langkah.

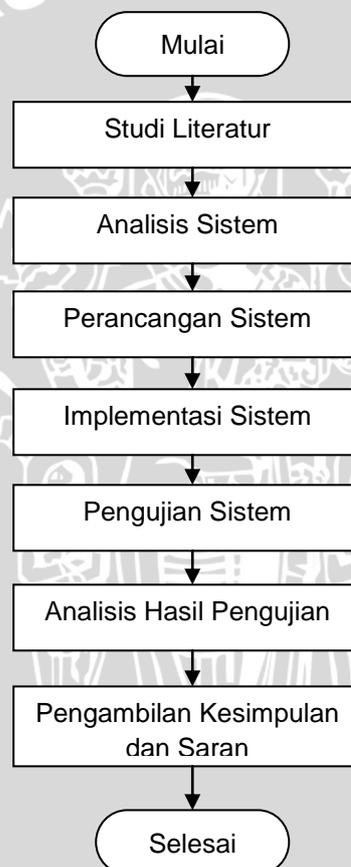
3. `onProgressUpdate` (`Progress. ..`), dipanggil di thread UI setelah panggilan ke `publishProgress` (`Progress. ..`). Waktu eksekusi tidak terdefinisi. Metode ini digunakan untuk menampilkan segala bentuk kemajuan dalam antarmuka pengguna sedangkan perhitungan latar belakang masih mengeksekusi. Sebagai contoh, dapat digunakan untuk menghidupkan sebuah progress bar atau menunjukkan log dalam kolom teks.
4. `onPostExecute` (`Hasil`), dipanggil di thread UI setelah latar belakang perhitungan selesai. Hasil perhitungan latar belakang akan diteruskan ke langkah ini sebagai parameter. [ANO – 14]



BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai metode yang digunakan dalam penelitian dimana terdiri dari beberapa tahap sebagai berikut : studi literatur, analisa kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, analisis hasil pengujian, pengambilan kesimpulan. Tahap-tahap tersebut dilakukan secara berurutan seperti yang digambarkan melalui diagram alir pada gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk memahamai secara mendalam hal-hal yang berkaitan dengan penelitian, baik dari segi konsep, dasar teori, maupun hal-hal yang membantu dalam proses perancangan, implementasi, dan pengujian sistem. Sumber-sumber yang digunakan pada penelitian ini sebagai bahan pembelajaran telah dilampirkan pada bagian daftar pustaka. Sumber pembelajaran pada penelitian ini didapatkan melalui buku tulis maupun media internet dalam bentuk jurnal ilmiah, maupun tugas akhir. Adapun yang dijadikan bahan studi literatur untuk mendukung penelitian ini antara lain:

1. e-KTP
 - a. Pengertian e-KTP
 - b. Teknologi e-KTP
2. Radio Frequency Identification (RFID)
 - a. Pengertian RFID
 - b. Macam-macam RFID
 - c. Pengertian Tag Writer dan Reader
3. Android
 - a. Penggunaan NFC Reader pada Android
 - b. Penggunaan Intent Filter
 - c. Penggunaan Asyncronus Task

3.2 Analisis Sistem

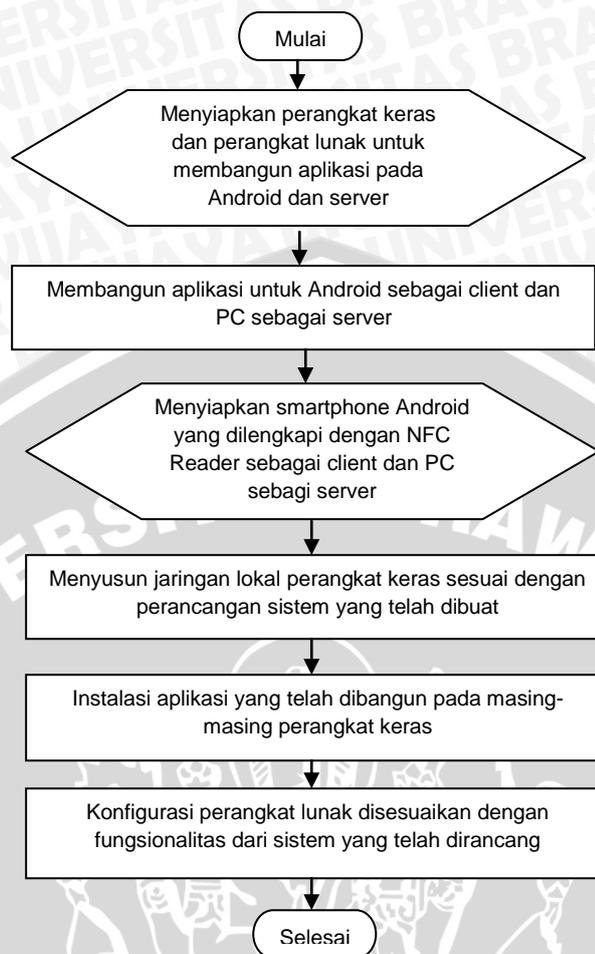
Analisis sistem dilakukan untuk dapat mengetahui kebutuhan-kebutuhan yang diperlukan dalam sistem yang akan dibangun, baik kebutuhan perangkat keras atau *hardware*, maupun kebutuhan perangkat lunak atau *software*. Analisis sistem pada perangkat keras akan digambarkan dengan diagram alir sistem dimana mencakup semua perangkat keras yang bekerja dan alur data yang masuk dan keluar. Analisis sistem pada perangkat lunak akan digambarkan menggunakan *Unified Modeling Language* (UML), khususnya diagram *use case* dimana mencakup semua fitur-fitur dan aktor yang terlibat dalam perangkat lunak yang akan dibangun.

3.3 Perancangan Sistem

Perancangan sistem dilakukan untuk mempermudah penelitian pada tahap implementasi sistem. Perancangan yang dilakukan dalam penelitian ini antara lain perancangan sistem penggunaan aplikasi pada rumah makan yang akan digambarkan menggunakan *activity diagram*, perancangan hubungan antar objek yang akan digambarkan menggunakan *sequence diagram*, perancangan *class* pada sistem yang akan digambarkan dengan *class diagram*, perancangan penyimpanan dan hubungan data dalam tabel yang digambarkan dengan *entity relationship diagram*, dan perancangan antarmuka pengguna untuk aplikasi *client*.

3.4 Implementasi Sistem

Implementasi dilakukan sesuai dengan tahap perancangan yang telah dibuat. Dalam penelitian ini proses implementasi dilakukan secara bertahap mulai dari tahapan mempersiapkan perangkat keras maupun perangkat lunak yang diperlukan untuk membangun aplikasi pada Android dan server, dilanjutkan dengan menyusun jaringan lokal perangkat keras sesuai dengan perancangan sistem yang telah dibuat, instalasi perangkat lunak pada masing-masing perangkat keras, konfigurasi perangkat lunak disesuaikan dengan fungsionalitas dari sistem yang telah dirancang. Proses implementasi dilakukan secara bertahap tujuannya agar ketika terjadi kesalahan pada sistem, masalah yang muncul dapat dengan mudah diketahui sehingga semakin efektif untuk memperbaiki kesalahan yang terjadi. Semua tahapan pada implementasi sistem digambarkan melalui diagram alir seperti dalam Gambar 3.2.



Gambar 3.2 Diagram Alir Implementasi Sistem

3.5 Pengujian Sistem

Pengujian dilakukan untuk mengetahui apakah sistem yang sudah diimplementasikan dapat mengatasi masalah dan sesuai dengan tujuan penelitian. Pengujian dilakukan dengan menggunakan metode pengujian validasi yang merupakan salah satu metode *blackbox testing* dalam menguji kinerja dari fitur-fitur yang telah dibuat apakah sesuai dengan hasil yang diharapkan dan metode pengujian unit yang merupakan salah satu metode *whitebox testing* dalam menguji kinerja algoritma yang telah diimplementasikan dalam aplikasi.

3.6 Analisis Hasil Pengujian

Setelah pengujian sistem dilakukan, akan ada tahap analisa hasil pengujian untuk mengetahui hasil dari pengujian sehingga dapat ditarik kesimpulan dari rancang bangun aplikasi yang telah dibuat.

3.7 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah melewati proses perancangan, implementasi, pengujian dan analisis hasil pengujian. Kesimpulan berisikan inti dari hasil penelitian yang telah dibangun berdasarkan tahap pengujian dan analisis hasil pengujian yang dilakukan pada penelitian ini untuk menjawab permasalahan pada penelitian. Tahap terakhir penulisan adalah saran berisikan pertimbangan dari penulis yang ditujukan kepada pembaca yang akan melakukan penelitian ini lebih lanjut.



BAB IV

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis sistem dan perancangan dalam pembangunan aplikasi rumah makan memanfaatkan NFC Reader pada Android dan aplikasi pada server. Bab ini akan dibagi menjadi dua tahap. Tahap pertama adalah analisis sistem yang berisi tentang gambaran umum sistem dan kebutuhan sistem yang diambarkan dengan diagram usecase, activity diagram dan sequence diagram. Selanjutnya tahap kedua adalah bagian perancangan aplikasi dibagi menjadi dua bagian, yaitu aplikasi pada client dan aplikasi pada server.

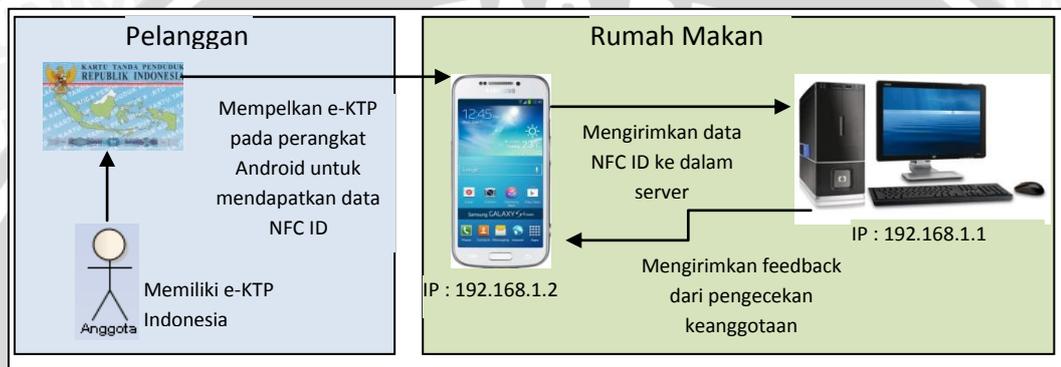
4.1 Analisis Sistem

Analisis sistem berisi gambaran umum sistem dan analisis kebutuhan yang digambarkan melalui diagram usecase, activity diagram, dan sequence diagram. Gambaran umum sistem menjelaskan bagaimana sistem bekerja pada ruang lingkup perangkat keras. Analisis kebutuhan menjelaskan kebutuhan dari aplikasi secara fungsional yang digambarkan dengan diagram usecase dan activity diagram nantinya akan menjadi fitur-fitur pada aplikasi. Segala aktifitas yang telah dibuat digambarkan sebagai hubungan antar objek dengan menggunakan *sequence diagram*. Perancangan hubungan objek bertujuan untuk menjelaskan bagaimana proses berlangsung yang melibatkan objek dalam aplikasi dengan fungsi-fungsi yang dimiliki objek tersebut.

4.1.1 Gambaran Umum Sistem

Aplikasi yang dibangun pada sistem ini terbagi menjadi dua, yaitu aplikasi pada sisi client menggunakan perangkat Android yang dilengkapi dengan perangkat NFC Reader, dan di sisi server menggunakan komputer atau PC. Sistem bertujuan untuk mengenali identitas pelanggan rumah makan dengan memanfaatkan e-KTP yang dimiliki oleh pelanggan sebagai pengganti kartu anggota. Pengecekan identitas dilakukan dengan menempelkan e-KTP pelanggan pada perangkat Android di sisi client dan akan mengirimkan data tersebut ke

dalam server untuk mengecek apakah pelanggan tersebut sudah terdaftar sebagai anggota. Jika belum terdaftar, pelanggan dapat mendaftarkan e-KTP-nya untuk menjadi anggota rumah makan dengan mengisi biodata pada sisi client yang selanjutnya akan dikirim ke sisi server. Selain itu pada sisi client juga melayani transaksi dimulai dari pemesanan menu makanan sampai proses pembayaran. Pada sisi server berupa aplikasi website untuk mengelola data anggota dan mencetak bukti pembayaran. Gambar 4.1 menjelaskan bagaimana sistem bekerja secara umum.



Gambar 4.1 Gambaran Umum Sistem

4.1.2 Analisis Kebutuhan

Analisis kebutuhan berisi tentang kebutuhan-kebutuhan yang harus terpenuhi dalam sistem. Kebutuhan fungsional berisi fungsi-fungsi utama dari sistem yang dapat dilakukan oleh aktor untuk memenuhi kebutuhannya terhadap sistem. Daftar kebutuhan fungsional ditampilkan pada Tabel 4.1.

Tabel 4.1 Daftar Kebutuhan Fungsional

Nomor	Kebutuhan	Use Case	Jenis
FR_001	Aplikasi mampu mengecek keanggotaan pelanggan dengan e-KTP	Mengecek Keanggotaan Pelanggan	Client
FR_002	Aplikasi mampu mendaftarkan pelanggan sebagai anggota baru	Mendaftarkan Pelanggan Menjadi Anggota	Client

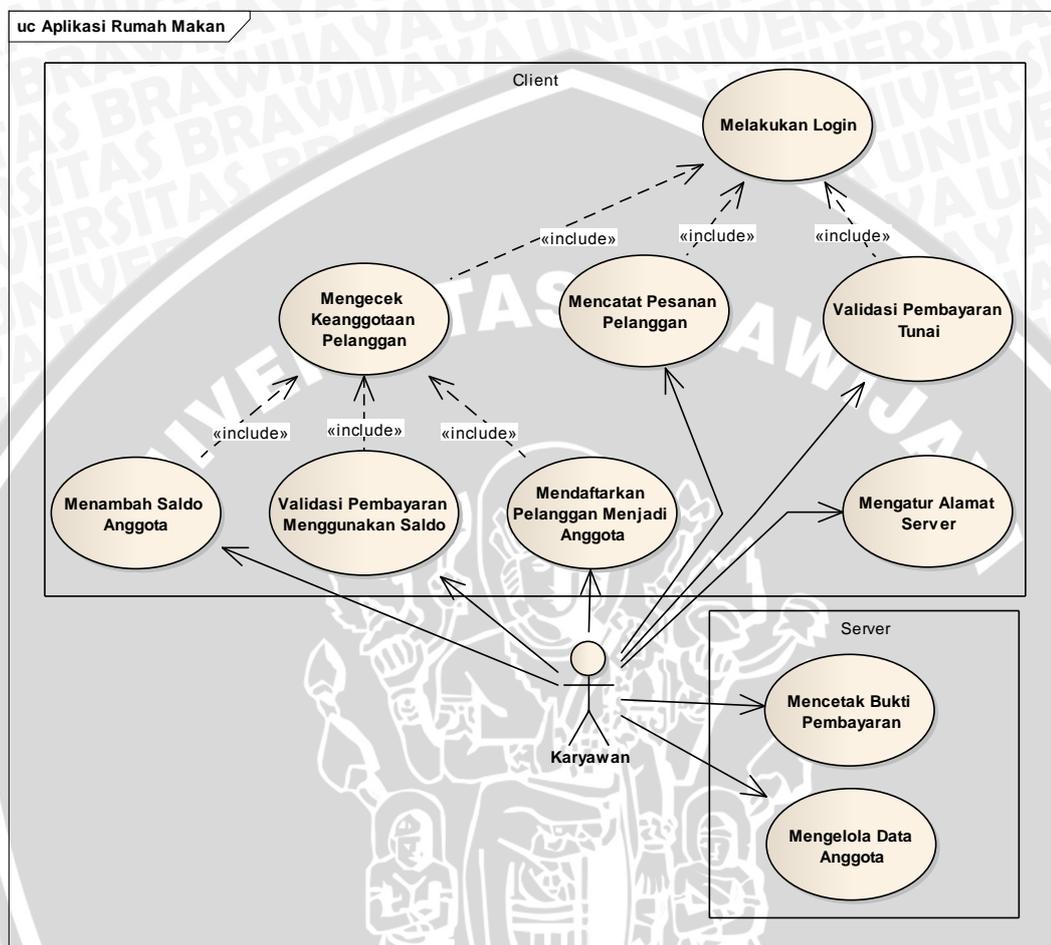
FR_003	Aplikasi mampu mengubah alamat server	Mengatur Alamat Server	Client
FR_004	Aplikasi mampu mencatat pesanan pelanggan	Mencatat Pesanan Pelanggan	Client
FR_005	Aplikasi mampu melakukan validasi pembayaran dengan menggunakan saldo anggota	Validasi Pembayaran Menggunakan Saldo	Client
FR_006	Aplikasi mampu melakukan validasi pembayaran dengan uang tunai	Validasi Pembayaran Tunai	Client
FR_007	Aplikasi mampu menambah saldo anggota melalui deposit	Menambah Saldo Anggota	Client
FR_008	Aplikasi mampu melakukan login karyawan	Melakukan Login	Client
FR_009	Aplikasi mampu mencetak bukti pembayaran	Mencetak Bukti Pembayaran	Server
FR_010	Aplikasi mampu mengelola data anggota berupa validasi, ubah dan hapus	Mengelola Data Anggota	Server

4.1.3 Diagram Use Case

Diagram *use case* merupakan salah satu cara untuk menggambarkan perilaku aktor terhadap sistem. Diagram *use case* terdiri dari aktor, *use case*, dan hubungan antara aktor dengan *use case* maupun *use case* dengan *use case* lainnya. Setiap *use case* memiliki sebuah skenario untuk menjelaskan aktifitas yang terjadi didalamnya. Diagram *use case* sistem ditunjukkan dalam Gambar 4.2.

Dalam Gambar 4.2 dijelaskan bahwa sistem dibagi menjadi 2, yaitu sistem client merupakan aplikasi yang berada pada perangkat Android dan sistem server merupakan aplikasi web yang berada pada PC. Setiap aplikasi akan memiliki fitur sesuai dengan kebutuhan yang sudah dianalisis sebelumnya. Aktor hanya diperbolehkan melakukan satu perintah atau satu *use case* di saat yang bersamaan,

misalnya pelanggan melakukan proses mengecek keanggotaan tidak dapat sekaligus melakukan proses mendaftar anggota baru.



Gambar 4.2 Diagram Use Case Sistem

Tabel 4.2 Skenario Use Case Mendaftarkan Pelanggan Menjadi Anggota

Nomor Use Case	FR_002
Nama	Mendaftarkan Pelanggan Menjadi Anggota
Prasyarat Konteks	- Karyawan telah melakukan pemasangan aplikasi pada perangkat Android - Pelanggan belum menjadi anggota - Karyawan telah login
Tujuan dalam Konteks	Mendaftar pelanggan yang belum mempunyai keanggotaan di rumah makan
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Karyawan mengirimkan biodata dan identitas e-KTP pelanggan ke dalam server untuk divalidasi oleh karyawan

	pada aplikasi server
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan memilih tombol daftar pada aplikasi client
Alur Utama	1. Karyawan memilih tombol daftar
	2. Sistem menampilkan halaman pengisian biodata
	3. Karyawan mengisi biodata pelanggan dan memilih tombol simpan
	4. Sistem menyimpan biodata pelanggan untuk proses validasi anggota oleh karyawan

Pada Tabel 4.2 menjelaskan bagaimana proses pendaftaran anggota melalui aplikasi client. Proses dapat dijalankan apabila aplikasi sudah terpasang pada perangkat dan server sudah siap menerima data yang dikirimkan oleh client, termasuk pengaturan alamat server pada sisi client. Proses pendaftaran membutuhkan proses pengecekan keanggotaan sebelum karyawan dapat menekan tombol daftar sebagai trigger proses pendaftaran. Setelah pengecekan, sistem akan menunjukkan status pelanggan, dan apabila belum terdaftar, karyawan dapat menekan tombol daftar. Proses selanjutnya adalah mengisi biodata pelanggan. Sistem akan menyimpannya dan menunggu validasi oleh karyawan sehingga akhirnya pelanggan dapat menjadi anggota.

Tabel 4.3 Skenario *Use Case* Validasi Pembayaran Menggunakan Saldo

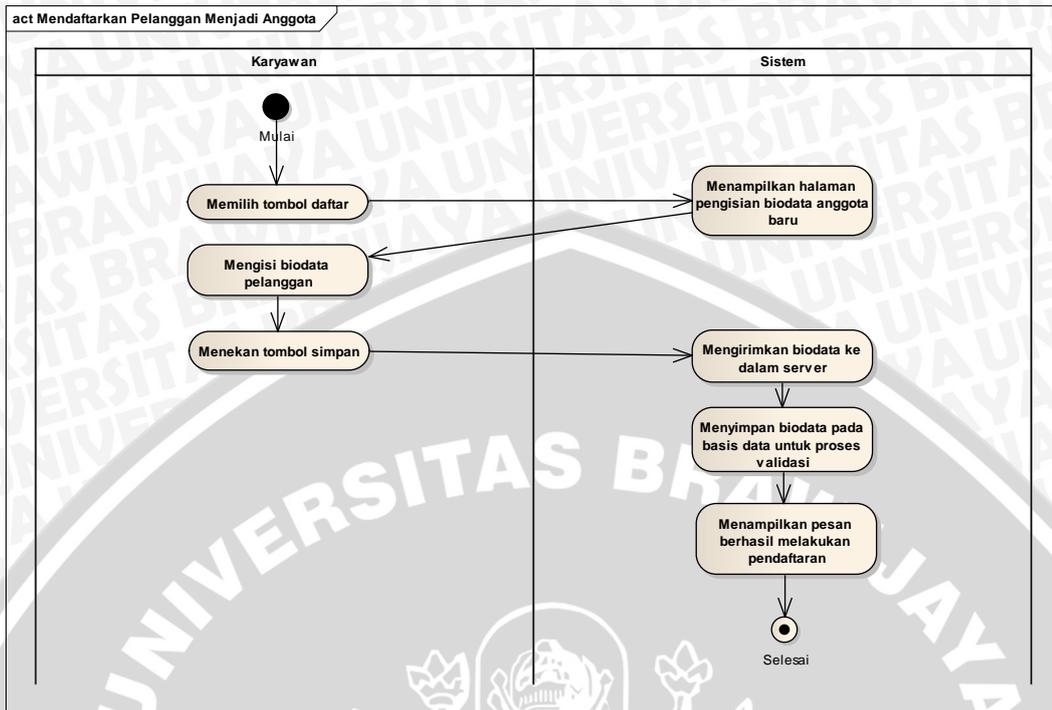
Nomor Use Case	FR_005
Nama	Validasi Pembayaran Menggunakan Saldo
Prasyarat Konteks	- Karyawan telah melakukan pemasangan aplikasi pada perangkat Android - Pelanggan telah terdaftar sebagai anggota - Karyawan telah login
Tujuan dalam Konteks	Melakukan proses pembayaran terhadap menu yang telah dipesan oleh pelanggan sebagai anggota dengan menggunakan saldo yang dimiliki oleh anggota
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Sistem melakukan validasi terhadap transaksi yang telah dilakukan dan mengurangi saldo anggota sebesar total pembayaran
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server atau saldo tidak mencukupi untuk melakukan pembayaran

Aktor	- Karyawan
Trigger	Karyawan memilih tombol pembayaran saldo pada aplikasi client
Alur Utama	1. Karyawan memilih tombol pembayaran saldo
	2. Sistem menampilkan halaman pembayaran saldo disertai dengan informasi saldo anggota
	3. Karyawan memasukkan nomor meja pelanggan dan menekan tombol cek
	4. Sistem menampilkan besarnya total pembayaran
	5. Karyawan menekan tombol bayar
	6. Sistem melakukan validasi pembayaran

Pada Tabel 4.3 menjelaskan bagaimana proses pembayaran menggunakan saldo anggota. Sebelum proses dilakukan, aplikasi harus sudah terpasang pada perangkat client dan pelanggan juga sudah menjadi anggota rumah makan. Proses pembayaran dengan menggunakan saldo anggota membutuhkan proses pengecekan keanggotaan sebelum karyawan dapat menekan tombol pembayaran saldo sebagai trigger proses pendaftaran Tahap awal yang dilakukan adalah karyawan memilih tombol pembayaran saldo. Proses selanjutnya, sistem akan menampilkan informasi saldo anggota serta kolom isian untuk memasukkan nomor meja pelanggan. Apabila saldo mencukupi, maka sistem akan melakukan validasi pembayaran dan mengurangi saldo anggota yang telah melakukan pembayaran.

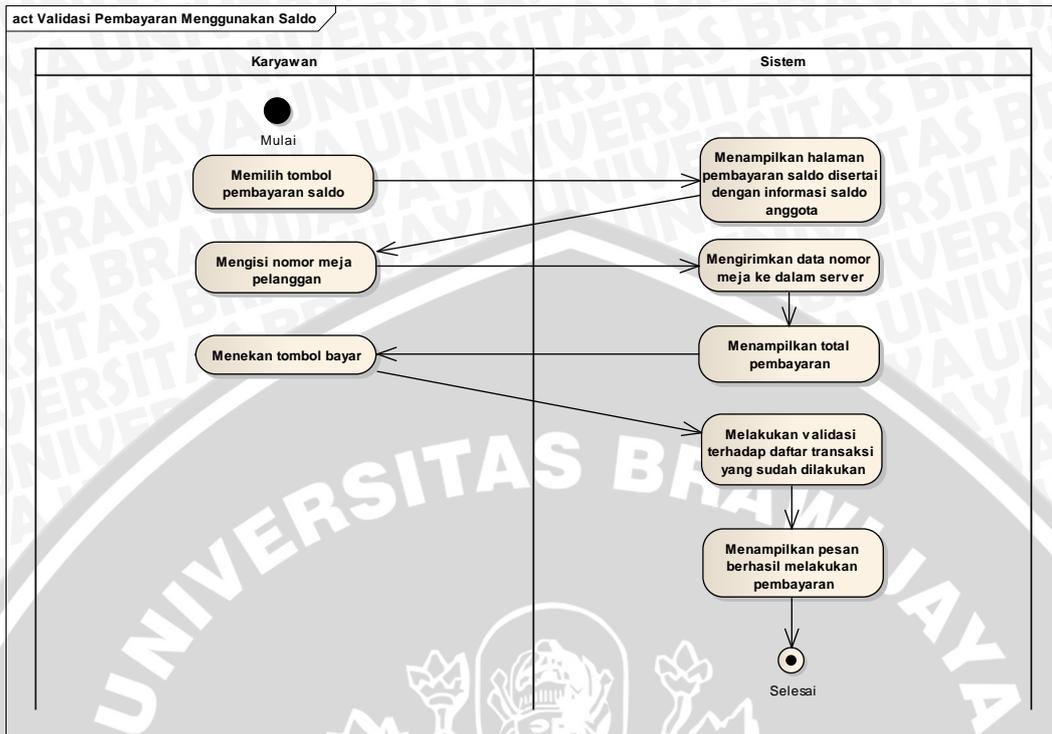
4.1.4 Activity Diagram

Pada *activity diagram* akan memodelkan bagaimana aktifitas-aktifitas yang terjadi di setiap skenario *use case* dapat melakukan proses tersebut. Gambar 4.3 menunjukkan *activity diagram* pada skenario *use case* mendaftarkan pelanggan menjadi anggota yang dilakukan oleh karyawan.



Gambar 4.3 Activity Diagram pada Skenario Mendaftarkan Pelanggan Menjadi Anggota

Dalam Gambar 4.3 menunjukkan aktifitas-aktifitas terjadi di dalam proses mendaftarkan pelanggan menjadi anggota. Aktifitas terbagi dibagi menjadi dua berdasarkan aktor yang terlibat pada proses tersebut. Sesuai dengan skenario *use case*, aktifitas dimulai dengan karyawan menekan tombol daftar pada aplikasi client. Sistem akan menampilkan form pengisian biodata anggota baru dan diisi oleh karyawan berdasarkan data pelanggan. Setelah pengisian selesai, karyawan menekan tombol simpan dan sistem akan melakukan proses penyimpanan biodata ke dalam basis data untuk proses validasi oleh server. Setelah proses sukses, sistem akan menampilkan berhasil melakukan pendaftaran.



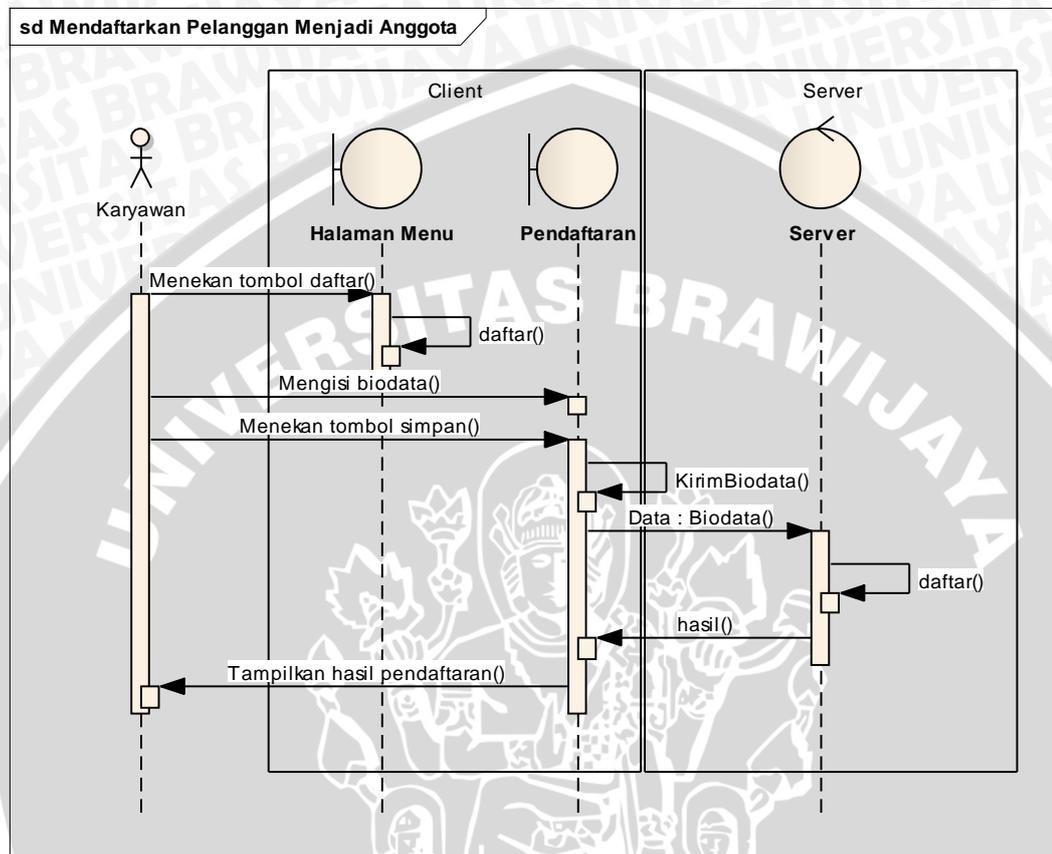
Gambar 4.4 Activity Diagram pada Skenario Validasi Pembayaran Menggunakan Saldo

Dalam Gambar 4.4 menunjukkan aktifitas-aktifitas terjadi di dalam proses validasi pembayaran menggunakan saldo. Aktifitas terbagi dibagi menjadi tiga berdasarkan aktor yang terlibat pada proses tersebut. Sesuai dengan skenario *use case*, aktifitas dimulai dengan karyawan memilih menu pembayaran anggota. Setelah sistem menampilkan tampilan pembayaran, pelanggan akan menempelkan e-KTP pada perangkat. Aplikasi pada server akan menampilkan data anggota. Karyawan memasukkan data menu yang telah dipesan oleh pelanggan dan menawarkan jenis pembayaran berupa tunai atau saldo yang dimiliki oleh akun pelanggan. Sistem akan mencetak bukti pembayaran dan menyimpan data transaksi. Untuk *activity diagram* lainnya dapat dilihat pada lampiran.

4.1.5 Sequence Diagram

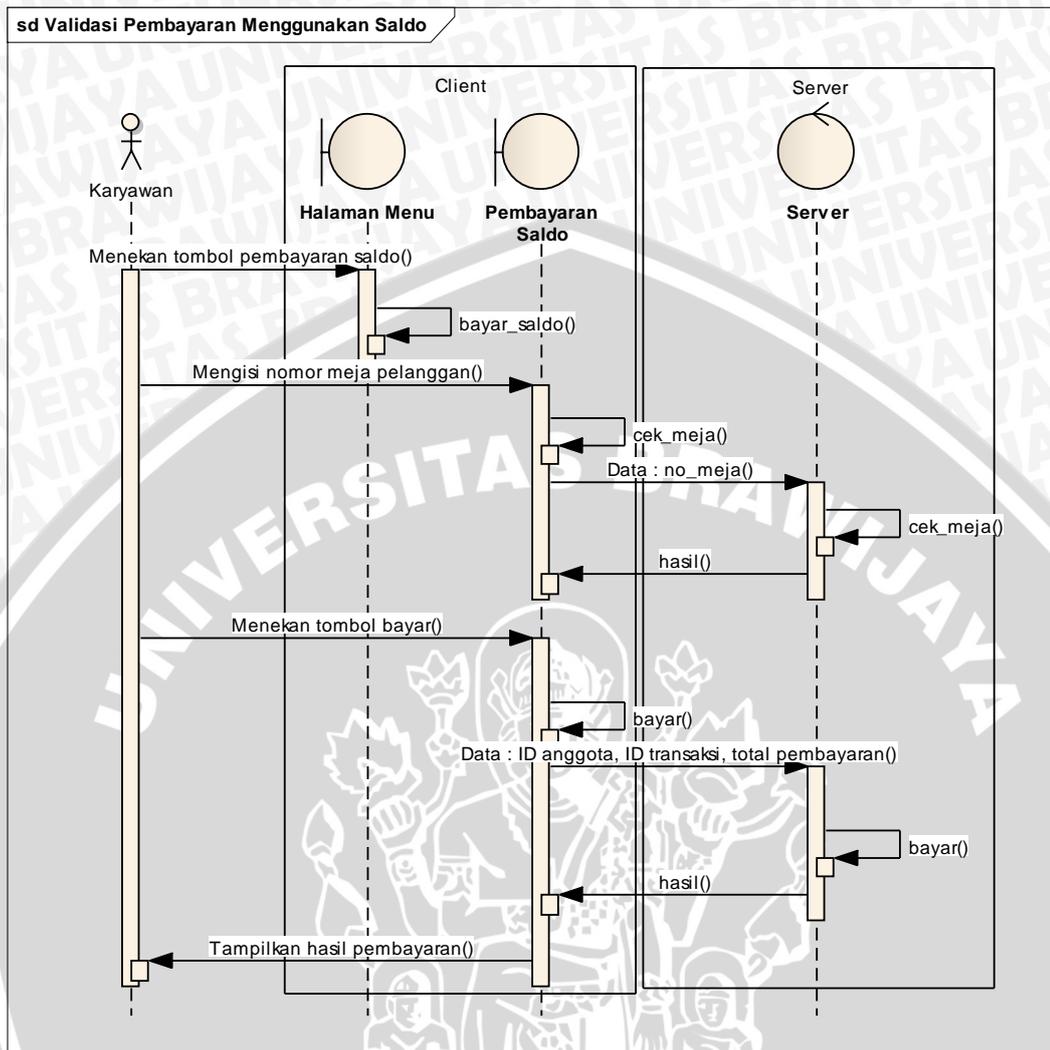
Perancangan *sequence diagram* dilakukan untuk mengetahui hubungan antar objek dalam aplikasi dan menerapkan fungsi-fungsi yang ada dalam objek.

Gambar 4.5 menunjukkan *sequence diagram* pada saat karyawan melakukan proses pendaftaran anggota baru.



Gambar 4.5 *Sequence Diagram* pada Skenario Mendaftarkan Pelanggan Menjadi Anggota

Dalam Gambar 4.5 menjelaskan hubungan antar objek sistem pada proses pendaftaran anggota baru. Objek yang terkait adalah karyawan sebagai aktor, halaman menu dan pendaftaran sebagai tampilan antarmuka pengguna sebagai jembatan jalannya proses, dan server sebagai pengendali proses pendaftaran yang berkaitan langsung dengan proses dalam basis data. Proses dimulai dengan karyawan menekan tombol daftar pada tampilan antarmuka halaman menu sehingga tampilan berubah menjadi antarmuka pendaftaran yang berisi form biodata anggota baru yang diisi oleh karyawan dan menekan tombol simpan untuk mengirim biodata ke dalam server. Selanjutnya server akan memasukkan biodata anggota untuk proses validasi.



Gambar 4.6 Sequence Diagram pada Skenario Validasi Pembayaran Menggunakan Saldo

Dalam Gambar 4.6 menjelaskan hubungan antar objek pada sistem pada proses pembayaran menggunakan saldo anggota. Objek yang terkait adalah karyawan sebagai aktor, halaman menu dan pembayaran saldo sebagai tampilan antarmuka pengguna, dan server sebagai pengendali proses. Proses diawali dengan karyawan menekan tombol pembayaran saldo pada halaman menu dan tampilan berubah menjadi antarmuka pembayaran saldo. Karyawan memasukkan nomor meja untuk mengetahui besarnya total pembayaran, sehingga dapat mengetahui apakah saldo yang dimiliki oleh anggota mencukupi untuk melakukan pembayaran. Saat saldo mencukupi, karyawan menekan tombol bayar dan sistem

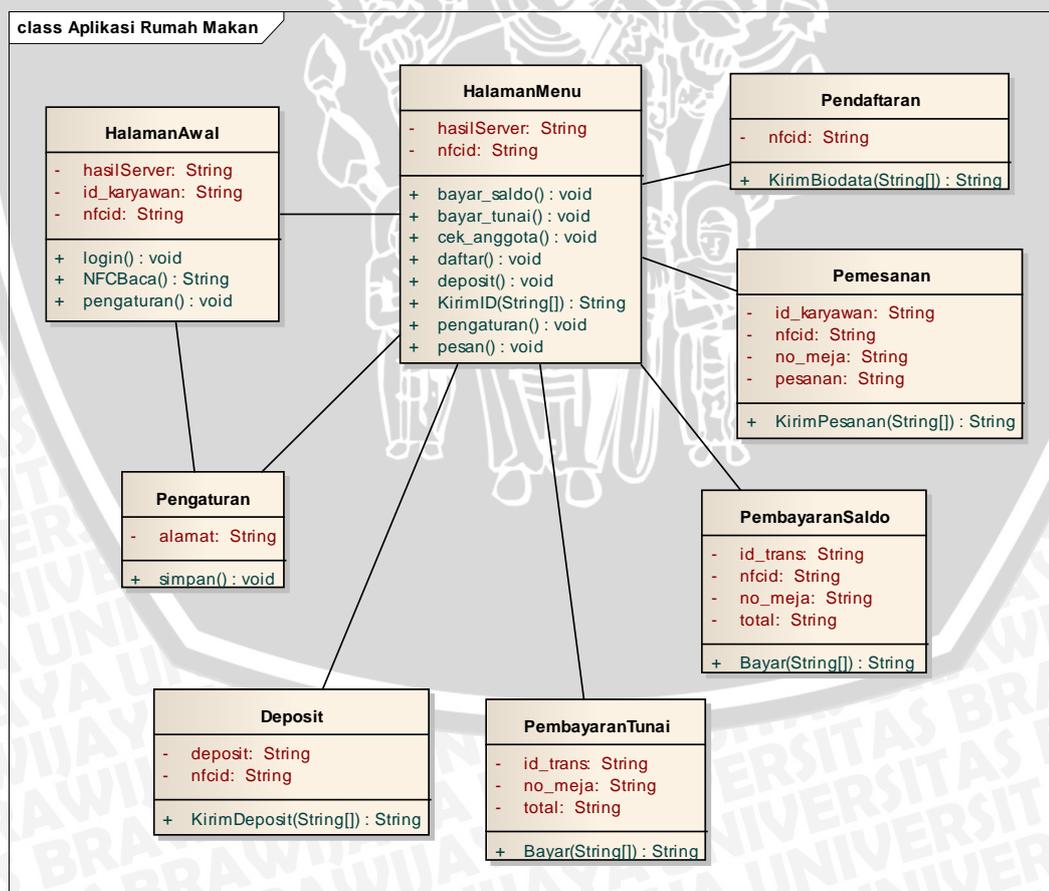
akan melakukan validasi pembayaran terhadap transaksi yang sudah dilakukan. Untuk *sequence diagram* pada proses lainnya dapat dilihat pada lampiran.

4.2 Perancangan Aplikasi Client

Perancangan aplikasi *client* terdiri dari perancangan kelas-kelas yang terlibat dimodelkan dengan *class diagram* dan perancangan antarmuka pengguna.

4.2.1 Perancangan Class Diagram

Perancangan *class diagram* bertujuan untuk mempermudah penelitian dalam membangun aplikasi di tahap implementasi. Setiap kelas menunjukkan proses yang berjalan pada layout atau tampilan antarmuka pengguna pada perangkat Android. Gambar 4.7 menunjukkan kelas-kelas yang terlibat dalam aplikasi client.



Gambar 4.7 Class Diagram pada Aplikasi Client

Tabel 4.4 Deskripsi kelas HalamanAwal

Nama Kelas : HalamanAwal
Deskripsi :
Kelas HalamanAwal merupakan kelas utama pada saat aplikasi dijalankan. HalamanAwal merupakan proses dari tampilan antarmuka halaman awal yang berisi login dan pengaturan. Fungsi utama kelas ini adalah melakukan pengecekan status pengguna sebagai karyawan untuk mengoperasikan fitur-fitur pada aplikasi
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembacaan ID dari NFC yang terpasang ada e-KTP hasilServer : berfungsi sebagai variabel untuk menyimpan hasil dari pembacaan identitas yang berasal dari server id_karyawan : berisi id karyawan yang telah melakukan proses login. Berfungsi untuk pengecekan identitas karyawan saat membuka aplikasi
Method :
NFCBaca : berfungsi untuk mengambil ID yang terdapat dalam NFC e-KTP dan menyimpannya ke dalam variabel nfcId login : berfungsi untuk mengirimkan ID yang telah didapat pada saat <i>method</i> NFCBaca sehingga pengguna dapat mengoperasikan fitur-fitur pada aplikasi pengaturan : berfungsi untuk mengubah tampilan menjadi tampilan pengaturan yang berisi pengaturan alamat server

Tabel 4.5 Deskripsi kelas HalamanMenu

Nama Kelas : HalamanMenu
Deskripsi :
Kelas HalamanMenu merupakan kelas dalam tampilan layout halaman menu, berfungsi untuk mengatur proses menu yang ditampilkan. Kelas HalamanMenu hanya dapat diakses setelah pengguna telah melakukan proses login pada halaman awal
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembacaan ID dari NFC yang terpasang ada e-KTP hasilServer : berfungsi sebagai variabel untuk menyimpan hasil dari pembacaan identitas yang berasal dari server.

Method :
bayar_saldo : berfungsi untuk membuka tampilan halaman pembayaran saldo
bayar_tunai : berfungsi untuk membuka tampilan halaman pembayaran tunai
cek_anggota : berfungsi untuk melakukan pengambilan nfcid pada e-KTP pelanggan
daftar : berfungsi untuk membuka tampilan pendaftaran
deposit : berfungsi untuk membuka tampilan penambahan saldo
KirimID : berfungsi untuk mengirimkan nfcid yang didapat pada proses cek_anggota ke dalam server untuk mendapatkan status keanggotaan pelanggan
pesan : berfungsi untuk membuka halaman pemesanan
pengaturan : berfungsi untuk mengubah tampilan menjadi tampilan pengaturan yang berisi pengaturan alamat server

Tabel 4.6 Deskripsi kelas Pendaftaran

Nama Kelas : Pendaftaran
Deskripsi :
Kelas Pendaftaran merupakan kelas dalam tampilan layout pendaftaran, berfungsi untuk mengatur proses pendaftaran anggota baru rumah makan. Kelas Pendaftaran diakses pada halaman menu
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembacaan ID dari NFC yang terpasang ada e-KTP
Method :
KirimBiodata : berfungsi untuk mengirimkan biodata pelanggan yang sudah diisi oleh karyawan ke dalam server

Tabel 4.7 Deskripsi kelas Pemesanan

Nama Kelas : Pemesanan
Deskripsi :
Kelas Pemesanan merupakan kelas dalam tampilan layout pemesanan, berfungsi untuk mengatur proses pemesanan menu yang dilakukan oleh karyawan. Kelas Pemesanan diakses pada halaman menu
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembacaan ID dari

NFC yang terpasang ada e-KTP
no_meja : berfungsi sebagai variabel untuk menyimpan nomor meja pelanggan untuk dikirim ke dalam server
pesanan : berfungsi sebagai variabel untuk menyimpan menu pesanan pelanggan untuk dikirim ke dalam server
id_karyawan : berisi id karyawan yang telah melakukan proses login. Berfungsi untuk mengenali identitas karyawan yang melakukan pencaatatan pesanan pelanggan
Method :
KirimID : berfungsi untuk mengirimkan data nomor meja, id karyawan, dan pesanan pelanggan ke dalam server

Tabel 4.8 Deskripsi kelas PembayaranSaldo

Nama Kelas : PembayaranSaldo
Deskripsi :
Kelas PembayaranSaldo merupakan kelas dalam tampilan layout pembayaran menggunakan saldo anggota, berfungsi untuk mengatur proses pembayaran anggota yang ingin menggunakan saldonya sebagai media pembayaran. Kelas PembayaranSaldo diakses pada halaman menu
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembacaan ID dari NFC yang terpasang ada e-KTP
no_meja : berfungsi sebagai variabel untuk menyimpan nomor meja pelanggan untuk dikirim ke dalam server
id_trans : berfungsi sebagai variabel untuk menyimpan id transaksi setelah pengecekan nomor meja
total : berfungsi sebagai variabel untuk menyimpan total harga menu yang telah dipesan setelah pengecekan nomor meja
Method :
Bayar : berfungsi untuk mengirimkan data pembayaran berupa id transaksi, nfcid anggota, dan total pembayaran ke dalam server

Tabel 4.9 Deskripsi kelas Pembayaran Tunai

Nama Kelas : Pembayaran Tunai
Deskripsi :
Kelas Pembayaran Tunai merupakan kelas dalam tampilan layout pembayaran menggunakan uang tunai, berfungsi untuk mengatur proses pembayaran menggunakan uang tunai sebagai media pembayaran. Kelas Pembayaran Tunai diakses pada halaman menu
Atribut :
no_meja : berfungsi sebagai variabel untuk menyimpan nomor meja pelanggan untuk dikirim ke dalam server id_trans : berfungsi sebagai variabel untuk menyimpan id transaksi setelah pengecekan nomor meja total : berfungsi sebagai variabel untuk menyimpan total harga menu yang telah dipesan setelah pengecekan nomor meja
Method :
Bayar : berfungsi untuk mengirimkan data pembayaran berupa id transaksi ke dalam server

Tabel 4.10 Deskripsi kelas Deposit

Nama Kelas : Deposit
Deskripsi :
Kelas Deposit merupakan kelas dalam tampilan layout penambahan saldo anggota, berfungsi untuk mengatur proses penambahan saldo anggota yang ingin melakukan deposit. Kelas Deposit diakses pada halaman menu
Atribut :
nfcid : berfungsi sebagai variabel untuk menyimpan hasil dari proses pembaruan ID dari NFC yang terpasang ada e-KTP deposit : berfungsi sebagai variabel untuk menyimpan besarnya deposit yang ingin dilakukan sebagai penambahan saldo
Method :
KirimDeposit : berfungsi untuk mengirimkan data penambahan deposit berupa nfcid anggota dan besarnya deposit ke dalam server

Tabel 4.11 Deskripsi kelas Pengaturan

Nama Kelas : Pengaturan
Deskripsi :
Kelas Pengaturan merupakan kelas dalam tampilan pengaturan alamat server, berfungsi untuk mengatur proses penyimpanan alamat server pada client. Kelas Pengaturan diakses pada halaman menu dan halaman awal
Atribut :
alamat : berfungsi sebagai variabel untuk menyimpan alamat server
Method :
simpan : berfungsi untuk menyimpan alamat server ke dalam memori perangkat berupa data tambahan aplikasi

4.2.2 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan rancangan awal pada bagian tampilan aplikasi. Tampilan antarmuka pengguna pada saat halaman menu dan halaman pendaftaran aplikasi client ditunjukkan dalam Gambar 4.8. Untuk tampilan antarmuka fitur-fitur lainnya dapat dilihat pada lampiran.

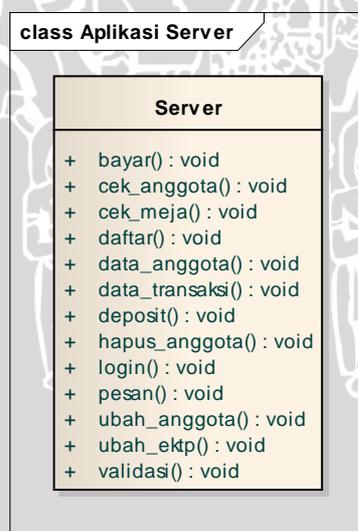
**Gambar 4.8** Tampilan Antarmuka Pengguna pada Pendaftaran Anggota

4.3 Perancangan Aplikasi Server

Aplikasi server berupa website dan dibuka melalui aplikasi browser. Perancangan aplikasi server meliputi perancangan kelas menggunakan *class diagram* dan perancangan basis data menggunakan *entity relationship diagram*. Perancangan basis data dilakukan untuk mempermudah proses implementasi saat membangun *database server* yang berfungsi sebagai penyimpanan data. Perancangan antarmuka pengguna bertujuan untuk memberikan gambaran terhadap tampilan aplikasi yang akan dibangun pada tahap implementasi.

4.3.1 Perancangan Class Diagram

Perancangan *class diagram* bertujuan untuk mempermudah penelitian dalam membangun aplikasi di tahap implementasi. Gambar 4.9 menunjukkan kelas Server yang memiliki satu atribut dan enam *method*. Tabel 4.12 menjelaskan atribut dan *method* yang terdapat pada kelas Server.



Gambar 4.9 Class Diagram pada Kelas Server

Tabel 4.12 Deskripsi kelas Server

Nama Kelas : Server
Deskripsi :
Kelas Server merupakan kelas yang mengatur semua proses data baik yang berasal dari

aplikasi client maupun aplikasi server.

Method :

bayar : fungsi ini bertugas untuk melakukan validasi pembayaran yang dilakukan oleh aplikasi client

cek_anggota : fungsi ini bertugas untuk menentukan apakah identitas yang terbaca oleh perangkat merupakan anggota atau bukan

cek_meja : fungsi ini bertugas untuk mengambil total biaya transaksi yang dilakukan pelanggan berdasarkan nomor meja pelanggan

daftar : fungsi ini bertugas untuk melakukan pendaftaran anggota baru pada aplikasi client dan membutuhkan validasi dari aplikasi server hingga pelanggan dapat menjadi anggota

data_anggota : fungsi ini digunakan untuk menampilkan semua data anggota pada aplikasi server

data_transaksi : fungsi ini digunakan untuk menampilkan semua data transaksi pada aplikais server

deposit : fungsi ini bertugas untuk melakuka penambahan saldo anggota yang dilakukan oleh aplikasi client

hapus_anggota : fungsi ini bertugas untuk menghapus data anggota yang dilakukan oleh aplikasi server

login : fungsi ini bertugas untuk melakukan validasi identitas karyawan saat melakukan proses login pada aplikasi client

pesan : fungsi ini bertugas untuk mencatat pesanan pelanggan

ubah_anggota : fungsi ini bertugas untuk melakukan perubahan data anggota pada aplikasi server

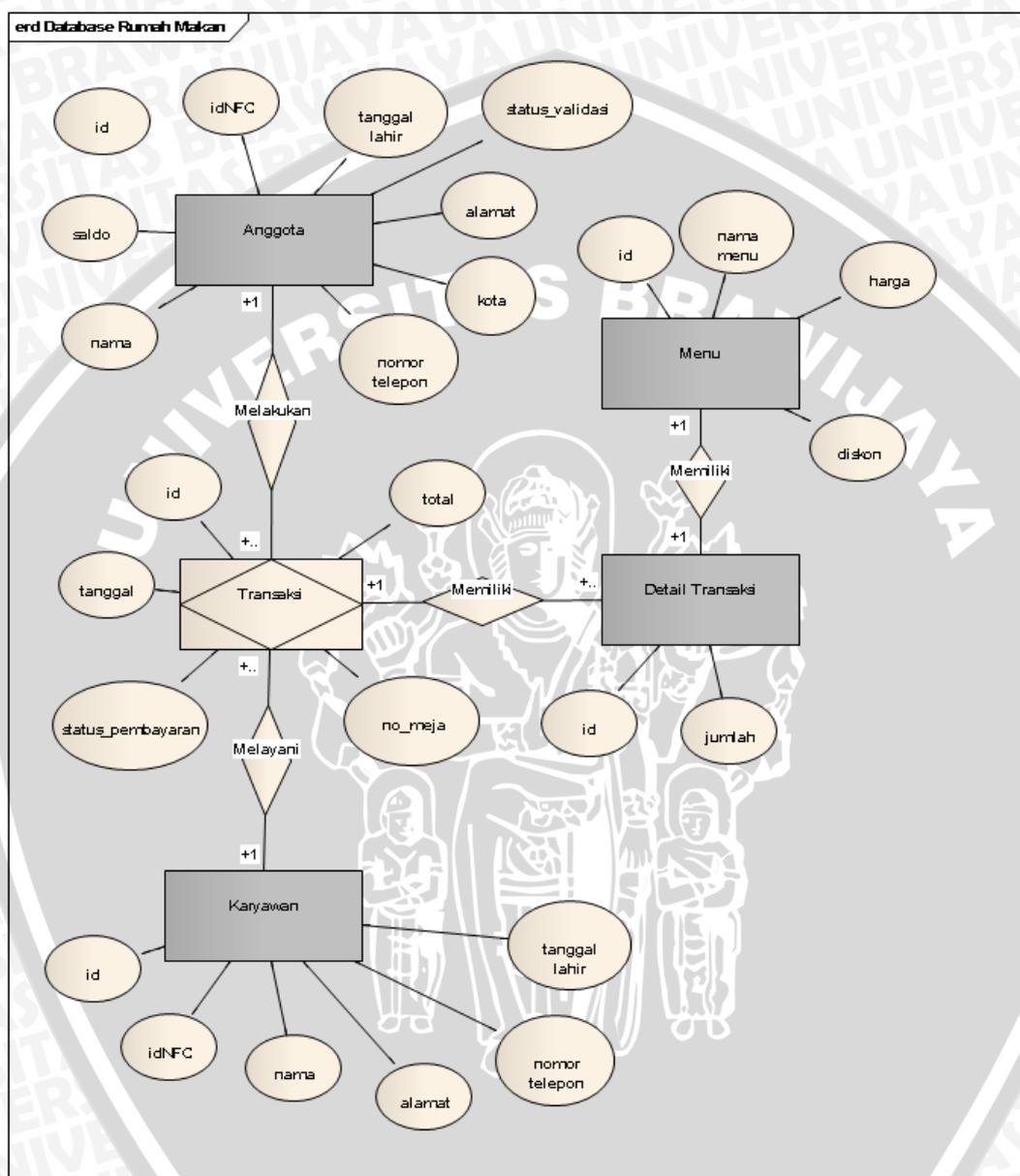
ubah_ektp : fungsi ini bertugas untuk mengambil data identitas nfc baru yang berasal dari e-ktip pada perangkat client apabila anggota ingin mengubah e-ktip yang dimilikinya

validasi : fungsi ini bertugas untuk melakukan validasi anggota baru pada aplikasi server

4.3.2 Perancangan Basis Data

Perancangan basis data digunakan untuk mempermudah proses implementasi saat membangun *database server* yang berfungsi menyimpan data. Basis data dimodelkan dalam bentuk tabel yang memiliki atribut data di setiap tabelnya dan menyimpan lebih dari satu data di setiap atribut serta memiliki relasi atau hubungan antara tabel satu dengan yang lainnya. Perancangan basis data

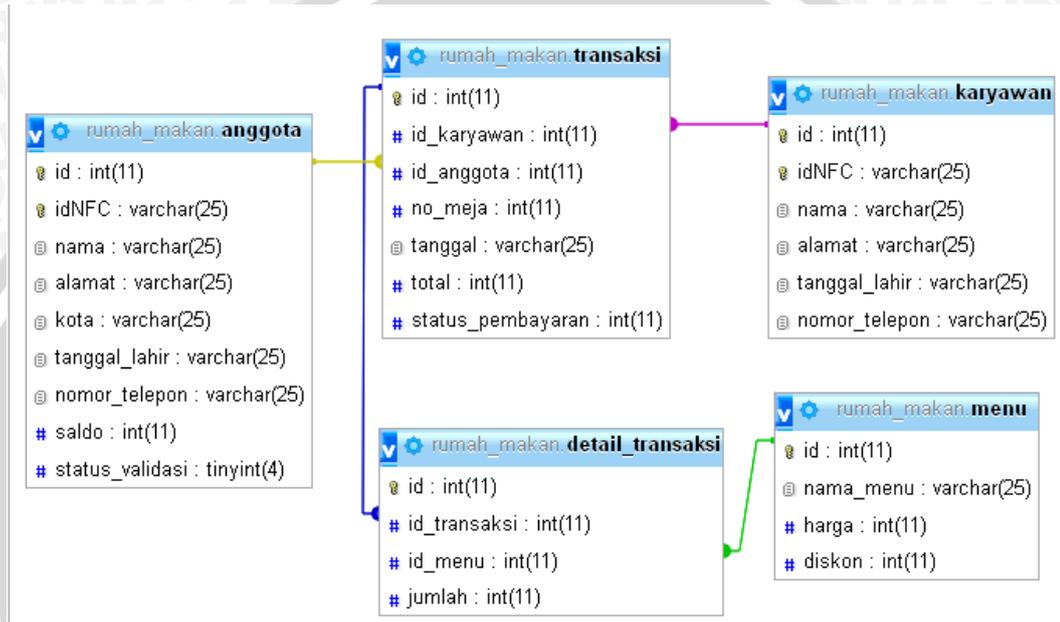
dimodelkan menggunakan *entity relationship diagram* dan *physical data model* yang ditunjukkan dalam Gambar 4.9 dan Gambar 4.10.



Gambar 4.10 Entity Relationship Diagram Basis Data Rumah Makan

Gambar 4.10 menjelaskan hubungan antar entitas yang mempunyai beberapa atribut. Entitas Anggota berelasi dengan entitas transaksi, relasi antara dua entitas tersebut adalah anggota melakukan transaksi dalam proses pembelian menu makanan. Entitas Karyawan berelasi dengan entitas transaksi, relasi antara

dua entitas adalah karyawan melayani transaksi yang dilakukan oleh anggota pada saat proses pembayaran. Entitas Detail Transaksi merupakan entitas pelengkap yang dimiliki oleh entitas Transaksi berisikan menu yang dibeli beserta jumlahnya. Entitas Menu berasosiasi dengan entitas Detail Transaksi.



Gambar 4.11 *Physical Data Model* Basis Data Rumah Makan

Gambar 4.11 menjelaskan tabel yang dibuat berdasarkan *Entity Relationship Diagram* yang ditunjukkan dalam Gambar 4.10. *Physical Data Model* berisikan tabel dan relasinya dengan tabel lain, setiap tabel memiliki atribut beserta tipe data yang dapat disimpan dalam atribut tersebut. Relasi antar tabel menggunakan fungsi *foreign key* sebagai penghubungnya.

BAB V

IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi sistem pada aplikasi client dan server berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan aplikasi. Pembahasan terdiri atas penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi client, dan implementasi server. Implementasi client dibagi menjadi implementasi kebutuhan sistem, implementasi kelas, dan implementasi layout. Implementasi server dibagi menjadi implementasi basis data dan implementasi kelas.

5.1 Spesifikasi Sistem

Sistem terbagi menjadi dua bagian, yaitu client dan server. Sehingga implementasi sistem bekerja pada dua lingkungan perangkat keras dan perangkat lunak yang berbeda antara satu dengan yang lainnya. Perangkat keras yang digunakan pada lingkungan client adalah *smartphone* berbasis Android dan perangkat keras yang digunakan pada lingkungan server adalah komputer berbasis Windows.

5.1.1 Spesifikasi Sistem Client

Tabel 5.1 Spesifikasi Perangkat Keras Client

Samsung Galaxy S4	
<i>CPU</i>	Quad-core 1.6 GHz Cortex-A15
<i>Memory(RAM)</i>	2 GB
<i>Internal Memory</i>	16 GB
<i>GPU</i>	PowerVR SGX544MP3
<i>Monitor</i>	1080 x 1920 pixels, 5" (~441 ppi pixel density)

Spesifikasi perangkat lunak pada client berupa Android OS v4.2.2 (Jelly Bean).

5.1.2 Spesifikasi Sistem Server

Tabel 5.2 Spesifikasi Perangkat Keras Server

Toshiba Satellite L510	
<i>Processor</i>	Intel® Core™2 Duo T6500 @2.10Ghz
<i>Memory(RAM)</i>	2 GB
<i>Harddisk</i>	300 GB
<i>Graphic Card</i>	Mobile Intel® 4 Series 1 GB VRAM
<i>Monitor</i>	14" Widescreen LED Backlit Display

Tabel 5.3 Spesifikasi Perangkat Lunak Server

Windows 8.1 Pro	
<i>DirectX Version</i>	DirectX 11
<i>Programming Language</i>	Java
<i>Integrated Development Environment</i>	Android Developer Tools Bundle (Eclipse + ADT Plugin, Android SDK Tools, Android Platform Tools)
<i>Localhost</i>	Apache v2.4.3
<i>Database</i>	MySQL v5.5.27
<i>PHP</i>	PHP v5.4.7

5.2 Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem baik pada client maupun server adalah sebagai berikut :

1. Aplikasi client hanya dapat dijalankan pada OS Android 2.3.3 Gingerbread (API versi 10) ke atas dan dilengkapi dengan perangkat atau sensor NFC.
2. Aplikasi client maupun server tidak mengelola data karyawan dan data menu makanan.
3. Tampilan antarmuka pengguna aplikasi server hanya berupa prototype.

5.3 Implementasi Client

Implementasi client dilakukan untuk membangun aplikasi client pada perangkat bergerak berbasis Android. Implementasi dibagi menjadi tiga bagian,

yaitu implementasi kebutuhan, implementasi kelas, dan implementasi layout. Implementasi kebutuhan diperlukan untuk menciptakan kondisi agar aplikasi dapat berjalan dengan lancar pada perangkat. Implementasi kelas merupakan realisasi terhadap perancangan kelas yang sudah dibuat pada bab perancangan. Implementasi layout adalah tampilan antarmuka pada perangkat.

5.3.1 Implementasi Kebutuhan

Daftar kebutuhan yang harus terpenuhi agar aplikasi dapat berjalan pada perangkat antara lain :

1. Dapat mengakses perangkat atau sensor NFC
2. Dapat mengakses aplikasi langsung saat tag NFC ditemukan
3. Dapat melakukan koneksi internet menggunakan jaringan Wifi

Kebutuhan-kebutuhan tersebut dapat terpenuhi dengan menambahkan kondisi pada file *AndroidManifest.xml*.

```
<uses-permission android:name="android.permission.NFC" />
```

Kondisi diatas dapat ditambahkan pada *AndroidManifest.xml* sebelum pendeklarasian *activity*, digunakan untuk memenuhi kebutuhan dalam mengakses perangkat atau sensor NFC. Kondisi dimana aplikasi dapat langsung berjalan saat terdapat tag NFC ditemukan dapat dilakukan dengan menambahkan *intent-filter* yang berguna sebagai kondisi saat suatu aksi ditemukan pada bagian *activity* utama.

```
<activity
    android:name="com.client.rumahmakan.NFCReader"
    android:label="@string/app_name" >
    <intent-filter>
        <action
            android:name="android.intent.action.MAIN" />

            <category
                android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <intent-filter>
            <action
                android:name="android.nfc.action.TAG_DISCOVERED" />

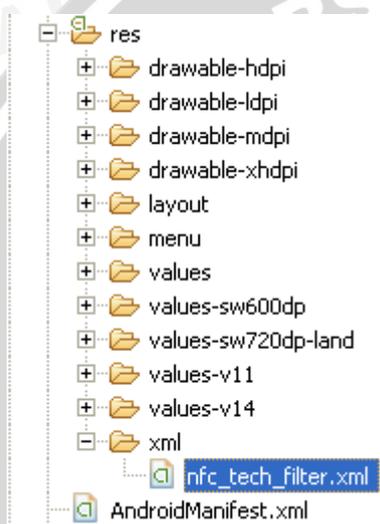
            <category
                android:name="android.intent.category.DEFAULT" />
        </intent-filter>
```

```

<meta-data
android:name="android.nfc.action.TAG_DISCOVERED"
    android:resource="@xml/nfc_tech_filter" />
</activity>

```

Pada aplikasi client ini, aksi yang di-filter atau disaring adalah TAG_DISCOVERED, yaitu aksi dimana perangkat mendeteksi adanya perangkat yang memiliki tag NFC. Untuk menyaring jenis tag NFC apa saja yang dapat dibaca oleh perangkat dapat menggunakan meta-data dengan menambahkan nfc_tech_filter.xml pada aplikasi.



```

<resources>
  <tech-list>
    <tech>android.nfc.tech.IsoDep</tech>

    <tech>android.nfc.tech.NfcA</tech>
    <tech>android.nfc.tech.NfcB</tech>
    <tech>android.nfc.tech.NfcF</tech>
    <tech>android.nfc.tech.NfcV</tech>
    <tech>android.nfc.tech.Ndef</tech>
    <tech>android.nfc.tech.NdefFormatable
</tech>
    <tech>android.nfc.tech.MifareClassic<
</tech>
    <tech>android.nfc.tech.MifareUltralig
ht</tech>
  </tech-list>
</resources>

```

e-KTP walaupun tidak menggunakan teknologi NFC, namun chip e-KTP dapat dibaca dengan teknologi nfc jenis NfcA, meskipun informasi yang dapat diambil hanya nomor identitas dari chip e-KTP tersebut, bukan data yang ada dalam chip. Nomor identitas yang dapat dibaca berupa kumpulan byte. Kondisi selanjutnya yang harus terpenuhi adalah koneksi internet untuk menghubungkan client dengan server pada proses pengiriman dan penerimaan data.

```

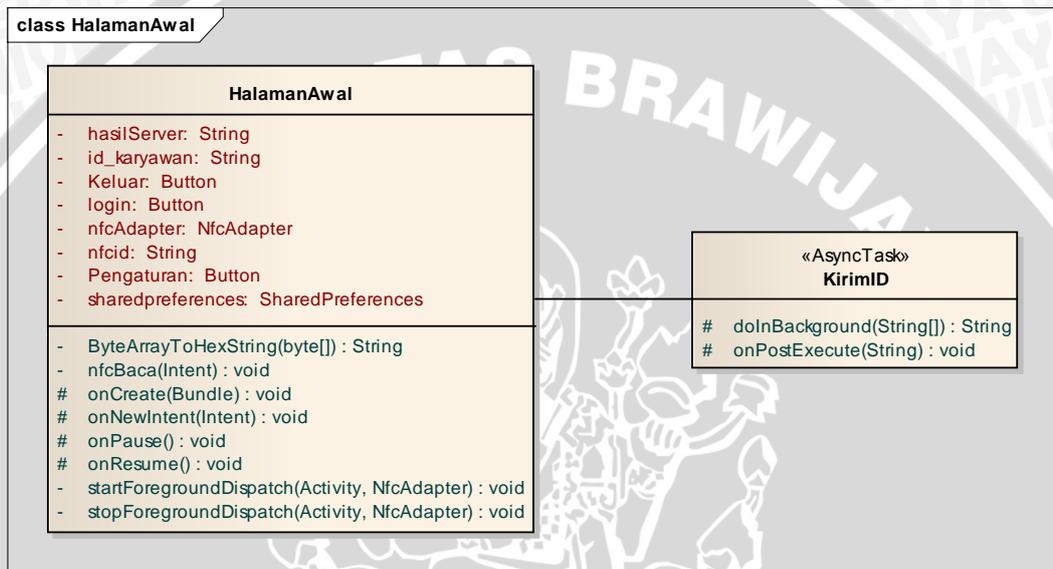
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

```

5.3.2 Implementasi Kelas

Implementasi kelas dilakukan setelah *permission* atau perizinan terhadap fitur-fitur tertentu sudah terpenuhi. Kelas berfungsi sebagai pengelola data pada aplikasi. Terdapat 8 kelas yang dibangun sesuai tahap perancangan, yaitu kelas HalamanAwal, HalamanMenu, Pemesanan, PembayaranSaldo, PembayaranTunai, Pendaftaran, Deposit, dan Pengaturan.

A. Kelas HalamanAwal



Gambar 5.1 Implementasi Kelas HalamanAwal

Kelas HalamanAwal berfungsi sebagai kelas utama pada aplikasi yang mengatur autentikasi pengguna sebagai karyawan. Kelas HalamanAwal mempunyai 8 fungsi atau *method* antara lain:

1. ByteArrayToHexString

Fungsi ini bertugas untuk mengkonversi atau mengubah data yang dibaca oleh perangkat NFC pada e-KTP. Hasil pembacaan awal berupa kumpulan data yang disimpan dalam *array* bertipe data *byte* sehingga perlu dikonversi menjadi data dengan satu nilai untuk mempermudah basis data dalam menyimpan nilai tersebut. Fungsi ini mengubah kumpulan data tersebut menjadi satu nilai bertipe data *String* dalam basis hexadesimal.

```

private String ByteArrayToHexString(byte [] inarray) {
    int i, j, in;
    String [] hex =
    
```



```

{"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E",
, "F"};
    String out= "";
    for(j = 0 ; j < inarray.length ; ++j)
    {
        in = (int) inarray[j] & 0xff;
        i = (in >> 4) & 0x0f;
        out += hex[i];
        i = in & 0x0f;
        out += hex[i];
    }
    return out;
}

```

2. nfcBaca

Fungsi ini bertugas untuk mengambil identitas NFC pada e-KTP dan menyimpannya dalam nfcid.

```

private void nfcBaca(Intent intent) {
    if
(intent.getAction().equals(NfcAdapter.ACTION_TAG_DISCOVERED)
) {

        nfcid=ByteArrayToHexString(intent.getByteArrayExtra(Nf
cAdapter.EXTRA_ID));
    }
}

```

3. onCreate

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanAwal terbentuk. Tugas dari fungsi ini antara lain menampilkan layout yang sesuai dengan aktifitas HalamanAwal, mengatur fungsi tiap tombol pada layout, dan mengatur nilai awal dari setiap atribut layout.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.halaman_awal);

    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);
    nfcid="";
    hasilServer="";
    id_karyawan="";

    if (sharedPreferences.contains("Karyawan"))

        id_karyawan=sharedPreferences.getString("Id_karyawan",
"");
}

```

```
        if(!id_karyawan.equals("")){
            Intent intent=new
            Intent(HalamanAwal.this,HalamanMenu.class);
            HalamanAwal.this.startActivity(intent);
            finish();
        }

        login=(Button) findViewById(R.id.tombol_login);
        login.setOnClickListener(new
        View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                if(nfcid.equals("")){

                    Toast.makeText(HalamanAwal.this, "Silahkan Tempelkan
                    e-KTP Pada Perangkat", Toast.LENGTH_LONG).show();
                }
                else{
                    new KirimID().execute(nfcid);
                }
            }
        });

        pengaturan=(Button)
        findViewById(R.id.tombol_pengaturan);
        pengaturan.setOnClickListener(new
        View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent=new
                Intent(HalamanAwal.this,Pengaturan.class);

                HalamanAwal.this.startActivity(intent);
            }
        });

        keluar=(Button)
        findViewById(R.id.tombol_keluar);
        keluar.setOnClickListener(new
        View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                finish();
            }
        });

        nfcAdapter = NfcAdapter.getDefaultAdapter(this);

        if (nfcAdapter == null) {
            Toast.makeText(HalamanAwal.this, "Perangkat Anda
            Tidak Mendukung NFC Reader", Toast.LENGTH_LONG).show();
            finish();
            return;
        }
    }
}
```

```
        if (!nfcAdapter.isEnabled()) {  
            Toast.makeText(HalamanAwal.this, "Silahkan  
Aktifkan NFC Reader Perangkat Anda",  
Toast.LENGTH_LONG).show();  
        }  
  
        nfcBaca(getIntent());  
    }
```

4. onNewIntent

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanAwal menemukan intent saat aplikasi masih berjalan dan mengirimkan intent tersebut pada fungsi aksiIntent.

```
protected void onNewIntent(Intent intent) {  
    nfcBaca(intent);  
}
```

5. onPause

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanAwal melakukan *pause* atau berhenti sementara, misalnya pada saat berpindah ke aktifitas lain. Bertugas untuk menghentikan *foreground dispatch system* saat aktifitas lain terbuka.

```
protected void onPause() {  
    stopForegroundDispatch(this, nfcAdapter);  
    super.onPause();  
}
```

6. onResume

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanAwal tidak lagi dalam kondisi *pause*, atau aktifitas lain sudah dijalankan dan kembali ke aktifitas HalamanAwal. Bertugas untuk mengaktifkan kembali *foreground dispatch system*.

```
protected void onResume() {  
    super.onResume();  
    startForegroundDispatch(this, nfcAdapter);  
}
```

```
}
```

7. startForegroundDispatch

Fungsi ini bertugas untuk mengatur *foreground dispatch system*, yaitu sebuah sistem yang memberikan prioritas tertentu saat aplikasi berjalan. Pada kasus ini apabila perangkat mempunyai beberapa aplikasi yang dapat membaca NFC Tag dan aktifitas HalamanAwal sedang berjalan, maka perangkat tidak menampilkan pilihan aplikasi untuk membaca intent tag NFC yang ditemukan, tetapi langsung tertuju pada aktifitas HalamanAwal.

```
private static void startForegroundDispatch(final Activity
activity, NfcAdapter adapter) {
    final Intent intent = new
Intent(activity.getApplicationContext(),
activity.getClass());
    intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);

    final PendingIntent pendingIntent =
PendingIntent.getActivity(activity.getApplicationContext(),
0, intent, 0);

    IntentFilter[] filters = new IntentFilter[1];
    String[][] techList = new String[][]{};

    filters[0] = new IntentFilter();

    filters[0].addAction(NfcAdapter.ACTION_TAG_DISCOVERED);
    filters[0].addCategory(Intent.CATEGORY_DEFAULT);

    adapter.enableForegroundDispatch(activity,
pendingIntent, filters, techList);
}
```

8. stopForegroundDispatch

Fungsi ini bertugas untuk menghentikan *foreground dispatch system*, terutama saat aktifitas HalamanAwal sedang tidak berjalan.

```
private static void stopForegroundDispatch(final Activity
activity, NfcAdapter adapter) {
    adapter.disableForegroundDispatch(activity);
}
```

Kelas HalamanAwal memiliki subclass KirimID yang bertugas untuk mengirimkan identitas e-ktip yang sudah terbaca ke dalam server secara asinkronus.

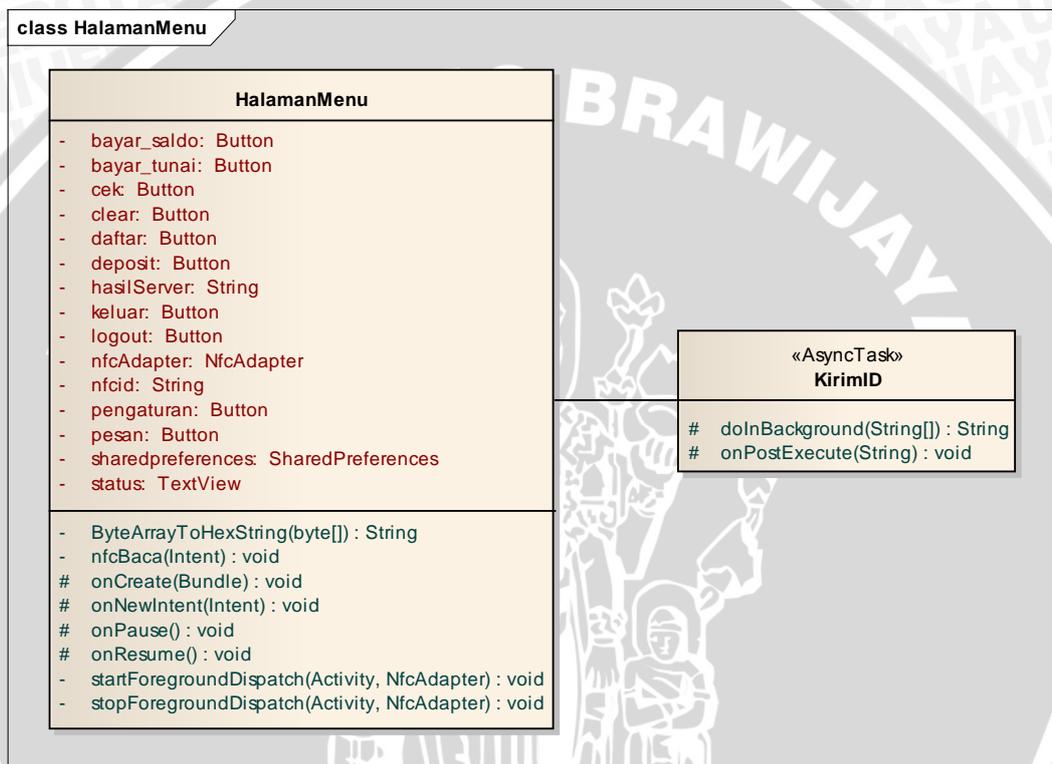
```
private class KirimID extends AsyncTask<String, Void, String> {  
    @Override  
    protected String doInBackground(String... params) {  
        try {  
            String ipaddr="10.10.10.1/rumahmakan";  
            if (sharedpreferences.contains("AlamatServer")){  
                ipaddr=sharedpreferences.getString("AlamatServer", "");  
            }  
            String link="http://"+ipaddr+"/proses.php";  
            String data = URLEncoder.encode("aksi", "UTF-8")  
+ "=" + URLEncoder.encode("login", "UTF-8");  
            data += "&" + URLEncoder.encode("nfcid", "UTF-8")  
+ "=" + URLEncoder.encode(params[0], "UTF-8");  
            URL url = new URL(link);  
            URLConnection conn = url.openConnection();  
            conn.setDoOutput(true);  
            OutputStreamWriter wr = new  
OutputStreamWriter(conn.getOutputStream());  
            wr.write(data);  
            wr.flush();  
            BufferedReader reader = new BufferedReader(new  
InputStreamReader(conn.getInputStream()));  
            StringBuilder sb = new StringBuilder();  
            String line = null;  
            while((line = reader.readLine()) != null)  
            {  
                sb.append(line);  
                break;  
            }  
            hasilServer = sb.toString();  
        } catch (Exception e) {  
            hasilServer="gagal";  
        }  
        return hasilServer;  
    }  
  
    @Override  
    protected void onPostExecute(String result) {  
        if(hasilServer.equals("gagal"))  
            Toast.makeText(HalamanAwal.this, "Koneksi Ke  
Server Gagal", Toast.LENGTH_LONG).show();  
        else if(hasilServer.equals("bukan")){  
            Toast.makeText(HalamanAwal.this, "Anda  
Tidak Terdaftar Sebagai Karyawan", Toast.LENGTH_LONG).show();  
        }  
        else{  
            Editor editor = sharedpreferences.edit();  
            String[] data=hasilServer.split("/");  
            editor.putString("Id_karyawan", data[0]);  
            editor.putString("Karyawan", data[1]);  
        }  
    }  
}
```

```

        editor.commit();
        Intent intent=new
Intent (HalamanAwal.this, HalamanMenu.class);
        HalamanAwal.this.startActivity(intent);
        finish();
    }
}
}

```

B. Kelas HalamanMenu



Gambar 5.2 Implementasi Kelas HalamanMenu

Kelas HalamanMenu berfungsi sebagai kelas utama saat pengguna sudah terautentikasi sebagai karyawan. Kelas ini berisi semua fitur utama dari aplikasi dan bertugas sebagai jembatan aktifitas antar fitur. Kelas HalamanMenu juga sebagai kelas yang melakukan pengecekan keanggotaan pelanggan. Kelas HalamanMenu mempunyai 8 fungsi atau *method* antara lain:

1. ByteArrayToHexString

Fungsi ini bertugas untuk mengkonversi atau mengubah data yang dibaca oleh perangkat NFC pada e-KTP. Hasil pembacaan awal berupa kumpulan

data yang disimpan dalam *array* bertipe data *byte* sehingga perlu dikonversi menjadi data dengan satu nilai untuk mempermudah basis data dalam menyimpan nilai tersebut. Fungsi ini mengubah kumpulan data tersebut menjadi satu nilai bertipe data *String* dalam basis hexadesimal.

```
private String ByteArrayToHexString(byte [] inarray) {
    int i, j, in;
    String [] hex =
    {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E",
    "F"};
    String out= "";

    for(j = 0 ; j < inarray.length ; ++j)
    {
        in = (int) inarray[j] & 0xff;
        i = (in >> 4) & 0x0f;
        out += hex[i];
        i = in & 0x0f;
        out += hex[i];
    }
    return out;
}
```

2. nfcBaca

Fungsi ini bertugas untuk mengambil identitas NFC pada e-KTP dan menyimpannya dalam *nfcid*.

```
private void nfcBaca(Intent intent) {
    if
    (intent.getAction().equals(NfcAdapter.ACTION_TAG_DISCOVERED)
    ) {

        nfcid=ByteArrayToHexString(intent.getByteArrayExtra(Nf
        cAdapter.EXTRA_ID));
    }
}
```

3. onCreate

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak *Activity* yang dijalankan apabila aktifitas *HalamanMenu* terbentuk. Tugas dari fungsi ini antara lain menampilkan layout yang sesuai dengan aktifitas *HalamanMenu*, mengatur fungsi tiap tombol pada layout, dan mengatur nilai awal dari setiap atribut layout.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.halaman_menu);
}
```

```
        sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

        String
karyawan=sharedPreferences.getString("Karyawan", "");
        TextView nama=(TextView)
findViewById(R.id.nama);
        nama.setText(" "+karyawan+" ");

        nfcid="";
        hasilServer="";

        status=(TextView) findViewById(R.id.status_anggota);

        bayar_saldo=(Button) findViewById(R.id.menu_pembayaran_
saldo);
        bayar_saldo.setClickable(false);
        bayar_saldo.setBackgroundColor(Color.BLACK);
        bayar_saldo.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent=new
Intent(HalamanMenu.this,PembayaranSaldo.class);
                String[]
data=hasilServer.split("/");
                intent.putExtra("nfcid", nfcid);
                intent.putExtra("nomor", data[0]);
                intent.putExtra("nama", data[1]);
                intent.putExtra("saldo", data[2]);

                HalamanMenu.this.startActivity(intent);
            }
        });

        bayar_tunai=(Button) findViewById(R.id.menu_pembayaran_
tunai);
        bayar_tunai.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent=new
Intent(HalamanMenu.this,PembayaranTunai.class);

                HalamanMenu.this.startActivity(intent);
            }
        });

        daftar=(Button) findViewById(R.id.menu_daftar);
        daftar.setClickable(false);
        daftar.setBackgroundColor(Color.BLACK);
```

```
daftar.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        Intent intent=new
Intent (HalamanMenu.this,Pendaftaran.class);
        intent.putExtra("nfcid", nfcid);

        HalamanMenu.this.startActivity(intent);
    }
});

deposit=(Button) findViewById(R.id.menu_deposit);
deposit.setClickable(false);
deposit.setBackgroundColor(Color.BLACK);
deposit.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        Intent intent=new
Intent (HalamanMenu.this,Deposit.class);
        String[]
data=hasilServer.split("/");
        intent.putExtra("nomor", data[0]);
        intent.putExtra("nama", data[1]);
        intent.putExtra("saldo", data[2]);

        HalamanMenu.this.startActivity(intent);
    }
});

cek_anggota=(Button) findViewById(R.id.menu_cek);
cek_anggota.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        if(nfcid.equals(""))
            status.setText("Silahkan
Tempelkan e-KTP Pada Perangkat");
        else
            new KirimID().execute(nfcid);
    }
});

clear=(Button) findViewById(R.id.menu_clear);
clear.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        nfcid="";
        daftar.setClickable(false);
        daftar.setBackgroundColor(Color.BLACK);
        deposit.setClickable(false);
```

```
deposit.setBackgroundColor(Color.BLACK);
bayar_saldo.setClickable(false);

bayar_saldo.setBackgroundColor(Color.BLACK);
status.setText("Status Keanggotaan
Pelanggan");
    }
});

pesan=(Button)findViewById(R.id.menu_pesan);
pesan.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        Intent intent=new
Intent(HalamanMenu.this,Pemesanan.class);
        intent.putExtra("nfcid", nfcid);

HalamanMenu.this.startActivity(intent);
    }
});

logout=(Button)findViewById(R.id.menu_logout);
logout.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        Editor editor =
sharedpreferences.edit();
        editor.putString("Karyawan", "");
        editor.putString("Id_karyawan", "");
        editor.commit();
        Toast.makeText(HalamanMenu.this, "Anda
Telah Logout", Toast.LENGTH_LONG).show();
        finish();
    }
});

pengaturan=(Button)findViewById(R.id.menu_pengaturan);
pengaturan.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        Intent intent=new
Intent(HalamanMenu.this,Pengaturan.class);

HalamanMenu.this.startActivity(intent);
    }
});

keluar=(Button)findViewById(R.id.menu_keluar);
keluar.setOnClickListener(new
View.OnClickListener() {
```

```

        @Override
        public void onClick(View arg0) {
            finish();
        }
    });

    nfcAdapter = NfcAdapter.getDefaultAdapter(this);

    if (nfcAdapter == null) {
        Toast.makeText(HalamanMenu.this, "Perangkat Anda
        Tidak Mendukung NFC Reader", Toast.LENGTH_LONG).show();
        finish();
        return;
    }

    if (!nfcAdapter.isEnabled()) {
        Toast.makeText(HalamanMenu.this, "Silahkan
        Aktifkan NFC Reader Perangkat Anda",
        Toast.LENGTH_LONG).show();
    }
}

```

4. onNewIntent

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanMenu menemukan intent, khususnya intent saat ditemukannya tag NFC, saat aplikasi masih berjalan dan mengirimkan intent tersebut pada fungsi nfcBaca.

```

protected void onNewIntent(Intent intent) {
    nfcBaca(intent);
}

```

5. onPause

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanMenu melakukan *pause* atau berhenti sementara, misalnya pada saat berpindah ke aktifitas lain. Bertugas untuk menghentikan *foreground dispatch system* saat aktifitas lain terbuka.

```

protected void onPause() {
    stopForegroundDispatch(this, nfcAdapter);
    super.onPause();
}

```

6. onResume

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas HalamanMenu tidak lagi dalam kondisi *pause*, atau aktifitas lain sudah dijalankan dan kembali ke aktifitas HalamanMenu. Bertugas untuk mengaktifkan kembali *foreground dispatch system*.

```
protected void onResume () {
    super.onResume ();
    startForegroundDispatch(this, nfcAdapter);
}
```

7. startForegroundDispatch

Fungsi ini bertugas untuk mengatur *foreground dispatch system*, yaitu sebuah sistem yang memberikan prioritas tertentu saat aplikasi berjalan. Pada kasus ini apabila perangkat mempunyai beberapa aplikasi yang dapat membaca NFC Tag dan aktifitas HalamanMenu sedang berjalan, maka perangkat tidak menampilkan pilihan aplikasi untuk membaca intent tag NFC yang ditemukan, tetapi langsung tertuju pada aktifitas HalamanMenu.

```
private static void startForegroundDispatch(final Activity
activity, NfcAdapter adapter) {
    final Intent intent = new
Intent(activity.getApplicationContext(),
activity.getClass());
    intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);

    final PendingIntent pendingIntent =
PendingIntent.getActivity(activity.getApplicationContext(),
0, intent, 0);

    IntentFilter[] filters = new IntentFilter[1];
    String[][] techList = new String[][]{};

    filters[0] = new IntentFilter();

    filters[0].addAction(NfcAdapter.ACTION_TAG_DISCOVERED);
    filters[0].addCategory(Intent.CATEGORY_DEFAULT);

    adapter.enableForegroundDispatch(activity,
pendingIntent, filters, techList);
}
```

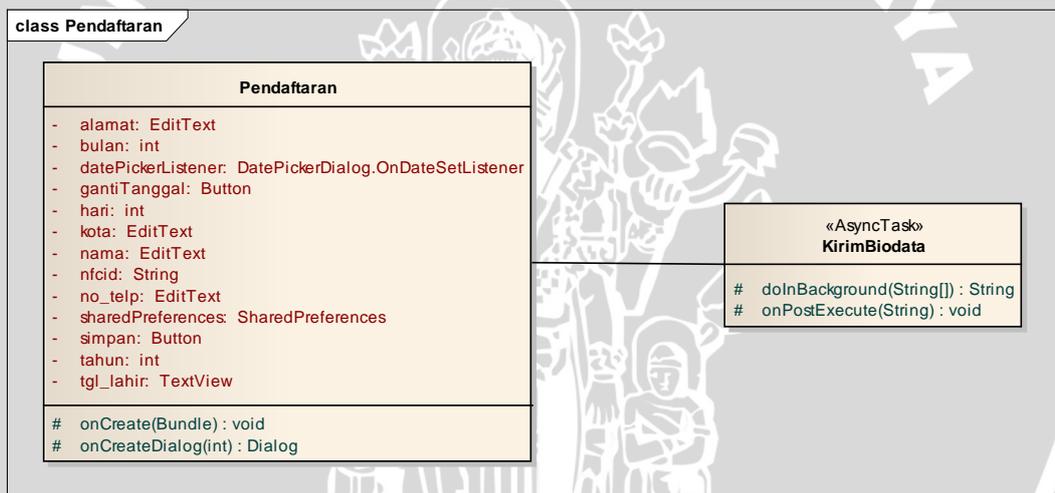
8. stopForegroundDispatch

Fungsi ini bertugas untuk menghentikan *foreground dispatch system*, terutama saat aktifitas HalamanMenu sedang tidak berjalan.

```
private static void stopForegroundDispatch(final Activity
activity, NfcAdapter adapter) {
    adapter.disableForegroundDispatch(activity);
}
```

Kelas HalamanMenu memiliki subclass KirimID yang sama dengan subclass pada HalamanAwal yang bertugas untuk mengirimkan identitas e-ktp yang sudah terbaca ke dalam server secara asinkronus.

C. Kelas Pendaftaran



Gambar 5.3 Implementasi Kelas Pendaftaran

Kelas Pendaftaran berfungsi sebagai aktifitas saat proses pendaftaran anggota melalui perangkat setelah identitas e-KTP pelanggan terbaca belum menjadi anggota rumah makan. Kelas Pendaftaran mempunyai dua fungsi atau *method* antara lain:

1. onCreate

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas Pendaftaran terbentuk. Tugas dari fungsi ini antara lain menampilkan layout yang sesuai dengan aktifitas



Pendaftaran, mengatur fungsi tiap tombol pada layout, dan mengatur nilai awal dari setiap atribut layout.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pendaftaran);
    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

    nama = (EditText) findViewById(R.id.nama);
    alamat = (EditText) findViewById(R.id.alamat);
    kota = (EditText) findViewById(R.id.kota);
    no_telp = (EditText) findViewById(R.id.no_telp);

    nfcid=getIntent().getExtras().getString("nfcid");

    tgl_lahir=(TextView)
findViewById(R.id.tgl_lahir);
    tahun=1990;
    bulan=0;
    hari=1;

    tgl_lahir.setText(new
StringBuilder().append(hari).append("-"
).append(bulan+1).append("-").append(tahun).append("
"));

    gantiTanggal=(Button)
findViewById(R.id.btnChangeDate);
    gantiTanggal.setOnClickListener(new
OnClickListener() {
        @Override
        public void onClick(View arg0) {
            showDialog(DATE_DIALOG_ID);
        }
    });

    simpan=(Button) findViewById(R.id.simpan);
    simpan.setOnClickListener(new OnClickListener()
{
        @Override
        public void onClick(View arg0) {
            new KirimBiodata().execute(
                nfcid,

                nama.getText().toString(),
                tgl_lahir.getText().toString(),
                alamat.getText().toString(),
                kota.getText().toString(),
                no_telp.getText().toString()
            );
        }
    });
}

```

```

        finish();
    }
});
}

```

2. onCreateDialog

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan sebelum aktifitas Pendaftaran menampilkan *dialog box*. Tugas fungsi untuk mengatur *dialog box* mana yang akan ditampilkan sesuai identitas *dialog* yang sudah ditentukan.

```

protected Dialog onCreateDialog(int id) {
    switch(id) {
        case DATE_DIALOG_ID : return new
DatePickerDialog(this, datePickerListener, tahun, bulan,
hari);
    }
    return null;
}

```

Kelas Pendaftaran memiliki subclass KirimBiodata yang bertugas untuk mengirimkan biodata pendaftaran untuk menjadi anggota secara asinkronus ke dalam server rumah makan.

```

private class KirimBiodata extends AsyncTask<String, Void, String>
{
    @Override
    protected String doInBackground(String... params) {
        try{
            String ipaddr="10.10.10.1/rumahmakan";
            if (sharedpreferences.contains("AlamatServer")){
                ipaddr=sharedpreferences.getString("AlamatServer", "");
            }
            String link="http://"+ipaddr+"/proses.php";
            String data = URLEncoder.encode("aksi", "UTF-8")
+ "=" + URLEncoder.encode("daftar", "UTF-8");
            data += "&" + URLEncoder.encode("nfcId", "UTF-8")
+ "=" + URLEncoder.encode(params[0], "UTF-8");
            data += "&" + URLEncoder.encode("nama", "UTF-8")
+ "=" + URLEncoder.encode(params[1], "UTF-8");
            data += "&" + URLEncoder.encode("tgl_lahir",
"UTF-8") + "=" + URLEncoder.encode(params[2], "UTF-8");
            data += "&" + URLEncoder.encode("alamat", "UTF-
8") + "=" + URLEncoder.encode(params[3], "UTF-8");
            data += "&" + URLEncoder.encode("kota", "UTF-8")
+ "=" + URLEncoder.encode(params[4], "UTF-8");
            data += "&" + URLEncoder.encode("no_telp", "UTF-

```

```

8") + "=" + URLEncoder.encode(params[5], "UTF-8");
    URL url = new URL(link);
    URLConnection conn = url.openConnection();
    conn.setDoOutput(true);
    OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
    wr.write(data);
    wr.flush();
    BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    StringBuilder sb = new StringBuilder();
    String line = null;
    while((line = reader.readLine()) != null)
    {
        sb.append(line);
        break;
    }
    return sb.toString();
} catch (Exception e) {
    return new String("Koneksi Ke Server Gagal");
}
}

@Override
protected void onPostExecute(String result) {
    Toast.makeText(Pendaftaran.this, result,
Toast.LENGTH_LONG).show();
}
}

```

D. Kelas Pengaturan



Gambar 5.4 Implementasi Kelas Pengaturan

Kelas Pengaturan berfungsi sebagai aktifitas saat mengatur alamat server rumah makan. Kelas Pengaturan hanya mempunyai satu fungsi atau *method*, yaitu *onCreate* yang berupa *abstract method* dari kelas abstrak *Activity*, bertugas untuk menampilkan layout dan mengatur fungsi tiap tombol pada layout.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

```

```

setContentView(R.layout.pengaturan);

sharedpreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

alamat=(EditText) findViewById(R.id.alamatserver);

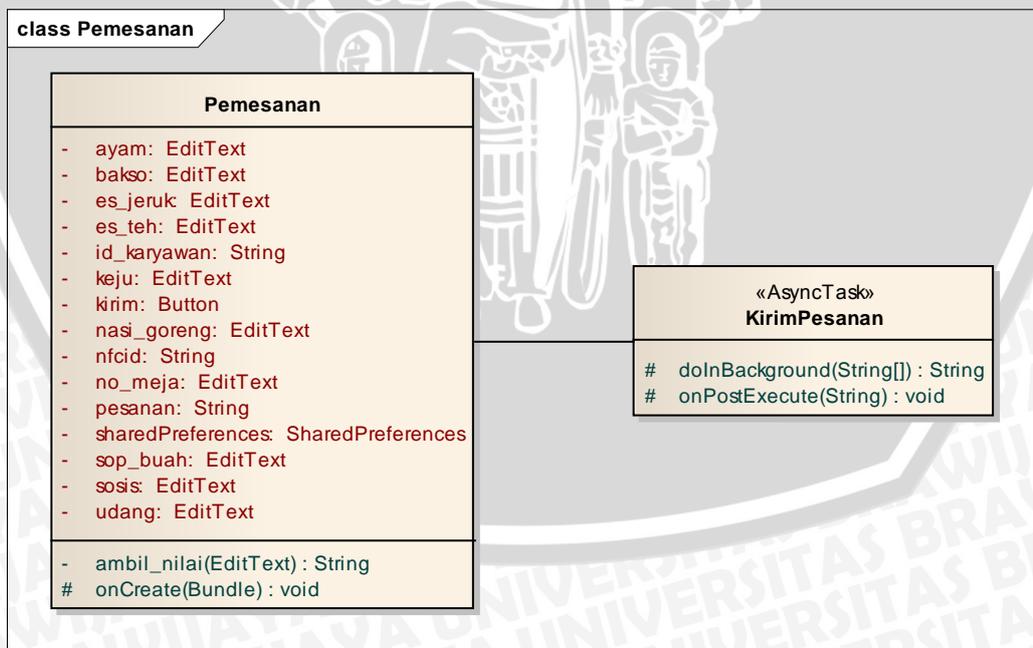
if (sharedpreferences.contains("AlamatServer"))
alamat.setText(sharedpreferences.getString("AlamatServer", ""));
else alamat.setText("10.10.10.1/rumahmakan");

simpan=(Button) findViewById(R.id.pengaturan_simpan);
simpan.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View arg0) {
    Editor editor = sharedpreferences.edit();
    editor.putString("AlamatServer",
alamat.getText().toString());
    editor.commit();
    Toast.makeText(Pengaturan.this, "Alamat Server
Berhasil Diubah", Toast.LENGTH_LONG).show();
    finish();
}
});
}

```

E. Kelas Pemesanan



Gambar 5.5 Implementasi Kelas Pemesanan



Kelas Pemesanan berfungsi sebagai aktifitas saat karyawan mencatat pesanan pelanggan dan mengirim pesanan ke dalam server. Kelas Pemesanan terdiri dari dua fungsi, yaitu :

1. ambil_nilai

Fungsi ini digunakan saat mengambil nilai dari jumlahnya menu yang dipesan di setiap pilihan menu yang disediakan.

```
private String ambil_nilai(EditText data){
    String hasil=data.getText().toString();
    if(hasil.equals("")) hasil="0";
    return hasil;
}
```

2. onCreate

Fungsi ini adalah fungsi bawaan atau *abstract method* dari kelas abstrak Activity yang dijalankan apabila aktifitas Pemesanan terbentuk. Tugas dari fungsi ini antara lain menampilkan layout yang sesuai dengan aktifitas Pemesanan, mengatur fungsi tiap tombol pada layout, dan mengatur nilai awal dari setiap atribut layout.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pemesanan);

    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);
    id_karyawan =
sharedPreferences.getString("Id_karyawan", "");

    nfcid=getIntent().getExtras().getString("nfcid");

    no_meja=(EditText) findViewById(R.id.pesan_no_meja);

    nasi_goreng=(EditText) findViewById(R.id.pesan_101);
    keju=(EditText) findViewById(R.id.pesan_201);
    bakso=(EditText) findViewById(R.id.pesan_202);
    udang=(EditText) findViewById(R.id.pesan_203);
    ayam=(EditText) findViewById(R.id.pesan_204);
    sosis=(EditText) findViewById(R.id.pesan_205);
    es_teh=(EditText) findViewById(R.id.pesan_301);
    es_jeruk=(EditText) findViewById(R.id.pesan_302);
    sop_buah=(EditText) findViewById(R.id.pesan_303);

    kirim=(Button) findViewById(R.id.pesan_kirim);
    kirim.setOnClickListener(new
View.OnClickListener() {
```

```

@Override
    public void onClick(View arg0) {
        pesanan="101,"+ambil_nilai(nasi_goreng)+"/"+
        "201,"+ambil_nilai(keju)+"/"+
        "202,"+ambil_nilai(bakso)+"/"+
        "203,"+ambil_nilai(udang)+"/"+
        "204,"+ambil_nilai(ayam)+"/"+
        "205,"+ambil_nilai(sosis)+"/"+
        "301,"+ambil_nilai(es_teh)+"/"+
        "302,"+ambil_nilai(es_jeruk)+"/"+
        "303,"+ambil_nilai(sop_buah);
        new
        KirimPesanan().execute(nfcid,id_karyawan,no_meja.getText().t
        oString(),pesanan);
    }
}

```

Kelas Pemesanan memiliki subclass KirimPesanan yang bertugas untuk mengirimkan menu pesanan pelanggan secara asinkronus ke dalam server rumah makan.

```

private class KirimPesanan extends AsyncTask<String, Void, String>
{
    @Override
    protected String doInBackground(String... params) {
        try{
            String ipaddr="10.10.10.1/rumahmakan";
            if (sharedpreferences.contains("AlamatServer")){
                ipaddr=sharedpreferences.getString("AlamatServer", "");
            }
            String link="http://"+ipaddr+"/proses.php";
            String data = URLEncoder.encode("aksi", "UTF-8")
+ "=" + URLEncoder.encode("pesan", "UTF-8");
            data += "&" + URLEncoder.encode("nfcid", "UTF-8")
+ "=" + URLEncoder.encode(params[0], "UTF-8");
            data += "&" + URLEncoder.encode("id_karyawan",
"UTF-8") + "=" + URLEncoder.encode(params[1], "UTF-8");
            data += "&" + URLEncoder.encode("no_meja", "UTF-
8") + "=" + URLEncoder.encode(params[2], "UTF-8");
            data += "&" + URLEncoder.encode("pesanan", "UTF-
8") + "=" + URLEncoder.encode(params[3], "UTF-8");
            URL url = new URL(link);
            URLConnection conn = url.openConnection();

```

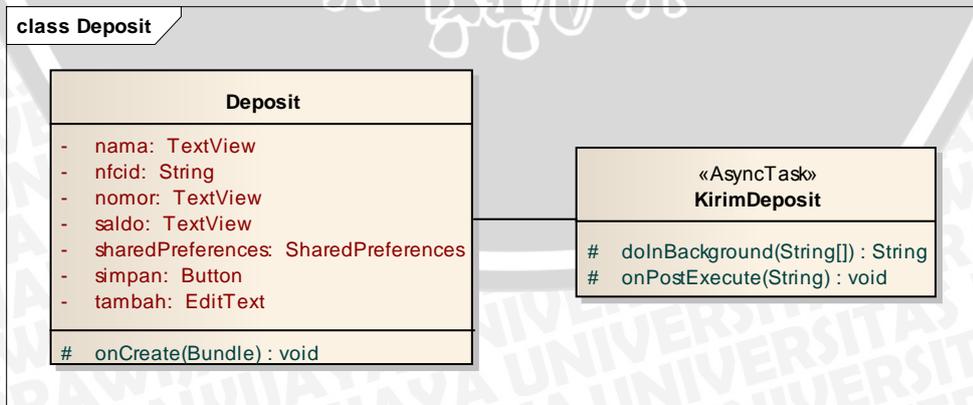
```

        conn.setDoOutput(true);
        OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
        wr.write(data);
        wr.flush();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        StringBuilder sb = new StringBuilder();
        String line = null;
        while((line = reader.readLine()) != null)
        {
            sb.append(line);
            break;
        }
        return sb.toString();
    } catch (Exception e) {
        return new String("Koneksi Ke Server Gagal");
    }
}

@Override
protected void onPostExecute(String result) {
    if(!result.equals("Koneksi Ke Server Gagal") &&
!result.equals("gagal")){
        Toast.makeText(Pemesanan.this, "Pemesanan
Berhasil", Toast.LENGTH_LONG).show();
        finish();
    }
    else if(result.equals("gagal")){
        Toast.makeText(Pemesanan.this, "Silahkan Cek
Kembali Data Anda", Toast.LENGTH_LONG).show();
    }
    else Toast.makeText(Pemesanan.this, result,
Toast.LENGTH_LONG).show();
}
}

```

F. Kelas Deposit



Gambar 5.6 Implementasi Kelas Deposit



Kelas Deposit berfungsi sebagai aktifitas saat karyawan ingin menambah saldo anggota. Kelas Deposit hanya mempunyai satu fungsi atau *method*, yaitu onCreate yang berupa *abstract method* dari kelas abstrak Activity, bertugas untuk menampilkan layout dan mengatur fungsi tiap tombol pada layout.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.deposit);

    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

    nfcid=getIntent().getExtras().getString("nfcid");

    nomor=(TextView) findViewById(R.id.deposit_nomor);
    nama=(TextView) findViewById(R.id.deposit_nama);
    saldo=(TextView) findViewById(R.id.deposit_saldo);
    tambah=(EditText) findViewById(R.id.deposit_deposit);

    nomor.setText(getIntent().getExtras().getString("nomor"));
    nama.setText(getIntent().getExtras().getString("nama"));
    saldo.setText(getIntent().getExtras().getString("saldo"));

    simpan=(Button) findViewById(R.id.deposit_simpan);
    simpan.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            new
            KirimDeposit().execute(nfcid,tambah.getText().toString());
        }
    });
}
```

Kelas Deposit memiliki subclass KirimDeposit yang bertugas untuk mengirimkan data anggota dan besarnya deposit yang akan ditambahkan ke dalam saldo anggota tersebut secara asinkronus ke dalam server rumah makan.

```
private class KirimDeposit extends AsyncTask<String, Void, String>
{
    @Override
    protected String doInBackground(String... params) {
        try{
            String ipaddr="10.10.10.1/rumahmakan";
            if (sharedpreferences.contains("AlamatServer")){
                ipaddr=sharedpreferences.getString("AlamatServer", "");
            }
        }
    }
}
```

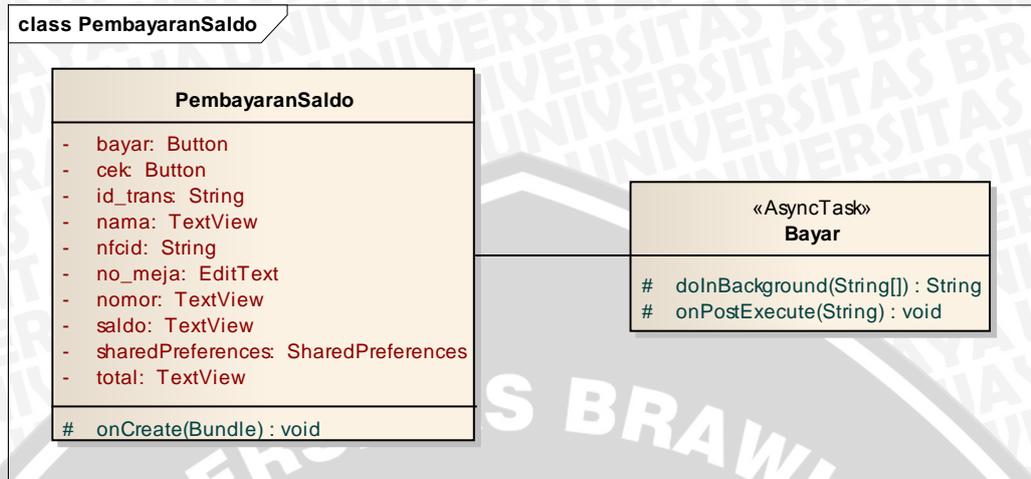
```

        String link="http://" + ipaddr + "/proses.php";
        String data = URLEncoder.encode("aksi", "UTF-8")
+ "=" + URLEncoder.encode("deposit", "UTF-8");
        data += "&" + URLEncoder.encode("nfcid", "UTF-8")
+ "=" + URLEncoder.encode(params[0], "UTF-8");
        data += "&" + URLEncoder.encode("deposit", "UTF-
8") + "=" + URLEncoder.encode(params[1], "UTF-8");
        URL url = new URL(link);
        URLConnection conn = url.openConnection();
        conn.setDoOutput(true);
        OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
        wr.write(data);
        wr.flush();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        StringBuilder sb = new StringBuilder();
        String line = null;
        while((line = reader.readLine()) != null)
        {
            sb.append(line);
            break;
        }
        return sb.toString();
    } catch (Exception e) {
        return new String("Koneksi Ke Server Gagal");
    }
}

@Override
protected void onPostExecute(String result) {
    if(!result.equals("Koneksi Ke Server Gagal") &&
!result.equals("gagal")) {
        Editor editor = sharedPreferences.edit();
        editor.putInt("Saldo",
Integer.parseInt(result));
        editor.commit();
        Toast.makeText(Deposit.this, "Deposit Berhasil",
Toast.LENGTH_LONG).show();
        finish();
    }
    else if(result.equals("gagal")) {
        Toast.makeText(Deposit.this, "Silahkan Cek
Kembali Data Anda", Toast.LENGTH_LONG).show();
    }
    else Toast.makeText(Deposit.this, result,
Toast.LENGTH_LONG).show();
}
}
}

```

G. Kelas PembayaranSaldo



Gambar 5.7 Implementasi Kelas PembayaranSaldo

Kelas `PembayaranSaldo` berfungsi sebagai aktifitas saat karyawan mencatat pembayaran yang dilakukan pelanggan sebagai anggota dan menggunakan saldo anggota sebagai media pembayaran. Kelas `PembayaranSaldo` hanya mempunyai satu fungsi atau *method*, yaitu `onCreate` yang berupa *abstract method* dari kelas abstrak `Activity`, bertugas untuk menampilkan layout dan mengatur fungsi tiap tombol pada layout.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pembayaran_saldo);

    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

    nfcid=getIntent().getExtras().getString("nfcid");

    nomor=(TextView) findViewById(R.id.pemb_saldo_nomor);
    nama=(TextView) findViewById(R.id.pemb_saldo_nama);
    saldo=(TextView) findViewById(R.id.pemb_saldo_saldo);
    total=(TextView) findViewById(R.id.pemb_saldo_total);

    nomor.setText(getIntent().getExtras().getString("nomor"));
    nama.setText(getIntent().getExtras().getString("nama"));
    saldo.setText(getIntent().getExtras().getString("saldo"));

    no_meja=(EditText) findViewById(R.id.pemb_saldo_no_meja);
  
```

```

cek=(Button)findViewById(R.id.pemb_saldo_cek);
cek.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        new
        Bayar().execute("cek_meja",no_meja.getText().toString());
    }
});

bayar=(Button)findViewById(R.id.pemb_saldo_bayar);
bayar.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        int
        bayar_saldo=Integer.parseInt(saldo.getText().toString());
        int
        bayar_total=Integer.parseInt(total.getText().toString());
        if(bayar_saldo>bayar_total)
            new
            Bayar().execute("bayar",nfcid,id_trans,total.getText().toString());
        ;
        else Toast.makeText(PembayaranSaldo.this,
        "Saldo Anda Tidak Mencukupi", Toast.LENGTH_LONG).show();
    }
});
}

```

Kelas PembayaranSaldo memiliki subclass Bayar yang bertugas untuk mengirimkan data nomer meja ke dalam server untuk mengambil total pembayaran transaksi pada nomor meja tersebut. Selain itu subclass Bayar juga bertugas untuk mengirimkan data transaksi ke dalam server untuk melakukan validasi pembayaran pada transaksi tersebut saat tombol bayar ditekan.

```

private class Bayar extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
        try{
            String ipaddr="10.10.10.1/rumahmakan";
            if (sharedpreferences.contains("AlamatServer")){
                ipaddr=sharedpreferences.getString("AlamatServer", "");
            }
            String link="http://" + ipaddr + "/proses.php";
            String data="";
            if(params[0].equals("cek_meja")){
                data = URLEncoder.encode("aksi", "UTF-8") +
                "=" + URLEncoder.encode("cek_meja", "UTF-8");
                data += "&" + URLEncoder.encode("no_meja",
                "UTF-8") + "=" + URLEncoder.encode(params[1], "UTF-8");
            }
        }
    }
}

```

```

        else{
            data = URLEncoder.encode("aksi", "UTF-8")
+ "=" + URLEncoder.encode("bayar", "UTF-8");
            data += "&" + URLEncoder.encode("nfcid",
"UTF-8") + "=" + URLEncoder.encode(params[1], "UTF-8");
            data += "&" + URLEncoder.encode("id_trans",
"UTF-8") + "=" + URLEncoder.encode(params[2], "UTF-8");
            data += "&" + URLEncoder.encode("total",
"UTF-8") + "=" + URLEncoder.encode(params[3], "UTF-8");
        }
        URL url = new URL(link);
        URLConnection conn = url.openConnection();
        conn.setDoOutput(true);
        OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
        wr.write(data);
        wr.flush();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        StringBuilder sb = new StringBuilder();
        String line = null;
        while((line = reader.readLine()) != null)
        {
            sb.append(line);
            break;
        }
        return sb.toString();
    } catch (Exception e) {
        return new String("Koneksi Ke Server Gagal");
    }
}

@Override
protected void onPostExecute(String result) {
    if(!result.equals("Koneksi Ke Server Gagal") &&
!result.equals("gagal")){
        String[] data=result.split("/");
        if(data[0].equals("cek_meja")){
            id_trans=data[1];
            total.setText(data[2]);
        }
        else{
            Editor editor = sharedPreferences.edit();
            editor.putInt("Saldo",
Integer.parseInt(data[1]));
            editor.commit();
            Toast.makeText(PembayaranSaldo.this,
"Pembayaran Berhasil", Toast.LENGTH_LONG).show();
            finish();
        }
    }
    else if(result.equals("gagal")){
        Toast.makeText(PembayaranSaldo.this, "Silahkan
Cek Kembali Data Anda", Toast.LENGTH_LONG).show();
    }
    else Toast.makeText(PembayaranSaldo.this, result,
Toast.LENGTH LONG).show();
}

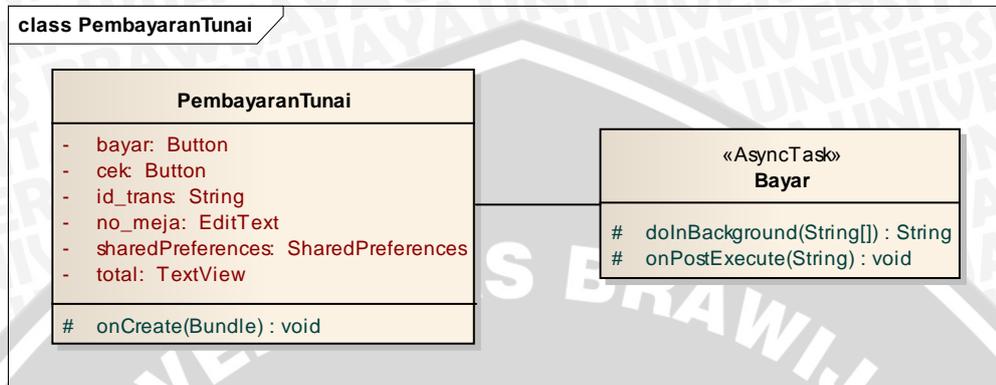
```

```

    }
}

```

H. Kelas PembayaranTunai



Gambar 5.8 Implementasi Kelas PembayaranTunai

Kelas PembayaranTunai berfungsi sebagai aktifitas saat karyawan mencatat pembayaran yang dilakukan pelanggan dengan uang tunai sebagai media pembayaran. Kelas PembayaranTunai hanya mempunyai satu fungsi atau *method*, yaitu onCreate yang berupa *abstract method* dari kelas abstrak Activity, bertugas untuk menampilkan layout dan mengatur fungsi tiap tombol pada layout.

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pembayaran_tunai);

    sharedPreferences = getSharedPreferences("DATA",
Context.MODE_WORLD_WRITEABLE);

    total=(TextView) findViewById(R.id.pemb_tunai_total);

    no_meja=(EditText) findViewById(R.id.pemb_tunai_no_meja);

    cek=(Button) findViewById(R.id.pemb_tunai_cek);
    cek.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            new
            Bayar().execute("cek_meja",no_meja.getText().toString());
        }
    });

    bayar=(Button) findViewById(R.id.pemb_tunai_bayar);
    bayar.setOnClickListener(new View.OnClickListener() {

        @Override
    
```



```

        public void onClick(View arg0) {
            new Bayar().execute("bayar",id_trans);
        }
    });
}

```

Kelas PembayaranTunai memiliki subclass Bayar yang bertugas untuk mengirimkan data nomer meja ke dalam server untuk mengambil total pembayaran transaksi pada nomor meja tersebut. Selain itu subclass Bayar juga bertugas untuk mengirimkan data transaksi ke dalam server untuk melakukan validasi pembayaran pada transaksi tersebut saat tombol bayar ditekan.

```

private class Bayar extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
        try{
            String ipaddr="10.10.10.1/rumahmakan";
            if (sharedpreferences.contains("AlamatServer")){
                ipaddr=sharedpreferences.getString("AlamatServer", "");
            }
            String link="http://"+ipaddr+"/proses.php";
            String data="";
            if(params[0].equals("cek_meja")){
                data = URLEncoder.encode("aksi", "UTF-8") +
                "=" + URLEncoder.encode("cek_meja", "UTF-8");
                data += "&" + URLEncoder.encode("no_meja",
                "UTF-8") + "=" + URLEncoder.encode(params[1], "UTF-8");
            }
            else{
                data = URLEncoder.encode("aksi", "UTF-8")
                + "=" + URLEncoder.encode("bayar", "UTF-8");
                data += "&" + URLEncoder.encode("id_trans",
                "UTF-8") + "=" + URLEncoder.encode(params[1], "UTF-8");
            }
            URL url = new URL(link);
            URLConnection conn = url.openConnection();
            conn.setDoOutput(true);
            OutputStreamWriter wr = new
            OutputStreamWriter(conn.getOutputStream());
            wr.write(data);
            wr.flush();
            BufferedReader reader = new BufferedReader(new
            InputStreamReader(conn.getInputStream()));
            StringBuilder sb = new StringBuilder();
            String line = null;
            while((line = reader.readLine()) != null)
            {
                sb.append(line);
                break;
            }
            return sb.toString();
        }
    }
}

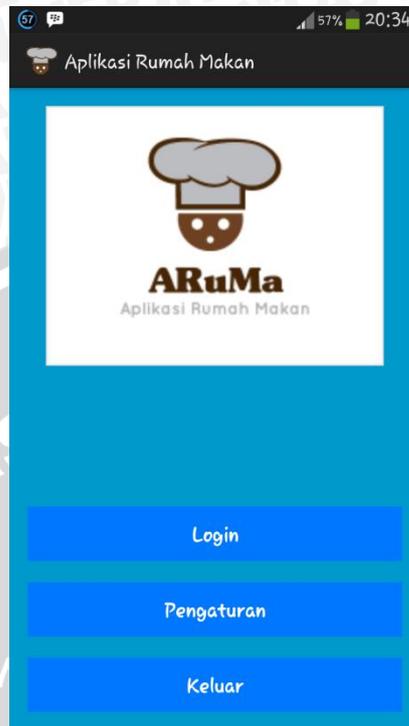
```

```
        } catch (Exception e) {  
            return new String("Koneksi Ke Server Gagal");  
        }  
    }  
  
    @Override  
    protected void onPostExecute(String result) {  
        if (!result.equals("Koneksi Ke Server Gagal") &&  
            !result.equals("gagal")) {  
            String[] data = result.split("/");  
            if (data[0].equals("cek_meja")) {  
                id_trans = data[1];  
                total.setText(data[2]);  
            }  
            else {  
                Toast.makeText(PembayaranTunai.this,  
                    "Pembayaran Berhasil", Toast.LENGTH_LONG).show();  
                finish();  
            }  
        }  
        else if (result.equals("gagal")) {  
            Toast.makeText(PembayaranTunai.this, "Silahkan  
            Cek Kembali Data Anda", Toast.LENGTH_LONG).show();  
        }  
        else Toast.makeText(PembayaranTunai.this, result,  
            Toast.LENGTH_LONG).show();  
    }  
}
```

5.3.3 Implementasi Layout

Layout adalah tampilan antarmuka pengguna pada perangkat Android. Layout berisikan label, kolom isian teks, angka, dan tanggal serta tombol. Layout dibentuk sesuai dengan aktifitas yang ada dalam aplikasi. Layout yang dibangun pada aplikasi ini antara lain layout halaman_awal, halaman_menu, pengaturan, pendaftaran, pemesanan, deposit, pembayaran_saldo, dan pembayaran_tunai.

A. Layout halaman_awal



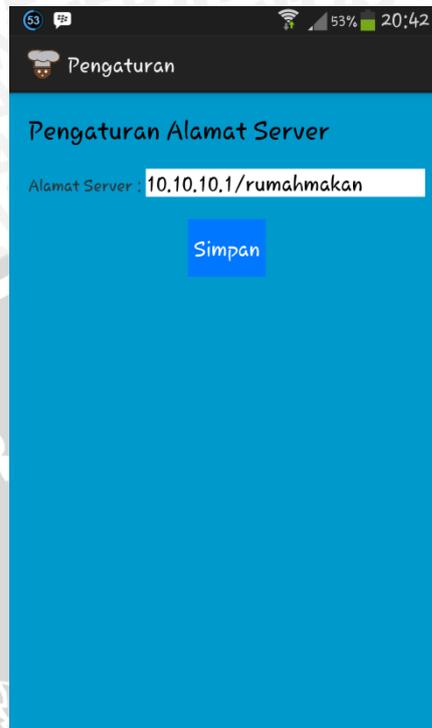
Gambar 5.9 Implementasi Layout halaman_awal

B. Layout halaman_menu



Gambar 5.10 Implementasi Layout halaman_menu

C. Layout pengaturan



Pengaturan

Pengaturan Alamat Server

Alamat Server : 10.10.10.1/rumahmakan

Simpan

Gambar 5.11 Implementasi Layout pengaturan

D. Layout pendaftaran



Biodata Pendaftaran

Biodata Anggota

Nama :

Tanggal Lahir : 1-1-1990 [Ubah](#)

Alamat :

Kota :

No. Telepon :

Simpan

Gambar 5.12 Implementasi Layout pendaftaran

E. Layout pemesanan



Gambar 5.13 Implementasi Layout pemesanan

F. Layout deposit



Gambar 5.14 Implementasi Layout deposit

G. Layout pembayaran_saldo

Pembayaran Saldo

No. Anggota : 6
Nama : fitri
Saldo : Rp.46900

No. Meja : 10 Cek
Total Harga : Rp.26550

Bayar

Gambar 5.15 Implementasi Layout pembayaran_saldo

H. Layout pembayaran_tunai

Pembayaran Tunai

No. Meja : 10 Cek
Total Harga : Rp.26550

Bayar

Gambar 5.16 Implementasi Layout pembayaran_tunai

5.3 Implementasi Server

Implementasi server dilakukan untuk membangun aplikasi server pada perangkat PC berupa aplikasi web. Implementasi dibagi menjadi tiga bagian, yaitu implementasi basis data, implementasi kelas, dan implementasi antarmuka. Implementasi basis data merupakan pembangunan tabel berdasarkan ERD dan PDM yang telah dirancang pada bab perancangan. Implementasi kelas merupakan realisasi terhadap perancangan kelas yang sudah dibuat pada bab perancangan.

5.3.1 Implementasi Basis Data

Pada aplikasi server terbentuk 5 tabel berdasarkan tahap perancangan, antara lain tabel anggota, detail transaksi, karyawan, menu, dan transaksi. Tiap tabel menyimpan data berdasarkan tipe data dan peran masing-masing.

A. Tabel Anggota

Tabel Anggota berperan untuk menyimpan data anggota rumah makan. Data yang disimpan antara lain id sebagai nomor anggota, idNFC sebagai identitas e-KTP, nama, alamat, kota, tanggal lahir, nomor telepon, saldo anggota, dan status validasi pendaftaran. Bentuk implementasi tabel Anggota dapat dilihat pada Tabel 5.4.

Tabel 5.4 Implementasi Tabel Anggota

No.	Nama Field	Key	Type	Deskripsi
1	id	PK	int	AUTO_INCREMENT
2	idNFC		varchar	NOT_NULL
3	nama		varchar	NOT_NULL
4	alamat		varchar	NOT_NULL
5	kota		varchar	NOT_NULL
6	tanggal_lahir		varchar	NOT_NULL
7	nomor_telepon		varchar	NOT_NULL
8	saldo		int	0
9	status_validasi		int	0

B. Tabel Detail Transaksi

Tabel Detail Transaksi berperan untuk menyimpan menu yang dipesan pada transaksi dan jumlahnya berdasarkan transaksi tertentu. Data yang disimpan antara lain id, id transaksi, id menu sebagai nomor menu yang dipesan, dan jumlah menu yang dipesan. Bentuk implementasi tabel Detail Transaksi dapat dilihat pada Tabel 5.5.

Tabel 5.5 Implementasi Tabel Detail Transaksi

No.	Nama Field	Key	Type	Deskripsi
1	id	PK	int	AUTO_INCREMENT
2	id_transaksi	FK	int	NOT_NULL
3	id_menu	FK	int	NOT_NULL
4	jumlah		int	NOT_NULL

C. Tabel Karyawan

Tabel Karyawan berperan menyimpan data karyawan atau pegawai kasir untuk mencatat transaksi dan mengelola anggota ataupun menu. Data yang disimpan antara lain id sebagai nomor karyawan, idNFC sebagai identitas e-KTP, nama, alamat, tanggal lahir, dan nomor telepon. Bentuk implementasi tabel Karyawan dapat dilihat pada Tabel 5.6.

Tabel 5.6 Implementasi Tabel Karyawan

No.	Nama Field	Key	Type	Deskripsi
1	id	PK	int	AUTO_INCREMENT
2	idNFC		varchar	NOT_NULL
3	nama		varchar	NOT_NULL
4	alamat		varchar	NOT_NULL
5	tanggal_lahir		varchar	NOT_NULL
6	nomor_telepon		varchar	NOT_NULL

D. Tabel Menu

Tabel Menu berperan untuk menyimpan menu makanan dan minuman yang dikelola oleh karyawan. Data yang disimpan antara lain id sebagai nomor menu, nama menu, harga, dan diskon sebagai diskon yang ditawarkan rumah makan pada anggota. Bentuk implementasi tabel Menu dapat dilihat pada Tabel 5.7.

Tabel 5.7 Implementasi Tabel Menu

No.	Nama Field	Key	Type	Deskripsi
1	id	PK	int	NOT_NULL
2	nama_menu		varchar	NOT_NULL
3	harga		int	NOT_NULL
4	diskon		int	NOT_NULL

E. Tabel Transaksi

Tabel Transaksi berperan untuk menyimpan data transaksi berdasarkan waktu. Data yang disimpan antara lain id, id karyawan yang mencatat transaksi, id anggota bila pembayaran sebagai anggota, tanggal transaksi, dan total harga. Bentuk implementasi tabel Detail Transaksi dapat dilihat pada Tabel 5.8.

Tabel 5.8 Implementasi Tabel Transaksi

No.	Nama Field	Key	Type	Deskripsi
1	id	PK	int	AUTO_INCREMENT
2	id_karyawan	FK	int	NOT_NULL
3	id_anggota	FK	int	0
4	no_meja		int	NOT_NULL
5	tanggal		varchar	NOT_NULL
6	total		int	NOT_NULL
7	status_pembayaran		int	0

5.3.2 Implementasi Kelas



Gambar 5.17 Implementasi Kelas Server

Implementasi kelas pada server hanya terdapat satu kelas yaitu kelas Server yang berfungsi sebagai penghubung antara aplikasi client maupun server dengan basis data sistem. Diagram implementasi kelas server ditunjukkan dalam Gambar 5.17. Kelas Server memiliki 14 fungsi, antara lain :

1. `__construct`

Merupakan konstruktor dari kelas Server yang menghubungkan kelas ke dalam *database*.

```

function __construct() {
    mysql_connect('localhost','root','');
    mysql_select_db('rumah_makan');
}

```

2. `bayar`

Fungsi ini bertugas untuk melakukan validasi pembayaran serta pengurangan saldo terhadap anggota apabila membayar menggunakan saldo.

```

function bayar($nfcid,$sid_trans,$total) {
    $hasil="bayar/";
    if($nfcid!=0) {

```

```

        $saldo=mysql_fetch_row(mysql_query("select saldo from
anggota where idNFC='$nfcid'"));
        $saldo=$saldo[0]-$total;
        $hasil.=$saldo;
        if(!mysql_query("update anggota set
saldo='$saldo' where idNFC='$nfcid'")) $hasil="gagal";
    }
    if(!mysql_query("update transaksi set
status_pembayaran='1' where id='$id_trans'"))
$hasil="gagal";
    return $hasil;
}

```

3. cek_anggota

Fungsi ini bertugas untuk mengecek keanggotaan dari identitas e-ktip yang diterima dari perangkat client.

```

function cek_anggota($nfcid){
    mysql_query("TRUNCATE TABLE baca");
    mysql_query("insert into baca values('$nfcid')");
    $hasil="bukan";
    $query=mysql_fetch_row(mysql_query("select * from
anggota where idNFC='$nfcid' and status_validasi=1"));
    if($query!="")
$hasil=$query[0]."/". $query[2]."/". $query[7];
    return $hasil;
}

```

4. cek_meja

Fungsi ini bertugas untuk mengecek data transaksi yang dilakukan berdasarkan nomor meja pelanggan.

```

function cek_meja($no_meja){
    $hasil="cek_meja/";
    $data=mysql_fetch_row(mysql_query("select
id,total from transaksi where no_meja='$no_meja' and
status_pembayaran='0'"));
}

```

```

$id=$data[0];
$total=$data[1];
if($id=="")$hasil="gagal";
else $hasil.=$id."/".$total;
return $hasil;
}

```

5. daftar

Fungsi ini bertugas untuk memasukkan biodata pelanggan ke *database* untuk proses validasi pendaftaran anggota.

```

function
daftar($nfcid,$nama,$alamat,$kota,$stgl_lahir,$no_telp){
    $hasil="Pendaftaran Gagal, Silahkan Cek Kembali Data
Anda";
    $query="insert into anggota values
('','$nfcid','$nama','$alamat','$kota','$stgl_lahir','$no_tel
p',',',',')";
    if(mysql_query($query)) $hasil="Terima Kasih Telah
Mendaftar Sebagai Anggota Kami";
    return $hasil;
}

```

6. data_anggota

Fungsi ini bertugas untuk mengambil data anggota sehingga dapat ditampilkan pada tampilan antarmuka pengguna pada aplikasi server.

```

function data_anggota($id_anggota){
    $hasil="";
    if($id_anggota=="")
        $query=mysql_query("select * from anggota order
by status_validasi ASC");
    else
        $query=mysql_query("select * from anggota where
id='$id_anggota'");
    while(($data=mysql_fetch_row($query))!=null){
        $hasil.=$data[0]."|".$data[2]."|".$data[3]."|".$
data[4]."|".$data[5]."|".$data[6]."|".$data[7]."|".$data[8].

```

```
"#";
}
if($hasil!="")
    $hasil=substr($hasil,0,-1);
return $hasil;
}
```

7. data_transaksi

Fungsi ini bertugas untuk mengambil data transaksi sehingga dapat ditampilkan pada tampilan antarmuka pengguna pada aplikasi server.

```
function data_transaksi(){
    $hasil="";
    $query=mysql_query("select * from transaksi where
status_pembayaran='0'");
    while(($data=mysql_fetch_row($query))!=null){
        $hasil.=$data[0]."|".$data[3]."|".$data[5]."#";
    }
    if($hasil!="")
        $hasil=substr($hasil,0,-1);
    return $hasil;
}
```

8. deposit

Fungsi ini bertugas untuk mengubah saldo anggota saat melakukan deposit atau penambahan saldo oleh karyawan.

```
function deposit($nfcid,$deposit){
    $hasil="gagal";
    $saldo=mysql_fetch_row(mysql_query("select saldo from
anggota where idNFC='$nfcid'"));
    $saldo=$saldo[0]+$deposit;

    if(mysql_query("update anggota set saldo='$saldo'
where idNFC='$nfcid'")) $hasil=$saldo;
    return $hasil;
}
```

9. hapus_anggota

Fungsi ini bertugas untuk menghapus anggota berdasarkan nomor anggota dari tabel anggota.

```
function hapus_anggota($id) {  
    mysql_query("delete from anggota where id='$id'");  
}
```

10. login

Fungsi ini bertugas untuk mengenali identitas e-ktip karyawan saat melakukan proses login pada aplikasi client.

```
function login($nfcid) {  
    $hasil="bukan";  
    $query=mysql_fetch_row(mysql_query("select * from  
karyawan where idNFC='$nfcid'"));  
    if($query!="") $hasil=$query[0]."/".$query[2];  
    return $hasil;  
}
```

11. pesan

Fungsi ini bertugas untuk mencatat pesanan pelanggan pada aplikasi client dan memasukkannya ke dalam data transaksi. Fungsi ini juga mengatur besarnya total pembayaran berdasarkan harga menu, jumlah, dan status keanggotaan pelanggan.

```
function pesan($nfcid,$id_karyawan,$no_meja,$pesanan) {  
    $hasil="sukses";  
    $id_anggota=0;  
    if($nfcid!="") {  
        $id_anggota=mysql_fetch_row(mysql_query("select id  
from anggota where idNFC='$nfcid'"));  
        $id_anggota=$id_anggota[0];  
    }  
    $tanggal=Date("d-M-Y");  
    $total=0;  
    $pesanan=explode('/', $pesanan);  
    foreach($pesanan as $menu) {
```

```

$menu=explode(',',$menu);
$id_menu=$menu[0];
$jumlah=$menu[1];
if($jumlah!=0){
    $data=mysql_fetch_row(mysql_query("select
harga,diskon from menu where id='$id_menu'"));
    if($nfcid!="") $total+=$data[0]*(1-
$data[1]/100)*$jumlah;
    else $total+=$data[0]*$jumlah;
}
}
if($total==0)die();
if(!mysql_query("insert into transaksi values
('','$id_karyawan','$id_anggota','$no_meja','$tanggal','$total',
'')"))$hasil="gagal";
$id_transaksi=mysql_fetch_row(mysql_query("select id
from transaksi where no_meja='$no_meja' order by id DESC"));
$id_transaksi=$id_transaksi[0];

foreach($pesanan as $menu){
    $menu=explode(',',$menu);
    $id_menu=$menu[0];
    $jumlah=$menu[1];
    if($jumlah!=0)
        if(!mysql_query("insert into
detail_transaksi values
('','$id_transaksi','$id_menu','$jumlah')")) $hasil="gagal";
}

return $hasil;
}

```

12. ubah_anggota

Fungsi ini bertugas untuk mengubah data anggota pada tabel anggota yang dilakukan oleh aplikasi server.

```

function
ubah_anggota($id,$nfcid,$nama,$salamat,$kota,$tgl,$nomor){

```

```
if($nfcid==0)
    mysql_query("update anggota set
nama='$nama',alamat='$alamat',kota='$kota',tanggal_lahir='$t
gl',nomor_telepon='$nomor' where id='$id'");
else
    mysql_query("update anggota set
idNFC='$nfcid',nama='$nama',alamat='$alamat',kota='$kota',ta
nggal_lahir='$tgl',nomor_telepon='$nomor' where id='$id'");
}
```

13. ubah_ektp

Fungsi ini bertugas untuk mengambil data e-ktip baru yang telah dibaca oleh perangkat client saat anggota ingin mengubah identitas e-ktip-nya.

```
function ubah_ektp(){
    $nfcid=mysql_fetch_row(mysql_query("select * from
baca"));
    if($nfcid!="") $nfcid=$nfcid[0];
    else $nfcid=0;
    return $nfcid;
}
```

14. validasi

Fungsi ini bertugas untuk melakukan validasi data anggota baru pada aplikasi server.

```
function validasi($id_anggota){
    mysql_query("update anggota set status_validasi='1'
where id='$id_anggota'");
}
```

BAB VI

PENGUJIAN DAN ANALISIS HASIL PENGUJIAN

Bab ini menjelaskan mengenai pengujian dan analisis hasil pengujian pada aplikasi yang sudah diimplementasikan. Pengujian dilakukan menggunakan uji validasi untuk menguji fungsionalitas dari aplikasi yang telah diimplementasikan pada perangkat client dan server dan pengujian identitas e-KTP dengan NFC Reader.

6.1 Pengujian Validasi

Pengujian validasi merupakan salah satu metode pengujian *blackbox*, yaitu pengujian tanpa memperhatikan alur jalannya algoritma yang berjalan pada suatu proses. Pengujian validasi dilakukan dengan membuat kasus uji pada tiap fungsi atau fitur. Pengujian validasi dibagi menjadi dua berdasarkan perangkat implementasi, yaitu pengujian validasi aplikasi client dan aplikasi server.

6.1.1 Pengujian Validasi Aplikasi Client

Berikut ini adalah skenario atau kasus uji pada aplikasi client :

A. Kasus Uji Mengecek Keanggotaan Pelanggan

Tabel 6.1 Kasus Uji pada Fitur Mengecek Keanggotaan Pelanggan

Nama Kasus Uji	Kasus Uji Mengecek Keanggotaan Pelanggan
Objek Uji	Kebutuhan Fungsional (FR_001)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengecek keanggotaan pelanggan pada aplikasi client saat karyawan menempelkan e-KTP pelanggan pada perangkat dan aplikasi menampilkan status pelanggan, sebagai anggota ataupun belum menjadi anggota
Prosedur Uji	1. Pengguna menempelkan e-KTP anggota rumah makan pada perangkat Android

	2. Pengguna menekan tombol cek keanggotaan
Hasil yang Diharapkan	Aplikasi dapat menampilkan status bahwa identitas e-ktip telah terdaftar sebagai anggota

B. Kasus Uji Mendaftarkan Pelanggan Menjadi Anggota

Tabel 6.2 Kasus Uji pada Fitur Mendaftarkan Pelanggan Menjadi Anggota

Nama Kasus Uji	Kasus Uji Mendaftarkan Pelanggan Menjadi Anggota
Objek Uji	Kebutuhan Fungsional (FR_002)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mendaftar pelanggan menjadi anggota dan memastikan data yang dikirim dari aplikasi client sama dengan data yang diterima oleh server
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menempelkan e-ktip anggota 2. Pengguna menekan tombol cek keanggotaan 3. Pengguna menekan tombol daftar 4. Pengguna memasukkan biodata dengan rincian sebagai berikut: Nama : Randyka Tgl. Lahir : 23-10-1990 Alamat : Griya Shanta Kota : Malang No.Telp : 034167676 5. Pengguna menekan tombol simpan
Hasil yang Diharapkan	Aplikasi dapat mengirimkan biodata yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat

	<p>data anggota baru dengan rincian sebagai berikut :</p> <p>Nama : Randyka</p> <p>Tgl. Lahir : 23-10-1990</p> <p>Alamat : Griya Shanta</p> <p>Kota : Malang</p> <p>No.Telp : 034167676</p>
--	---

C. Kasus Uji Mengatur Alamat Server

Tabel 6.3 Kasus Uji pada Fitur Mengatur Alamat Server

Nama Kasus Uji	Kasus Uji Mengatur Alamat Server
Objek Uji	Kebutuhan Fungsional (FR_003)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengatur alamat server dengan aplikasi dapat menyimpan alamat server pada memori perangkat
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol pengaturan 2. Pengguna mengisi alamat server sebagai berikut: 10.10.10.2/rumahmakan 3. Pengguna menekan tombol simpan
Hasil yang Diharapkan	Aplikasi dapat menampilkan alamat server yang sudah tersimpan, yaitu 10.10.10.2/rumahmakan saat pengguna menekan tombol pengaturan

D. Kasus Uji Mencatat Pesanan Pelanggan

Tabel 6.4 Kasus Uji pada Fitur Mencatat Pesanan Pelanggan

Nama Kasus Uji	Kasus Uji Mencatat Pesanan Pelanggan
Objek Uji	Kebutuhan Fungsional (FR_004)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan

	<p>aplikasi dapat memenuhi kebutuhan fungsional mencatat pesanan pelanggan dengan mengisi daftar pesanan dan menampilkan transaksi tersebut pada aplikasi server</p>
<p>Prosedur Uji</p>	<ol style="list-style-type: none"> 1. Pengguna menekan tombol pemesanan 2. Pengguna memasukkan nomor meja dan jumlah menu yang dipesan sebagai berikut: <ul style="list-style-type: none"> No. Meja : 10 Nasi Goreng : 1 Topping Keju : 1 Es Teh : 1 3. Pengguna menekan tombol kirim
<p>Hasil yang Diharapkan</p>	<p>Aplikasi dapat mengirimkan pesan pelanggan yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat data transaksi dengan rincian sebagai berikut :</p> <ul style="list-style-type: none"> No. Meja : 10 Nasi Goreng : 1 Topping Keju : 1 Es Teh : 1

E. Kasus Uji Validasi Pembayaran Menggunakan Saldo

Tabel 6.5 Kasus Uji pada Fitur Validasi Pembayaran Menggunakan Saldo

<p>Nama Kasus Uji</p>	<p>Kasus Uji Validasi Pembayaran Menggunakan Saldo</p>
<p>Objek Uji</p>	<p>Kebutuhan Fungsional (FR_005)</p>
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional validasi pembayaran menggunakan saldo yang dilakukan oleh karyawan kepada anggota yang</p>

	ingin menggunakan saldo mereka sebagai media pembayaran
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menempelkan e-ktp anggota 2. Pengguna menekan tombol cek keanggotaan 3. Pengguna menekan tombol pembayaran saldo 4. Pengguna memasukkan nomor meja 10 dan menekan tombol cek sehingga muncul total pembayaran 5. Pengguna menekan tombol bayar
Hasil yang Diharapkan	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server dan saldo anggota berkurang sebanyak total pembayaran

F. Kasus Uji Validasi Pembayaran Tunai

Tabel 6.6 Kasus Uji pada Fitur Validasi Pembayaran Tunai

Nama Kasus Uji	Kasus Uji Validasi Pembayaran Tunai
Objek Uji	Kebutuhan Fungsional (FR_006)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional validasi pembayaran tunai yang dilakukan oleh karyawan kepada pelanggan yang ingin menggunakan uang tunai sebagai media pembayaran
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol pembayaran tunai 2. Pengguna memasukkan nomor meja 10 dan menekan tombol cek sehingga muncul total pembayaran

	3. Pengguna menekan tombol bayar
Hasil yang Diharapkan	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server

G. Kasus Uji Menambah Saldo Anggota

Tabel 6.7 Kasus Uji pada Fitur Menambah Saldo Anggota

Nama Kasus Uji	Kasus Uji Menambah Saldo Anggota
Objek Uji	Kebutuhan Fungsional (FR_007)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional menambah saldo anggota yang dilakukan oleh karyawan kepada anggota yang ingin menambah saldo mereka
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menempelkan e-ktp anggota 2. Pengguna menekan tombol cek keanggotaan 3. Pengguna menekan tombol deposit 4. Pengguna memasukkan jumlah deposit sebesar 10000 5. Pengguna menekan tombol simpan
Hasil yang Diharapkan	Saldo anggota bertambah sebesar 10000

H. Kasus Uji Melakukan Login

Tabel 6.8 Kasus Uji pada Fitur Melakukan Login

Nama Kasus Uji	Kasus Uji Melakukan Login
Objek Uji	Kebutuhan Fungsional (FR_008)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional melakukan login karyawan untuk masuk ke pilihan menu transaksi pada aplikasi client

Prosedur Uji	1. Pengguna menempelkan e-ktp karyawan 2. Pengguna menekan tombol login
Hasil yang Diharapkan	Halaman antarmuka pengguna berpindah menjadi halaman pilihan menu

Hasil pengujian dan validitas fitur dari uji kasus yang telah dijabarkan dapat dilihat pada Tabel 6.9.

Tabel 6.9 Hasil Pengujian Validasi *Client*

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validitas
1	Kasus Uji Mengecek Keanggotaan Pelanggan	Aplikasi dapat menampilkan status bahwa identitas e-ktp telah terdaftar sebagai anggota	Aplikasi dapat menampilkan status bahwa identitas e-ktp telah terdaftar sebagai anggota	Valid
2	Kasus Uji Mendaftarkan Pelanggan Menjadi Anggota	Aplikasi dapat mengirimkan biodata yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat data anggota baru dengan rincian sebagai berikut : Nama : Randyka Tgl. Lahir : 23-10-1990 Alamat : Griya Shanta	Aplikasi dapat mengirimkan biodata yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat data anggota baru dengan rincian sebagai berikut : Nama : Randyka Tgl. Lahir : 23-10-1990 Alamat : Griya Shanta	Valid
3	Kasus Uji Mengatur Alamat Server	Aplikasi dapat menampilkan alamat server yang sudah tersimpan, yaitu	Aplikasi dapat menampilkan alamat server yang sudah tersimpan, yaitu	Valid

		10.10.10.2/rumahmakan saat pengguna menekan tombol pengaturan	10.10.10.2/rumahmakan saat pengguna menekan tombol pengaturan	
4	Kasus Uji Mencatat Pesanan Pelanggan	Aplikasi dapat mengirimkan pesanannya pelanggan yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat data transaksi dengan rincian sebagai berikut : No. Meja : 10 Nasi Goreng : 1 Topping Keju : 1 Es Teh : 1	Aplikasi dapat mengirimkan pesanannya pelanggan yang telah diisi ke dalam server dan pada tampilan antarmuka pengguna aplikasi server terdapat data transaksi dengan rincian sebagai berikut : No. Meja : 10 Nasi Goreng : 1 Topping Keju : 1 Es Teh : 1	Valid
5	Kasus Uji Validasi Pembayaran Menggunakan Saldo	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server dan saldo anggota berkurang sebanyak total pembayaran	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server dan saldo anggota berkurang sebanyak total pembayaran	Valid
6	Kasus Uji Validasi Pembayaran Tunai	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server	Status pembayaran pada transaksi nomor meja 10 berubah dari belum bayar menjadi sudah bayar pada aplikasi server	Valid

7	Kasus Uji Menambah Saldo Anggota	Saldo anggota bertambah sebesar 10000	Saldo anggota bertambah sebesar 10000	Valid
8	Kasus Uji Melakukan Login	Halaman antarmuka pengguna berpindah menjadi halaman pilihan menu	Halaman antarmuka pengguna berpindah menjadi halaman pilihan menu	Valid

6.1.2 Pengujian Validasi Aplikasi Server

Berikut ini adalah skenario atau kasus uji pada aplikasi server :

A. Kasus Uji Mencetak Bukti Pembayaran

Tabel 6.10 Kasus Uji pada Fitur Mencetak Bukti Pembayaran

Nama Kasus Uji	Kasus Uji Mencetak Bukti Pembayaran
Objek Uji	Kebutuhan Fungsional (FR_009)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mencetak bukti pembayaran pada aplikasi server
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih menu data transaksi 2. Pengguna memilih menu cetak pada data transaksi nomor meja 10
Hasil yang Diharapkan	Aplikasi dapat menampilkan halaman bertipe data pdf dengan data transaksi pada nomor meja 10

B. Kasus Uji Validasi Data Anggota Baru

Tabel 6.11 Kasus Uji pada Fitur Validasi Data Anggota Baru

Nama Kasus Uji	Kasus Uji Validasi Data Anggota Baru
Objek Uji	Kebutuhan Fungsional (FR_010)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan

	aplikasi dapat memenuhi kebutuhan fungsional mengelola data anggota pada saat pengguna ingin melakukan validasi data anggota baru
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih menu data anggota 2. Pengguna memilih menu validasi pada data anggota baru dengan nomor anggota 1
Hasil yang Diharapkan	Status validasi anggota baru dengan nomor anggota 1 berubah dari belum tervalidasi menjadi anggota

C. Kasus Uji Mengubah Data Anggota

Tabel 6.12 Kasus Uji pada Fitur Mengubah Data Anggota

Nama Kasus Uji	Kasus Uji Mengubah Data Anggota
Objek Uji	Kebutuhan Fungsional (FR_010)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengelola data anggota pada saat pengguna ingin melakukan perubahan data pada anggota
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memilih menu data anggota 2. Pengguna memilih menu ubah pada data anggota dengan nomor anggota 1 yang akan diubah datanya 3. Pengguna mengisi data baru pada kolom nama dengan “Uji Validasi” 4. Pengguna memilih tombol simpan
Hasil yang Diharapkan	Data anggota dengan nomor anggota 1 akan memiliki nama “Uji Validasi”

D. Kasus Uji Menghapus Data Anggota

Tabel 6.13 Kasus Uji pada Fitur Menghapus Data Anggota

Nama Kasus Uji	Kasus Uji Menghapus Data Anggota
Objek Uji	Kebutuhan Fungsional (FR_010)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional mengelola data anggota pada saat pengguna ingin melakukan menghapus data pada anggota
Prosedur Uji	1. Pengguna memilih menu data anggota 2. Pengguna memilih menu hapus pada data anggota dengan nomor anggota 1
Hasil yang Diharapkan	Data anggota dengan nomor anggota 1 akan terhapus dan tidak ditampilkan pada data anggota

Hasil pengujian dan validitas fitur dari uji kasus yang telah dijabarkan dapat dilihat pada Tabel 6.14.

Tabel 6.14 Hasil Pengujian Validasi *Server*

No	Nama Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validitas
1	Kasus Uji Mencetak Bukti Pembayaran	Aplikasi dapat menampilkan halaman bertipe data pdf dengan data transaksi pada nomor meja 10	Aplikasi dapat menampilkan halaman bertipe data pdf dengan data transaksi pada nomor meja 10	Valid
2	Kasus Uji Validasi Data Anggota Baru	Status validasi anggota baru dengan nomor anggota 1 berubah dari belum	Status validasi anggota baru dengan nomor anggota 1 berubah dari belum	Valid

		tervalidasi menjadi anggota	tervalidasi menjadi anggota	
3	Kasus Uji Mengubah Data Anggota	Data anggota dengan nomor anggota 1 akan memiliki nama "Uji Validasi"	Data anggota dengan nomor anggota 1 akan memiliki nama "Uji Validasi"	Valid
4	Kasus Uji Menghapus Data Anggota	Data anggota dengan nomor anggota 1 akan terhapus dan tidak ditampilkan pada data anggota	Data anggota dengan nomor anggota 1 akan terhapus dan tidak ditampilkan pada data anggota	Valid

6.2 Pengujian Identitas e-KTP dengan NFC Reader

Pengujian identitas e-ktp dengan NFC Reader dilakukan dengan metode *whitebox* yaitu menguji berdasarkan algoritma yang berjalan pada suatu proses. Salah satu proses yang akan diuji yaitu proses konversi kumpulan data berupa byte yang tergabung dalam array saat proses pembacaan identitas tag NFC e-KTP dan diubah menjadi hexadesimal pada fungsi yang terdapat pada aplikasi client, yaitu `byteArrayToHexString`. Pengujian dilakukan untuk membuktikan apakah proses konversi dapat diberlakukan sehingga identitas tag NFC dapat menjadi nilai *unique* (tidak ada yang menyamai) atau dapat menimbulkan nilai kembar. Pengujian dilakukan pada 5 e-KTP berbeda.

Contoh kasus konversi pada salah satu e-KTP :

data byte dalam array = [4, 22, 126, 82, 33, 37, -128]

Tiap byte akan dikonversi menjadi nilai hexadesimal, sehingga nilai identitas menjadi data hexadesimal dalam array=[04, 16, 7E, 52, 21, 25, 80] dengan proses konversi sebagai berikut :

$$16\sqrt{4} = 0 \text{ sisa } 4$$

$$04$$

$16\sqrt{22} = 1 \text{ sisa } 6$	16
$16\sqrt{126} = 7 \text{ sisa } 14$	7E
$16\sqrt{82} = 5 \text{ sisa } 2$	52
$16\sqrt{33} = 2 \text{ sisa } 1$	21
$16\sqrt{37} = 2 \text{ sisa } 5$	25
$16\sqrt{256 - 128} = 8 \text{ sisa } 0$	80

Hasil Konversi = 04167E52212580

Tabel 6.15 Hasil Konversi Data Identitas e-KTP

No	Data Byte	Data Hexadesimal
1	4 22 126 82 33 37 -128	04 16 7E 52 21 25 80
2	4 91 40 70 -54 42 -128	04 5B 28 BA CA 2A 80
3	4 21 57 102 94 42 -128	04 15 39 9A 5E 2A 80
4	4 35 37 58 118 43 -128	04 23 25 3A 76 2B 80
5	4 94 89 50 -52 40 -128	04 5E 59 32 CC 28 80

6.3 Analisis Hasil Pengujian

Pengujian validasi pada perangkat client memberikan hasil valid berjumlah 8 dan tidak adanya hasil tidak valid. Membuktikan bahwa implementasi pada perangkat client dapat memenuhi semua kebutuhan fungsional yang ada pada tahap perancangan dan layak untuk digunakan dalam melakukan proses transaksi.

Pengujian validasi pada perangkat server memberikan hasil valid berjumlah 4 dan tidak adanya hasil tidak valid. Membuktikan bahwa implementasi pada perangkat server dapat memenuhi semua kebutuhan fungsional yang ada pada tahap perancangan dan layak untuk digunakan dalam melakukan proses transaksi.

Pengujian identitas e-KTP dengan NFC Reader pada perangkat Android memberikan hasil bahwa tidak adanya data yang sama atau kembar antara e-KTP satu dengan lainnya setelah dilakukan proses konversi dari data mentah yang berupa *array* ke dalam bentuk *string*. Membuktikan bahwa identitas tersebut layak digunakan sebagai identitas keanggotaan rumah makan.

BAB VII

PENUTUP

Bab ini berisi kesimpulan dari penelitian dan saran bagi pembaca. Kesimpulan berisi analisis hasil pengujian untuk menjawab rumusan masalah yang telah dijelaskan pada tahap pendahuluan. Saran ditujukan bagi pembaca yang ingin melanjutkan penelitian ke tahap yang lebih baik.

7.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dan telah diuji, maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan analisis hasil pengujian pada uji validasi pada perangkat client, aplikasi rumah makan yang sudah diimplementasikan dapat melakukan proses transaksi pada rumah makan berupa cek keanggotaan, pendaftaran anggota, pemesanan menu makanan dan minuman, pembayaran dengan uang tunai, pembayaran dengan saldo anggota, dan penambahan saldo anggota.
2. Berdasarkan analisis hasil pengujian pada uji validasi pada perangkat server, aplikasi server yang sudah diimplementasikan dapat melakukan proses transaksi pada rumah makan berupa pencetakan bukti pembayaran dan pengelolaan data anggota.
3. Berdasarkan analisis hasil pengujian identitas e-KTP dengan NFC Reader, e-KTP, khususnya e-KTP Indonesia, dapat dijadikan kartu identitas pada sistem keanggotaan rumah makan karena tidak adanya identitas yang sama atau kembar antara e-KTP satu dengan yang lainnya.

7.2 Saran

Bagi pembaca, baik perorangan maupun instansi, yang ingin mengembangkan penelitian dapat memperhatikan hal-hal berikut ini:

1. Jika pembaca memiliki hak akses untuk mengambil data yang ada di dalam e-KTP, maka tidak perlu adanya proses pendaftaran bagi pelanggan yang ingin menjadi anggota.
2. Jika pembaca dapat mengambil *digital signature* berupa data sidik jari pemilik e-KTP, maka proses login karyawan dapat ditambahkan dengan proses pengecekan sidik jari untuk mengurangi jumlah kecurangan.
3. Pada konsep *Customer Relationship Management*, keuntungan yang didapat anggota tidak hanya pengurangan harga atau diskon pada menu, tetapi keuntungan yang didapat dapat berupa *event* atau kegiatan, seperti pengundian hadiah.



DAFTAR PUSTAKA

- [ANO – 14] Anonymous. 2014. “*Android Developers*”. <http://developer.android.com/> diakses tanggal 10 Juni 2014 pukul 20.00.
- [KHA – 12] Khatwani, Jaikishan, dkk. 2012. “*Haki-NFC Based Android Application*”. International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), ISSN No. 2248-9738 , Vol-2, Iss-1,2.
- [KUR – 13] Kurniawan, R. I. dan Sujianto. 2013. “*Efektivitas Implementasi Program e-KTP*”, Jurnal Administrasi Pembangunan, Vol. 1, No. 2, Hal. 101-218.
- [PRA –12] Prasetya, Angga, Efri S., dan Sholekan. 2012. “*Implementasi Radio Frequency Identification (Rfid) Sebagai Sistem Inventaris Dan Transaksi Pada Rental Video Menggunakan Visual Basic 6.0*”. Institut Teknologi Telkom : Bandung.
- [SAR – 13] Sarinanto, Mustofa. 2013. “*Teknologi e-KTP*”. <https://plus.google.com/+RizqiDjamaluddin/posts/AUFaS4Zk73v> diakses tanggal 24 Maret 2014 pukul 01.00.
- [SHO – 99] Shoemaker, S., dan Lewis, R. C. 1999. “*Customer loyalty: The future of hospitality marketing*”, Hospitality Management, vol. 18, pp. 345-370.
- [UTO – 13] Utomo, Yunanto. 2013. “*E-KTP, Apa Saja Teknologi di Dalamnya?*”. <http://sains.kompas.com/read/2013/05/15/18315786/EKTP..Apa.Saja.Teknologi.di.Dalamnya> diakses tanggal 10 Juni 2014 pukul 20.00.
- [WIJ – 08] Wijaya, S. dan Thio, S. 2008. “*Implementasi Membership Card dan Pengaruhnya dalam Meningkatkan Loyalitas Pengunjung Restoran di Surabaya*”, Eksekutif, Vol.5, No.1.
- [YAN – 09] Yan, Zhang. 2009. “*RFID and Sensor Networks: Architectures, Protocols, Security, and Integrations*”. CRC Press : Florida.

LAMPIRAN

Nomor Use Case	FR_001
Nama	Mengecek Keanggotaan Pelanggan
Prasyarat Konteks	- Karyawan telah melakukan pemasangan aplikasi pada perangkat Android -Karyawan telah melakukan proses login
Tujuan dalam Konteks	Melakukan pengecekan terhadap e-KTP pelanggan apakah pelanggan sudah menjadi anggota atau belum
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi menampilkan status keanggotaan pelanggan
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan menempelkan e-KTP pelanggan pada perangkat
Alur Utama	1. Karyawan menempelkan e-KTP pelanggan pada perangkat
	2. Sistem membaca identitasn e-KTP pelanggan
	3. Karyawan menekan tombol cek keanggotaan
	4. Sistem mengirimkan identitas e-KTP pelanggan
	5. Sistem menerima status keanggotaan pelanggan
	6. Sistem menampilkan status keanggotaan pelanggan

Nomor Use Case	FR_003
Nama	Mengatur Alamat Server
Prasyarat Konteks	Karyawan telah melakukan pemasangan aplikasi pada perangkat Android
Tujuan dalam Konteks	Mengatur alamat server, berupa alamat IP ataupun alamat web untuk proses mengirim dan menerima data pada perangkat client
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi menyimpan alamat server yang telah ditentukan dan menampilkan pesan simpan alamat server berhasil
Kondisi Akhir Gagal	-
Aktor	Karyawan
Trigger	Karyawan menekan tombol pengaturan
Alur Utama	1. Karyawan menekan tombol pengaturan
	2. Sistem menampilkan halaman pengaturan
	3. Karyawan memasukkan alamat server pada kolom
	4. Karyawan menekan tombol simpan
	5. Sistem menyimpan alamat server pada memori
	6. Sistem menampilkan simpan berhasil

Nomor Use Case	FR_004
Nama	Mencatat Pesanan Pelanggan
Prasyarat Konteks	-Karyawan telah melakukan pemasangan aplikasi pada perangkat Android -Karyawan telah melakukan proses login
Tujuan dalam Konteks	Mencatat pesanan pelanggan dalam bertransaksi
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi mengirimkan pesan ke dalam server dan menampilkan pesan berhasil melakukan pemesanan
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan menekan tombol pemesanan
Alur Utama	<ol style="list-style-type: none"> 1. Karyawan menekan tombol pemesanan 2. Sistem menampilkan halaman pemesanan 3. Karyawan memasukkan nomor meja pelanggan dan menu makanan yang dipesan oleh pelanggan 4. Karyawan menekan tombol kirim 5. Sistem mengirim data transaksi 6. Sistem menampilkan pemesanan berhasil

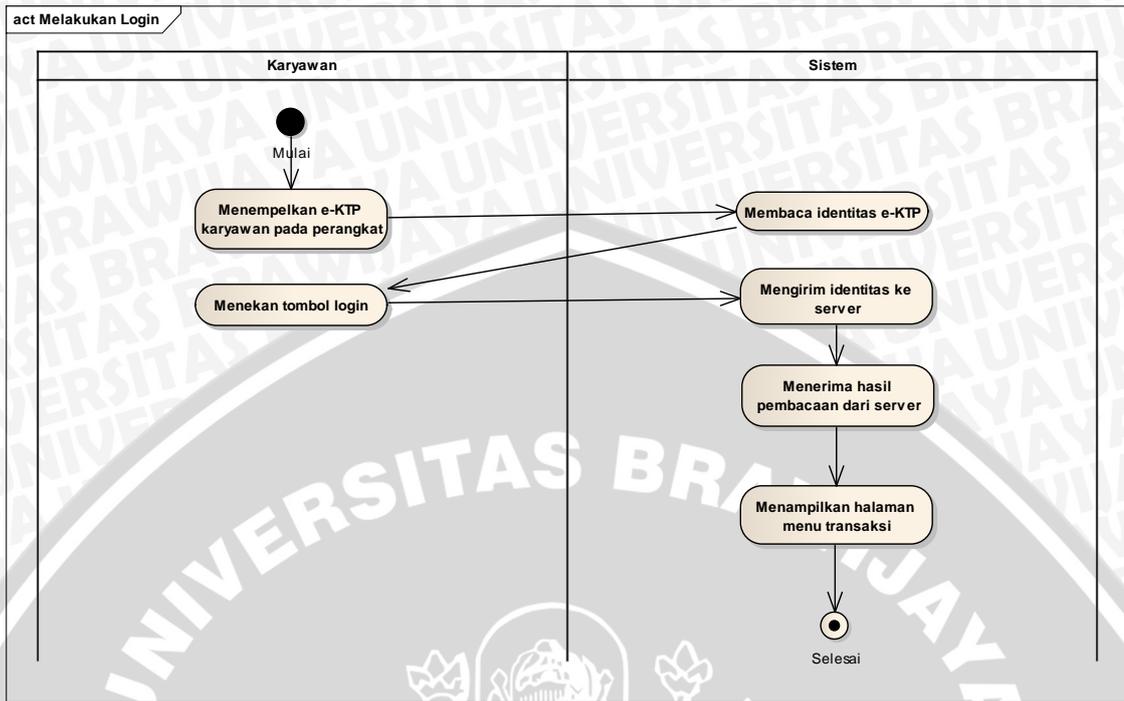
Nomor Use Case	FR_006
Nama	Validasi Pembayaran Tunai
Prasyarat Konteks	-Karyawan telah melakukan pemasangan aplikasi pada perangkat Android -Karyawan telah melakukan proses login
Tujuan dalam Konteks	Melakukan validasi pembayaran terhadap transaksi yang sudah dilakukan
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi mengirimkan data transaksi ke dalam server dan menampilkan pesan berhasil melakukan pembayaran
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan menekan tombol pembayaran tunai
Alur Utama	<ol style="list-style-type: none"> 1. Karyawan memilih tombol pembayaran tunai 2. Sistem menampilkan halaman pembayaran tunai 3. Karyawan memasukkan nomor meja pelanggan dan menekan tombol cek 4. Sistem menampilkan besarnya total pembayaran 5. Karyawan menekan tombol bayar 6. Sistem melakukan validasi pembayaran
Nomor Use Case	FR_007
Nama	Menambah Saldo Anggota

Prasyarat Konteks	- Karyawan telah melakukan pemasangan aplikasi pada perangkat Android - Pelanggan telah terdaftar sebagai anggota
Tujuan dalam Konteks	Melakukan penambahan saldo anggota dengan melakukan transaksi deposit
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi mengirimkan data transaksi ke dalam server dan menampilkan pesan berhasil melakukan deposit
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan menekan tombol deposit
Alur Utama	1. Karyawan memilih tombol pembayaran tunai
	2. Sistem menampilkan halaman deposit
	3. Karyawan memasukkan besarnya deposit
	4. Karyawan menekan tombol simpan
	5. Sistem mengirimkan data anggota dan besarnya deposit ke dalam server
	6. Sistem mengubah jumlah saldo anggota
	7. Sistem menampilkan pesan telah berhasil melakukan penambahan saldo

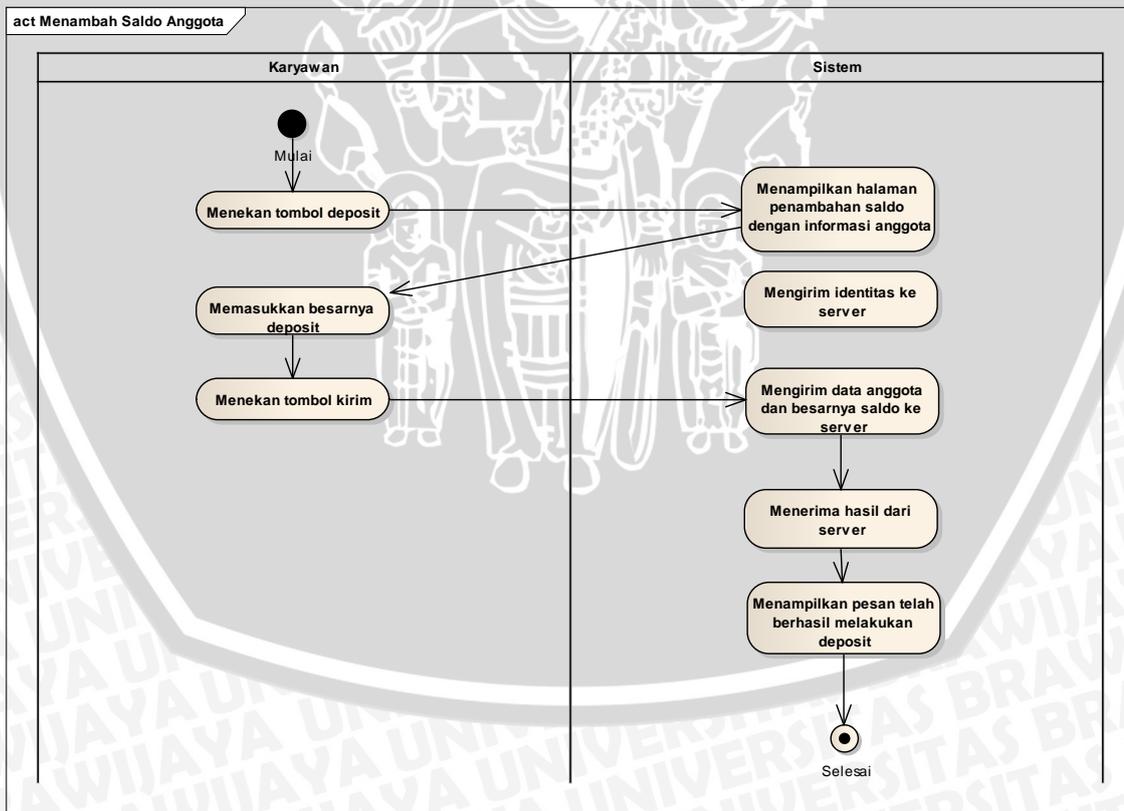
Nomor Use Case	FR_008
Nama	Melakukan Login
Prasyarat Konteks	-Karyawan telah melakukan pemasangan aplikasi pada perangkat Android -e-KTP karyawan sudah terdaftar dalam basis data
Tujuan dalam Konteks	Melakukan proses identifikasi karyawan berdasarkan e-KTP untuk dapat menggunakan menu dalam aplikasi
Prakondisi	Aplikasi telah terpasang pada perangkat client
Kondisi Akhir Sukses	Aplikasi menampilkan halaman pilihan menu
Kondisi Akhir Gagal	Karyawan menerima peringatan apabila tidak dapat terkoneksi dengan server
Aktor	Karyawan
Trigger	Karyawan menempelkan e-KTP karyawan pada pearangkat client
Alur Utama	1. Karyawan menempelkan e-KTP karyawan pada pearangkat client
	2. Sistem membaca identitas e-KTP
	3. Karyawan menekan tombol login
	4. Sistem mengirim identitas e-KTP ke dalam server untuk mendapatkan status kepegawaian
	5. Sistem mendapat status kepegawaian
	6. Sistem menampilkan halaman pilihan menu

Nomor Use Case	FR_009
Nama	Mencetak Bukti Pembayaran
Prasyarat Konteks	Telah melakukan proses transaksi pemesanan pada perangkat client
Tujuan dalam Konteks	Melakukan pencetakan bukti pembayaran terhadap transaksi yang sudah dilakukan
Prakondisi	Terdapat data transaksi pada server
Kondisi Akhir Sukses	Aplikasi menampilkan halaman berisi bukti pembayaran untuk dicetak
Kondisi Akhir Gagal	-
Aktor	Karyawan
Trigger	Karyawan menekan tombol cetak pada transaksi yang dipilah
Alur Utama	1. Karyawan menekan tombol cetak pada transaksi yang dipilah
	2. Sistem mengambil data transaksi dari basis data
	3. Sistem menampilkan data transaksi dalam bentuk <i>printable</i>

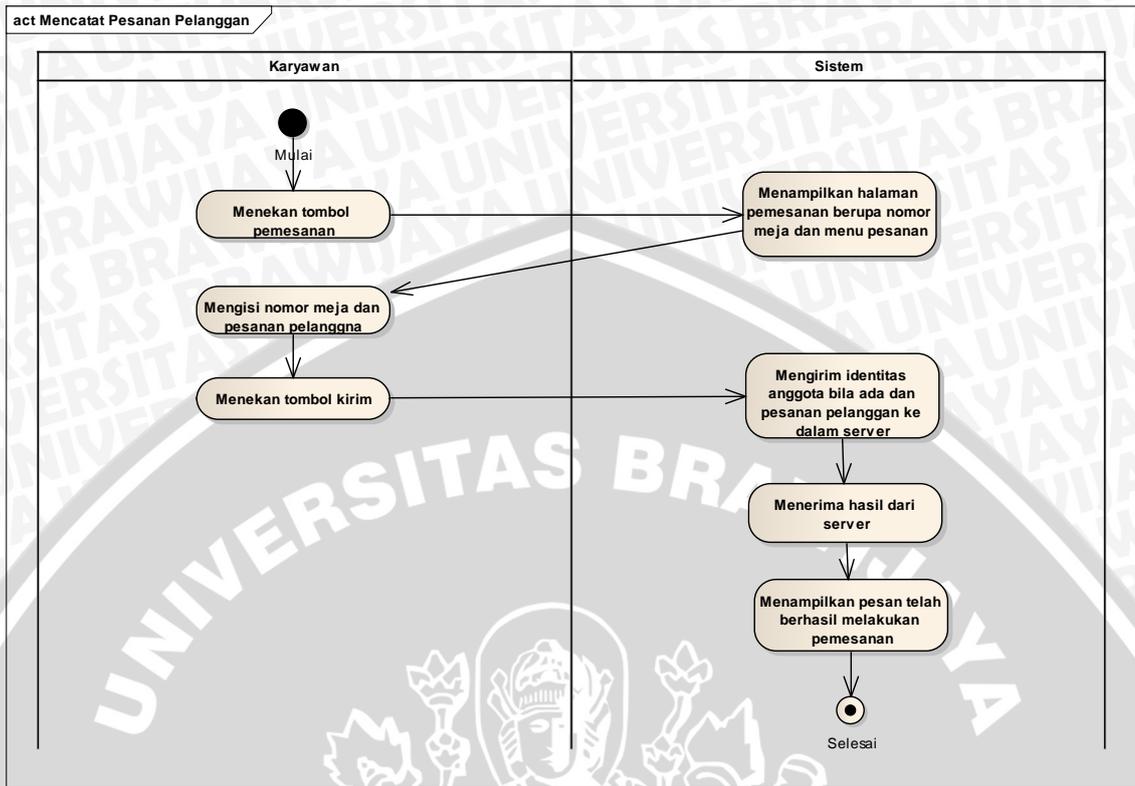
Nomor Use Case	FR_010
Nama	Mengelola Data Anggota
Prasyarat Konteks	Terdapat data anggota pada server
Tujuan dalam Konteks	Melakukan proses validasi anggota baru, mengubah, dan menghapus data anggota
Prakondisi	Terdapat data anggota pada server
Kondisi Akhir Sukses	Aplikasi menampilkan pesan berhasil melakukan aksi validasi, ubah, atau hapus
Kondisi Akhir Gagal	-
Aktor	Karyawan
Trigger	Karyawan memilih menu anggota
Alur Utama	1. Karyawan memilih menu anggota
	2. Karyawan menekan tombol validasi, ubah, atau hapus pada data anggota yang dipilih
	3. Sistem menampilkan pesan berhasil dalam melakukan validasi, ubah, atau hapus



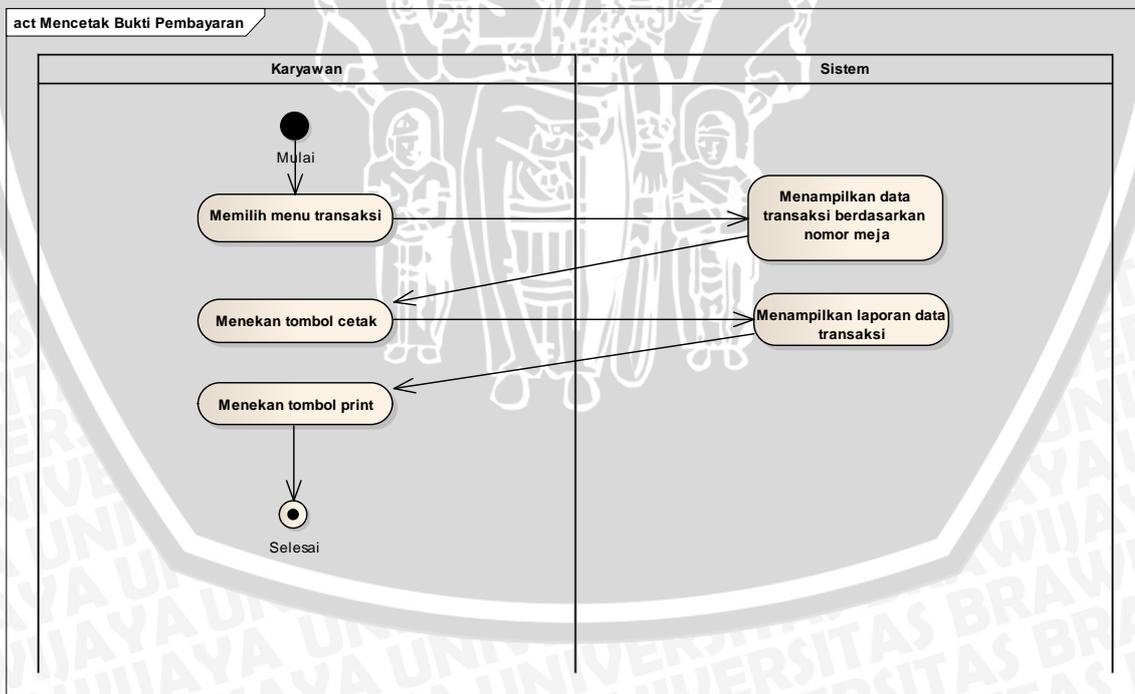
Gambar Activity Diagram pada Skenario Melakukan Login



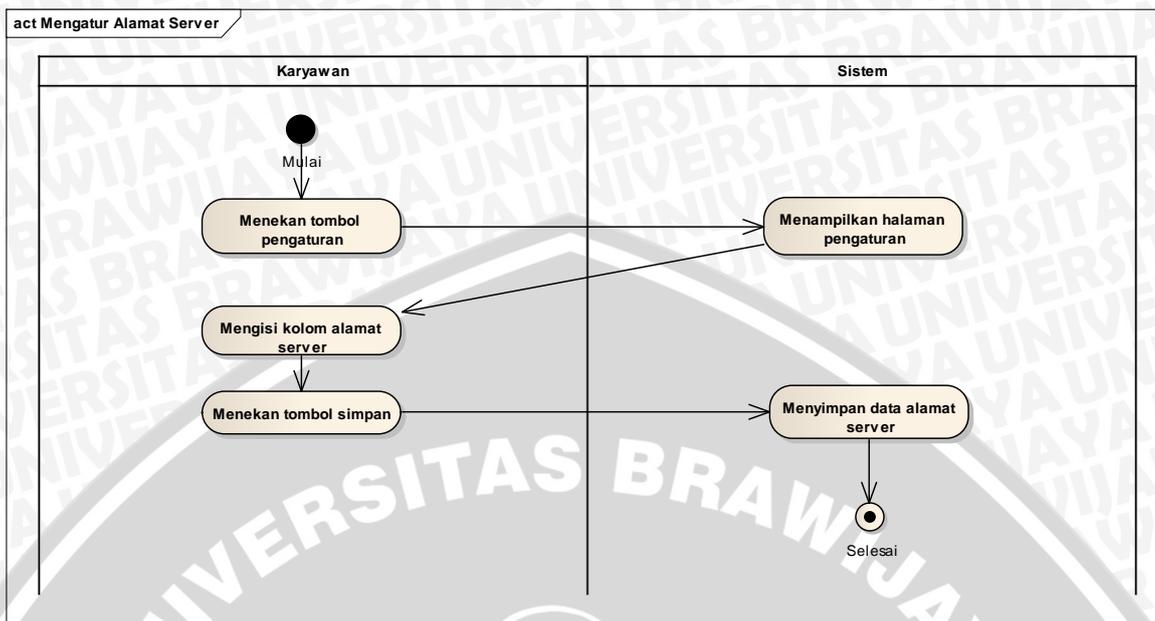
Gambar Activity Diagram pada Skenario Menambah Saldo Anggota



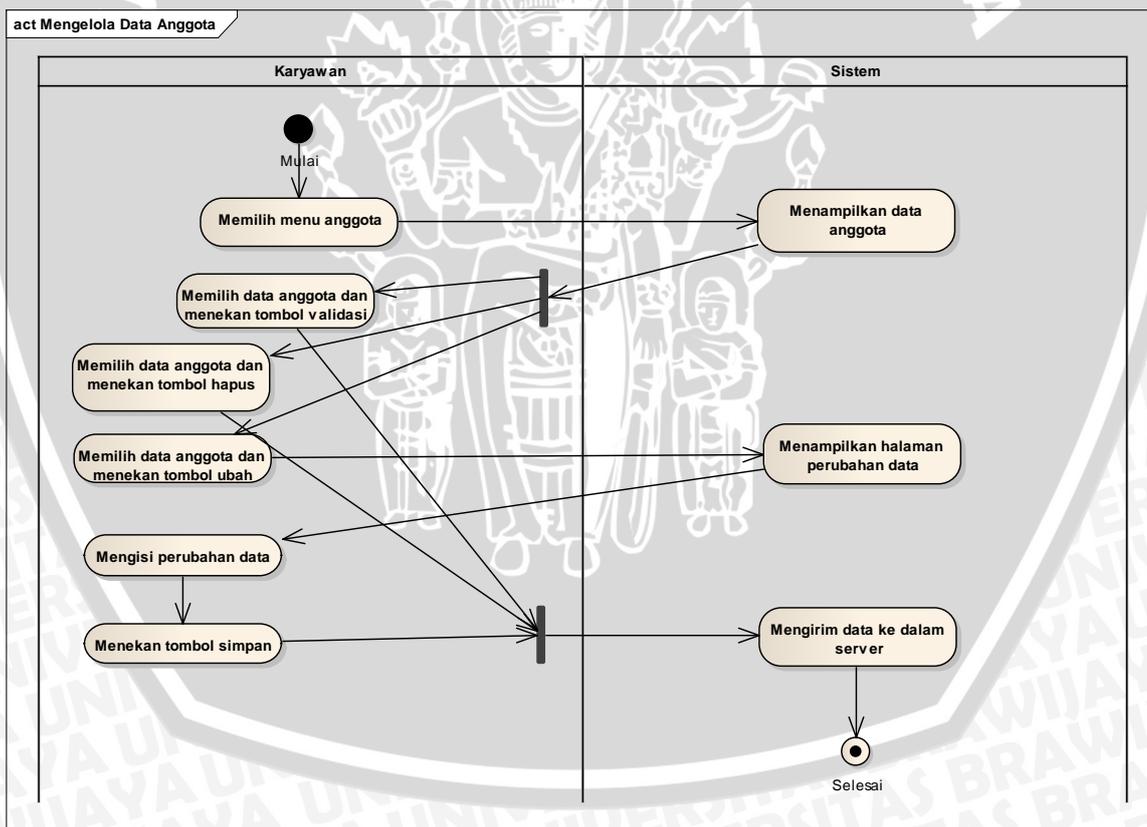
Gambar Activity Diagram pada Skenario Mencatat Pesanan Pelanggan



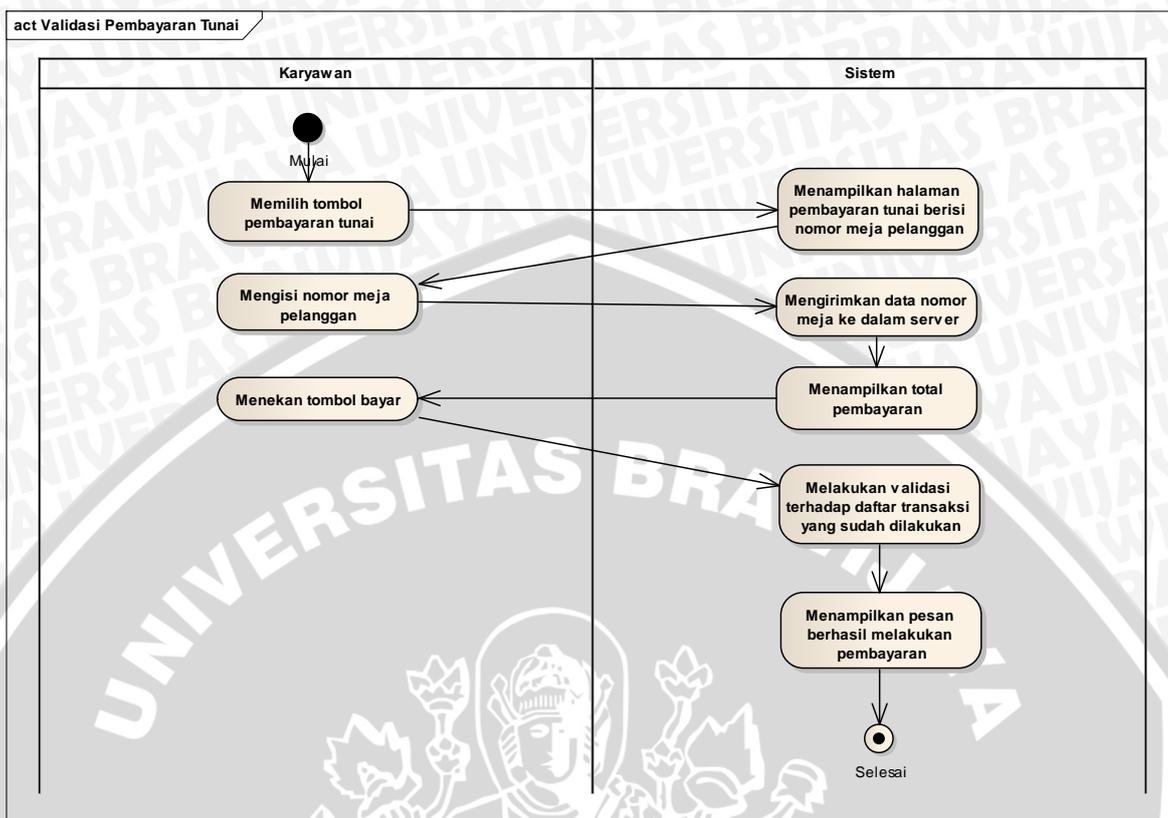
Gambar Activity Diagram pada Skenario Mencetak Bukti Pembayaran



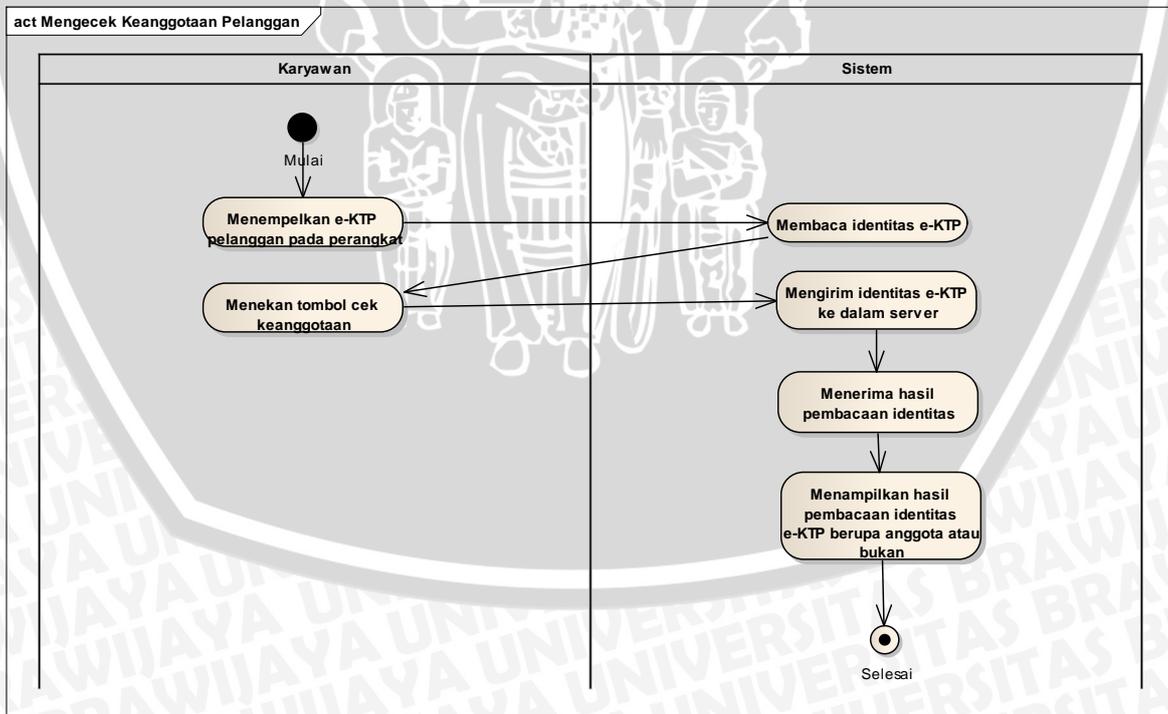
Gambar Activity Diagram pada Skenario Mengatur Alamat Server



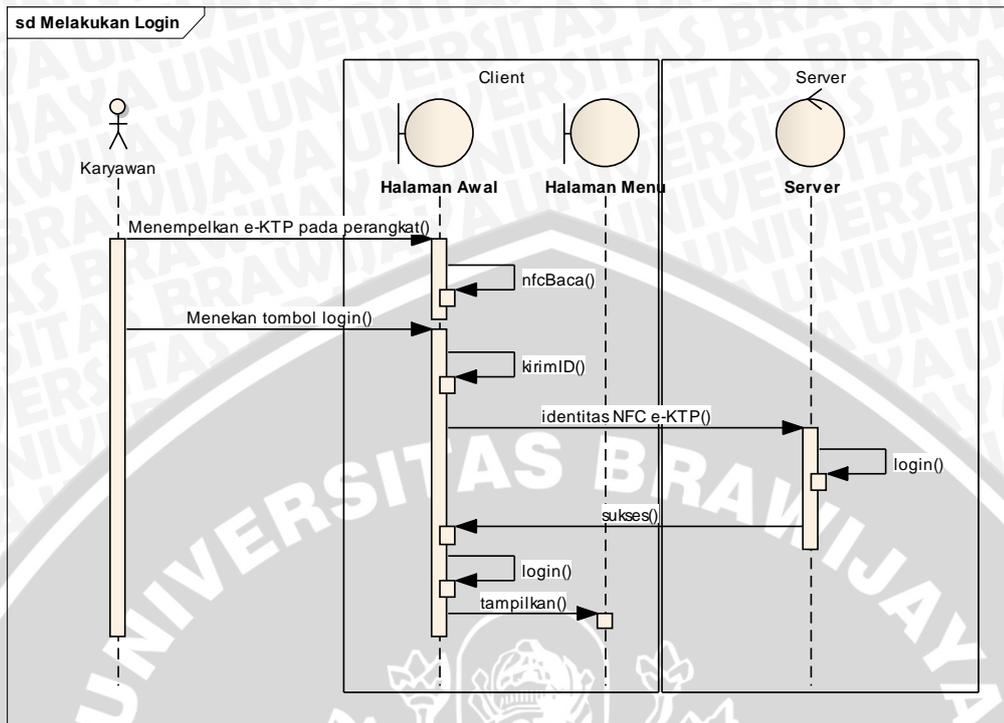
Gambar Activity Diagram pada Skenario Mengelola Data Anggota



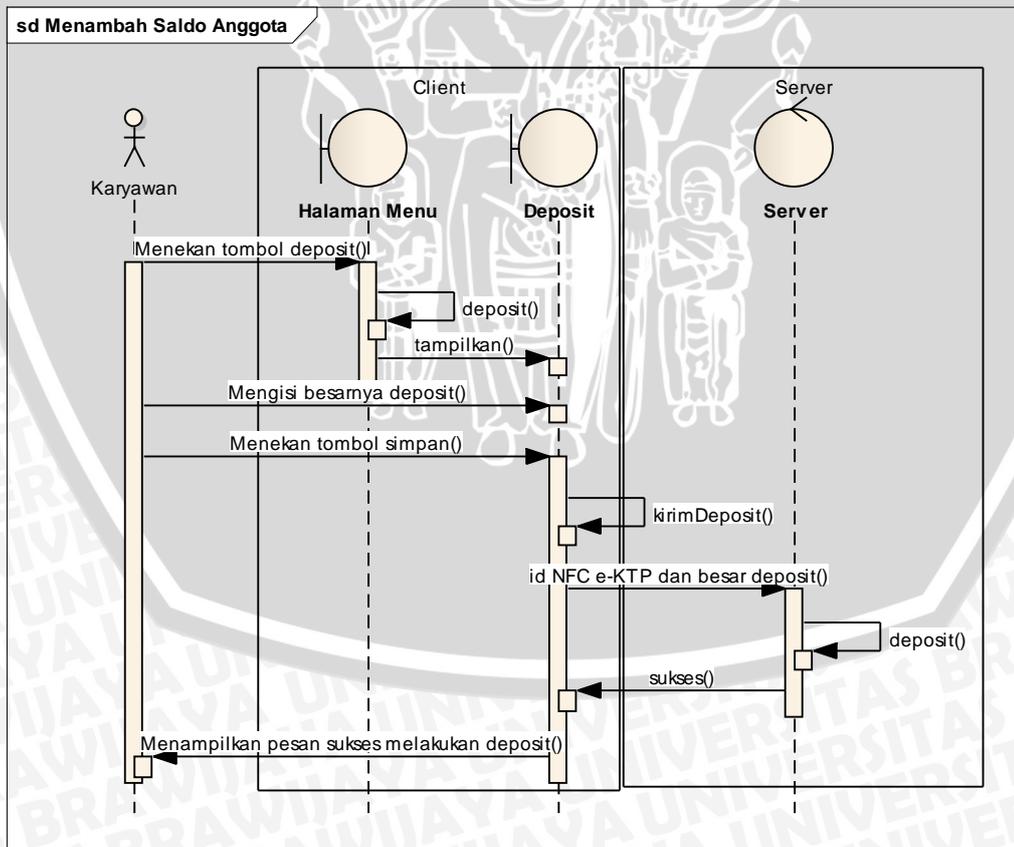
Gambar Activity Diagram pada Validasi Pembayaran Tunai



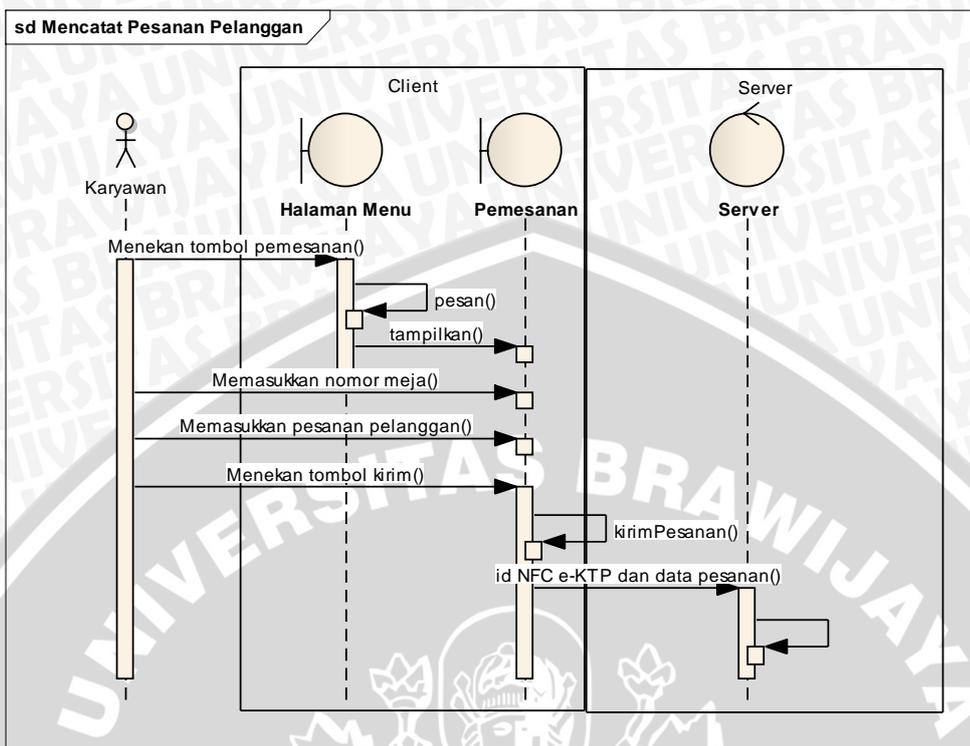
Gambar Activity Diagram pada Skenario Mengecek Keanggotaan Pelanggan



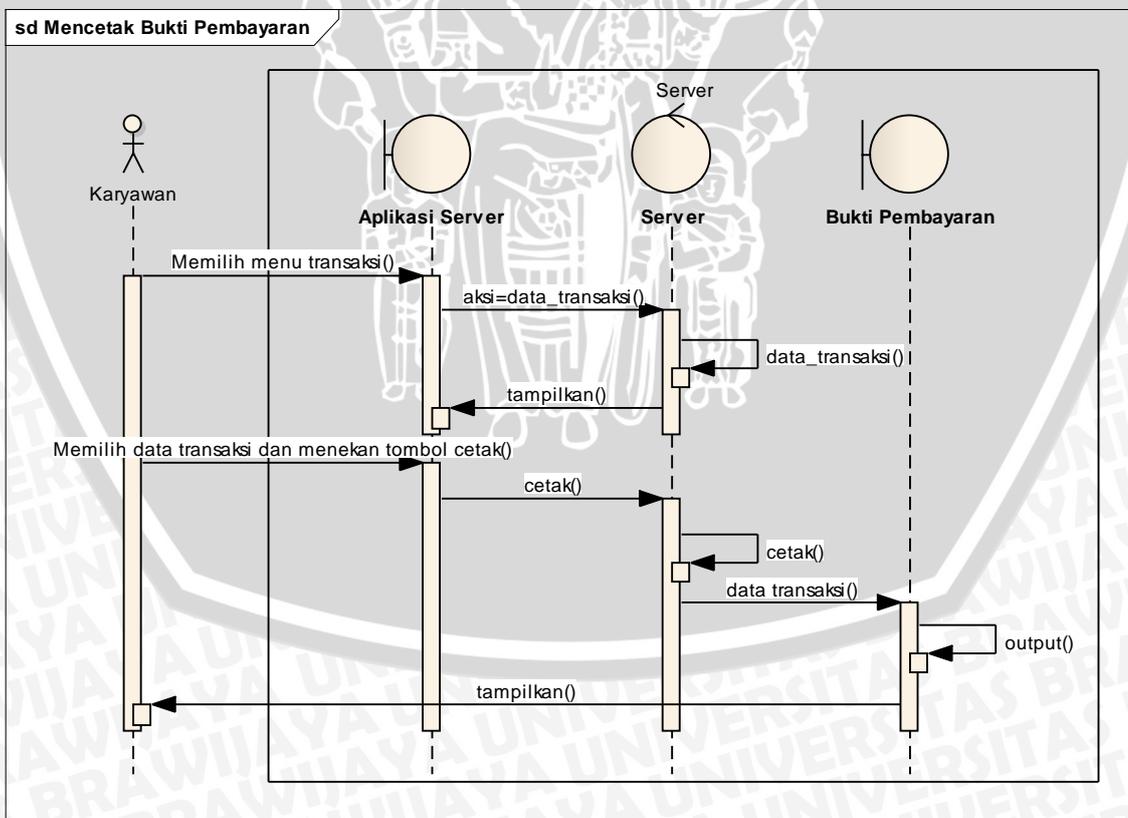
Gambar Sequence Diagram pada Skenario Melakukan Login



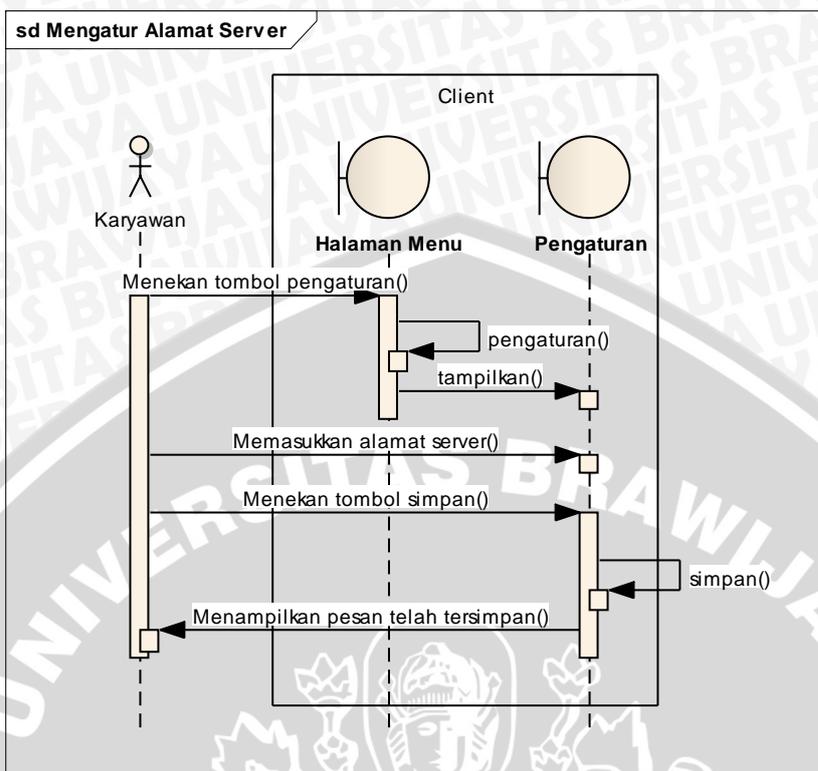
Gambar Sequence Diagram pada Skenario Menambah Saldo Anggota



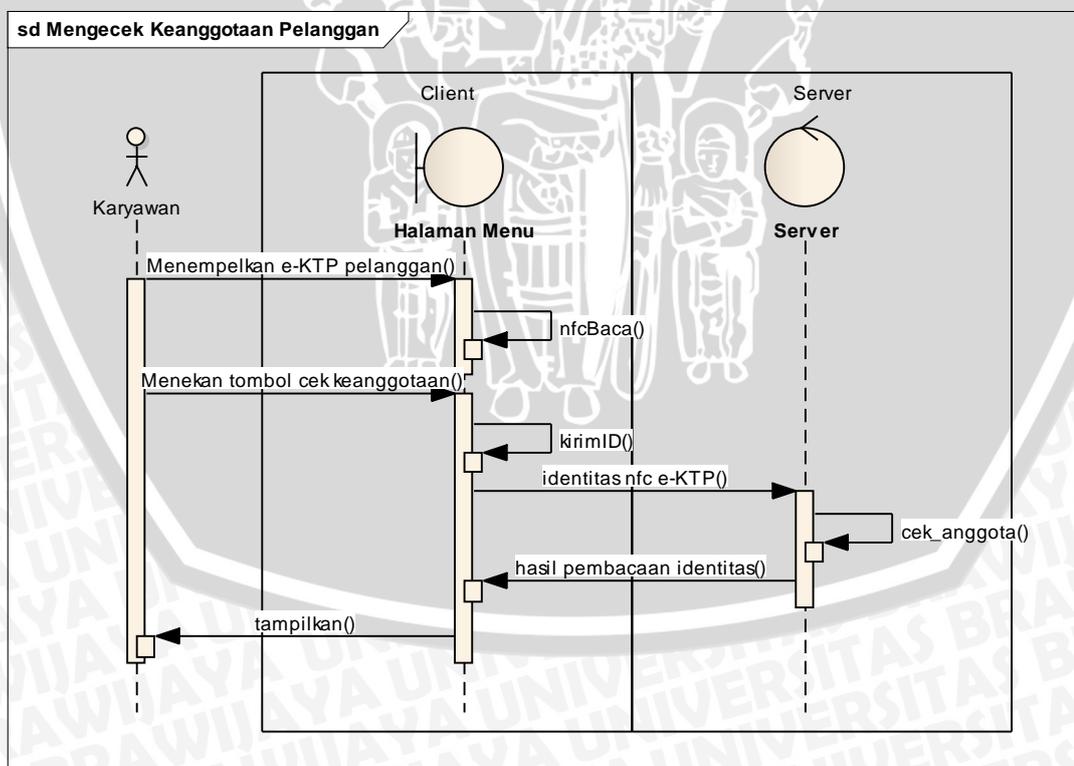
Gambar Sequence Diagram pada Skenario Mencatat Pesanan Pelanggan



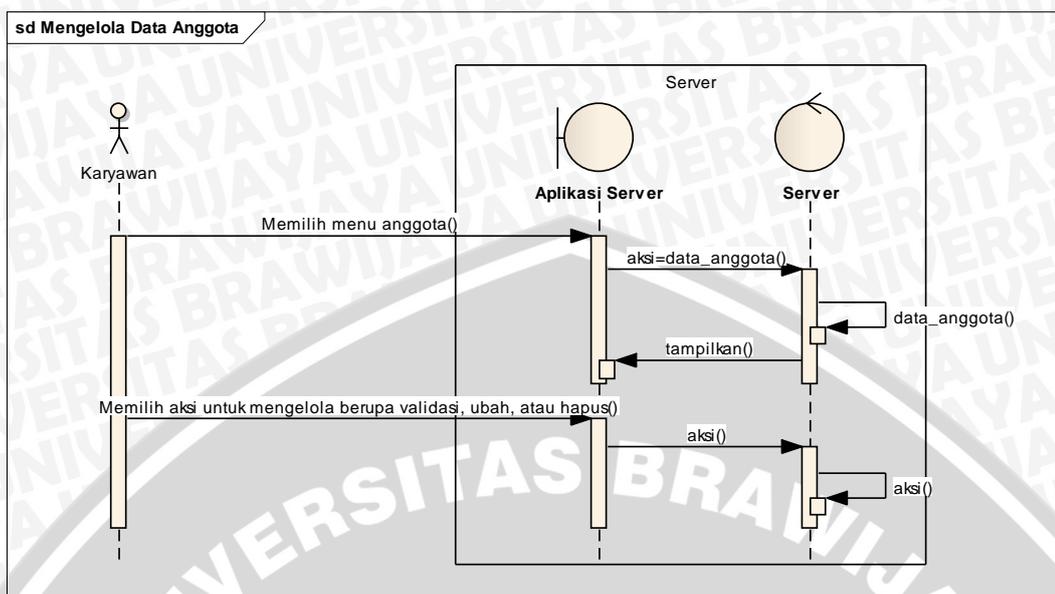
Gambar Sequence Diagram pada Skenario Mencetak Bukti Pembayaran



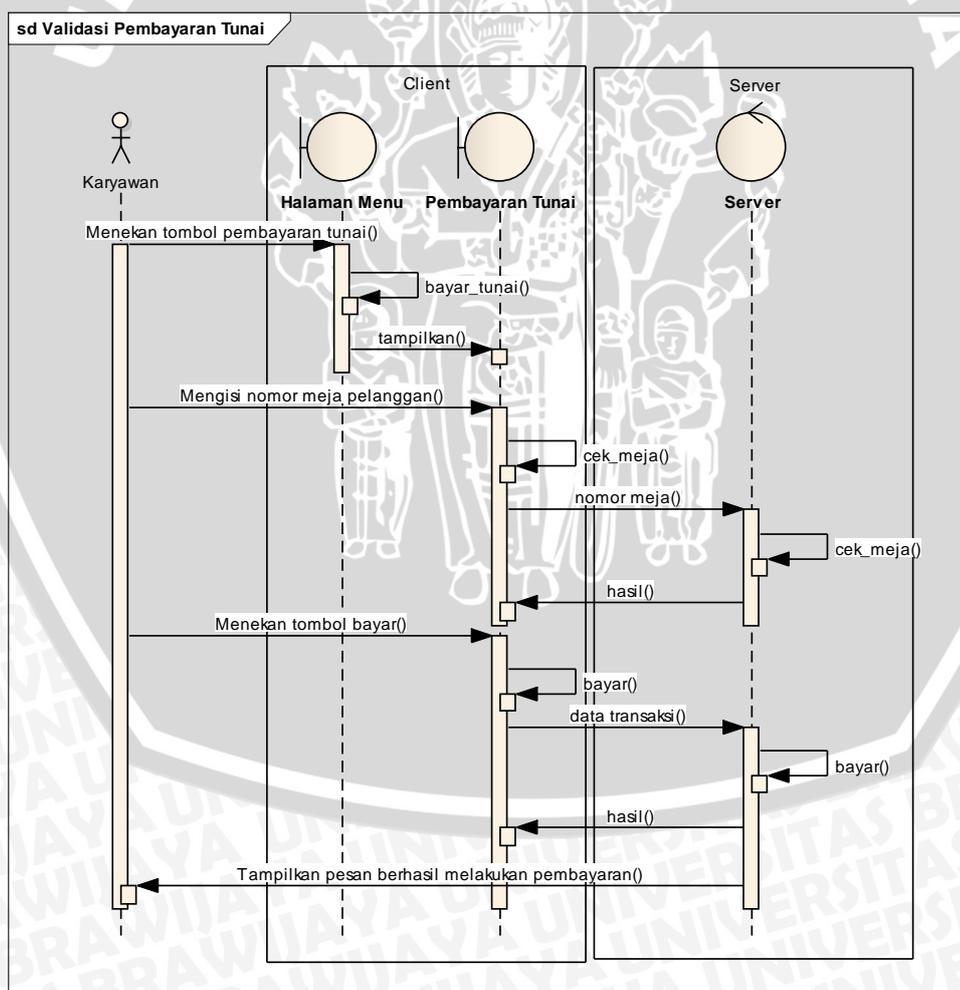
Gambar Sequence Diagram pada Skenario Mengatur Alamat Server



Gambar Sequence Diagram pada Skenario Mengecek Keanggotaan Pelanggan



Gambar Sequence Diagram pada Skenario Mengelola Data Anggota



Gambar Sequence Diagram pada Skenario Validasi Pembayaran Tunai