

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Dalam melaksanakan penelitian ini diambil beberapa referensi dari jurnal terkait sebagai dasar penelitian aplikasi E-Kasir di Antero Distro desa Kecapi diantaranya:

Menurut Heryanto dkk pada tahun 2014 komputer adalah salah satu sarana terpenting bagi badan usaha dimana kita dapat mengolah dan mengakses data yang diperlukan dengan cepat dan tepat sehingga dapat memanfaatkan waktu yang tersisa dengan kegiatan-kegiatan yang lain. Sedangkan android adalah sistem operasi yang memang khusus dirancang untuk *smartphone* dan *tablet*. Sistem android ini memiliki basis *linux* sendiri merupakan sistem operasi yang khusus dirancang untuk komputer[1].

Untuk memenuhi kebutuhan informasi, memerlukan pengolahan yang sistematis dengan cara membentuk suatu sistem informasi. Sistem persediaan barang sangat dibutuhkan oleh perusahaan, karena dengan sistem tersebut perusahaan dapat mendukung operasional usaha suatu perusahaan. Menurut Rahadi dkk pada tahun 2014 Dalam penerapan sistem informasi perlu ditunjang dengan teknologi informasi yaitu pemanfaatan komputer atau *gadget* beserta aplikasi-aplikasinya dan penggunaan jaringan (*network*) komputer atau gadget sebagai alat untuk mempercepat pendistribusian data dan informasi[1].

Menurut David Wahyu Kuncoro dalam jurnal Speed Journal - Sentra penelitian Engineering dan Edukasi yang berjudul Analisis dan Perancangan Sistem Kasir dan Pendataan Stok Barang pada Tata Distro Pacitan dalam jurnal ilmiah ini dijelaskan Tata Distro masih menggunakan sistem pelayanan kasir yang cukup sederhana. Dengan hanya dicatat pada sebuah buku, pelayanan kepada pelanggan menjadi kurang maksimal. Penghitungan hasil transaksi yang hanya mengandalkan sebuah kalkulator dan tanpa dukungan sebuah komputer maupun laptop. Sistem penjualan memiliki banyak kelemahan, kebanyakan terjadi kesalahan pada pencatatan, dan pencarian data yang sulit karena setiap dilakukan pencarian data penjual pada buku besar. Tidak ada informasi khusus

yang menginformasikan tentang jumlah stok barang sehingga tidak jarang ketika stok sudah habis pemilik toko baru mengetahui ketika terjadi proses transaksi sehingga mengecewakan pelanggan. Dengan dilakukan penelitian ini diharapkan mampu menciptakan rancangan pengembangan dari sistem sebelumnya yang berjalan pada Tata Distro sehingga pada bagian kasir agar proses transaksi dengan pelanggan bisa dilakukan dengan cepat. Mampu meningkatkan efisiensi kinerja kasir serta pemanfaatan komputerisasi yang optimal sehingga lebih produktif serta pendataan stok barang lebih terjamin [2].

Menurut Ahmad Jamal Lies Yulianto dalam jurnal on Computer Science - Speed -IJCSS - Volume 10 No 4 yang berjudul Rancang Bangun Sistem Informasi Aplikasi Kasir Menggunakan *Barcode Reader* Pada Toko Dan Jasa Widodo Computer Ngadirojo Kabupaten Pacitan. Dalam jurnal ilmiah ini dijelaskan tentang perancangan dan penerapan aplikasi yang menyediakan informasi mengenai aplikasi Electronic Kasir. Aplikasi ini bertujuan mempermudah karyawan dalam mengolah informasi tentang sistem informasi aplikasi kasir menggunakan barcode reader pada toko dan jasa toko widodo komputer Ngadirojo Kabupaten Pacitan. Metode penelitian yang digunakan adalah SDLC (*System Development Live Cycle*) *prototype* yang menjelaskan secara sistematis mengenai waktu dan tempat dilaksanakan penelitian, sampe dan populasi, teknik pengumpulan data, dan metode analisis data yang digunakan dalam penelitian. Aplikasi yan dihasilkan berupa aplikasi berbasis android yang dirancang menggunakan bahasa pemrograman Java web API menggunakan PHP [3].

Menurut Menurut Rina Br Bukit, SE AK, M. Si; dalam jurnal yang berjudul “Penelitian Implementasi Sistem Informasi :Pendekatan Yang Lebih Integratif”; Fakultas Ekonomi; Program Studi Akuntansi dari Universitas Sumatera Utara Tahun 2002, memaparkan dalam banyak hal, penerapan sistem yang baru atau perbaikan sistem dibuat pada basis data dan jaringan yang telah ada. Jika penerapan sistem yang baru pada sistem kasir pada PT. Indoniaga Lestari memerlukan basis data dan jaringan yang baru atau dimodifikasi maka sistem yang baru ini biasanya harus diimplementasikan sebelum pemasangan program komputer. Tahap kedua adalah membuat dan menguji program. Tahap ini merupakan tahap pertama untuk siklus pengembangan sistem yang spesifik bagi

programer. Tahap kedua ini bertujuan untuk mengembangkan rencana yang lebih rinci dalam pengembangan dan pengujian program komputer yang baru atau program perbaikan komputer yang lama serta mengembangkan program komputer yang secara akurat memenuhi kebutuhan pemrosesan bisnis. Pada tahap ketiga, software package dipasang dan diuji. Tahap ini dilakukan untuk menyakinkan bahwa kebutuhan integrasi system baru terpenuhi. Rencana konversi juga dikembangkan pada tahap ini agar pengiriman system baru kedalam operasi dapat berjalan dengan sukses. Tahap terakhir adalah mengirim system baru ke dalam operasi. Tujuan tahap ini adalah untuk mengubah secara perlahan – lahan sistem lama (konvensional) menjadi sistem baru (komputerisasi) sehingga perlu dilakukan pemasangan basis data yang akan digunakan pada system baru, menyediakan pelatihan dan dokumentasi untuk individu yang akan menggunakan sistem baru tersebut, dan evaluasi proyek dan sistem. Perbedaan penelitian ini dengan kedua jurnal yang diambil pada kajian pustaka dalam penelitian ini, maka praktikan menyimpulkan bahwa dengan dibuatnya sistem informasi dan database yang sudah terkomputerisasi, diharapkan dapat bermanfaat bagi pengguna dalam proses pengelolaan stok barang dan menghasilkan informasi yang tepat pada perusahaan ini juga tepat diaplikasikan pada PT. Indoniaga Lestari sehingga efisiensi kinerja kasir dapat meningkatkan produktifitas [5].

2.2 Tinjauan Pustaka

2.2.1 Profil Distro Antero

Antero Distro adalah sebuah distro sandang atau biasa disebut pakaian yang menyediakan desain-desain pakaian yang dikhususkan untuk remaja, baik yang belum mempunyai pasangan maupun yang memiliki pasangan, untuk sekedar bergaya mengikuti gaya anak muda masa kini. Distro ini aktif dan bergerak dalam bidang sandang yang terdiri dari celana, kemeja, kaos & sandal berbagai macam merk, didirikan pada tanggal 18 September 2014 oleh Mas Vendi berlokasi di tengah-tengah keramaian desa, tepatnya di desa Kecapi RT02 RW 05 Tahunan Jepara. Distro ini baru membuka satu distro dan rencana akan membuka cabang. Membahas soal pemasaran yang dilakukan oleh Antero Distro adalah secara *online* dan *offline*, untuk pemasaran online menggunakan media jejaring sosial seperti Facebook, twitter, serta *website* (Blog). Distro ini menyediakan galeri di

Facebook sehingga pelanggan bisa memilih desain yang mereka sukai. Jika pelanggan ingin menanyakan soal stok yang tersedia di halaman Facebook terdapat kontak yang bisa dihubungi berupa nomor ataupun E-mail. Secara *offline* untuk pemasaran produk bisa dari orang-orang sekitar Antero Distro dan dari mulut ke mulut jika konsumen ingin memesan atau membeli produk dari distro tersebut. Visi Antero distro ini adalah menyediakan berbagai jenis sandang berkualitas dengan desain bagus, unik, kreatif, dan sesuai dengan gaya anak muda masa kini. Sedangkan misi pada Antero Distro adalah menciptakan suasana nyaman dan memberikan layanan yang terbaik kepada konsumen (pembeli).

2.2.2 E-Kasir

E-Kasir atau kasir elektronik adalah salah satu bentuk digitalisasi data transaksi pada sebuah toko atau sarana jual beli dengan tujuan untuk memudahkan pencatatan data transaksi. Pada sistem kasir konvensional, seluruh rekam transaksi dicatat dengan cara manual, seperti mencatat jumlah dan *stock* barang, jumlah barang yang terjual, data harga dan lain sebagainya dicatat dalam pembukuan toko tersebut. Karena semua dilakukan dengan cara manual maka rawan terjadi kesalahan penulisan dan hal lainnya yang menyebabkan pembukuan keuangan tersebut menjadi tidak valid. Sistem E-Kasir menawarkan kemudahan dalam hal pembukuan transaksi sebuah toko. Dimana, semua transaksi baik itu *stock* maupun penjualan direkam secara digital dan *real time*.

2.2.3 Android

Android adalah salah satu sistem operasi (OS) yang dibuat khusus untuk perangkat seluler seperti *smarthphone* dan tablet. Sistem operasi ini berbasis *linuk* yang dikembangkan langsung oleh Android Inc. Dengan dukungan penuh dari Google Finance namun kemudian resmi dibeli Google pada tahun 2005 dan resmi dirilis pada 5 November 2007. Kelebihan OS Android dibanding OS lain yaitu memiliki sistem operasi yang bersifat *open source* yang berarti Google mengizinkan pengembang untuk mengembangkan sistem operasi tersebut. Sehingga sistem operasi yang ada di *smarthphone* sekarang bisa menyesuaikan mulai dari kelas *low-end* hingga *high-end* yang menjadikan sistem operasi android meningkat cukup tajam. [1]

Kelebihan android dibanding sistem operasi lainnya:

1. *User friendly*

Sistem android sangat mudah dijalankan oleh siapapun sama halnya pada sistem operasi windows yang ada pada komputer. Butuh waktu sebentar untuk bisa menggunakan dan mempelajari *smartphone* berbasis android.

2. Kemudahan dalam notifikasi

Setiap ada pesan, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di *Home Screen smartphone* Android beserta bunyi atau getar yang bisa diatur, sehingga Anda tidak akan terlewatkan satu pemberitahuan sekalipun.

3. Aplikasi

Di dalam android telah disajikan jutaan aplikasi yang menarik mulai dari yang gratis hingga berbayar. Dan bisa didapatkan dengan mendownload pada google play.

4. *Multitasking*

Ponsel android bisa menjalankan berbagai aplikasi sekaligus sama seperti komputer.

5. *Open Source*

Android merupakan sistem operasi yang bersifat open source yang berarti pengembang dapat dengan bebas mengembangkan sistem androidnya sendiri sesuai dengan kebutuhan.

2.2.4 **Android Studio**

Android Studio adalah sebuah Integrated Development Environment (IDE) yang dikembangkan oleh Google dan dipakai untuk pengembangan aplikasi Android. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE java populer, yaitu intellij IDEA.

Sebagai pengembangan dari Eclipse, Android Studio mempunyai beberapa fitur baru dibandingkan Eclipse IDE, yaitu:

1. Bisa mem-*build multiple* APK.
2. *Layout editor* yang lebih bagus.
3. Menggunakan *Gradle-based build* sistem yang fleksibel.[11]

2.2.5 Extensible Markup Language (XML)

XML merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. Pada *level* paling detail *web service* secara keseluruhan dibentuk diatas XML. Fungsi utama dari XML adalah komunikasi antar aplikasi, komunikasi aplikasi eksternal dengan *partner* luaran dan integrasi data. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain

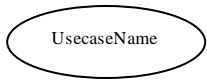
2.2.6 Unified Modelling Language (UML)

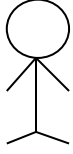



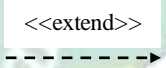
Unified Modelling Language atau bisa disingkat UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk merancang, memvisualisasikan dan mendokumentasikan sistem perangkat lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis perangkat lunak, dimana aplikasi tersebut dapat berjalan pada sistem operasi, perangkat lunak dan jaringan apapun, serta dapat ditulis dalam bahasa pemrograman apapun. Ketika membuat model menggunakan konsep UML ada beberapa aturan yang harus dipatuhi. dan membuat bagaimana elemen-elemen pada model saling terhubung satu sama lain mengikuti standar yang ada. [2]

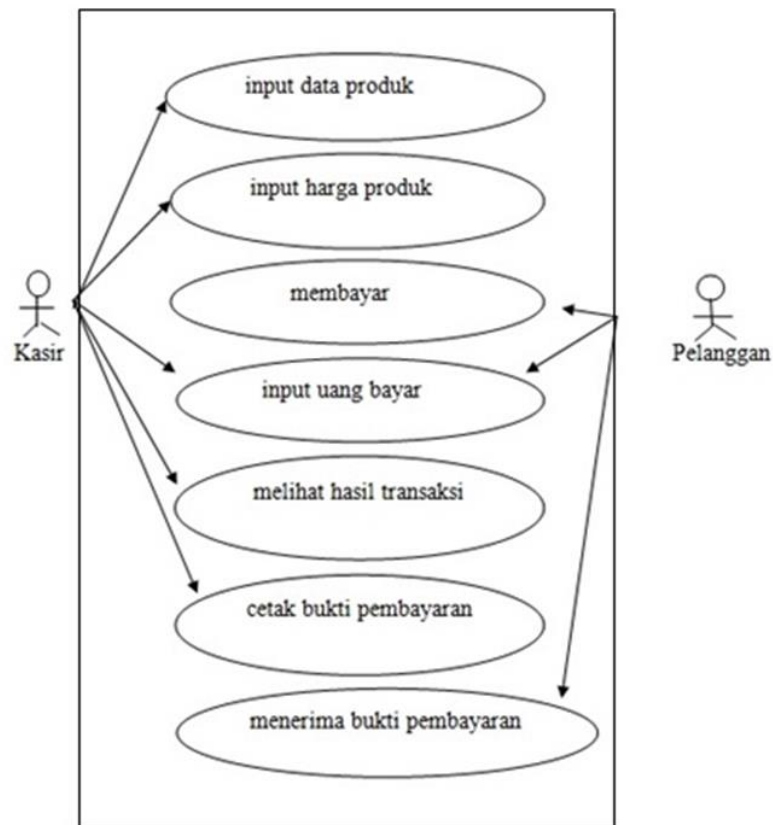
1.2.6.1 Usecase Diagram

Usecase diagram berisi penjelasan tentang manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Diagram ini menjelaskan fungsionalitas, kelas sistem dan bagaimana sistem berinteraksi dengan dunia luar. Usecase merupakan sebuah lambang pekerjaan tertentu, semisal *login*, *create* dan sebagainya.

Tabel 2. 1 Simbol Usecase Diagram

No	Nama	Deskripsi	Simbol
1.	Use Case	Usecase adalah gambaran fungsionalitas atau kegunaan dari suatu sistem (entitas lain) yang berinteraksi dengan <i>actor</i> sistem	

2.	Actor	Actor menggambarkan orang, sistem atau <i>external</i> entitas yang menyediakan atau menerima informasi sistem	 ActorName
3.	System Boundary	Kotak yang menetapkan lingkup sistem untuk <i>usecase</i> . Semua <i>usecase</i> di luar kotak akan dianggap di luar lingkup sistem itu.	
4.	Association	Partisipasi suatu <i>actor</i> dalam <i>usecase</i> , yaitu menghubungkan <i>actor</i> dan <i>instance</i> dari <i>usecase</i> berkomunikasi satu sama lain	
5.	Generalization	Hubungan taksonomi antara <i>usecase</i> yang lebih umum dan <i>usecase</i> yang lebih spesifik	
6.	Extend	Hubungan dari <i>usecase</i> ekstensi ke <i>usecase</i> dasar, menentukan bagaimana perilaku untuk <i>usecase</i> ekstensi dapat dimasukkan ke dalam perilaku yang ditentukan untuk <i>usecase</i> dasar.	
7.	Include	Suatu hubungan dari <i>usecase</i> dasar ke <i>usecase</i> inklusi, menentukan bagaimana perilaku untuk <i>usecase</i> inklusi dimasukkan ke dalam perilaku yang ditentukan untuk <i>usecase</i> dasar.	




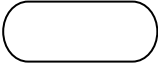
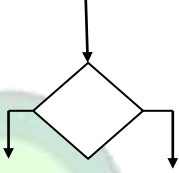

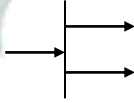
Gambar 2. 1 Contoh Usecase Diagram Sistem Kasir

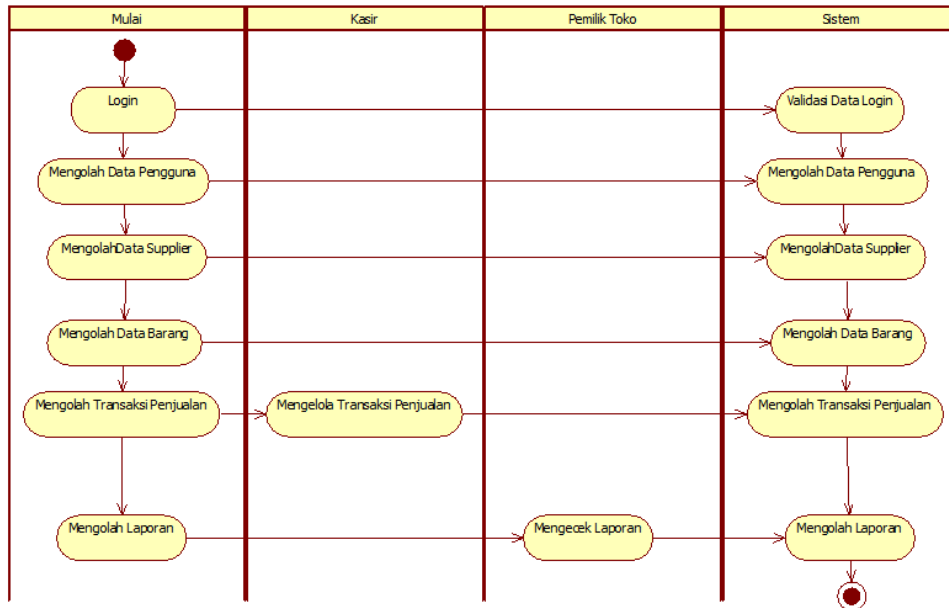
1.2.6.2 Activity Diagram

Activity Diagram merupakan diagram yang bersifat dinamis, diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini penting terutama dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Tabel 2. 2 Simbol Activity Diagram

No	Nama	Deskripsi	Simbol
1.	Initial State	Initial state adalah awal mulainya suatu aliran kerja pada <i>activity diagram</i> dan dalam setiap <i>diagram activity</i> hanya terdapat satu <i>initial state</i>	●

2.	Final State	Final state adalah bagian akhir dari suatu aliran kerja pada sebuah <i>activity diagram</i> , dan pada setiap <i>activity diagram</i> dapat memiliki lebih dari satu <i>final state</i> .	
3.	Activity	Activity adalah pekerjaan atau aktivitas yang dilakukan aliran kerja pada sebuah <i>activity diagram</i> .	
4.	Decision	Decision berfungsi untuk menggambarkan pilihan kondisi dimana terdapat kemungkinan perbedaan transisi, serta untuk memastikan bahwa aliran kerja dapat mengalir ke lebih dari satu jalur.	
5.	Merge	Merge berfungsi untuk menggabungkan kembali aliran kerja yang sebelumnya telah dipecah oleh <i>decision</i> .	
6.	Transition / Association	Transition berfungsi untuk menghubungkan aktivitas selanjutnya setelah aktivitas sebelumnya.	
7.	Synchronization Fork	Synchronization Fork digunakan untuk memecah <i>behavior</i> menjadi aktivitas yang paralel.	
8.	Synchronization Join	Synchronization Join digunakan untuk menggabungkan kembali aktivitas yang paralel.	




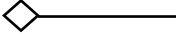



Gambar 2. 2 Contoh Activity Diagram Sistem Kasir

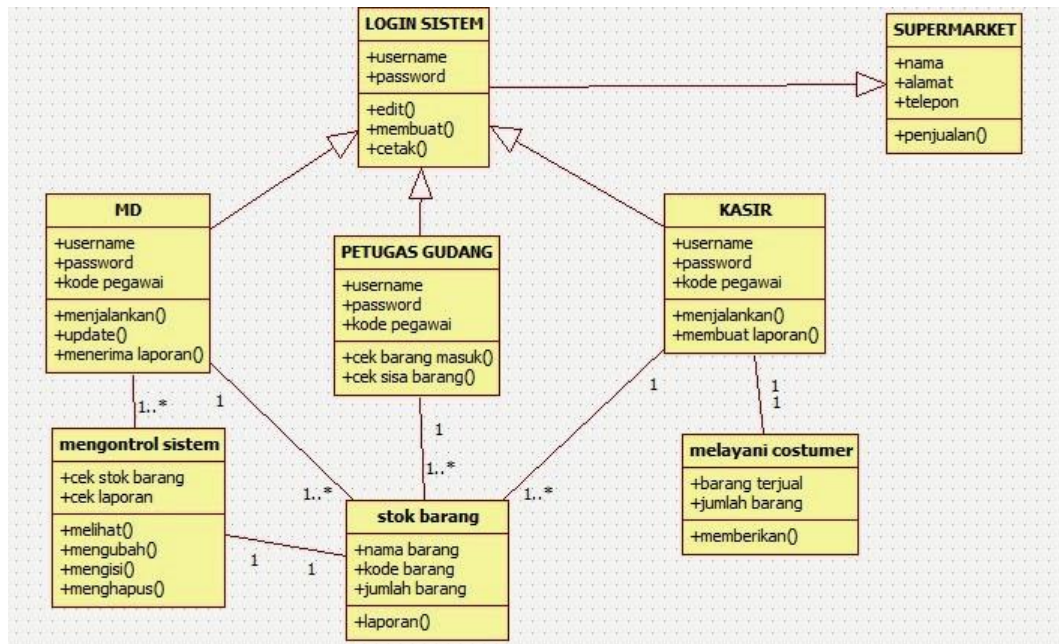
1.2.6.3 Class Diagram

Class Diagram merupakan sebuah spesifikasi yang ketika dibuatkan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (properti atau atribut) suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi) Class diagram menggambarkan struktur dan deskripsi class package dan objek beserta hubungan satu sama lain seperti pewarisan containment, asosiasi dan lain lain [3]

Tabel 2. 3 Simbol Class Diagram

No	Nama	Deskripsi	Simbol			
1.	Class	Class adalah penggambaran keadaan (atribut/ <i>property</i>) suatu sistem, sekaligus layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">ClassName</td> </tr> <tr> <td style="text-align: center;">Attributes</td> </tr> <tr> <td style="text-align: center;">Operations (Methods)</td> </tr> </table>	ClassName	Attributes	Operations (Methods)
ClassName						
Attributes						
Operations (Methods)						
2.	Association	Association adalah hubungan statis antar <i>class</i> . Umumnya				

		menggambarkan <i>class</i> yang memiliki atribut berupa <i>class</i> lain. Arah panah <i>navigability</i> menunjukkan arah <i>query</i> antar <i>class</i> .	
3.	Aggregation	Aggregation adalah hubungan yang menyatakan bagian (“terdiri atas”) atau biasa disebut relasi mempunyai sebuah	
4.	Composition	Composition adalah sebuah <i>class</i> yang tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain.	
5.	Dependency	Dependency adalah hubungan yang menunjukkan operasi pada suatu kelas yang menggunakan <i>class</i> yang lain.	
6.	Generalization	Generalization atau pewarisan adalah hubungan hirarkis antar <i>class</i> . Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda <i>class</i> asalnya dan menambahkan fungsionalitas baru, sehingga disebut anak <i>class</i> yang diwarisinya	




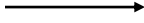
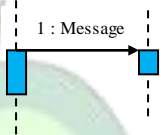
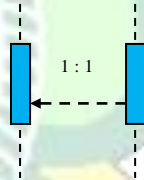
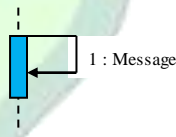
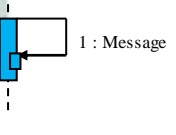
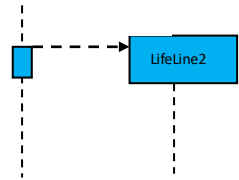
Gambar 2. 3 Contoh Class diagram Sistem Kasir

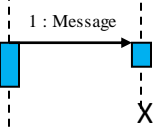
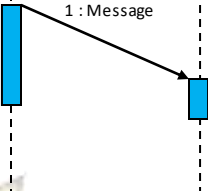

1.2.6.4 Sequence Diagram

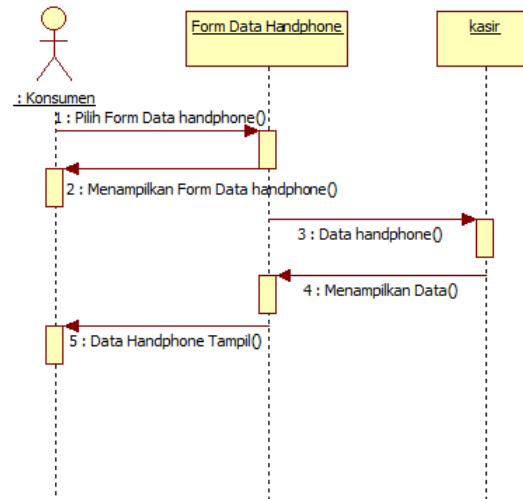
Menjelaskan interaksi dari objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan usecase. Sequence diagram memperlihatkan tahapan-tahapan apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam usecase. Diagram sequence sebaiknya digunakan di awal tahap desain atau analisis karena sifatnya yang sederhana dan mudah dimengerti.

Tabel 2. 4 Simbol Sequence Diagram

No	Nama	Deskripsi	Simbol
1.	Object	Object adalah <i>instance class</i> dan dituliskan tersusun secara <i>horizontal</i>	
2.	Actor	Actor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem utama. Actor berada diluar sistem utama itu sendiri	
3.	Lifeline	Lifeline adalah <i>symbol</i> yang mengindikasikan keberadaan sebuah obyek dalam basis waktu.	

4.	Activation	Activation adalah <i>symbol</i> yang mengindikasikan sebuah obyek yang melakukan sebuah aksi.	
5.	Message	Message adalah <i>symbol</i> yang mengindikasikan adanya komunikasi antar obyek satu dengan yang lainnya.	
6.	Call Message	Call message atau pesan panggilan adalah sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifelines dari sebuah interaksi. Call message adalah jenis pesan yang mewakili permintaan operasi dari target Lifeline	
7.	Return Message	Return Message atau pesan balik merupakan sebuah pesan yang mewakili informasi yang dikirim kembali kepada pengirim pesan atau pemanggil berdasarkan pesan sebelumnya.	
8.	Self message	Self message merupakan jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama	
9.	Recursive Message	Recursive message adalah jenis pesan yang mewakili permohonan pesan dari <i>lifeline</i> yang sama, dan menargetkan aktivasi di aktivasi tempat pesan itu berasal.	
10.	Create Message	Create message atau membuat pesan merupakan jenis pesan yang mewakili instansiasi (target) Lifeline.	

11.	Destroy Message	Destroy Message merupakan sebuah pesan yang mewakili permintaan untuk menghancurkan siklus hidup target.	
12.	Duration Message	Pesan durasi menunjukkan jarak antara dua batasan waktu untuk permohonan pesan	
13.	Note	Note atau catatan adalah catatan tau lampiran berbagai komentar ke elemen. Sebuah komentar tidak memiliki kekuatan semantik, tetapi dapat berisi informasi yang berguna bagi <i>user</i> .	



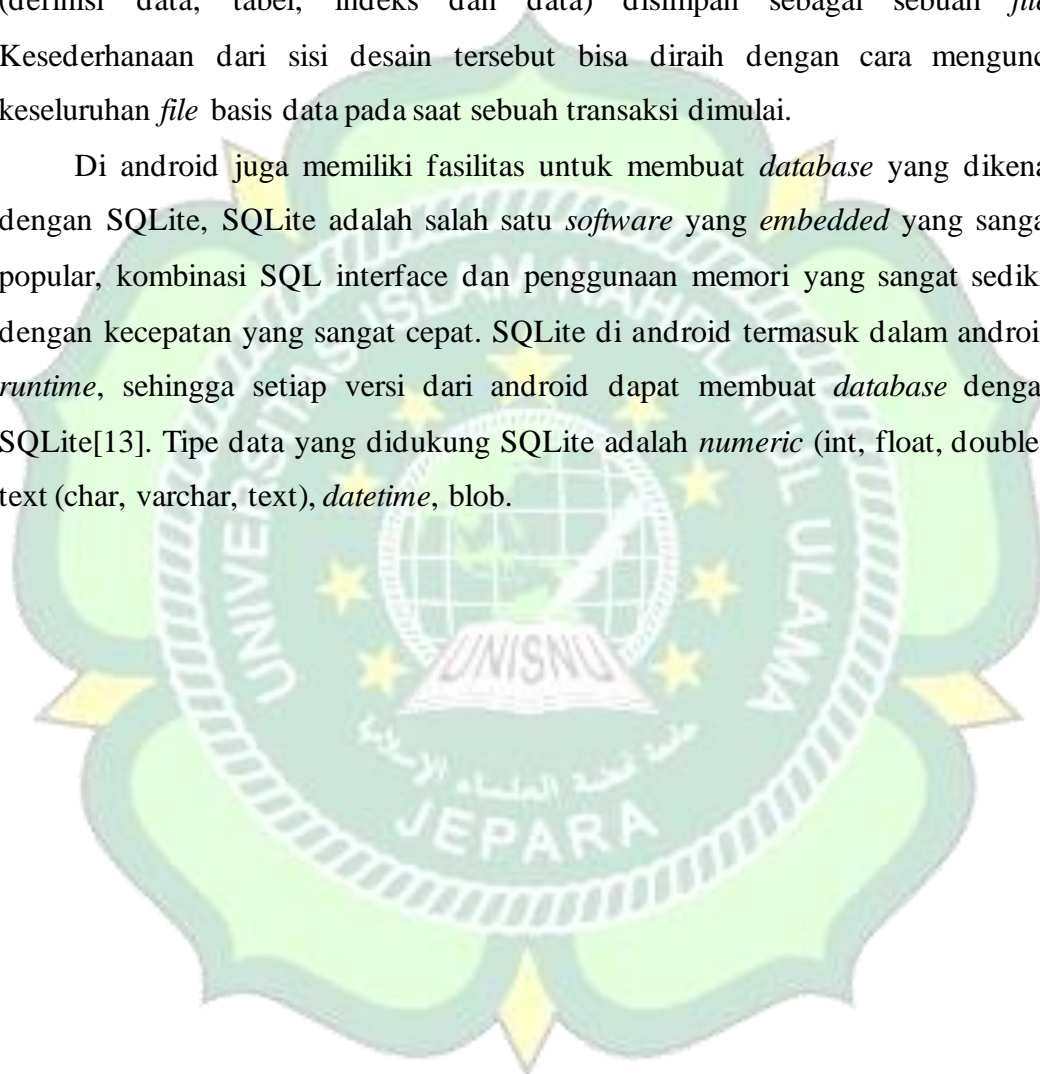
Gambar 2. 4 Contoh Sequence Diagram Kasir

2.2.7 SQLite

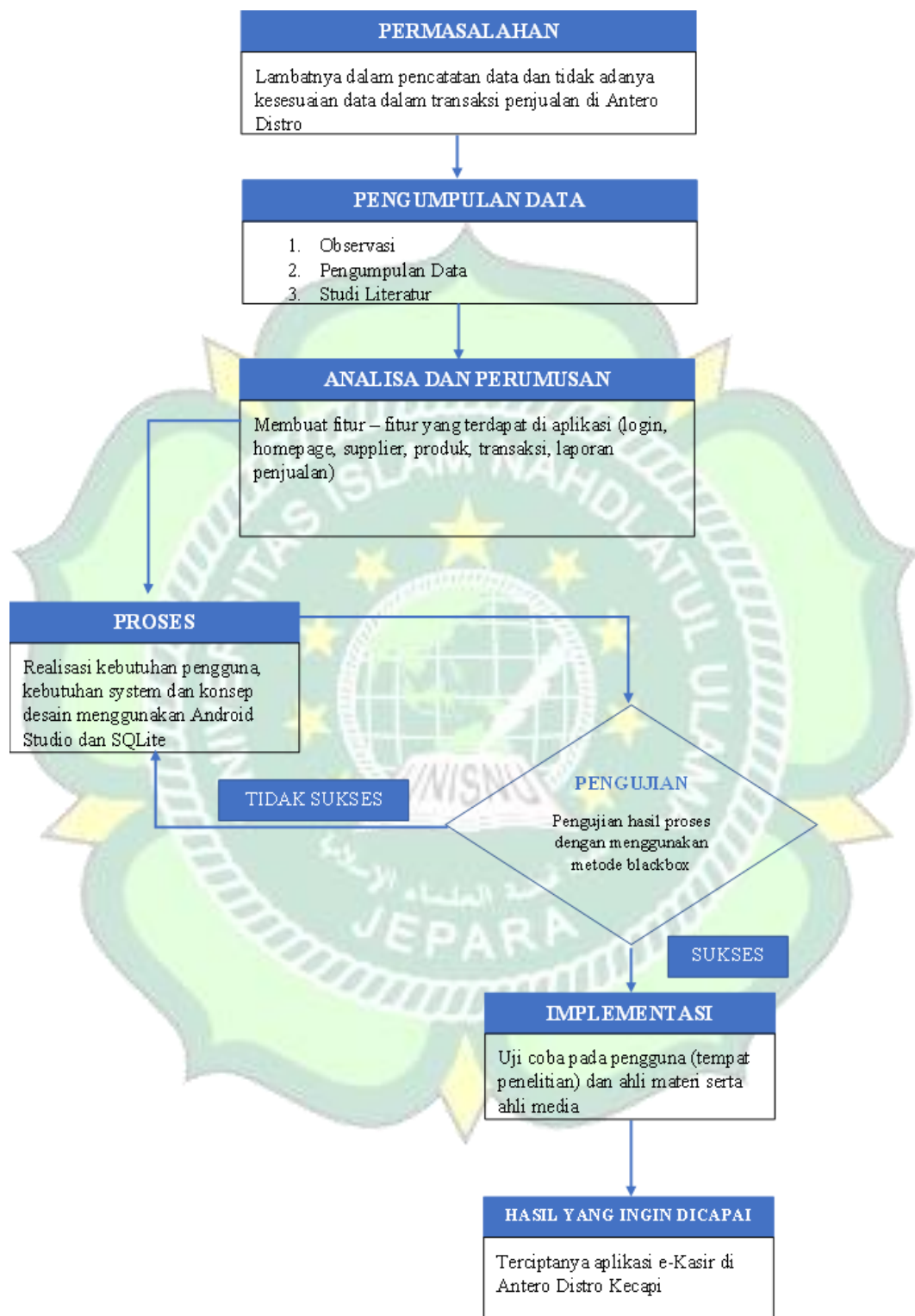
SQLite merupakan sebuah sistem manajemen *database* relasional yang bersifat ACID-Compliant dan memiliki ukuran pustaka kode yang relatif kecil, serta ditulis dalam bahasa C. SQLite merupakan salah satu database yang bersifat *open source*. Tidak seperti paradigma *client-server* umumnya, inti SQLite

bukanlah sebuah sistem yang mandiri yang mampu berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protocol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times* dan secara keseluruhan lebih sederhana. Seluruh elemen basis data (definisi data, tabel, indeks dan data) disimpan sebagai sebuah *file*. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan *file* basis data pada saat sebuah transaksi dimulai.

Di android juga memiliki fasilitas untuk membuat *database* yang dikenal dengan SQLite, SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi SQL interface dan penggunaan memori yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di android termasuk dalam android *runtime*, sehingga setiap versi dari android dapat membuat *database* dengan SQLite[13]. Tipe data yang didukung SQLite adalah *numeric* (int, float, double), text (char, varchar, text), *datetime*, blob.



2.3 Kerangka Pemikiran



Gambar 2. 5 Kerangka Pemikiran