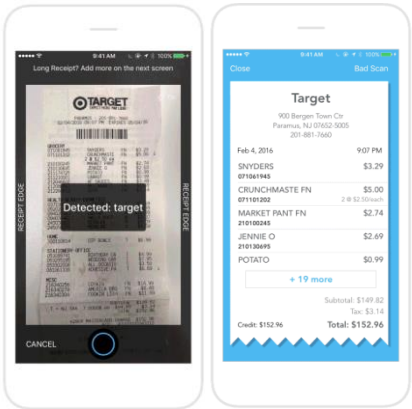
BAB 2
LANDASAN TEORI

2.1 Optical Character Recognition

OCR (*Optical Character Recignition*) adalah sebuah aplikasi yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk di konversi ke dalam bentuk *file* tulisan[2]. Sistem pengenal huruf ini dapat meningkatkan fleksibilitas atau kemampuan dan kecerdasan sistem komputer. Sistem pengenal huruf yang cerdas sangat membantu usaha besar-besaran yang saat ini dilakukan banyak pihak yakni usaha digitalisasi informasi dan pengetahuan.

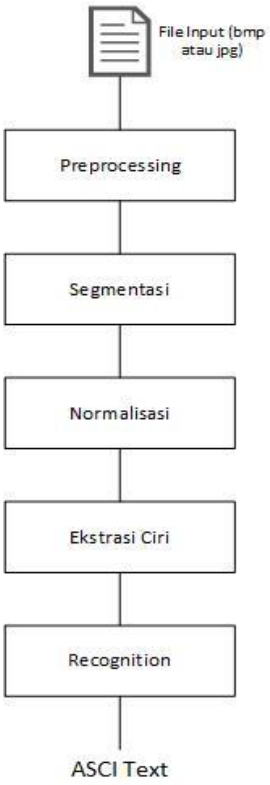
1.2 BlinkReceipt SDK

BlinkReceipt adalah sebuah SDK (*Software Development Kit*) yang dapat digunakan untuk melakukan scanning berbagai jenis faktur pembelajaan dan melakukan ekstrasi semua data didalam faktur tersebut. *BlinkReceipt* berjalan secara lokal di perangkat (tidak ada pemprosesan sisi server), memiliki akurasi tinggi, skala tanpa batas. Cara kerjanya mirip OCR (*Optical Character Recognition*) sebuah teknik pendeteksi dari sebuah *image* digital[5].



Gambar 2.1 *BlinkReceipt*

Secara umum proses *BlinkReceipt* sama halnya dengan OCR dapat dilihat pada gambar 2.1[3], dengan penjelasan sebagai berikut :



Gambar 2.2 Proses OCR *BlinkReceipt*

a. *File Input*

File input berupa *file* citra digital dengan format bmp dan jpg.

b. *Preprocessing*

Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang diperlukan pada gambar *input* untuk proses selanjutnya.

c. Segmentasi

Segmentasi adalah proses memisahkan area pengamatan (*region*) pada tiap karakter yang dideteksi.

d. Normalisasi

Normalisasi adalah proses merubah dimensi *region* tiap karakter dan ketebalan **karakter.**

e. Ekstrasi Ciri

Ekstrasi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.

f. *Recognition*

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.

2.3 Convolutional Neural Network

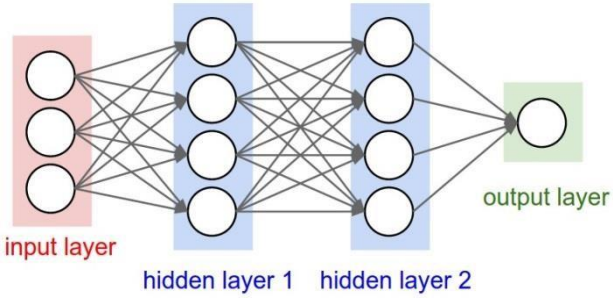
Convolutional Neural Network (CNN) adalah pengembangan dari *MultiLayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network*, karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra[6].

Convolutional Neural Network (CNN) pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Reserch Laboratories, Kinuta,Setagaya, Tokyo, Jepang[7]. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet barhasil diterapkan oleh Le Chin pada penelitiannya mengenai pengenalan angka dan tulisan tangan[8].

Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetensi ImageNet *Large Scale Visual Recognitio*n *Challenge* 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, Khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra.

2.3.1 Konsep CNN

Cara Kerja *Convolutional Neural Network* (CNN) memiliki kesamaan pada *Multilayer Percepton* (MLP), namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



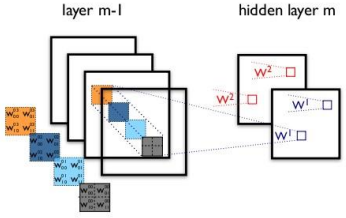
Gambar 2.3 Arsitektur MLP Sederhana

Sebuah MLP seperti pada gambar 2.2. memiliki *layer* (kotak merah dan biru) dengan masing-masing *layer* berisi neuron (lingkaran putih). MLP menerima *input* data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan *output*. Setiap hubungan antar neuron pada dua *layer* yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Di setiap data *input* dan *layer* dilakukan *linear* dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasikan menggunakan operasi *non linear* yang disebut sebagai fungsi aktivasi[6].

Pada *Convolutional Neural Network* (CNN), data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti Gambar 2.4 Dimensi bobot pada CNN adalah :

neuron input x neuron output x tinggi *x* lebar

Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara[6].



Gambar 2.4 Proses Konvolusi Pada CNN

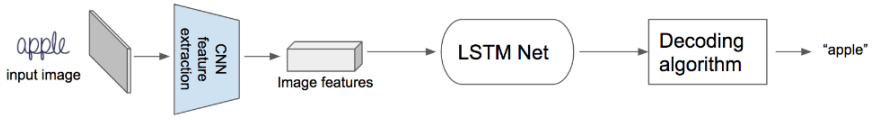
2.4 Machine Learning

Secara definisi, m*achine learning* adalah cabang dari ilmu kecerdasan buatan yang berfokus pada pembangunan dan studi sebuah sistem agar mampu belajar dari data-data yang diperolehnya, Menurut Arthur Samuel, *machine learning* adalah bidang studi yang memberikan kemampuan program komputer untuk belajar tanpa secara eksplisit di program[9].

Untuk bisa mengaplikasikan teknik-teknik *machine learning* maka harus ada data. Tanpa data maka algoritma *machine learning* tidak dapat bekerja. Data yang ada biasanya dibagi menjadi dua, yaitu data *training* dan data *testing* digunakan untuk mengetahui performa algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah di lihat[9].

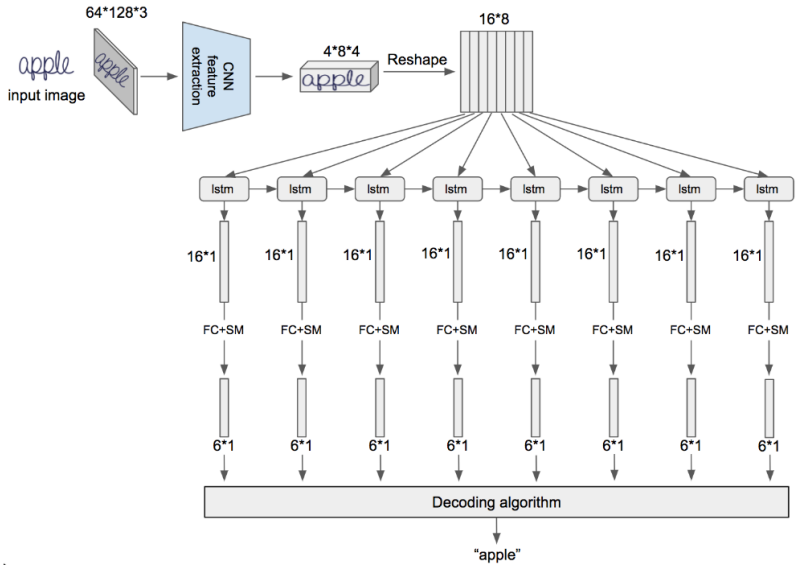
2.4.1 Cara Kerja Machine Learning

Pertama, gambar diinputkan ke CNN untuk mengekstrak fitur gambar. Langkah selanjutnya adalah menerapkan *Recurrent Neural Network* ke fitur-fitur ini, kemudian masuk ke proses algoritma *decoding* khusus. Algoritma ekstraksi ini mengambil output LSTM dari setiap waktu dan menghasilkan pelabelan akhir.



Gambar 2.5 Proses Image Yang Sedang Diekstrak Melalui CNN

Gambar memiliki bentuk berikut: tinggi = 64, lebar = 128, jumlah channel=3. Gambar yang telah melalui proses CNN diubah ke bentuk tensor 4 * 8 * 4. Sehingga sekarang gambar memiliki bentuk tinggi = 4, lebar = 8 (ini adalah dimensi spasial) dan jumlah channel = 4. Disini terjadi perubahan jumlah channel gambar dari 3 menjadi 4.

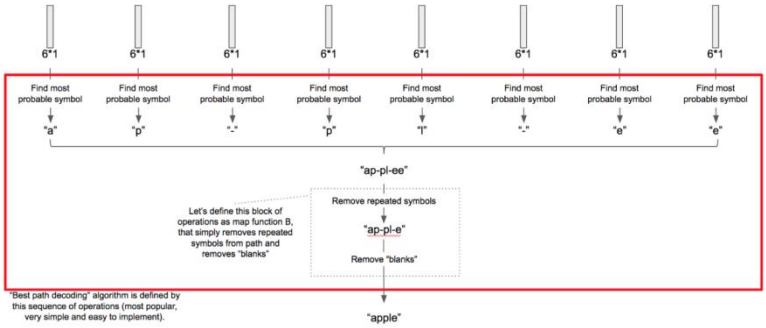


Gambar 2.6 Proses *Image* Oleh CNN

Selanjutnya lakukan operasi *reshape* (pembentukan kembali). Setelah itu diperoleh urutan 8 vektor dari 16 elemen. Setelah itu input 8 vektor ini ke jaringan LSTM dan mendapatkan outputnya - juga vektor dari 16 elemen.

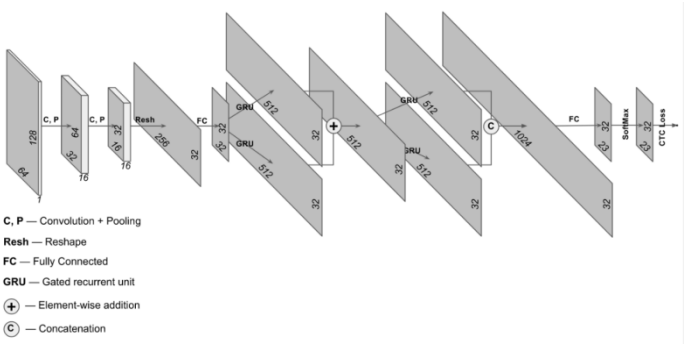
2.4.2 Cara Kerja Algoritma Decoding

Pada gambar 2.6 terdapat 8 vektor probabilitas pada setiap langkah waktu LSTM. Ambil simbol yang paling mungkin pada setiap langkah waktu. Sebagai hasilnya, akan memperoleh string delapan karakter - satu kemungkinan surat yang paling pada setiap langkah waktu. Proses selanjutnya merekatkan semua karakter berulang berurutan menjadi satu. Dalam contoh ini, dua huruf "e" dilekatkan pada yang satu. Karakter kosong khusus memungkinkan untuk memisahkan simbol yang diulang dalam label asli. Ditambahkan simbol kosong ke alfabet untuk mengajarkan jaringan saraf k untuk memprediksi kosong antara simbol-simbol kasus tersebut. Lalu menghapus semua simbol kosong.



Gambar 2.7 Cara Kerja Alogritma Decoding

Ketika menguji coba jaringan ini dengan mengganti algoritma dekode dengan layer CTC Loss. Arsitektur CNN yang kompleks digunakan dalam implementasinya. Arsitekturnya adalah sebagai berikut, tetapi prinsip utamanya masih tetap sama.



Gambar 2.8 Arsitektur CNN

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone.* Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan perantikeras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia[4].

Pada saat perilisan perdana android 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada

perangkat mobile. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangakat seluler[4].

Didunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapatkan dukungan penuh dari Google atau Google *Mail Services* (GMS) dan kedua adalahyang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau dikenal sebagai *Open Handset Alliance* (OHD)[4].

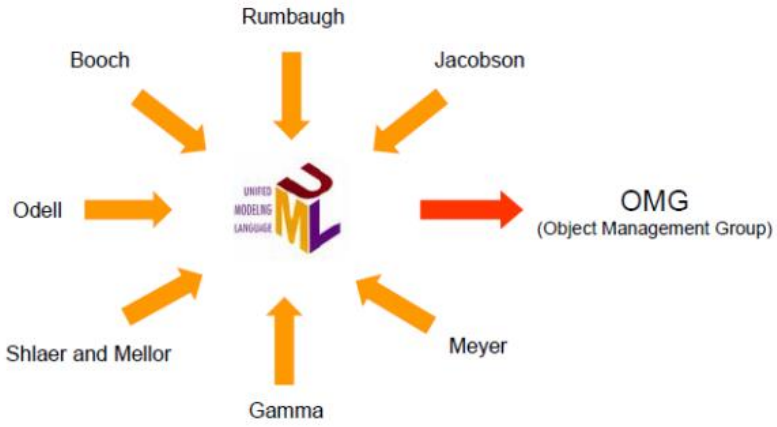
Sekitar September 2007 Google mengenalkan Nexus One, Salah satu jenis smartphone yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia dipasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru[4].

Pada masa saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO,Asus dan masih banyak lagi vendor *smartphone* didunia yang memproduksi Android. Hal ini karena Android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun[4].

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Table Pc. Pesatnya pertumbuhan Android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, Aplikasi dan tool pengembangan, Market aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* didunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada didunia[4].

2.6 Unified Modeling Language

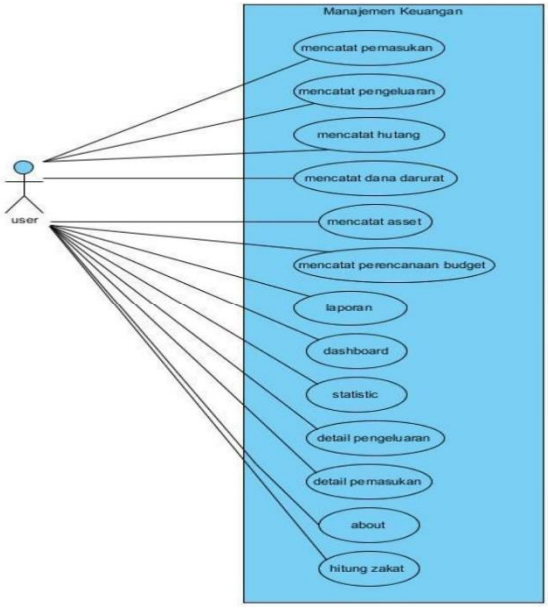
Unified Modeling Language (UML) adalah termasuk kedalam rumpun jenis permodelan notasi grafis yang didukung oleh model-model tunggal. Permodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang di bangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh open Management Group (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka. UML merupakan suatu bahasa permodelan yang tediri banyak model[10].



Gambar 2.9 Contoh UML

2.6.1 Use Case Diagram

Use case diagram merupakan suatu permodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. *Use case* diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, use case diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponen dalam *use case* diagram antaranya[10] :



Gambar 2.10 Contoh *Use Case*

1. Aktor

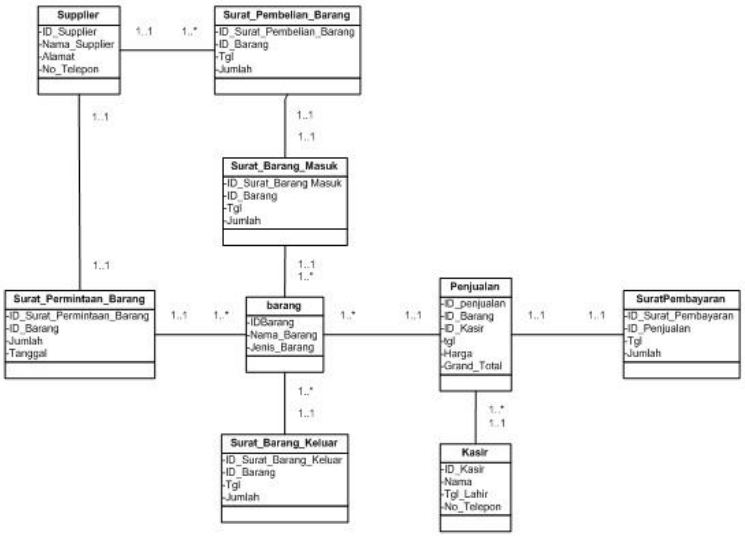
Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berbeda diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Actor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasikan tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut[10].

2. Use Case

Use case merupakan gambaran umum dari fungsi proses utaman yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses sistem yang akan di modelkan. Tidak semua proses digambarkan secara fungsional pada *use case*, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. *Use case* menitik beratkan bagaimana sistem dapat berinteraksi baik antar sistem maupun diluar sistem[10].

2.6.2 Class Diagram

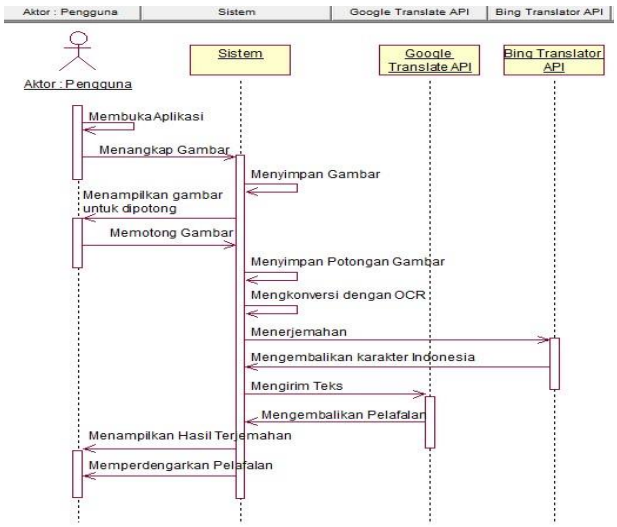
Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapi suatu tujuan. Kelas pada class diagram terdiri dari 3 bagian utama yaitu nama kelas, isi property dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas[10].



Gambar 2.11 Contoh *Class* Diagram

2.6.3 Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case* : interaksi terjadi antar kelas, operasi apa saja yang terlibat, urutan antar operasi dan informasi yang diperlukan oleh masuk-masing operasi[10].



Gambar 2.12 Contoh *Sequence* Diagram

2.7 Perangkat Lunak Pendukung

Berikut adalah beberapa perangkat lunak pendukung untuk membangun aplikasi yang akan dibuat :

2.7.1 Java

Java merupakan bahasa pemgrograman yang disusun oleh James Goslin yang dibantu oleh rekan-rekannya disuatu perusahaan perangakt lunak yang bernama *Sun Microsystems,* pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama "*Oak*", namun pada tahun 1995 diganti namanya menjadi "*Java*"[4].

Java dikembangkan oleh perusahaan *Sun Microsystem. Java* menurut definisi dari *Sun Microsystem* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari *java platform*[4].

Java merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat internet/jaringan komunikasi. Melalui teknologi *java,* dimungkinkan perangakt *audio stereo* dirumah terhubung jaringan komputer. *Java* tidak lagi hanya membuat *applet* yang memperintah halaman *web* tapi *java* telah menjadi bahasa untuk pengembangan aplikasi skala *interprise* berbasis jaringan besar[4].

2.7.2 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan disemua platform (platform-independent). Berikut ini adalah sifat dari *Eclipse*[4].

1. Multi-*platform* : Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris,AIX,HP-UX dan Mac OS X.
2. Multi-*language* : *Eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP dan lain sebagainya.
3. Multi-*role* : Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.