

BAB 2

LANDASAN TEORI

2.1 Studi Literatur

Tabel 2-1 Review Literatur

Review Literatur Ke-1	
Judul Jurnal	A Statistical Approach for Orchid Disease Identification using RGB Color
Penulis	Nurul Hidayah Tuhid, Noor Ezan Abdullah, N.M Khairi*, M.F.Saaid, Shahrizam M.S.B, & Hadzli Hashim Faculty of Electrical Engineering Universiti Teknologi MARA
Volume / Halaman	1 / 1-4
Tahun Penerbitan	2012)
Masalah Utama yang diangkat	Identifikasi daun tanaman anggrek menggunakan metode RGB
Objek Penelitian	Daun Tanaman Anggrek
Hasil Penelitian dan Kesimpulan	Metode rgb dalam digunakan dalam mengklasifikasi penyakit, Terdapat beberapa penyakit yang memiliki kemiripan dan membuat hasil pendeteksian tida akurat
(Paparkan persamaan dengan research)	Persamaan - Orchid Disease Identification -

Review Literatur Ke-2

Judul Jurnal	Classification of Orchid Species using Neural Network
Penulis	Maizura Mohd Sani , Suhaili Beeran Kutty, Hasmila Akmar Omar, Ili Nadia Md Isa Faculty of Electrical Engineering Universiti Teknologi Mara
Volume / Halaman	1 / 1-4
Tahun Penerbitan	2013
Masalah Utama yang diangkat	Pengklasifikasian tanaman anggrek menggunakan neural network Wajah Menggunakan Metode Viola-Jones
Objek Penelitian	Tanaman anggrek
Hasil Penelitian dan Kesimpulan	Metode neural network berhasil mendeteksi anggrek
(Paparkan persamaan dengan research)	Persamaan - Mendeteksi tanaman anggrek

Review Literatur Ke-3	
Judul Jurnal	Identifikasi Citra Daun Teh Menggunakan Metode Histogram untuk Deteksi Dini Serangan Awal Hama Empoasca
Penulis	Christophorus Candra Kusumadewa ¹ , Supatman ² Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta Jl. Wates Km. 10 Yogyakarta 55753, Indonesia
Volume / Halaman	1 / 1-10
Tahun Penerbitan	2018

Masalah Utama yang diangkat	Identifikasi citra daun teh menggunakan metode histogram untuk deteksi dini serangan awal hama empoasca
Objek Penelitian	Daun teh dan hama empoasca
Hasil Penelitian dan Kesimpulan	Penelitian ini berhasil mengimplementasikan metode histogram ke dalam sistem dan memperoleh akurasi sebesar 95,45%
(Paparkan persamaan dengan research)	Persamaan - Deteksi daun

Review Literatur Ke-4	
Judul Jurnal	Orchid Leaf Disease Detection using Border Segmentation Techniques
Penulis	Wan Mohd Fadzil W.M.N*, Shah Rizam M.S.B, R. Jailani, Nooritawati M.T Faculty of Electrical Engineering Universiti Teknologi MARA
Volume / Halaman	1/ 1-6
Tahun Penerbitan	2014
Masalah Utama yang diangkat	Mendeteksi penyakit pada daun anggrek menggunakan metode border segmentation
Objek Penelitian	Penyakit pada daun tanaman anggrek
Hasil Penelitian dan Kesimpulan	Dari hasil pengujian tingkat keberhasilan sistem sebesar 86.36%.
(Paparkan persamaan dengan research)	Persamaan - Mendeteksi penyakit pada daun tanaman anggrek

Review Literatur Ke-5

Judul Jurnal	Digital Image Processing Techniques for Object Detection From Complex Background Image
Penulis	R. Hussin*, M. Rizon Juhari, Ng Wei Kang, R.C.Ismail, A.Kamarudin School of Microelectronic Engineering, University Malaysia Perlis (UniMAP)
Volume / Halaman	1/ 1-5
Tahun Penerbitan	2012
Masalah Utama yang diangkat	Mendeteksi objek menggunakan metode image processing
Objek Penelitian	Buah mangga
Hasil Penelitian dan Kesimpulan	Sitem berhasil mendeteksi buah mangga
(Paparkan persamaan dengan research)	Persamaan - Metode image processing

2.2 Landasan Teori

Landasan Teori adalah kumpulan berbagai informasi informasi yang bisa membantu dalam penelitian.

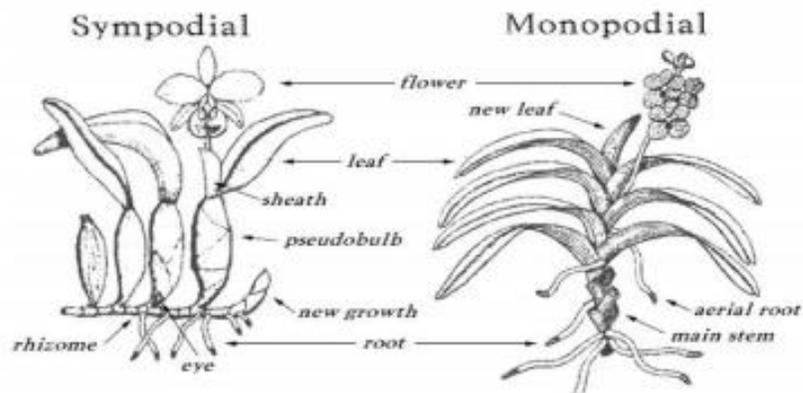
2.2.1 Pengertian Tanaman Anggrek

Anggrek (Orchidaceae) merupakan tanaman yang mudah beradaptasi dengan lingkungannya [4]. Tak heran, apabila tanaman anggrek bisa dijumpai hampir di seluruh bagian di dunia. Lokasi tumbuh mereka juga bisa sangat beragam. Mulai dari daerah dataran rendah hingga dataran tinggi, mulai dari kawasan yang bersuhu dingin hingga bersuhu panas.

2.2.2 Tanaman Anggrek

Tanaman Anggrek umumnya dapat dibedakan menjadi dua tipe berdasarkan pada pola pertumbuhannya, yaitu tipe simpodial dan tipe monopodial (Gambar 2.1).

1. Anggrek tipe simpodial yaitu anggrek yang tidak memiliki batang utama, bunga akan keluar dari ujung batang dan biasanya bunga akan kembali tumbuh pada anakan atau tunas. Pertumbuhan batang tipe simpodial yaitu *Dendrobium* sp. Dan *Cattleya* sp.
2. Anggrek Tipe monopodial yaitu anggrek yang memiliki batang dengan titik tumbuh yang ada diujungnya, sehingga pertumbuhannya akan lurus dan bunga akan keluar dari sisi batang di antara kedua sisi daun. Contoh anggrek tipe monopodial ialah *Aranthera* sp. Dan *Phalaenopsis* sp.



Gambar 2.1 Tipe anggrek simpodial dan monopodial (Widiastoety et al., 2010)

2.2.3 Struktur Tanaman Anggrek

Bunga Anggrek termasuk kedalam bunga yang berkelamin dua atau hermaprodit dan tersusun kedalam karangan bunga dengan jumlah satu hingga banyak kuantum. Bunga anggrek terdiri dari lima bagian utama yaitu :

1. Daun Mahkota (petal)
2. Daun kelopak (sepal)
3. Benang sari (stamen)

4. Putik (pistil)
5. Bakal Buah (ovari)

Daun Kelopak kerap kali berwarna serupa daun mahkota, tetapi mudah dibedakan. Daun kelopak atau sepal berjumlah 3 kelopak, yakni bagian atas atau dinamakan *sepalum dorsale*. Sedangkan dua lainnya dinamakan *sepalum laterale* (Gambar 2.2). Jumlah daun mahkota (petal) berjumlah tiga buah yakni dua buah sama persis dan terpisah, sedangkan yang paling bawah termodifikasi menjadi bibir atau *labellum*. Labellum umumnya dijadikan sebagai karakter pembeda antara satu jenis anggrek dengan jenis anggrek yang lain



Gambar 2.2 Struktur bunga anggrek *Dendrobium sonia* (Abdullakasim et al., 2015) Keterangan: (a) *sepalum dorsale*, (b) *Petal*, (c) *collumna*, (d) *sepalum laterale*, (e) *labellum*

Anggrek memiliki bentuk daun yang sangat beragam antara lain yakni agak bulat, lonjong hingga lanset. Selain itu, daun anggrek juga mempunyai ketebalan yang berbeda tergantung jenisnya, yaitu ada yang memiliki daun tipis hingga yang berdaging dengan tekstur yang rata dan sedikit kaku. Anggrek memiliki posisi tulang daun sejajar, tidak bertangkai dan menyebar seperti tanaman monokotil lainnya. Susunan daun kerap kali berhadapan dan berselangseling [5].

2.2.4 Data

Data merupakan sekumpulan informasi atau juga keterangan - keterangan dari suatu hal yang diperoleh dengan melalui pengamatan atau juga pencarian ke sumber sumber tertentu. Data yang diperoleh tersebut dapat menjadi sebuah anggapan atau fakta disebabkan karena memang belum diolah dengan lebih lanjut. Setelah diolah dengan melalui penelitian atau percobaan maka suatu data tersebut dapat menjadi bentuk yang lebih kompleks seperti suatu database, informasi atau juga bahkan solusi untuk menyelesaikan masalah tertentu

2.2.5 Informasi

Pengertian Informasi adalah sekumpulan data atau fakta yang telah diproses dan diolah sedemikian rupa sehingga menghasilkan sesuatu yang bisa dipahami dan memberikan manfaat bagi penerimanya. Data dan fakta adalah “bahan baku” informasi, tetapi tidak semuanya bisa diolah menjadi informasi. Istilah “informasi” berasal dari bahasa Perancis kuno, “informacion,” yang mengambil dari bahasa Latin, *informare* yang artinya “aktivitas dalam pengetahuan yang dikomunikasikan”. Informasi bisa dikatakan sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi

2.2.6 Image Processing

Citra adalah kombinasi antara titik, garis, bidang, dan warna untuk menciptakan suatu imitasi dari suatu objek—biasanya objek fisik atau manusia. Citra bisa berwujud gambar (picture) dua dimensi, seperti lukisan, foto, dan berwujud tiga dimensi, seperti patung. Pengolahan Citra atau image processing adalah proses bagaimana komputer mengolah citra yang berupa data data yang nantinya bisa membuat sebuah objek berbentuk yang umumnya dua dimensi. Salah satu tahapan dalam image processing adalah tahapan grey scaling atau tahapan mengubah citra menjadi abu abu [6]

2.2.7 Javascript

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis [7]. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Google Chrome, Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT. JavaScript merupakan salah satu teknologi inti World Wide Web selain HTML dan CSS. JavaScript membantu membuat halaman web interaktif dan merupakan bagian aplikasi web yang esensial[7].

2.2.8 Tensorflow JS

Tensorflow.js adalah antarmuka tingkat tinggi open source yang ramah untuk library untuk menangani operasi matematika yang dipercepat GPU dan manajemen memori untuk algoritme pembelajaran mesin[8].

2.2.9 JSON

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data[9].

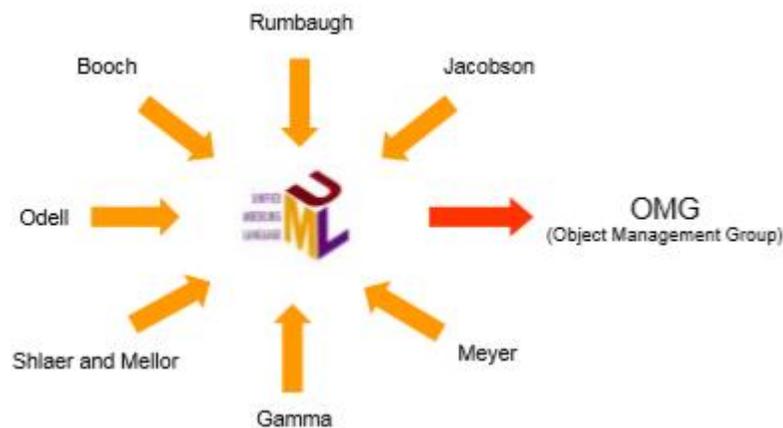
Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*. Daftar nilai terurutkan (*an ordered list of values*). Dalam kebanyakan bahasa ini direpresentasikan sebagai array, vektor, daftar atau string[9].

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini[9].

2.2.10 Unified Modeling Language

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering) [10].

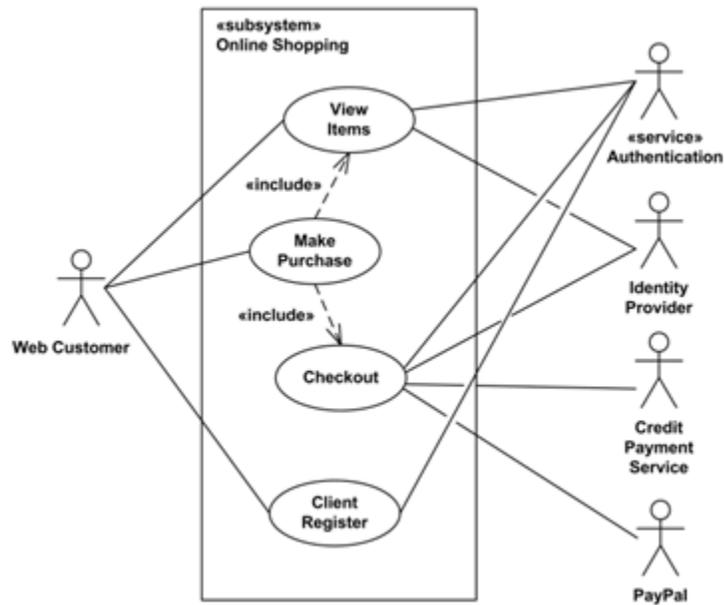
Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch , metodologi coad , metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan



Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999 . Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek[10].

2.2.11 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu[11].



Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common[11].

2.2.12 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi)[10].

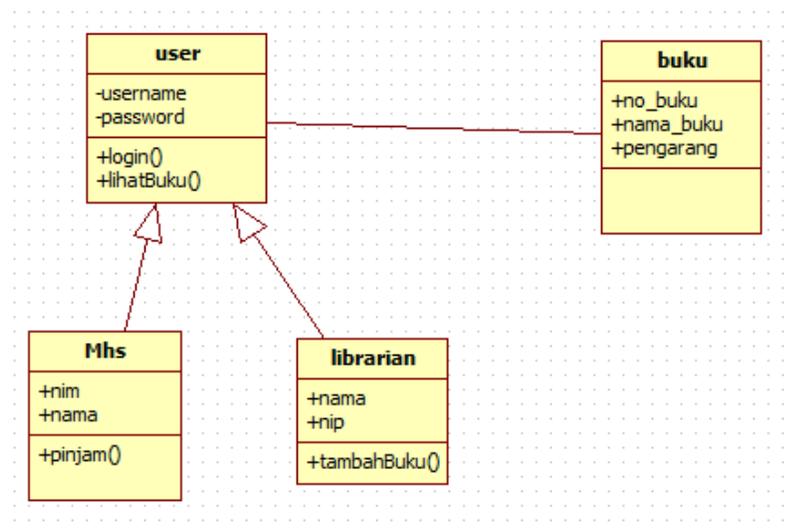
Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja

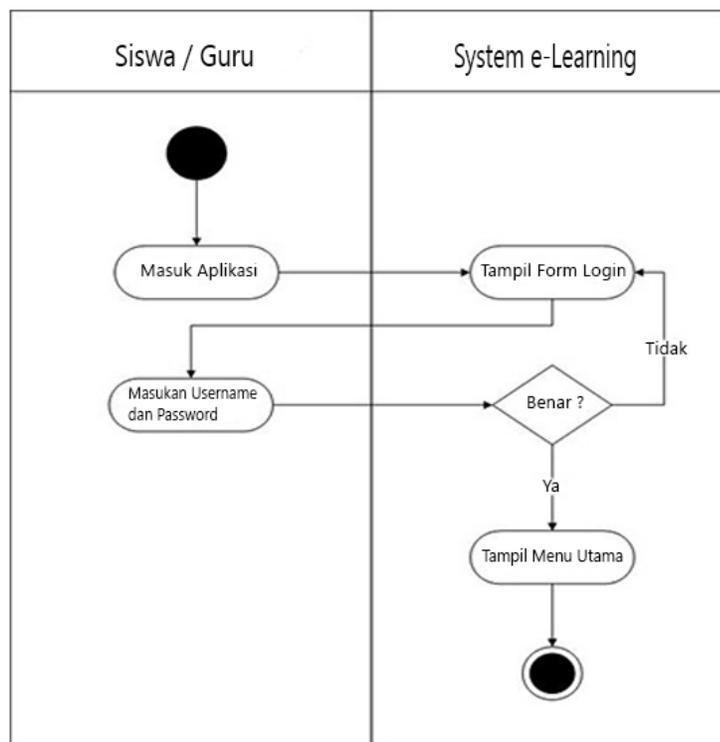


2.2.13 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin

terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[12].

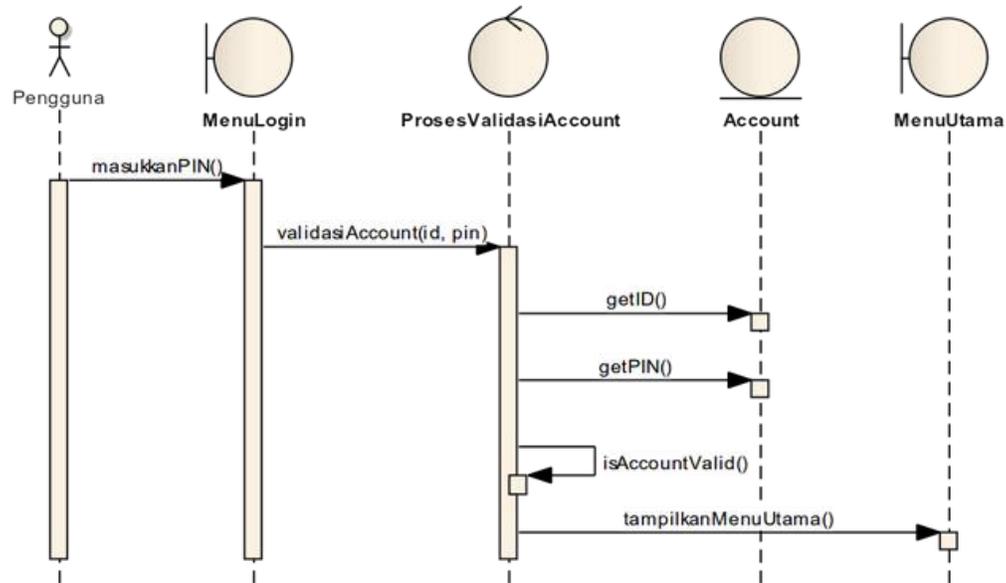
Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.



Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

2.2.14 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message[10].



2.2.15 Machine Learning

Machine Learning (ML) telah menjadi alat yang sangat diperlukan dalam sistem perangkat lunak, meningkatkan aplikasi yang sudah ada dan memungkinkan pembuatan aplikasi yang baru. Namun, platform perangkat lunak yang tersedia untuk pembelajaran mesin mencerminkan akar akademis dan industri dari teknologi tersebut. Pustaka pembelajaran mesin berkualitas produksi biasanya ditulis untuk pengembang Python dan C++. Namun, ada komunitas JavaScript front-end dan back-end (JS) besar yang berkembang pesat. Pada tahun 2017, JS memiliki 2,3 juta permintaan unduhan GitHub, dibandingkan dengan 1 juta di Python (GitHub.com, 2017)[8].

Machine Learning menjawab pertanyaan tentang komputer yang secara otomatis membuat mesin belajar dari pengalaman. Ini adalah salah satu bidang teknis yang tumbuh paling cepat, yang berada di persimpangan ilmu komputer dan statistik, dan merupakan inti dari kecerdasan buatan dan ilmu data. Kemajuan terbaru dalam mesin pembelajaran didorong oleh pengembangan algoritma dan teori pembelajaran baru dan ledakan berkelanjutan dalam ketersediaan data online dan tokoh-tokoh yang menguntungkan. Persetujuan metode pembelajaran intensif data dapat ditemukan di semua sains, teknologi dan perdagangan, yang mengarah pada pengambilan keputusan

berbasis bukti di berbagai perusahaan, termasuk perawatan kesehatan, produksi, pendidikan, pemodelan keuangan, polisi dan pemasaran[13].

