

BAB II

LANDASAN TEORI

2.1 Plagiarisme

Menurut (Feri Sulianta, 2007) di dalam bukunya yang berjudul “Konten Internet”. Plagiarisme adalah bentuk penyalahgunaan hak kekayaan intelektual milik orang lain, yang mana karya tersebut dipresentasikan dan diakui secara tidak sah sebagai hasil karya pribadi.

2.2 Dokumen

Menurut (Dr. Ghansham Anand, S.H., M.Kn. , 2018) di dalam bukunya yang berjudul “Karakteristik Jabatan Notaris di Indonesia”. Dokumen menurut **Louis Gottschalk** istilah dokumen seringkali digunakan para ahli dalam dua pengertian, yaitu pertama, berarti sumber tertulis dan arkeologis, kedua diperuntukan bagi surat – surat resmi dan surat – surat negara seperti surat perjanjian, undang – undang, dan sebagainya.

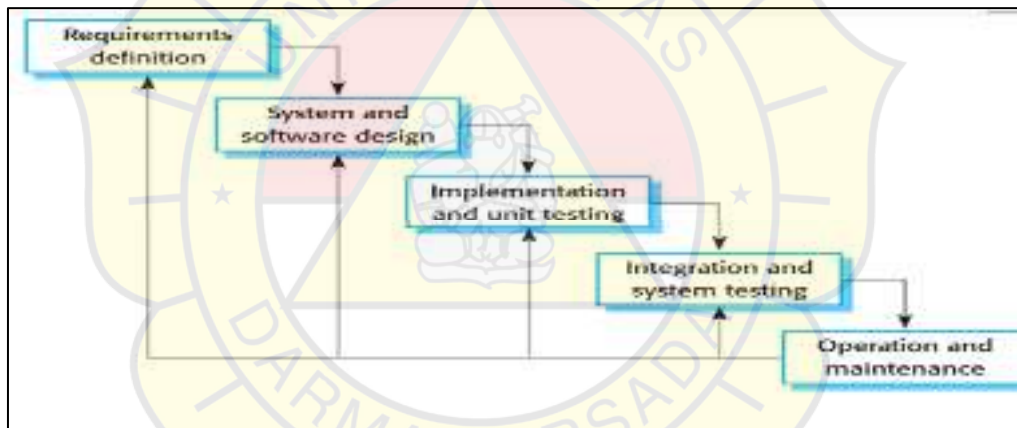
2.3 Skripsi

Menurut (Muliadi Anangkota, 2018) di dalam bukunya yang berjudul “Jangan Takut Menulis Skripsi”. Skripsi merupakan karya tulis ilmiah berdasarkan hasil penelitian sarjana (S1) yang membahas suatu permasalahan/fenomena dalam bidang ilmu tertentu. Penelitian adalah keseluruhan kegiatan baik di dalam pikiran maupun

dalam kegiatan nyata yang dilakukan oleh mahasiswa untuk menyelesaikan suatu masalah di bidang ilmu pengetahuan ilmiah dalam rangka penyusunan skripsi.

2.4 Model *Waterfall*

Menurut (Ian Sommerville, 2011) dalam buku "*Software Engineering, 9th Edition*". Metodologi *waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian.



Gambar 2.1 Fase Metode *Waterfall* (Ian Sommerville, 2011)

Metode Waterfall memiliki tahapan-tahapan sebagai berikut :

1. *Requirements analysis and definition* Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design* Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.
3. *Implementation and unit testing* Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.
4. *Integration and system testing* Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer

Operation and maintenance Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata.

2.5 Winnowing

Menurut (Diana Purwitasari, Putu Yuwono Kusmawan, Umi Laili Yuhana, 2011) di dalam jurnalnya yang berjudul “Deteksi Keberadaan Kalimat Sama Sebagai Indikasi Penjiplakan dengan Algoritma Hashing Berbasis N-Gram”. Ada banyak metode yang bisa digunakan dalam mendeteksi penjiplakan dalam file teks dengan mengenali kalimat – kalimat yang mirip. Namun ada kebutuhan mendasar yang harus dipenuhi oleh algoritma pendeteksi tersebut yaitu :

2.5.1 *Whitespace Insensitivity*

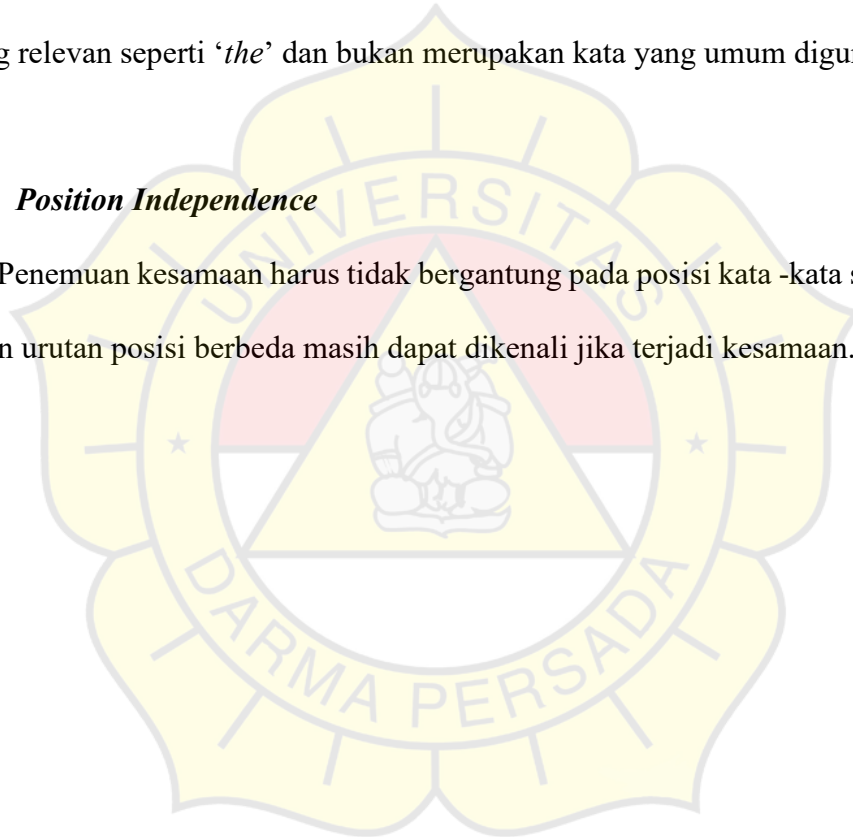
Pencarian kalimat mirip seharusnya tidak terpengaruh oleh spasi, jenis huruf (kapital atau normal), tanda baca, karakter – karakter unik, dan sebagainya.

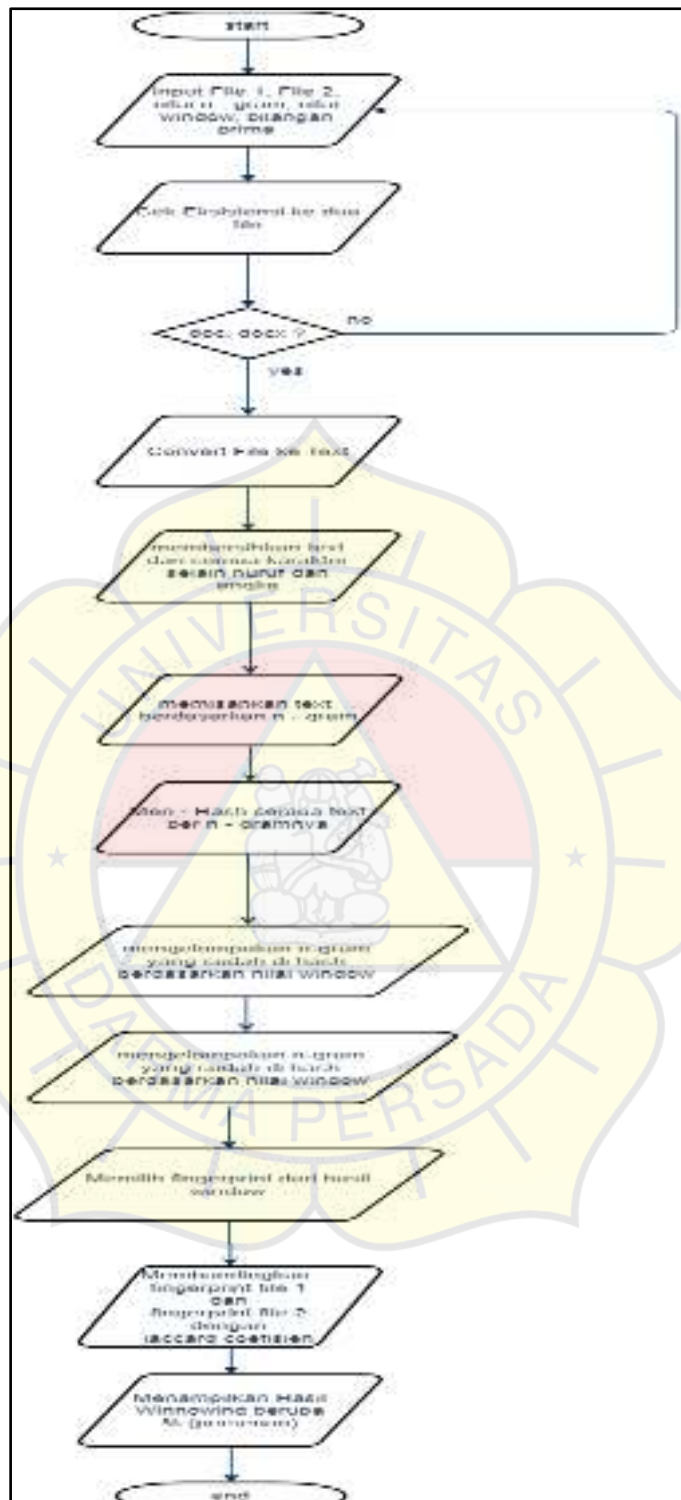
2.5.2 *Noise Supression*

Menghindari penemuan kecocokan dengan panjang kata yang terlalu kecil atau kurang relevan seperti '*the*' dan bukan merupakan kata yang umum digunakan.

2.5.3 *Position Independence*

Penemuan kesamaan harus tidak bergantung pada posisi kata -kata sehingga kata dengan urutan posisi berbeda masih dapat dikenali jika terjadi kesamaan.





Gambar 2.2 Flowchart Winnowing

2.6 Aplikasi dan Pemrograman *Web*

2.6.1 Internet, Aplikasi *Web*, dan Pemrograman *Web*

2.6.1.1 Internet

Menurut (Sandra Weber, 2004) dalam buku elektroniknya yang berjudul “*The Internet*”. Internet adalah koneksi ribuan jaringan di seluruh dunia yang menghubungkan jutaan komputer.

2.6.1.2 Aplikasi *Web*

Menurut (Janner Simarmata, 2010) dalam buku elektroniknya yang berjudul “*Rekayasa WEB*”. Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis Web.

2.6.1.3 Pemrograman *Web*

Menurut (Rintho Rante Rerung, 2018) dalam buku elektroniknya yang berjudul “*Pemrograman Web Dasar*”. Pemrograman Web diambil dari 2 kata yaitu pemrograman dan web. Pemrograman yang dalam bahasa *English* adalah *programming* dan diartikan proses, cara, perbuatan program. Sedangkan definisi web adalah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *hypertext transfer protokol*.

2.6.2 Kelebihan dan Kelemahan *Web*

2.6.2.1 Kelebihan

Menurut (Rintho Rante Rerung, 2018) dalam buku elektroniknya yang berjudul “*Pemrograman Web Dasar*”. Beberapa Kelebihan menggunakan web yaitu

Akses Informasi yang lebih mudah, *setup server* lebih mudah, informasi mudah didistribusikan, dan bebas *platform*, informasi dapat di sajikan oleh *browser* web pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

2.6.2.2 Kelemahan

Menurut (Erick Kurniawan, 2010) dalam bukunya yang berjudul “Cepat Mahir ASP.NET 3.5 Untuk Aplikasi Web Interaktif”. Salah satu kelemahan aplikasi berbasis web adalah sifatnya yang kurang responsif dibandingkan dengan aplikasi desktop, misalnya ketika hendak mengklik sebuah link pada website, maka pengguna harus menunggu beberapa detik hingga muncul tampilan halaman yang dikehendaki (proses *refresh* halaman).

2.6.3 Tools Pemrograman Web

2.6.3.1 HTML

Menurut (Taryana Suryana & Koesheryatin, 2014) dalam bukunya yang berjudul “Aplikasi Internet Menggunakan HTML, CSS, Javascript”. *HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman web. HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu *Standard Generalized Markup Language* (SGML). HTML pada dasarnya merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN, dan dipopulerkan pertama kali oleh *browser* Mosaic. Selama awal tahun 1990, HTML mengalami perkembangan yang sangat pesat. Setiap pengembangan HTML, Pasti akan menambahkan kemampuan dan fasilitas yang lebih baik dari versi sebelumnya.

2.6.3.2 CSS

Menurut (Jubilee Enterprise, 2016) dalam bukunya yang berjudul “Pengenalan HTML dan CSS Referensi langkah demi langkah memahami HTML dan CSS untuk desain website”. Secara definitif, CSS adalah “kumpulan kode untuk mendefinisikan desain dari bahasa markup”. Karena ada kata bahasa markup pada definisi CSS, maka relasi antara CSS dan HTML sangatlah dekat. Dengan CSS-lah, sebuah desain website yang dibangun menggunakan HTML akan menjadi lebih menarik dan variatif.

Jika didefinisikan secara bebas, CSS merupakan kumpulan kode untuk mendesain atau mempercantik tampilan halaman website. Dengan arti lain, dengan memanfaatkan CSS kita bisa mengubah desain standar yang dihasilkan oleh HTML menjadi variasi – variasi yang lebih kompleks. Sebagai contoh, kita bisa mewarnai kotak teks pada sebuah form dimana jika hanya mengandalkan HTML, hal ini tidak bisa terwujud.

2.6.3.3 Javascript

Menurut (Menurut Vivian Siahian dan Rismon Hasiholan Sianipar, 2018) dalam bukunya yang berjudul “JAVASCRIPT Dari A Sampai Z”. Javascript

merupakan bahasa skript populer yang dipakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. Javascript merupakan perekat yang menyatukan halaman – halaman web. Akan sangat susah menjumpai halaman web komersial yang tidak membuat kode Javascript.

2.6.3.4 PHP

Menurut (ANHAR, 2010) dalam bukunya yang berjudul “Panduan Menguasai PHP & MySQL secara Otodidak”. PHP singkatan dari PHP: Hypertext Preprocessor yaitu bahasa pemrograman web server-side yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan.

2.6.3.5 PostgreSQL

Menurut (ANHAR, 2010) dalam bukunya yang berjudul “Panduan Menguasai PHP & MySQL secara Otodidak”. PostgreSQL adalah *Object – Relational Database Management System* (ORDBMS) berdasarkan POSTRES, *Version 4.2*, yang di develop di *University of California* di *Computer Science Departement* di Berkeley. POSTGRES yang merintis banyak konsep yang nantinya akan di terapkan di beberapa *database*

sistem komersial. PostgreSQL adalah turunan *open source* dari kode original Berkeley. PostgreSQL mendukung sebagian besar aturan standar SQL dan menawarkan banyak fitur – fitur modern dan dikernakan lisensi liberalnya PostgreSQL bisa digunakan, merubah, dan didistribusikan oleh siapapun secara gratis baik untuk di gunakan sendiri, untuk komersial, ataupun untuk belajar mengajar.

2.6.4 Web Server, Browser, dan Text Editor

2.6.4.1 Web Server

Menurut (Rohi Abdullah, 2018) dalam buku elektroniknya yang berjudul “7 in 1 Pemrograman Web Untuk Pemula”. *Web Server* merupakan perangkat lunak yang berfungsi untuk menerima permintaan (*request*) melalui protokol HTTP atau HTTPS dari *client* kemudian mengirimkan kembali dalam bentuk halaman – halaman web. Contoh yang termasuk *web server* adalah Apache.

2.6.4.2 Browser

Menurut (Rohi Abdullah, 2018) dalam buku elektroniknya yang berjudul “7 in 1 Pemrograman Web Untuk Pemula”. *Browser* adalah software atau perangkat lunak yang digunakan untuk menampilkan dan mengetes hasil program. Contoh dari *browser* adalah Firefox, Chrome, dan lain – lain.

2.6.4.3 Text Editor

Menurut (Rohi Abdullah, 2018) dalam buku elektroniknya yang berjudul “7 in 1 Pemrograman Web Untuk Pemula”. *Text Editor* memiliki fungsi yang sama

dengan web *authoring* yaitu untuk mengetikkan skrip program, hanya saja tidak bersifat *WYSIWYG* (*What You See What You Get*), sehingga untuk menampilkan hasilnya memerlukan *web browser*. Contoh dari *text editor* yaitu seperti Visual Studio Code atau Sublime Text.

2.7 UML

Menurut (Sinan Si Alhir, 2003) dalam bukunya yang berjudul “*Learning UML*”. UML (*Unified Modeling Language*) adalah bahasa *visual* untuk memodelkan sistem atau komunikasi tentang sistem menggunakan diagram dan teks.

2.7.1 *Use Case Diagram*

Menurut (Muhamad Muslihudin Oktafianto, 2016) dalam bukunya yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”. *Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya.

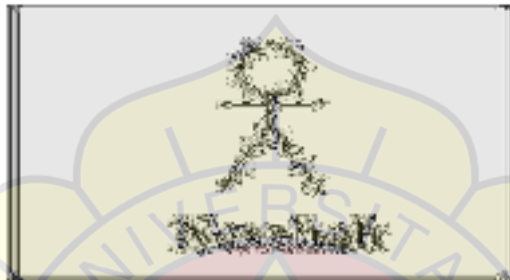


Gambar 2.3 Contoh *Use Case Diagram* (Muslihudin, 2016)

Komponen pembentukan *use case diagram* adalah :

2.7.1.1 Aktor

Menurut (Muhamad Muslihudin Oktafianto, 2016) dalam bukunya yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”. Aktor (*Actor*), menggambarkan pihak – pihak yang berperan dalam sistem.



Gambar 2.4 Contoh Aktor (Muslihudin, 2016)

2.7.1.2 Use Case

Menurut (Muhamad Muslihudin Oktafianto, 2016) dalam bukunya yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”. *Use case* menggambarkan aktivitas/sarana yang disiapkan oleh bisnis/sistem. *Use case* digambarkan dalam bentuk elips/oval.



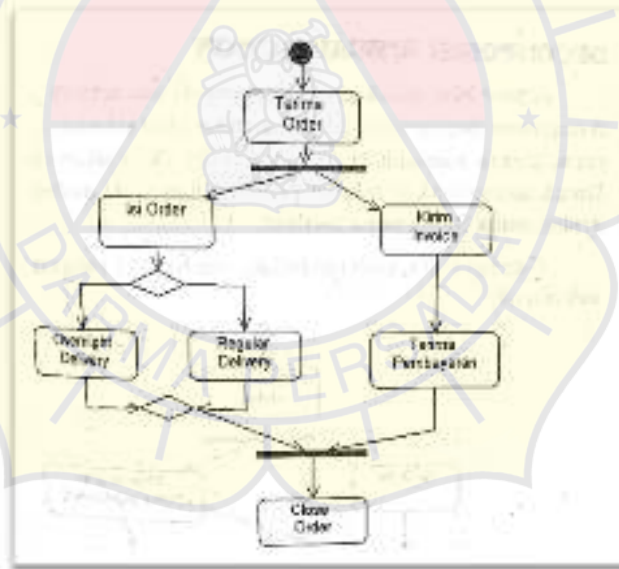
Gambar 2.5 Contoh *Use Case* (Muslihudin, 2016)

2.7.1.3 Hubungan (*Link*)

Menurut (Muhamad Muslihudin Oktafianto, 2016) dalam bukunya yang berjudul “Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML”. Hubungan (*Link*) merupakan sebuah penghubung dari aktor mana saja yang terlibat dalam *use case*.

2.7.2 *Activity Diagram*

Menurut (Munawar, 2005) dalam bukunya yang berjudul “Pemodelan Visual dengan UML”. *Activity* diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus.



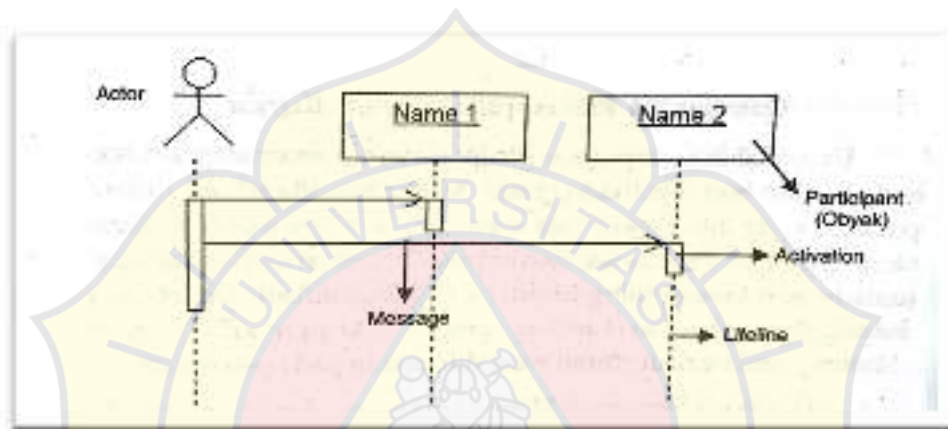
Gambar 2.6 Contoh *Activity Diagram* (Munawar, 2005)

2.7.3 *Sequence Diagram*

Menurut (Munawar, 2005) dalam bukunya yang berjudul “Pemodelan Visual dengan UML”. *Sequence* diagram digunakan untuk menggambarkan perilaku pada

sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakan diantara obyek – obyek ini di dalam *use case*.

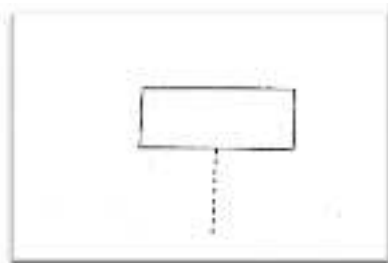
Komponen utama *sequence* diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.



Gambar 2.7 Contoh *sequence* diagram (Munawar, 2005)

2.7.3.1 Obyek/*Participant*

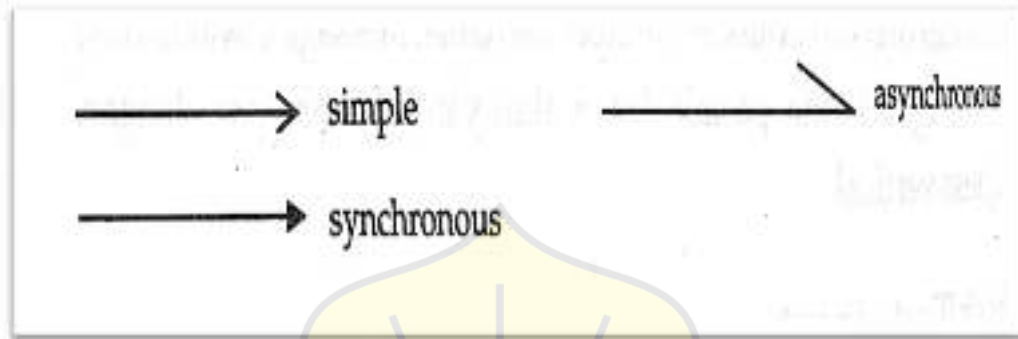
Obyek diletakan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram.



Gambar 2.8 Contoh *participant* atau obyek di *sequence* diagram (Muhawar, 2005)

2.7.3.2 Message

Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline* yang lain.



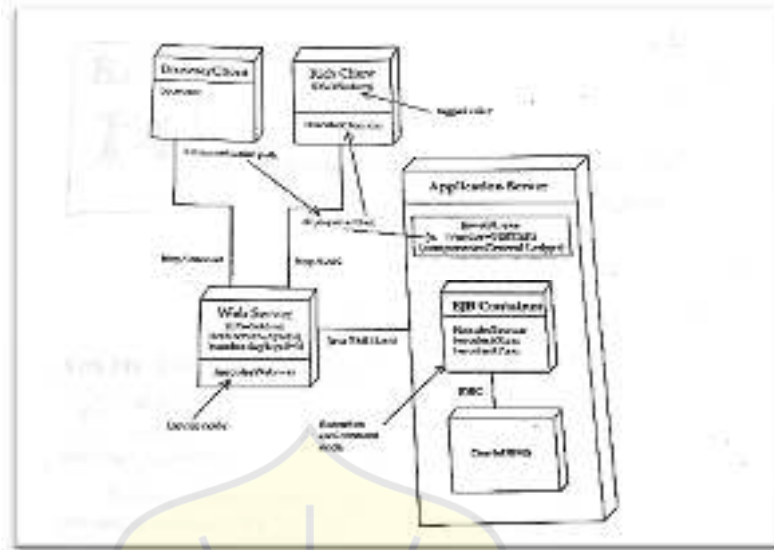
Gambar 2.9 Simbol – simbol *message* pada *sequence* diagram (Munawar, 2005)

2.7.3.3 Time

Time adalah diagram yang mewakili waktu pada arah *vertikal*. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat ke bawah.

2.7.4 Deployment Diagram

Menurut (Munawar, 2005) dalam bukunya yang berjudul “Pemodelan Visual dengan UML”. *Deployment* Diagram menunjukkan tata letak sebuah sistem secara fisik, menampilkan bagian – bagian software yang berjalan pada bagian – bagian hardware.



Gambar 2.10 Contoh *deployment* diagram (Munawar, 2005)

