



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB 2

LANDASAN TEORI

2.1 Teori Pengumpulan Data

2.1.1 Studi Lapangan

Pada studi lapangan di bagi menjadi 2 cara yaitu:

1. Observasi

Observasi atau pengamatan (*observation*) merupakan salah satu teknik pengumpulan fakta/data yang cukup efektif untuk mempelajari suatu sistem. Observasi adalah pengamatan langsung suatu kejadian yang sedang dilakukan. (Jogiyanto, 2005).

2. Wawancara (*interview*)

Wawancara (*interview*) telah diakui sebagai teknik pengumpulan data atau fakta yang penting dan banyak dilakukan dalam pengembangan sistem informasi. Wawancara memungkinkan analisis sistem sebagai pewawancara untuk mengumpulkan data secara tatap muka langsung dengan orang yang diwawancarainya. (Jogiyanto, 2005).

2.1.2 Studi Literatur

Studi literatur adalah mencari suatu masalah untuk diteliti. Dalam arti bukti-bukti atau pernyataan bahwa masalah yang akan diteliti itu belum terjawab atau belum terjawab atau belum terpecahkan secara memuaskan atau belum pernah diteliti orang mengenai tujuan, data dan metode analisa dan hasil untuk waktu dan tempat yang sama (Bintarto, 2002). Dalam mencari studi literature, peneliti mencari permasalahan yang sama dalam pengembangan sistem penjadwalan.

2.1.3 Internet

Menurut (Kamus Besar Bahasa Indonesia) internet dikenal sebagai jaringan komunikasi elektronik yang menghubungkan jaringan komputer dengan fasilitas komputer lainnya yang terintegrasi di seluruh dunia melalui telepon maupun satelit internet.

Internet sendiri memberikan beberapa jasa umum seperti aplikasi *World Wide Web (WWW)*, *e-mail*, *newsgroup*, dan lain-lain. Jasa internet yang paling sering digunakan saat ini adalah penggunaan aplikasi WWW untuk pencarian data maupun informasi yang dibutuhkan untuk tugas, pekerjaan, maupun pengetahuan serta *e-mail* yang digunakan untuk pengiriman file maupun *chatting* antar penggunanya. Selain kedua hal tersebut, adapula file sharing dimana pihak-pihak pengguna internet saling mengirimkan data terhadap satu sama lain. Dalam hal pengiriman data perusahaan yang bersifat pribadi, umumnya memerlukan *encryption*

sebagai kode - kode biner yang menyulitkan pihak tidak bertanggungjawab yang ingin menyalahgunakan data perusahaan tersebut.

2.2 Teori Pengolahan Data

2.2.1 PHP : *Hypertext Preprocessor* (PHP)

Berdasarkan (w3schools.com) PHP merupakan sebuah bahasa pemrograman dari sisi server yang digunakan untuk perancangan web yang dinamis serta interaktif, yang digunakan secara bebas, serta sebagai bahasa pemrograman yang efisien bila dihadapkan terhadap pesaing seperti Microsoft ASP.

Kemudian menurut (Oracle, 1997, 2016) PHP juga dapat digunakan untuk membuat halaman web yang dinamis, serta bersifat fleksibel untuk digunakan untuk rata-rata sistem operasi maupun *web server*. PHP dapat digunakan pula sebagai kunci pengakses *database*, dan salah satunya adalah MySQL.



Gambar 2.1 PHP's Logo

2.2.2 *HyperText Markup Language 5* (HTML5)

Menurut (Andreas Johan, 2013) *HyperText Markup Language* (HTML) merupakan standar yang digunakan secara luas dalam menampilkan halaman web. Hingga saat ini, HTML digunakan sebagai standar internet yang diartikan maupun dikendalikan penggunaannya melalui *World Wide Web*. Tanpa adanya HTML, internet masih dalam keadaan berbasis *text*, dimana sebuah halaman web hanya berisikan format *text* datar tanpa adanya format dokumen berbasis visual secara menarik.

Pengertian HTML sendiri bila dijabarkan berdasarkan kata-kata penyusunannya, dapat dijabarkan lebih dalam lagi menjadi:

- *Hypertext*

Link hypertext merupakan kata yang dapat menunjukkan relasi suatu naskah dokumen dengan naskah lainnya. Bila kita mengklik pada kata untuk mengikuti link tersebut, maka sebuah *web browser* akan menampilkan bagian dari dokumen yang dikehendaki.

- *Markup*

Sesuai definisinya, *markup* menunjukkan bahwa sebuah file HTML memiliki suatu intruksi tertentu yang dapat memberikan suatu format pada dokumen yang akan ditampilkan pada *World Wide Web*.

- *Language*

HTML dapat berupa sebuah kumpulan dari beberapa intruksi yang dapat digunakan untuk mengubah-ubah format suatu dokumen.

HTML sendiri sudah memiliki banyak versi. Versi terbaru dari HTML saat ini adalah HTML5. HTML5 merupakan versi terbaru teknologi

hypertext yang saat ini sedang dalam tahap *development*. Berikut adalah beberapa tujuan dibutanya HTML5:

- Fitur baru harus didasarkan pada HTML, CSS, DOM , dan JavaScript.
- Mengurangi kebutuhan *plugin* eksternal (contoh: Adobe Flash).
- Penanganan kesalahan yang lebih baik.
- Lebih banyak *markup* untuk menggantikan scripting.
- HTML5 merupakan perangkat mandiri.

Saat ini HTML5 sudah mulai diterapkan pada beberapa situs besar sebagai *test* kelayakan fitur serta teknologi baru yang masih terus dikembangkan dari versi sebelumnya yaitu HTML 4.01. Fitur serta perbaikan yang ditambahkan dalam teknologi ini diantaranya adalah video dan audio, *database* lokal dan CSS3. Berikut ini adalah beberapa *browser* yang dapat mendukung HTML5 yaitu:

- Opera Web Browser
- Safari
- FireFox
- Google Chrome
- Internet Explorer

2.3 Konsep Dasar Sistem

Menurut (Jogiyanto, 2005), sistem adalah kumpulan dari komponen atau elemen yang saling berhubungan satu dengan lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Selain itu (Sutabri, 2004) menyimpulkan sistem adalah sekelompok unsur yang erat hubungannya

satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

2.4 Konsep Dasar Informasi

Menurut Davis dalam (Ladjamudin, 2005) informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Sedangkan menurut McLeod dalam (Ladjamudin, 2005) informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya. Sumber informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu. Informasi diperoleh setelah data-data mentah diproses atau diolah.

2.5 Pengertian Sistem Informasi

Suatu sistem dapat dikatakan baik jika mampu menyediakan informasi yang bermanfaat bagi penggunanya. Menurut (O'Brien, 2010) Sistem Informasi adalah kombinasi dari unit setiap orang, *hardware* (perangkat keras), *software* (perangkat lunak), jaringan komputer dan jaringan komunikasi data (komunikasi), dan *database* (basis data) yang terkumpul, berubah, dan menyebarkan informasi dalam sebuah organisasi.

Menurut (Sutarman, 2012) Sistem Informasi adalah sistem yang dapat didefinisikan dengan proses pengumpulan, pemrosesan, penyimpanan, penganalisis, penyebaran informasi untuk tujuan tertentu.

Jadi pengertian Sistem Informasi menurut penulis adalah rangkaian suatu sistem terdiri dari sumber daya, *hardware*, *software*, perangkat, jaringan, dan lain-lain yang saling bekerja sama untuk untuk mencapai tujuan organisasi.

2.6 Pengembangan Sistem

2.6.1 Pengertian Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Menurut (Sutabri, 2004) sistem yang lama perlu diperbaiki atau diganti, disebabkan karena beberapa hal, yaitu sebagai berikut:

1. Adanya permasalahan-permasalahan yang timbul di sistem yang lama seperti: adanya ketidak beresan atau adanya pertumbuhan organisasi.
2. Untuk meraih kesempatan-kesempatan. Kesempatan ini dapat berupa peluang bisnis, pelayanan yang meningkat pada langganan.
3. Adanya instruksi-instruksi dari batasan atau pimpinan atau dari luar organisasi, seperti peraturan pemerintah.

2.6.2 Prinsip Pengembangan Sistem

Adapun prinsip dari pengembangan sistem adalah sebagai berikut (Sutabri, 2004):

- a. Sistem yang dikembangkan adalah untuk manajemen.
- b. Sistem yang dikembangkan adalah investasi modal yang besar.
- c. Sistem yang dikembangkan memerlukan orang yang terdidik.
- d. Proses pengembangan sistem tidak harus urut.
- e. Jangan takut membatalkan proyek.
- f. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem.

2.7 Perangkat Lunak

Menurut (Suryatmo dan Rusmadi, 2006:65), Perangkat lunak (*software*) merupakan suatu komponen di dalam suatu sistem data berupa program atau instruksi untuk mengontrol suatu sistem. Tujuan dari perancangan perangkat lunak, antara lain:

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, handal, cepat dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *platform*.
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

2.8 Peti Kemas

Peti kemas adalah peti yang terbuat dari logam yang memuat barang-barang yang lazim disebut muatan umum yang dikirimkan melalui laut (Amir, 1997). Menurut (Kramadibrata, 2002), peti kemas adalah suatu bentuk kemasan satuan muatan yang terbaru yang mulai diperkenalkan pada

tahun 1960 dan diawali dengan ukuran 20 kaki (*twenty feet container*). Pada umumnya peti kemas dibuat dari bahan-bahan yang berupa baja, aluminium, dan *polywood* atau FRP (*fiber glass reinforced plastics*). Pemilihan bahan peti kemas ini berdasarkan pada pemakaian peti kemas bersangkutan. Ukuran peti kemas didasarkan pada *International Standard Organization (ISO)*. Unit ukuran yang lazim digunakan adalah TEU's (*Twenty Feet Square Units*). Peti kemas dengan ukuran 20 *feet* kuadrat sama dengan 1 TEU's, sedangkan peti kemas dengan ukuran 40 *feet* kuadrat sama dengan dua TEU's. Dalam pencatatan di lapangan seringkali juga digunakan istilah *BOX* yang menunjukkan satu kotak peti kemas dengan ukuran tertentu. Ukuran ini lebih mudah dipakai daripada penggunaan ukuran TEU's.

2.9 UML

Menurut (P.H. Schmitt, 2002) *Unified Modelling Language (UML)* adalah bahasa untuk menggambarkan, menspesifikasikan, membangun, dan mendokumentasikan sistem perangkat lunak yang berorientasi pada objek. UML telah diterima secara luas sebagai standar untuk sistem perangkat lunak modeling dan didukung oleh banyak bantuan perangkat *Computer Aided Software Engineering Tools (CASE)*.


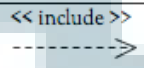

Menurut (Julius Hermawan, 2014) *Unified Modelling Language* atau dikenal dengan UML, merupakan sebuah metode permodelan visual sebagai alat untuk merancang sistem berorientasi objek, atau dapat diartikan sebagai sebuah bahasa pemrograman yang sudah menjadi standar secara visual, perancangan serta pendokumentasian sistem perangkat lunak. UML sendiri terdiri dari beberapa jenis, yang terdiri dari:

- *Use Case Diagram*

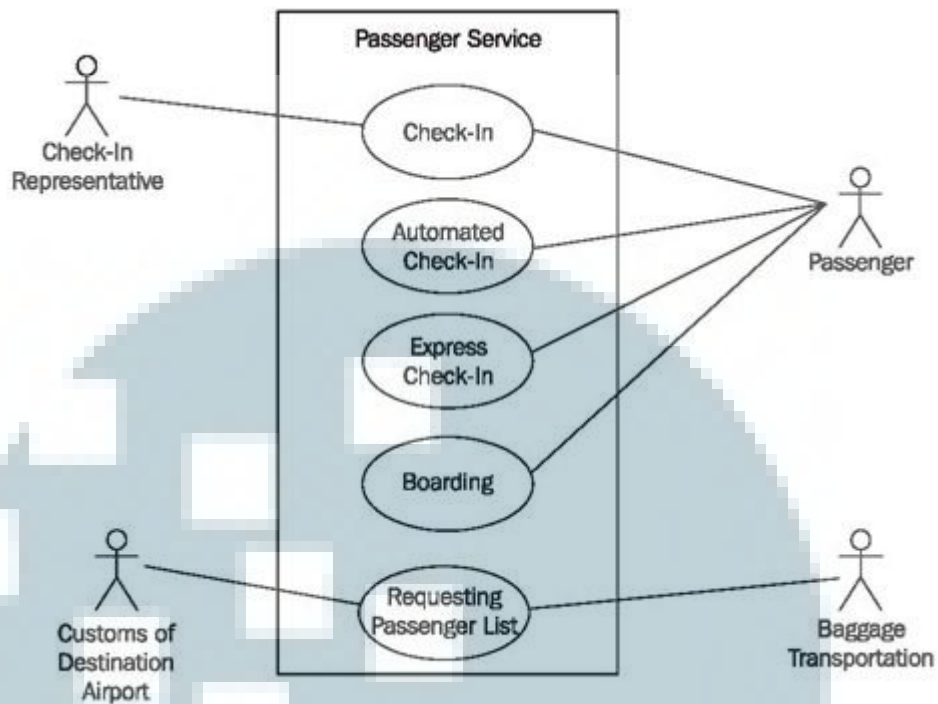
Use case diagram merupakan sebuah jenis diagram pada UML yang menggambarkan interaksi antara sistem dengan aktor atau penggunanya.

Use case diagram umumnya bersifat menjelaskan secara detail, tipe interaksi antara si *user* dengan sistemnya.

Tabel 2.1 Keterangan simbol *use case diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

UMMN



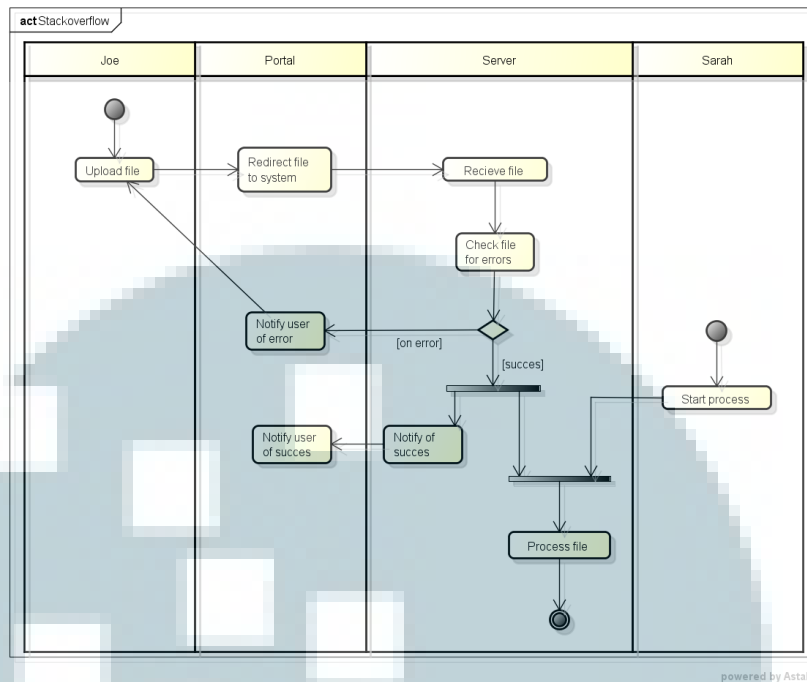
Gambar 2.2 Contoh use case diagram

- *Activity diagram*

Activity diagram merupakan diagram yang menggambarkan model-model aktivitas apa saja yang terjadi dalam suatu sistem.

Tabel 2.2 Simbol *activity diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya



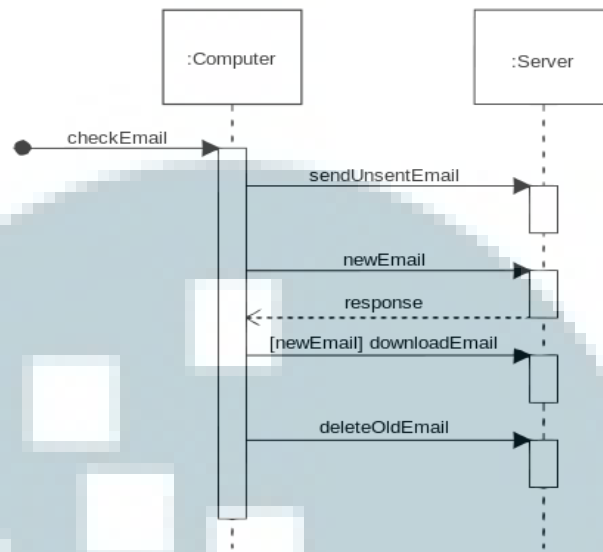
Gambar 2.3 Contoh activity diagram

- *Sequence Diagram*

Sequence diagram merupakan jenis diagram pada UML yang mendeskripsikan interaksi antar objek berdasarkan urutan waktu. *Sequence diagram* sendiri juga memiliki fungsi untuk menggambarkan langkah yang harus dilakukan untuk dapat menghasilkan sesuatu.

Tabel 2.3 Simbol *sequence diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi



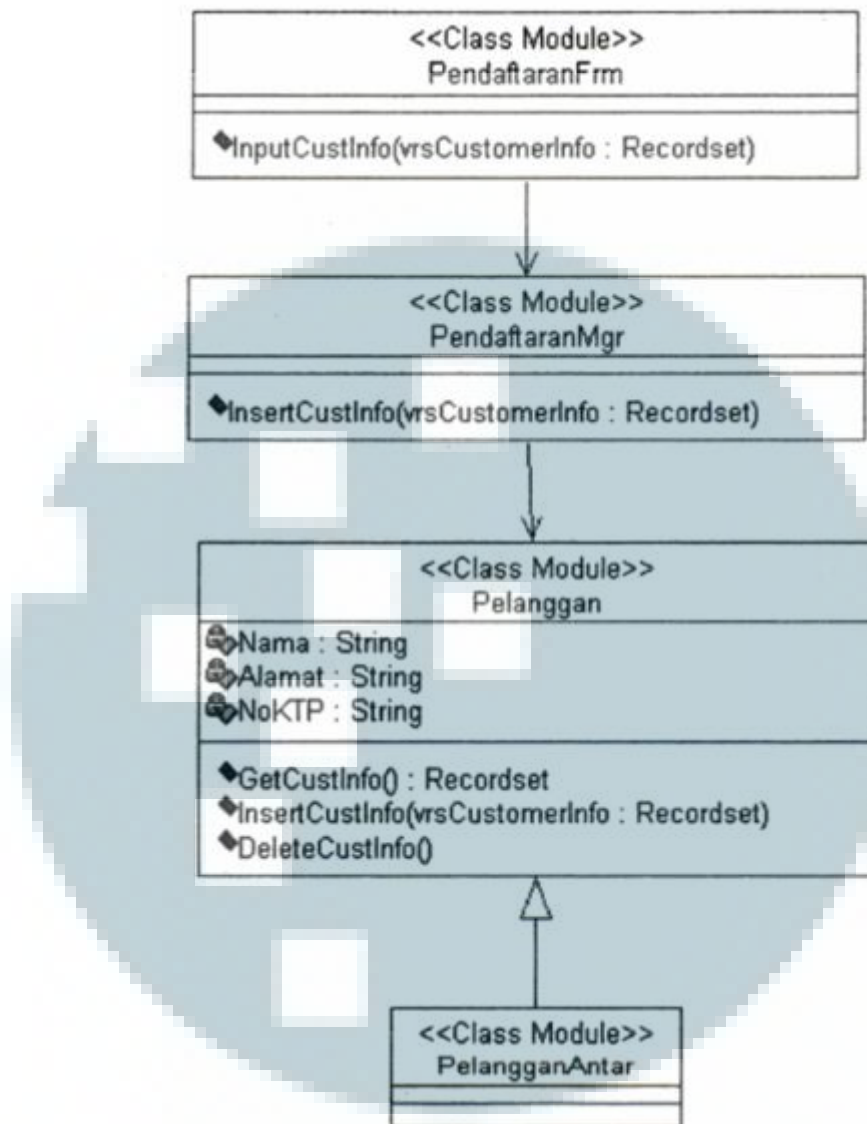
Gambar 2.4 Contoh *sequence diagram*

- *Class Diagram*

Class diagram merupakan sebuah diagram pada UML yang berfungsi untuk menampilkan kelas-kelas yang ada pada suatu sistem yang nantinya akan digunakan, sehingga *class diagram* memungkinkan untuk memberikan gambaran sistem maupun relasi-relasi yang terdapat dalam sistem tersebut

Tabel 2.4 Simbol *class diagram*

No.	Simbol	Fungsi
1		<ol style="list-style-type: none"> 1. <i>Class Name</i> 2. <i>Attribute</i> adalah properti dari sebuah class. <i>Attribute</i> ini melukiskan batas nilai yang mungkin ada pada objek dari class. Sebuah class mungkin mempunyai nol atau lebih <i>attribute</i>. 3. <i>Operation</i> adalah sesuatu yang dapat dilakukan oleh sebuah class atau yang anda (class yang lain) dapat lakukan untuk sebuah class.
2	<p style="text-align: center;"><i>Asosiation</i></p>	Dapat diartikan sebagai relasi. Digambarkan sebagai garis lurus antara dua kelas. Namun tidak berarti bahwa kelas satu memiliki kelas yang lain, tetapi kelas yang lain dapat berelasi juga dengan kelas yang sama.
3		<i>Cardinality</i> atau <i>multiplicity</i> adalah simbol yang menunjukkan jumlah keterhubungan dari satu kelas dengan kelas lainnya.



Gambar 2.5 Contoh class diagram

2.10 Model *Prototype*

Proses pengembangan sistem seringkali menggunakan pendekatan *prototype* (*prototyping*). Metode ini sangat baik digunakan untuk menyelesaikan masalah kesalahpahaman antara *user* dan analis yang timbul akibat *user* tidak mampu mendefinisikan secara jelas kebutuhannya. (Mulyanto, Rekayasa Perangkat Lunak, 2009).

Prototype adalah suatu versi sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran bagaimana kira-kira sistem tersebut akan berfungsi bila telah disusun dalam bentuk yang lengkap. (Jr. McLeod Raymond dan George P. Schell, Sistem Informasi Manajemen, 2008:188).

Sebagian *user* kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan *user* dan menerjemahkannya ke dalam bentuk model (*prototype*). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan *user*.

2.11 *Interface*

Menurut (*Oxford American Dictionaries*) *interface* adalah cara dari sebuah program komputer dalam menyajikan informasi kepada pengguna atau menerima informasi dari pengguna, terutama tata letak layar dan menu. Berdasarkan hal tersebut, maka *User Interface* adalah bentuk tampilan grafis yang berhubungan langsung dengan pengguna. *User Experience* buku “Ergonomics of Human System Interaction” dijelaskan UX adalah persepsi dan respon seseorang yang dihasilkan dari penggunaan dari sebuah produk, sistem, ataupun jasa. Kemudian *User Centered Design* merupakan paradigma baru dalam pengembangan sistem berbasis web. Perancangan berbasis pengguna UCD adalah istilah yang yang digunakan untuk menggambarkan filosofi perancangan. Konsep dari UCD adalah pengguna sebagai pusat dari proses pengembangan sistem, dan tujuan/sifat-sifat,

konteks dan lingkungan sistem semua didasarkan dari pengalaman pengguna.

2.12 *User Interface (UI)*

Menurut (Dmitry Fadeyev, 2009) *user interface* merupakan interaksi yang tercipta antara pengguna dengan aplikasi. Hal ini berarti, *user interface* bukan sekedar mengenai penampilan produk, tetapi lebih kepada produk tersebut bekerja. Fadeyev menjelaskan hal-hal yang perlu diperhatikan untuk merancang sebuah *user interface* yang baik dan berkualitas adalah sebagai berikut:

- *Concision*
Tampilan yang ringkas, padat, dan mudah dimengerti sehingga tidak membingungkan pengguna
- *Familiarity*
Tampilan yang akrab dengan cara menggunakan elemen umum yang mudah dipahami pengguna.
- *Responsiveness*
Memiliki respon atau ketanggapan yang baik dalam memberikan umpan balik kepada pengguna.
- *Efficiency*
Menerapkan efisiensi yang tinggi dan menjunjung tinggi produktivitas.
- *Clarity*
Menghindari sifat ambigu, dengan memperjelas, bahasa, alur, serta hirarki tampilan tersebut.

- *Concistency*

Menggunakan elemen-elemen yang terpola dengan baik secara konsisten.

- *Aesthetics*

Tampilan yang indah, sehingga membuat pengguna tertarik dan nyaman dalam menggunakannya.

- *Forgiveness*

Menyediakan fasilitas untuk pengguna ketika melakukan kesalahan, misal ketika pengguna salah menekan tombol, user *interface* yang baik menyediakan tombol untuk undo atau back.

2.13 Metode *LAFF*

Sebuah algoritma yang digunakan untuk menempatkan kotak dengan luas permukaan terbesar yang pertama dengan meminimalkan tinggi dari bawah wadah. (M.Zahid Gurbuz, 2009)

2.13.1 Penginputan Untuk Algoritma *LAFF*

Pertama, masukkan jumlah ukuran *box* yang berbeda ditandai dengan N. Kedua, masukkan dimensi untuk setiap jenis *box* yang berbeda ukuran. Kita tandai parameter ini dengan empat nilai (a_n , b_n , c_n , k_n) dimana a_n adalah lebar *box*, b_n adalah ukuran dalam *box*, c_n adalah tinggi dan k_n adalah jumlah *box* yang sudah ditentukan ukurannya.

Tergantung bagaimana kalian memperhatikan *box* itu, setiap dimensi bisa dipertimbangkan seperti lebar, tinggi, atau ukuran dalam. Jika kalian memperkirakan b_n sebagai

lebar, dimensi yang lain seperti a_n dan c_n bisa diterima sebagai tinggi atau ukuran dalam. Karena *box* memiliki tiga dimensi dan bisa diputar juga dilihat dari perspektif yang berbeda. Berikut ini adalah daftar parameter yang diusulkan untuk memasukkan algoritma:

Jumlah ukuran *box* yang berbeda..... : N

Lebar *box*..... : a_n

Ukuran dalam *box*..... : b_n

Tinggi *box*..... : c_n

Jumlah *box*..... : k_n

2.13.2 Output Untuk Algoritma *LAFF*

Setelah mengeksekusi *LAFF* Algoritma, program menghasilkan daftar output:

O1: Volume dari kontainer

O2: Ruang yang digunakan (volume dari *box* yang sudah tersimpan)

O3: Ruang yang tidak terpakai

2.13.3 Bagaimana Cara Kerja Algoritma *LAFF*

Seperti yang sudah disebutkan sebelumnya, masalah *packing* 3D adalah masalah kombinasi dengan kompleksitas ketajaman NP. Itu berarti hasil optimal untuk solusi dari masalah tersebut dapat ditemukan dengan mencoba semua kombinasi yang mungkin memiliki solusi berbeda. Namun, jika jumlah *box* meningkat, jumlah iterasi akan meningkat

begitu banyak sehingga tidak dapat diselesaikan dalam waktu polinomial bahkan dengan computer tercepat yang tersedia dengan teknologi saat ini. Dibawah beberapa asumsi dasar dan kendalanya. Beberapa masalah ini dapat diselesaikan dengan algoritma heuristic yang menyediakan solusi dekat dan optimal.

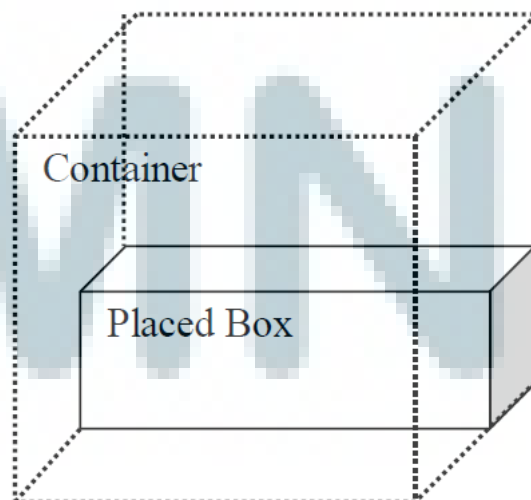
Algoritma yang diusulkan disebut dengan *Largest Area First-Fit(LAFF)* meminimalkan tinggi. Ini menggunakan heuristic yang menempatkan *box* dengan luas permukaan terbesar pertama dengan meminimalkan ketinggian dari dasar wadah. Algoritma ini bekerja sebagai berikut:

Pertama-tama, lebar dan kedalaman wadah harus ditentukan. Mengingat satu set *box* ukuran yang berbeda, kita menghitung lebar dan kedalaman wadah dengan menemukan dua sisi terpanjang *box* yang diberikan. Lebar dan kedalaman wadah ditentukan pada awal algoritma dan tetap sepanjang eksekusi algoritma. Ketinggian meningkatkan algoritma berjalan.

Artinya, lebar (a_k) dan kedalaman (b_k) dari wadah ditentukan dengan memilih tepi pertama dan kedua terpanjang *box* yang diberikan (a_i , b_i , dan c_i). Tepi terpanjang diambil sebagai lebar (a_k) dan tepi terpanjang kedua diambil sebagai kedalaman (b_k). Jadi, a_k dan b_k tidak pernah berubah.

Setelah menentukan nilai dari lebar dan kedalaman wadah, kotak yang diberikan dapat ditempatkan ke dalam wadah. Dalam algoritma ini, dua jenis metode penempatan digunakan. Metode penempatan pertama mengalokasikan ruang untuk kotak yang meningkatkan ketinggian wadah. Metode penempatan kedua mengalokasikan ruang untuk kotak yang tersisa jika ada sebuah kotak cocok dalam ruang yang tersedia di sekitar kotak ditempatkan tanpa meluap ketinggian kotak ditempatkan

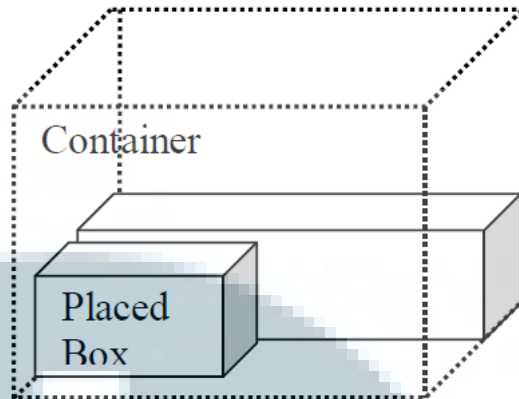
Dalam metode penempatan pertama, kotak dengan luas permukaan terbesar ditentukan dan kotak yang dipilih untuk menemukan sebuah kotak yang memiliki ketinggian minimum dari semua kotak yang dipilih. Kemudian, kotak dengan tinggi minimum ditempatkan dalam wadah. Permukaan terbesar dari kotak yang dipilih harus sejajar dengan bagian bawah wadah (Gbr. 1).



Gambar 2.6 Jenis pertama dari metode penempatan

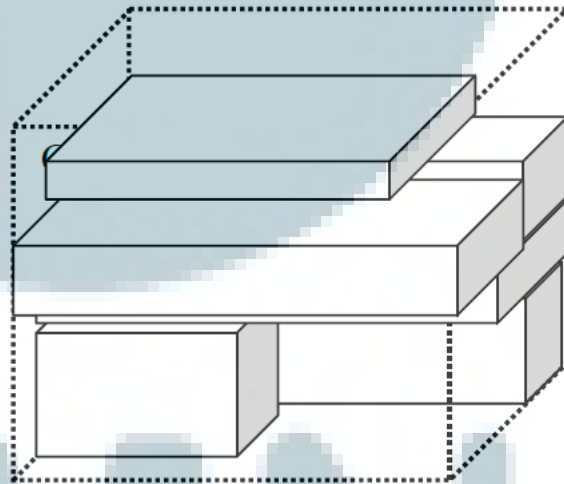
Dalam metode penempatan kedua, algoritma mencoba untuk mengalokasikan ruang untuk kotak yang tersisa di sekitar kotak yang ditempatkan dengan metode pertama. Artinya, jika beberapa ruang tetap sekitar kotak ditempatkan seperti pada Gambar. 1, algoritma mencoba untuk mengisi ruang itu dengan kedua jenis metode penempatan. Dalam metode ini, mengingat bahwa kotak yang ditempatkan memiliki dimensi a_i , b_i , dan c_i pada tingkat, maka wadah akan memiliki dua ruang kosong di sekitar kotak yang ditempatkan dengan dimensi $((a_k - a_i), b_k, c_i)$ dan $(a_k, (b_k - b_i), c_i)$. Jadi, jika ada kotak pas di ruang-ruang kosong, yang dapat menjadi salah satu atau lebih, kita menempatkan kotak yang memiliki volume maksimum dari kotak pas seperti ditunjukkan pada Gambar. 2. iterasi ini berlanjut sampai tidak ada kotak atau tidak ada kotak bisa masuk ke dalam ruang tersebut.

Dalam metode penempatan kedua, jika tidak ada ruang di sekitar kotak ditempatkan, maka algoritma berlanjut dengan jenis pertama metode penempatan, akibatnya.



Gambar 2.7 Jenis kedua dari metode penempatan

Setiap kali ada kotak yang belum ditempatkan, algoritma akan langsung bekerja dengan jenis metode pertama dan seterusnya sampai semua kotak ditempatkan. Pada akhir algoritma, solusi yang mungkin seperti yang ditunjukkan pada Gambar. 3 dapat dihasilkan.



Gambar 2.8 Solusi yang memungkinkan dari algoritma

2.13.4 Algoritma *LAFF*

Langkah-langkah utama dari algoritma *LAFF* adalah sebagai berikut:

Langkah 1: Input dimensi kotak dan angka.

N: jumlah kotak yang khusus.

Step2: Tentukan lebar (a_k) dan kedalaman (b_k) dari Wadah.

a_k : tepi terpanjang pertama dari semua kotak.

b_k : tepi terpanjang kedua dari semua kotak.

c_k : 0 (nol).

Step3: Memilih kotak yang memiliki luas permukaan terluas. Jika ada lebih dari satu, pilih kotak yang memiliki tinggi terendah. Tempatkan kotak ini (i th) pada paralel permukaan terbesar ke dasar container.

Step3.1: Tentukan ketinggian kontainer dan pengurangan jumlah kotak engan.

$$c_k = c_k + c_i$$

$$k_i = k_i - 1$$

Step3.2: jika jumlah *box* adalah nol kemudian akhiri.

$$\sum_p k_p = 0$$

Langkah 3.3: jika ruang $(a_k - a_i) = 0$ dan $(b_k - b_i) = 0$ kemudian kembali ke Langkah 3. Jika tidak, pilihlah kotak yang pas pada ruang container. Jika tidak ada kotak yang pas dengan ruang container, lanjutkan ke Langkah 3. Jika ada lebih dari satu kotak yang pas dengan ruang ini, pilih kotak yang memiliki volume terbesar. Ini disebut kotak j .

Step3.3.1: Tentukan dimensi ruang.

$$a_s = a_k - a_i - a_j \text{ ve } b_s = \max(b_k - b_i, b_k - b_j)$$

atau

$as = \max (ak-ai, ak-aj)$ dan $bs = bk-bi-bj$

Step3.3.2: Jumlah Pengurangan kotak (ith)

$kj = kj-1$

Step3.3.3: jika jumlah *box* adalah nol kemudian akhiri.

Jika tidak, lanjutkan ke langkah 3.3.

$$\sum_p k_p = 0$$

2.14 *Bin Packing Problem*

Bin Packing Problem adalah sebuah permasalahan optimalisasi kombinatorial yang termasuk dalam jenis *Non-deterministic Polynomial-time* (NP) hard yang pertama kali diperkenalkan oleh Garey dan Johnson pada tahun 1979. *Bin Packing* memiliki peran penting dalam menyelesaikan beberapa persoalan di dunia nyata seperti perencanaan transportasi, pemuatan kontainer, alokasi sumber daya, penjadwalan, serta kargo pesawat dan kapal (Yesodha & Amudha, 2013). Pada permasalahan *bin packing*, diberikan sebuah bin dengan kapasitas tertentu V yang digunakan sebagai tempat dari seluruh objek n yang memiliki ukuran berbeda-beda, dan tujuannya adalah untuk meminimalkan bin yang digunakan agar seluruh objek n dapat ditempatkan ke dalam *bin* tersebut (Swain et al., 2014).

2.14.1 *Two Dimensional Bin Packing Problem (2DBPP)*

Pada 2DBPP, sebuah *item* atau objek hanya memiliki dua buah variabel seperti panjang dan lebar. Penyusunan *item* atau objek pada 2DBPP hanya berdasarkan kepada dua variabel tersebut. 2DBPP biasanya digunakan untuk menyusun barang pada lantai

ruang dan tidak adanya penumpukan barang. 2DBPP bisa juga digunakan untuk menyusun barang dengan adanya penumpukan, tetapi semua barang tersebut harus memiliki variabel tinggi yang sama, sehingga variabel tinggi tersebut tidak akan mempengaruhi penyusunan barang yang memiliki variabel panjang dan lebar yang berbeda.

2.14.2 Three Dimensional Bin Packing Problem (3DBPP)

Pada 3DBPP, satu atau lebih *bin* yang tersedia dipilih untuk memuat barang-barang secara tiga dimensi sehingga ruang pada *bin* dapat dimaksimalkan (Li et al., 2014). Berbeda dengan 2DBPP yang hanya menggunakan dua variabel, 3DBPP menggunakan tiga variabel yaitu panjang, lebar, dan tinggi barang dalam melakukan penyusunan barang. Hal ini menyebabkan tingkat kesulitan 3DBPP lebih tinggi dari 2DBPP. Setiap *item* atau objek pada 3DBPP harus disusun sedemikian rupa agar *item* tersebut dapat dimuat ke dalam *bin* yang juga memiliki batasan panjang, lebar, dan tinggi. Jika pada 2DBPP penyusunan barang lebih ditekankan kepada penyusunan bidang segi empat pada dasar ruang (*rectangle-to-floorplan packing*), 3DBPP lebih ditekankan kepada penyusunan bangun segi empat pada ruang (*box-to-room packing*) (Sweep, 2003). 3DBPP juga termasuk ke dalam permasalahan pemuatan kontainer (*Container Loading Problem*). Pada 3DBPP penyusunan barang dapat dibedakan menjadi *single bin* atau *multiple bins*. Pada *single bin*, penyusunan barang yang dilakukan hanya menggunakan sebuah

bin, sehingga tujuan penyusunan hanya untuk meminimalkan sisa ruang kosong pada *bin* tersebut. Sementara pada *multiple bins*, penyusunan barang yang dilakukan menggunakan lebih dari satu *bin*, sehingga tujuan penyusunannya adalah untuk meminimalkan jumlah *bin* yang digunakan.

2.15 *Cascading Style Sheet (CSS)*

Menurut (W3schools.com) CSS mendeskripsikan bagaimana unsur-unsur dalam HTML ditampilkan dalam layar, kertas, maupun media lainnya. Dalam CSS, terdapat sebuah *tools* bernama *materialize CSS*, dimana menurut (materializecss.com) *tools* ini berguna untuk menciptakan kerangka yang menggabungkan unsur serta animasi yang lebih menyediakan *feedback* kepada pengguna.

2.16 Simulasi

2.16.1 Definisi Simulasi

Simulasi dapat diartikan sebagai meniru suatu sistem nyata yang kompleks dengan penuh dengan sifat probabilistik, tanpa harus mengalami keadaan yang sesungguhnya. Hal ini dapat dilakukan dengan membuat sebuah miniatur yang *representative* dan *valid* dengan tujuan *sampling* dan *survey* statistik pada sistem nyata, sehingga perilaku sistem dapat diprediksi untuk dipelajari. Jadi simulasi secara sederhana dapat diartikan sebagai proses peniruan.

Beberapa pendapat tentang definisi simulasi :

1. Simulasi adalah proses perancangan model dari suatu sistem nyata dan pelaksanaan eksperimen-eksperimen dengan model ini untuk tujuan memahami tingkah laku sistem.
2. Simulasi adalah tiruan dari proses dunia nyata atau sistem. Simulasi menyangkut pembangkitan proses serta pengamatan dari proses untuk menarik kesimpulan dari sistem yang diwakili.
3. Simulasi adalah teknik numerik untuk melakukan eksperimen pada komputer, yang melibatkan jenis matematika dan model tertentu yang menjelaskan perilaku bisnis atau ekonomi pada suatu periode waktu tertentu.
4. Simulasi adalah teknik untuk membuat konstruksi model matematika untuk suatu proses atau situasi, dalam rangka menduga secara karakteristik atau menyelesaikan masalah berkaitan dengan menggunakan model yang diajukan.

2.16.2 Kekurangan Dan Kelebihan Simulasi

Tidak semua pengolahan data untuk melihat karakteristik sistem cocok menggunakan simulasi, hanya sistem yang cukup kompleks yang baik dikerjakan dengan simulasi, sementara untuk sistem yang sederhana lebih baik menggunakan cara analitik dibanding harus membuat simulasinya.

Kelebihan Simulasi :

1. Sistem nyata sulit diamati secara langsung.
2. Mampu memberikan perkiraan sistem yang lebih nyata sesuai operasional dari kumpulan pekerjaan.
3. Pengamatan sistem secara langsung tidak dimungkinkan karena :
 - a. Sangat mahal
 - b. Memakan waktu terlalu lama
 - c. Akan merusak sistem yang sedang berjalan
4. Solusi analitik tidak dapat dikembangkan, karena sistem yang digunakan di dunia kerja sangat kompleks. Jadi simulasi dapat memberi solusi apabila model analitik gagal.
5. Memudahkan pengontrolan lebih banyak kondisi dari suatu percobaan sehingga dimungkinkan untuk dicoba diterapkan secara nyata pada sistem tersebut.

2.17 Algoritma Genetika

2.17.1 Pengertian Algoritma Genetika

Menurut (Desiani dan Arhami, 2006) Algoritma Genetika (AG) diperkenalkan pertama kali oleh John Holland (1975) dari Universitas Michigan, John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan ke dalam terminologi genetika.

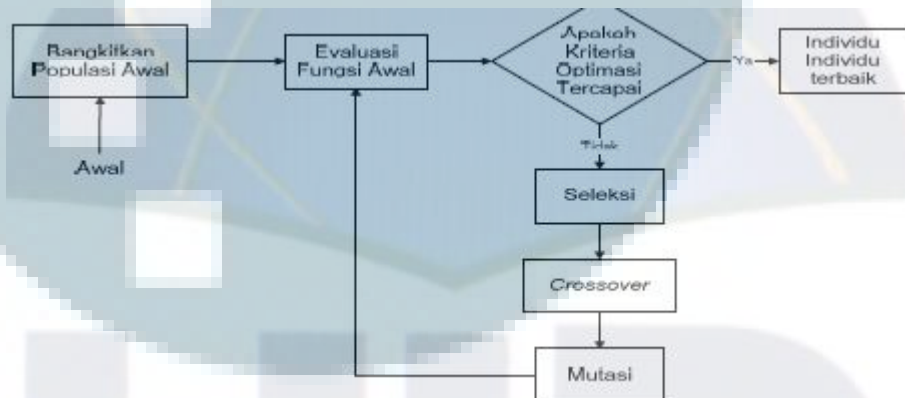
Kemudian menurut Goldberg (1989) mendefinisikan algoritma genetika ini sebagai suatu pencarian algoritma berdasarkan pada mekanisme seleksi alam dan genetika alam. Bauer (1993) mendefinisikan algoritma genetika sebagai perangkat lunak, prosedur yang dimodelkan setelah genetika dan evolusi. Selain itu juga Algoritma Genetika mempunyai karakteristik-karakteristik yang perlu diketahui sehingga dapat terbedakan dari prosedur pencarian atau optimisasi yang lain, yaitu :

1. AG bekerja dengan pengkodean dari himpunan solusi permasalahan berdasarkan parameter yang telah ditetapkan dan bukan parameter itu sendiri.
2. AG melakukan pencarian pada sebuah populasi dari sejumlah individu-individu yang merupakan solusi permasalahan bukan hanya dari sebuah individu.
3. AG merupakan informasi fungsi objektif (*fitness*), sebagai cara untuk mengevaluasi individu yang mempunyai solusi terbaik, bukan turunan dari suatu fungsi.
4. AG menggunakan aturan-aturan transisi peluang, bukan aturan-aturan deterministic.

2.17.2 Struktur Umum Algoritma Genetika

Menurut (Kusumadewi, 2003) Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah **populasi**. Individu yang terdapat dalam satu

populasi individu disebut dengan istilah **kromosom**, Charles L Karr (1999). Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasinya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah **generasi**. Pada setiap generasi kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut **fungsi fitness**. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Proses ini dapat direpresentasikan dalam algoritma sederhana Algoritma Genetika sebagai berikut :



Gambar 2.9 Diagram Alir *Genetic Algorithms* Sederhana

U M N