BAB II
TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Sistem Pengukuran Kaki

Pengukuran kaki dilakukan untuk memperoleh ukuran sepatu yang sesuai dengan dan nyaman ketika digunakan. Pengukuran dapat dilakukan dengan 2 (dua) cara yaitu [8] :

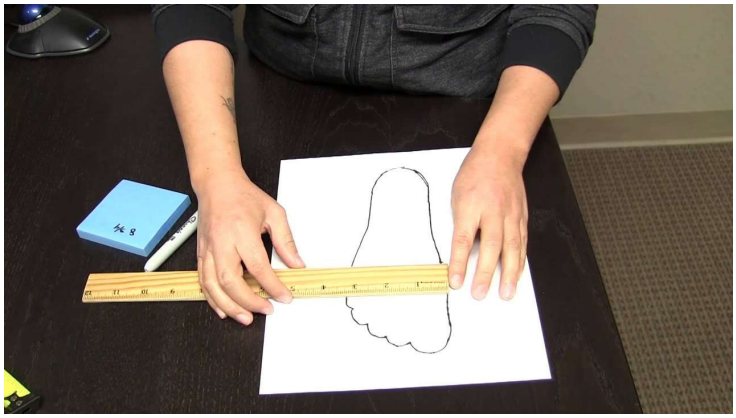
1. Pengukuran menggunakan perangkat *Brannock*

   Brannock adalah alat bantu untuk mengukur kaki dengan standar internasional, penggunaan alat ini hanya dengan menggeser pembatas untuk mengukur panjang dan lebar kaki. Cara penggunaannya, semula kaki diletakkan diatas *Brannock* dengan posisi tumit berada di ujung batas alat kemudian pembatas panjang kaki digeser hingga ujung jari terpanjang setelah itu menggeser pembatas lebar ke bagian kaki terlebar yaitu antara ruas kaki dan tulang telapak. Ilustrasi untuk pengukuran kaki menggunakan *Brannock* ditunjukkan pada gambar 2.1

Gambar 2. 1 Pengukuran Menggunakan Brannock
(Sumber : https://www.footfiles.com/beauty/shoes/article/how-to-properly-measure-your-feet)

2. Pengukuran Manual

   Pengukuran kaki secara manual dilakukan dengan cara yang sederhana, yaitu menggambar bentuk kaki kita diatas kertas polos yang kemudian memberikan garis pada bagian jari terpanjang dan mengukur panjang tersebut menggunakan penggaris. Begitupun dengan mengukur lebarnya. Ilustrasi untuk pengukuran kaki secara manual ditunjukkan pada gambar 2.2



Gambar 2. 2 Pengukuran Secara Manual
(Sumber : http://www.shoecity.co.za/index/shoe-guide)

Setelah hasil pengukuran didapatkan kemudian tambahkan hasil pengukuran dengan nilai 0.5 cm. Hal ini bertujuan untuk memberikan sedikit ruang pada saat menggunakan sepatu. Kemudian hasil dari penjumlahan tersebut dibandingkan dengan *size chart* untuk mencari tahu ukuran sepatu yang akan digunakan. *Size chart* adalah daftar acuan untuk standar ukuran sepatu, namun dalam *size chart* ini yang dipakai hanyalah panjang dari kaki saja dan setiap negara memiliki standar ukuran tersendiri[9].

Berikut adalah tabel ukuran sepatu standar internasional berdasarkan panjang kaki yang akan digunakan :

Tabel 2. 1 Tabel Ukuran Sepatu

SYSTEM		SIZE															
Europe		35	35 1/2	36	37	37 1/2	38	38 1/2	39	40	41	42	43	44	45	46 1/2	48 1/2
Mexico							4.5	5	5.5	6	6.5	7	7.5	9	10	11	12.5
Japan	M	21.5	22	22.5	23	23.5	24	24.5	25	25.5	26	26.5	27.5	28.5	29.5	30.5	31.5
	W	21	21.5	22	22.5	23	23.5	24	24.5	25	25.5	26	27	28	29	30	31
U.K.	M	3	3 1/2	4	4 1/2	5	5 1/2	6	6 1/2	7	7 1/2	8	8 1/2	10	11	12	13 1/2
	W	2 1/2	3	3 1/2	4	4 1/2	5	5 1/2	6	6 1/2	7	7 1/2	8	9 1/2	10 1/2	11 1/2	13
Australia	M	3	3 1/2	4	4 1/2	5	5 1/2	6	6 1/2	7	7 1/2	8	8 1/2	10	11	12	13 1/2
	W	3 1/2	4	4 1/2	5	5 1/2	6	6 1/2	7	7 1/2	8	8 1/2	9	10 1/2	11 1/2	12 1/2	14
U.S & Canada	M	3 1/2	4	4 1/2	5	5 1/2	6	6 1/2	7	7 1/2	8	8 1/2	9	10 1/2	11 1/2	12 1/2	14
	W	5	5 1/2	6	6 1/2	7	7 1/2	8	8 1/2	9	9 1/2	10	10.5	12	13	14	15.5
Russia & Ukraina	W	33 1/2	34		35		36		37		38		39				
Kore (mm.)		228	231	235	238	241	245	248	251	254	257	260	267	273	279	286	292
Inches		9	9 1/8	9 1/4	9 3/8	9 1/2	9 5/8	9 3/4	9 7/8	10	10 1/8	10 1/4	10 1/2	10 3/4	11	11 1/4	11 1/2
Cm		22.8	23.1	23.5	23.8	24.1	24.5	24.8	25.1	25.4	25.7	26	26.7	27.3	27.9	28.6	29.2

(Sumber : https://www.blitzresults.com/en/shoe-size-conversion/)

Berikut adalah tabel ukuran lebar berdasarkan pengukuruan sistem EU ditunjukkan pada tabel 2.2.

Tabel 2. 2 Tabel Lebar Kaki

EU Size	Narrow	Medium	Wide	Extra-Wide
35.5	7.1 cm	8 cm	8.9 cm	9.9 cm
36	7.2 cm	8.1 cm	9 cm	10 cm
37.5	7.5 cm	8.5 cm	9.4 cm	10.3 cm
38	7.7 cm	8.6 cm	9.5 cm	10.4 cm
38.5	7.9 cm	8.8 cm	9.7 cm	10.6 cm

39	8 cm	8.9 cm	9.8 cm	10.8 cm
40	8.3 cm	9 cm	10 cm	11 cm
40.5	8.4 cm	9.2 cm	10.2 cm	11.1 cm
41	8.6 cm	9.4 cm	10.3 cm	11.3 cm
42	8.7 cm	9.5 cm	10.4 cm	11.5 cm
43	9 cm	9.9 cm	10.8 cm	11.7 cm
44	9.1 cm	10 cm	11 cm	11.9 cm
45	9.6 cm	10.3 cm	11.2 cm	12.3 cm
46	9.7 cm	10.5 cm	11.4 cm	12.5 cm

(Sumber : https://www.blitzresults.com/en/shoe-size-conversion/)

2.1.2 Pengertian Pengolahan Citra

2.1.2.1 Definisi Citra

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 (dua) dimensi menggunakan computer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 (dua) dimensi. Citra digital merupakan sebuah larik (arry) yang berisi nilai-nilai maupun komplek yang di representasikan dengan deretan bit tertentu[10].

Citra didefinisikan sebagai fungsi intensitas cahaya dua dimensi f (x,y) dimana x dan y menunjukkan koordinat spasial. Dan nilai f pada suatu titik (x,y) sebanding dengan tingkat kecerahan (gray level) dari citra di titik tersebut [11].

Citra sebagai out put dari suatu sistem perekaman data dapat bersifat :

1.Optik berupa foto.

2.Analog berupa sinyal video seperti gambar pada monitor televisi.

3.Digital yang dapat langsung disimpan pada media penyimpanan magnetik.

Citra dapat dikelompokkan menjadi dua yaitu citra diam (still image) dan citra bergerak (moving image). Citra diam yang ditampilkan secara beruntun (sekuensial), sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra didalam rangkaian tersebut disebut frame. Gambar-gambar yang

tampak pada film layar lebar atau televisi pada hakekatnya terdiri dari ratusan sampai ribuan frame[11].

Citra juga dapat dikelompokkan menjadi dua yaitu citra yang tampak seperti foto, gambar, lukisan, apa yang nampak dilayar monitor/televisi, hologram, dan lain sebagainya. Sedangkan citra tidak tampak seperti data foto/gambar dalam file, citra yang direpresentasikan dalam fungsi matematis[12].

Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa Citra digital yang tersusun dari 3 komponen warna yakni R (red), G (green), B (blue) di dalam matlab direpresentasikan dalam bentuk :

```
R = citra_rgb(:,:,1)
G = citra_rgb(:,:,2)
B = citra_rgb(:,:,3)
```

Maksudnya adalah bahwa tiap komponen warna terpisah dalam bentuk koordinat spasial (x,y) dan angka ketiga berfungsi sebagai urutan apakah warna tersebut R, G atau B. Urutan ini sudah standar, jadi jangan sampai tertukar. Untuk mengubah warna RGB ke skala warna abu-abu atau grayscale dapat digunakan rumus standar berikut:

```
gray_R = .2989*citra_rgb(:,:,1)
gray_G = .5870*citra_rgb(:,:,2)
gray_B = .1140*citra_rgb(:,:,3)
```

Citra dalam skala abu-abu memiliki variasi warna 0-255, dan berukuran 1x8 bit. Tentu saja berbeda dengan citra biner yang hanya memiliki warna hitam dan putih berarti hanya memiliki komponen warna 0 dan 1 saja.

2.1.2.2 Pengolahan Citra Dalam Komputer Vision

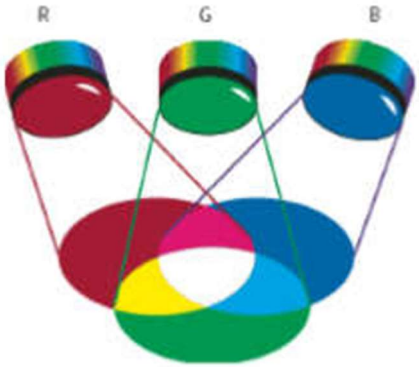
Computer vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, pengenalan dan membuat keputusan. Computer vision mencoba meniru cara kerja sistem visual manusia (Human Vision) yang sesungguhnya sangat kompleks. Untuk itu, computer vision diharapkan memiliki kemampuan tingkat tinggi sebagaimana human visual[10].

Kemampuan tersebut diantara adalah :

1. Object Detection → Apakah sebuah objek ada pada scene? Jika begitu, dimana batasan-batasannya..?
2. Recognation → Menempatkan label pada objek.
3. Description → Menugaskan properti kepada objek.
4. 3D Inference → Menafsirkan adegan 3D dari 2D yang dilihat.
5. Interpreting motion → Menafsirkan gerakan.

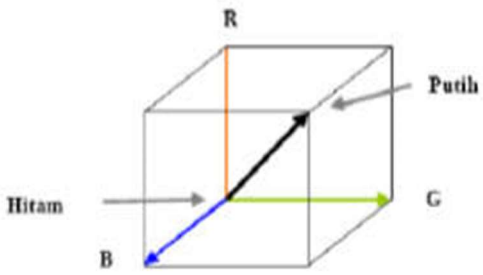
2.1.2.3 RGB (Red Green Blue)

Model warna Red Green Blue (merah, hijau dan biru). Apabila pada ruangan tidak terdapat cahaya (gelap) maka tidak ada gelombang cahaya yang dapat dihasilkan oleh mata sehingga nilai RGB (0,0,0) dan apabila kita tambahkan sebuah cahaya yang berwarna merah maka akan menghasilkan nilai RGB (255,0,0) dan semua benda yang ada dalam ruangan tersebut akan berwarna atau terlihat berwarna merah, begitu pun sebaliknya.



Gambar 2. 3 Warna RGB
(Sumber : https://idseducation.com/articles/yuk-mengenal-warna-rgb/)

Menurut Terori Tri Stimulus manusia dalam melihat warna yaitu dengan cara membandingkan cahaya yang diterima dengan sensor – sensor peka cahaya dalam retina. Sensor – sensor dapat melihat dengan panjang gelombang 630 nm (merah), 530 nm (hijau), dan 450 nm (biru)[13]. Berikut contoh gambar yang dapat menerangkan sudut – sudut dari sumbu R,G, dan B



Gambar 2. 4 Sudut Warna RGB
(Sumber : https://www.ristofa.com/2017/01/perbedaan-warna-rgb-dan-cmyk-pada-desain-grafis.html)

Model warna RGB adalah salah satu model additive dimana intensitas dalam suatu warna primer dijumlahkan sehingga akan menghasilkan warna lainnya. Model warna RGB dapat digabungkan dengan halftoning yang dapat menghasilkan warna yang lebih banyak. Dalam sistem on/off, model warna RGB hanya terdapat 8 warna, dimana nilai halftoning 2 x 2 yang menghasilkan 125 (5 warna merah x 5 warna hijau x 5 warna biru)[14].

2.1.2.4 Grayscale

Suatu citra grayscale adalah suatu citra yang hanya memiliki warna tingkat keabuan. Penggunaan citra grayscale dikarenakan membutuhkan sedikit informasi yang diberikan pada tiap piksel dibandingkan dengan citra berwarna. Warna abu-abu pada citra grayscale adalah warna R (Red), G (Green), B (Blue) yang memiliki intensitas yang sama. Sehingga dalam grayscale image hanya membutuhkan nilai intensitas tunggal dibandingkan dengan citra berwarna membutuhkan tiga intensitas untuk tiap pikselnya. Intensitas dari citra grayscale disimpan dalam 8 bit integer yang memberikan 256 kemungkinan yang mana dimulai dari level 0 sampai dengan 255 (0 untuk hitam dan 255 untuk putih dan nilai diantaranya adalah derajat keabuan). Tingkat keabuan atau Grayscale level dapat dilihat pada gambar

Gambar 2. 5 Grayscale Level
(Sumber : https://uxplanet.org/designing-systematic-colors-b5d2605b15c)

2.1.2.5 Canny Edge Detection

Batas garis atau Lane Boundaries didefinisikan oleh kontras antara permukaan lintaan dan garis pada jalan. Kontras ini adalah edges dari gambar. Oleh karena itu *edge detectors* sangat penting untuk menentukan dimana batas garis atau lane boundaries. Hal itu juga mengurangi jumlah pembelajaran data yang dibutuhkan untuk menyederhanakan gambar, jika outline dari sebuah lintsan dapat diekstrak dari sebuah gambar.

Edge detector yang digunakan pada algoritma ini dan satu-satunya yang dapat menghasilkan gambar edge terbaik dari semua edge detector adalah 'canny' edge detector. Sangat penting memiliki pendeteksi pinggiran atau edge detection yang dapat memilih threshold secara otomatis, Tetapi *threshold* otomatis yang ada pada standar canny menghasilkan informasi pinggiran atau edge yang terlalu banyak. Dengan membuat sedikit modifikasi terhadap *edge detection* yang dihasilkan oleh canny memberikan hasil yang lebih baik. Canny *edge detector* juga memiliki karakteristik yang membuatnya tidak menghasilkan noise seperti pendekatan pendekatan yang lainnya[16]

2.1.2.6 Ekstraksi Ciri Suatu Gambar

Ekstraksi ciri adalah pemrosesan indeks suatu database citra dengan isinya atau merupakan tahapan mengekstrak ciri/informasi dari objek didalam citra yang ingin dikenali atau dibedakan dengan objek lainnya. Ekstrasi ciri dapat dibedakan menjadi 3 jenis yaitu, *low*, *middle*, dan *high level. Low level* merupakan ekstraksi secara visual seperti warna dan tekstur. *Middle Level* merupakan ekstraksi berdasarkan wilayah citra atau disebut segmentasi. *High level* merupakan ekstraksi berdasarkan informasi *semantic* yang tersimpan dalam citra[15], maca-macam ektraksi berdasarkan ciri sebagai berikut :

1. **Warna**

   Warna merupakan salah satu ciri visual yang dipakai dalam *Content Based Image Retreival.* Warna sangat baik jiga digunakan dalam suatu citra karena memiliki hubungan yang sangat kuat dengan objek dan menjadi

latarbelakang untuk penggabungan *background*, skala, orientasi, perspektif dan ukuran suatu citra.

2. **Bentuk**

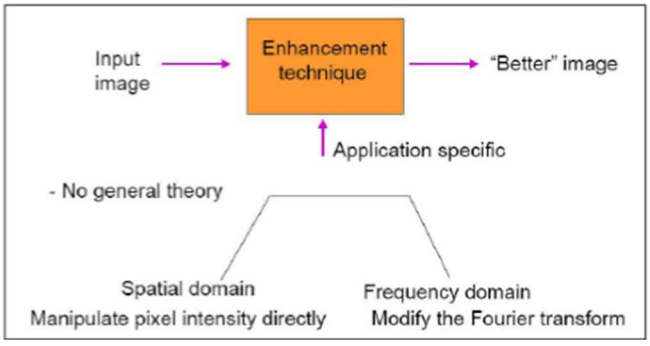
   Bentuk dapat didefinisikan sebagai gambaran suatu objek dalam ukuran, posisi dan oritentasi. Ciri bentuk suatu citra sangat intrinsic untuk segmentasi pada suatu citra karena dapat mendeteksi objek atau batas wilayah.

3. **Tekstur**

   Tekstur merupakan ciri citra yang sangat menarik digunakan untuk menentukan karakterisasi suatu citra dengan aplikasi CBIR. Hal ini dikarenakan tekstur mengandung informasi penting mengenai susunan struktur permukaan suatu citra[17].

2.1.2.7 Image Enhancement (Peningkatan Mutu Citra)

Berbagai macam operasi dapat dilakukan dalam pengolahan citra salah satunya adalah *Image Enhancement* (peningkatan mutu citra) yaitu, suatu operasi atau teknik untuk mendapatkan citra yang lebih detail. Proses peningkatan mutu citra bertujuan untuk memperoleh citra yang dapat memberikan informasi sesuai dengan tujuan atau kepentingan pengolahan citra[18].



Gambar 2. 6 Diagram Teknik Image Enhancement

(Sumber : Priyawati, Diah.2011.Teknik pengolahan citra digital berdomain spasial untuk peningkatan citra sinar-x. KomuniTi, Vol. II, No.6)

2.1.3 Pengertian Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Hingga saat ini terdapat

dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GSM) dan kedua adalah yang bebas pendistribusiannya tanpa dukungan langsung Google atau diketahui sebagai *Open Handset Distribution* (OHD). Pada saat ini umumnya semua vendor *smartphone* telah memproduksi smartphone berbasis android, antara lain HTC, Motorola, Samsung, LG, Sony, Xiaomi dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi android. Hal ini karena android itu adalah sistem operasi yang bersifat *open source* sehingga bebas didistribusikan dan digunakan oleh vendor manapun.

Seiring bertumbuhnya android selain faktor yang disebutkan sebelumnya adalah karena android adalah *platform* yang cukup lengkap baik sistem operasinya, aplikasi dan *Tools* Pengembangan, *Market* aplikasi android serta mendapat dukungan yang tinggi dari komunitas *open source* di dunia, sehingga android terus berkembang pesat baik dari segi teknologi yang diterapkan maupun dari segi jumlah device yang diproduksi hingga saat ini.

2.1.3.1 Aplikasi Android

Android mengijinkan penggunanya untuk menyematkan aplikasi pihak ketiga, baik yang diperoleh dari *market* aplikasi seperti Google Play, Amazon Appstore, ataupun dengan mengunduh dan memasang berkas berekstensi APK dari situs pihak ketiga. Di Google Play, pengguna bisa mencari, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh Google dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas Google. Google Play akan menyaring daftar aplikasi yang tersedia berdasarkan kompatibilitasnya dengan perangkat pengguna.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, *emulator handset* yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools

(ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.

2.1.3.2 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. Bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi android terbaru yaitu Android versi 3.0. juga sudah bergabung dengan beberapa *smart mobile* seperti Nokia, Sony Ericsson, dan lainnya[24].

2.2 Teori Khusus

2.2.1 UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

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).

2.2.1.1 Konsep Dasar UML

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan pada gambar

Tabel 2. 3 Konsep Dasar UML

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Abstraksi konsep dasar UML yang terdiri dari *structural classification, dynamic behavior, dan model management,* bisa dipahami dengan mudah apabila dilihat pada gambar diatas dari diagram. *Main concepts* bisa dipandang sebagai *term* yang akan muncul pada saat membuat diagram. Dan *view* adalah kategori dari diagram

tersebut. Untuk menguasai UML, sebenarnya cukup dua hal yang harus di perhatikan:

1. Menguasai pembuatan diagram UML
2. Menguasai langkah-langkah dalam analisa dan pengembangan dengan UML

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

1. use case diagram
2. class diagram
3. activity diagram
4. sequence diagram

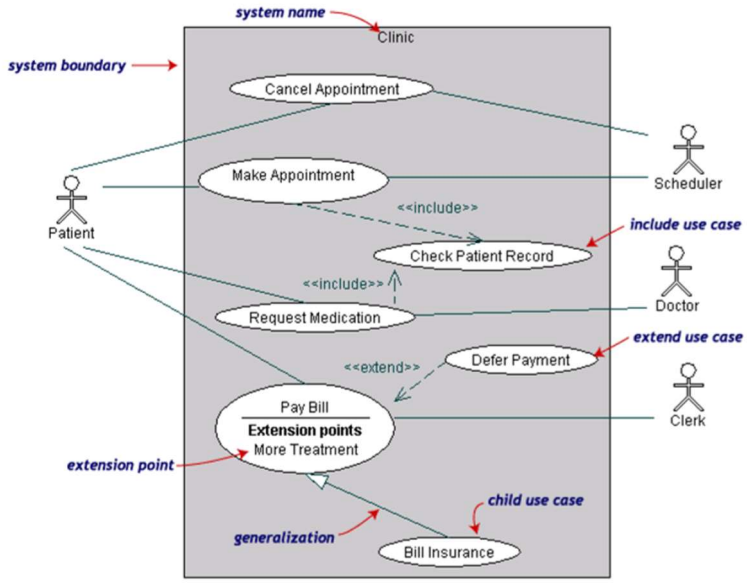
2.2.1.2 Use Case Diagram

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.

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*.

Sebuah use case juga dapat meng-extend use case lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain[26].

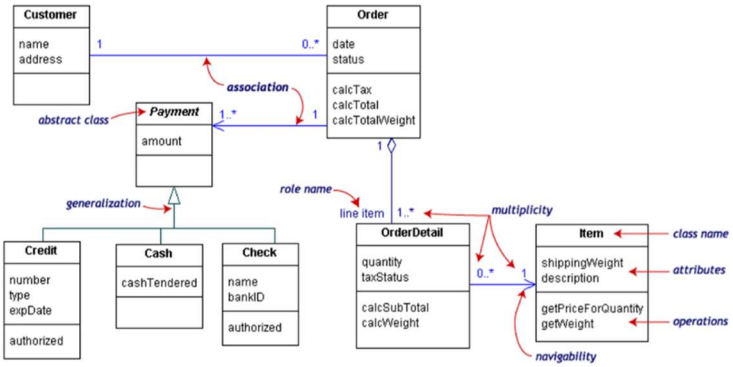


Gambar 2. 7 Contoh Use Case Diagram
(Sumber : https://www.dumetschool.com/blog/Mengenal-Use-Case-Diagram)

2.2.1.3 Class Diagram

Class diagram 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).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time



Gambar 2. 8 Contoh Class Diagram

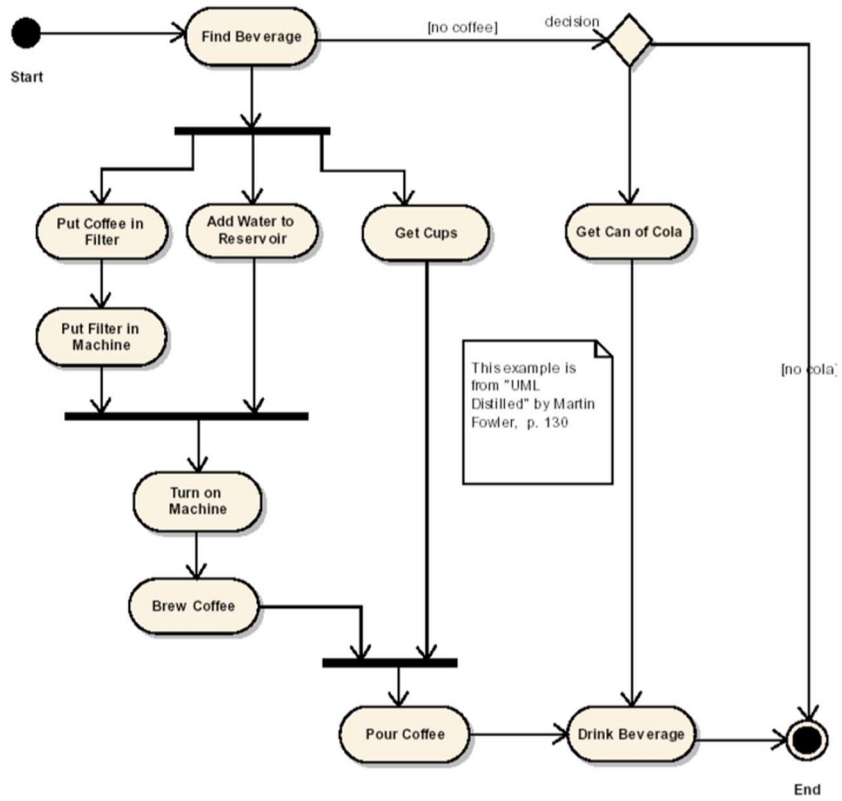
(Sumber : http://www.tutorialkampus.com/2014/06/perancangan-website-mts-nuur.html)

2.2.1.4 Activity Diagram

Activity diagram 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.

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.



Gambar 2. 9 Contoh Activity Diagram
(Sumber : http://www.tutorialkampus.com/2014/06/perancangan-website-mts-nuur.html)

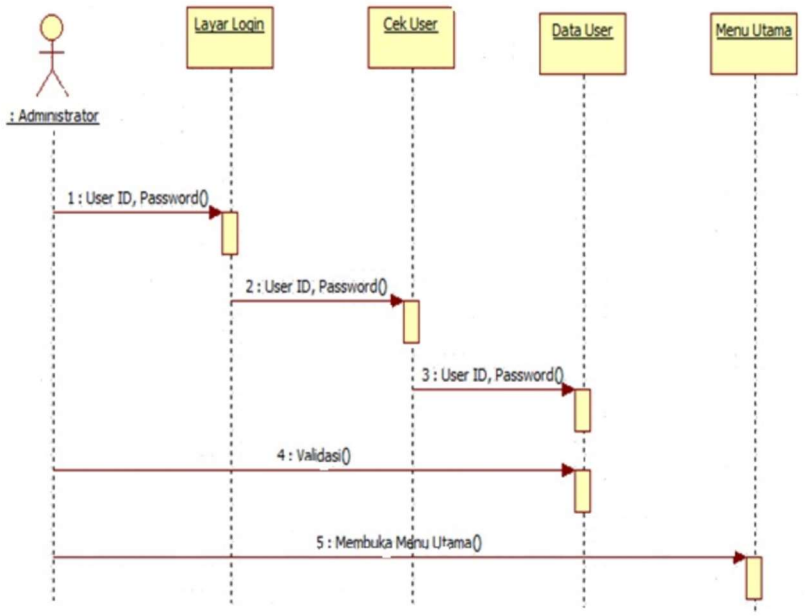
2.2.1.5 Sequence Diagram

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 atar 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.



Gambar 2. 10 Contoh Sequence Diagram
(Sumber : http://www.tutorialkampus.com/2014/06/perancangan-website-mts-nuur.html)

2.2.2 Basis Data

Sistem Manajemen basis data (*Database Management System*) atau DBMS adalah perangakat lunak yang dirancang untuk membantu dalam hal pemeliharaan dan penggunaan kumpulan data dalam skala besar. Model yang digunakan dalam desain basis data adalah teknik normalisasi. Proses normalisasi memiliki beberapa istilah yang dipakai yaitu:

1. Entity

   Entity adalah orang, tempat, kejadian atau konsep yang informasinya terekam.
2. Attribute

   Attribute adalah label yang digunakan untuk mewakili suatu *entity*
3. Data Value (Isi Data)

Data value adalah data aktual atau informasi yang disimpan pada setiap data *element* atau *attribute*.

4. Record

   Record adalah kelompok elemen yang saling terkait untuk menginformasikan suatu entity secara lengkap.

5. File

   File adalah kelompok dari beberapa record yang sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, namun meliki data *value* yang berbeda.

2.3 Teori Pengujian

2.3.1 Black Box Testing

Black box testing adalah pengujian pada perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian bertujuan untuk mengetahui apakah fungsi, masukan (*input*), dan keluaran (*output*) dari perangkat lunak yang dibangun sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan menyiapkan kasus uji yang berperan untuk mencoba semua fungsi dengan menggunakan perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang disiapkan untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah[22].

Black box testing juga disebut sebagai pengujian tingkah laku, yang memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan mendapatkan serangkaian kondisi masukan (*input*) yang seluruhnya memakai semua persyaratan fungsional untuk suatu program[23]. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data, kesalahan performasi, kesalahan inisialisasi dan akhir program.

Pada black box testing terdapat jenis teknik design tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, diantaranya sebagai berikut:

1. Equivalence Class Partitioning

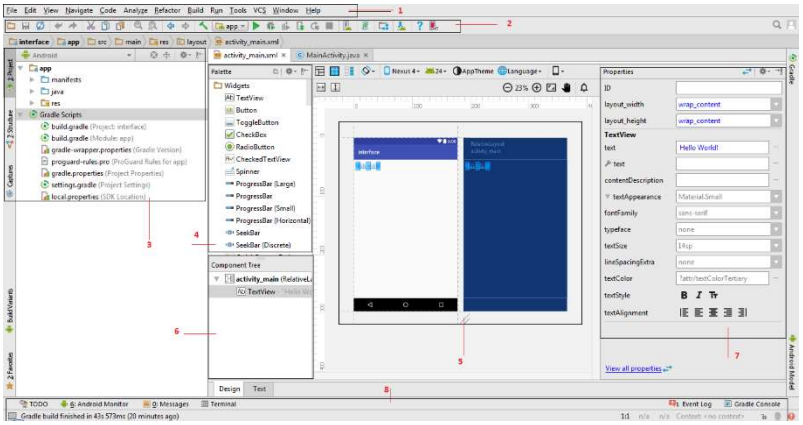
2. Boundary Value Analysis
3. State Transitions Testing
4. Cause-Effect Graphing

2.4 Teori Program

2.4.1 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment* atau IDE) yang resmi untuk pengembangan aplikasi *android*, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang mumpuni, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas programmer dalam membuat aplikasi *android*, antara lain[27]:

- Sistem build berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan terpadu sebagai tempat mengembangkan aplikasi untuk semua perangkat *android*
- Menerapakan perubahan untuk melakukan push pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Template kode dan integrasi GitHub untuk membantu membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan fitur pengujian yang lengkap
- Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Mendukung C++ dan NDK
- Mendukung bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine
- Halaman ini menyediakan pengantar fitur-fitur dasar Android Studio. Untuk ringkasan perubahan terbaru, lihat catatan rilis Android Studio.



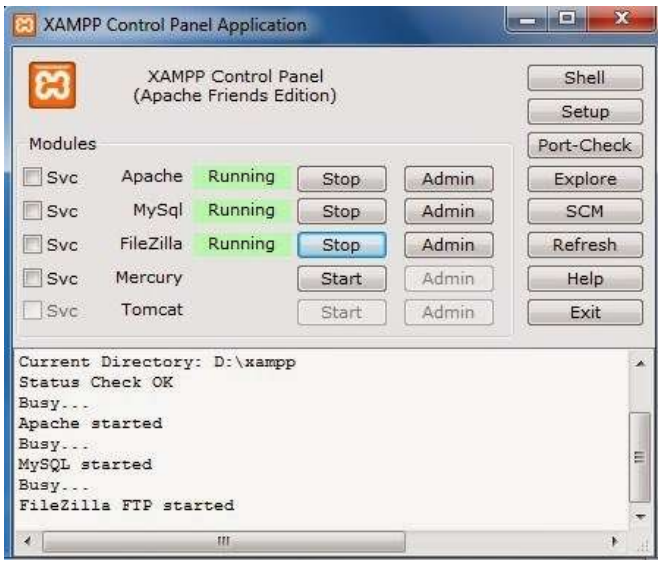
Gambar 2. 11 Tampilan Antarmuka Android Studio

(Sumber : www.badoystudio.com)

2.4.2 XAMPP

XAMPP adalah paket program *web* lengkap yang dapat dipakai untuk belajar pemograman *web*, khususnya PHP dan MySQL. XAMPP adalah perangkat lunak bersifat bebas, yang mendukung cukup banyak sistem operasi, merupakan kompilasi dari beberapa program. Berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Bagian penting dari XAMPP yang diasa digunakan[28]:

- Htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
- PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat http://localhost/phpMyAdmin, maka akan muncul halaman phpMyAdmin.
- Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*)



Gambar 2. 12 Tampilan XAMPP

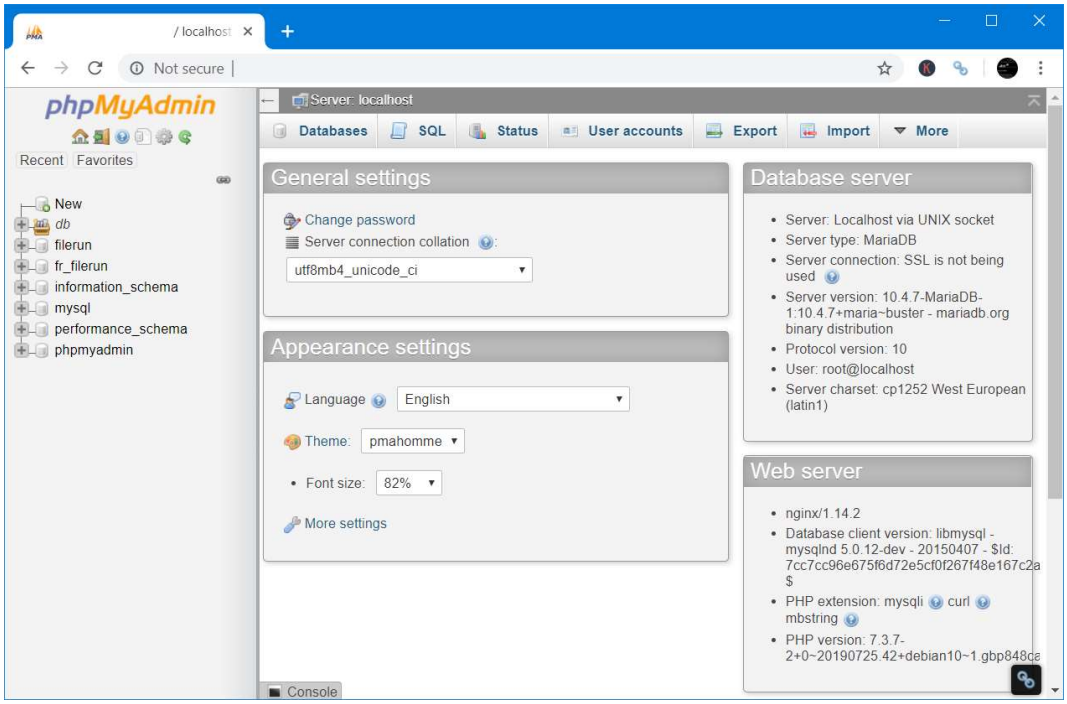
(Sumber : www.davaradise.com)

2.4.3 PhpMyAdmin

PhpMyAdmin adalah aplikasi manajemen *database server* MySQL berbasis *web*. Dengan aplikasi phpMyAdmin pengguna dapat mengelola database sebagai root (pemilik *server*) atau juga sebagai user biasa. Pengguna bisa membuat *database* baru, mengelola *database* dan melakukan operasi berbagai macam perintah database secara lengkap.

PhpMyAdmin adalah interface web yang dibuat untuk mengelola database MySQL. PhpMyAdmin dibuat menggunakan bahasa PHP dan bersifat *open source*. Dengan phpMyAdmin, *administrator web server* dapat mengelola *database* tanpa harus menguasai perintah berbasis baris teks (*command line*) dari SQL (*Structure Query Language*). PhpMyAdmin pada umumnya digunakan pengembang *web* untuk menyiapkan *database* dari aplikasi *web* seperti CMS, Blog dan sebagainya. Beberapa fitur penting dari phpMyAdmin antara lain:

1. Membuat, menghapus dang mengedit baik database, tabel, record, struktur.
2. Membuat pencarian sederhana dan kompleks
3. Mengimpor CVS (bisa digunakan untuk mengimpor data *spreedsheet*)
4. Mengekspor ke CVS, XML, Pdf, *spreedsheet*.



Gambar 2. 13 Tampilan Antarmuka PhpMyAdmin

(Sumber : www.nesabamedia.com)

2.4.4 Web Server

Web adalah antarmuka pada *browser* dengan alamat *domain* khusus. *Web* dapat dirancang dengan menggunakan bahasa HTML dan PHP menggunakan *style* tampilan menggunakan bahasa CSS. *Web* yang terlah dirancang tersebut disimpan pada satu komputer yang dinamakan *server*. *Server* menyimpan program *web* dan *database* untuk mendapatkan akses oleh admin atau *client* dari *browser*. Program notepad/notepad++ atau adobe dreasweaver dapat digunakan untuk membangun *website*. *Web server* adalah sebuah *software* yang menyediakan layanan berbasis data yang berguna untuk menerima permintaan dari HTTP atau HTTPS pada *client* yang dikenal dengan nama *web browser* dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen HTML. Dalam bentuk sederhana *web server* akan mengirim data HTML kepada permintaan *web browser* sehingga akan terlihat seperti pada umumnya yaitu sebuah tampilan *website*. Kegunaan utama dari *web server* adalah untuk melakukan *tranfer* berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sebelumnya. Halaman *web* yang diminta terdiri dari berkas teks, video,

gambar, file dan banyak lagi. *Web server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk yang di dalam berupa teks, video, gambar atau banyak lagi. Beberapa Jenis *Web Server* di antanya adalah:

1. Apache Web Server / The HTTP Web Server
2. Apache Tomcat
3. Microsoft windows Server 2008 IIS (Internet Information Services)
4. Lighttpd
5. Zeus Web Server
6. Sun Java System Web Server