

BAB II

LANDASAN TEORI

II.1 Konsep Dasar Sistem

II.1.1 Pengertian Sistem

Secara umum arti sistem adalah suatu kesatuan atau kelompok dari bagian-bagian tertentu yang saling berhubungan dengan mencapai tujuan tertentu.

Menurut Andri Kristanto^[4]

“Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.

Suatu sistem yang baik harus mempunyai tujuan dan sasaran yang tepat karena hal ini akan sangat menentukan dalam mendefinisikan masukan yang dibutuhkan sistem dan juga keluaran yang dihasilkan menurut Andri Kristanto^[4]

II.1.2 Elemen-Elemen Sistem

Sistem mempunyai elemen-elemen yang mendukung sebuah sistem itu sendiri berupa:

1. Tujuan Sistem

Tujuan sistem merupakan tujuan dari sistem tersebut disebut. Tujuan sistem dapat berguna tujuan organisasi, kebutuhan organisasi, permasalahan yang ada dalam suatu organisasi maupun urutan prosedur untuk mencapai tujuan organisasi.

2. Batasan Sistem

Batasan sistem merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem. Batasan sistem dapat berupa peraturan-peraturan yang ada dalam suatu organisasi, biaya-biaya yang dikeluarkan, orang-orang yang ada dalam organisasi, fasilitas baik itu sarana dan prasarana maupun batasan yang lain.

3. Kontrol Sistem

Kontrol atau pengawasan sistem merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol sistem dapat berupa kontrol terhadap pemasukan data (*input*), control

terhadap keluaran data (*output*), kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

4. Input

Input merupakan elemen dari sistem yang bertugas untuk menerimaseseluruh masukan data, dimana masukan tersebut dapat berupa jenisdata, frekuensi pemasukan data dan sebagainya.

5. Proses

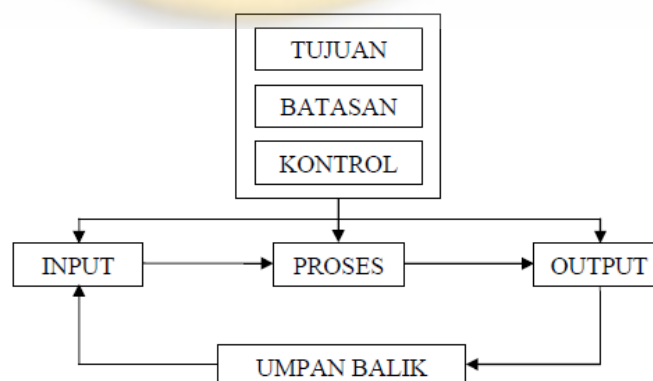
Proses merupakan elemen dari sistem yang bertugas untuk mengolahatau memproses seluruh masukan data menjadi suatu yang lebihberguna. Misalkan sistem produksi akan mengolah bahan baku yangberupa bahan mentah menjadi bahan jadi yang siap untuk digunakan.

6. Output

Output merupakan hasil dari inout yang telah diproses oleh bagianpengolah dan merupakan tuijuan akhir sistem.Output ini bisa berupalaporan grafik, diagram batang dan sebagainya.

7. Umpan Balik

Umpan balik merupakan elemen dalam sistem yang bertugasmengevaluasi bagian dari output yang dikeluarkan, dimana elemenini sangat penting demi kemajuan sebuah sistem. Umpan balik inidapat merupakan perbaikan sistem, pemeliharaan sistem dansebagainya.



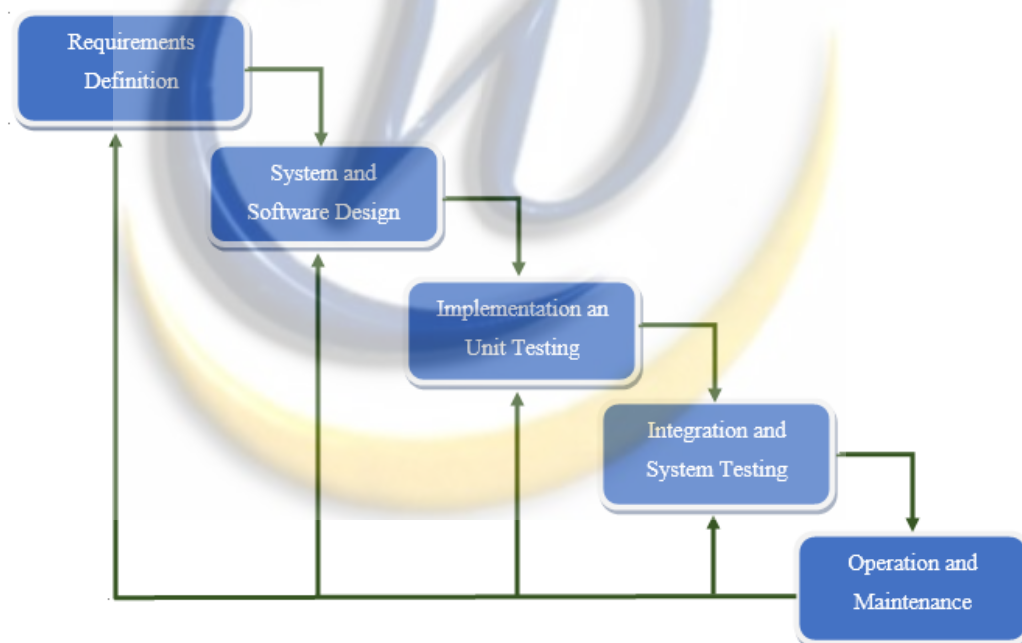
Gambar II.1 – Siklus Informasi^[4]

II.2 Konsep Dasar Perancangan

II.2.1 *Waterfall Model*

Waterfall Model merupakan dasar dari aktivitas proses yang terdiri dari spesifikasi, pengembangan, validasi, evolusi dan semua direpresentasikan dalam tahapan proses yang terpisah seperti spesifikasi kebutuhan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya^[5].

Model dari *software development process* ini adalah model yang pertama kali dipublikasikan yang diperoleh dari *system engineering process* yang umum. Karena satu tahap ke tahap lainnya mengalir ke bawah, model ini disebut sebagai *Waterfall Model*. *Waterfall Model* ini adalah contoh dari sebuah proses yang bersifat *plan-driven* - secara prinsip, semua aktivitas proses harus direncanakan dan diproses terlebih dahulu sebelum mulai mengerjakannya^[5].



Gambar II.2 – *Waterfall Model*^[5]

Tahapan dari *Waterfall Model* merefleksikan pokok-pokok dari aktivitas pengembangan^[5]:

1. *Requirements Analysis and Definition*

Layanan yang diberikan oleh sistem, batasan sistem, dan tujuan ditetapkan setelah melakukan konsultasi dengan pengguna sistem. Semua didefinisikan

secara rinci dan dibuat sebagai spesifikasi dari sistem.

2. *System and Software Design*

Proses perancangan sistem menyediakan kebutuhan hardware atau software dengan menyediakan arsitektur dari keseluruhan sistem. Perancangan sistem melibatkan pengidentifikasian dan penjelasan dari abstraksi sistem dan hubungannya.

3. *Implementation and Unit Testing*

Pada tahap ini, perancangan sistem direalisasikan menjadi sebuah program atau unit program. Pengujian unit melibatkan verifikasi untuk memastikan apakah setiap unit memenuhi spesifikasi sistem.

4. *Integration and System Testing*

Setiap unit program dan program-program yang sudah ada diintegrasikan dan diuji sebagai satu keutuhan sistem untuk memastikan apakah kebutuhan sistem sudah terpenuhi. Setelah melakukan pengujian, sistem baru disembarkan ke pengguna.

5. *Operation and Maintenance*

Dilakukan instalasi terhadap sistem dan digunakan dalam prakteknya. *Maintenance* melibatkan koreksi terhadap *error* yang tidak ditemukan pada tahap sebelumnya, memperbaiki implementasi dari unit sistem dan meningkatkan layanan yang diberikan oleh sistem sebagai kebutuhan baru yang ditemukan.

Waterfall model digunakan hanya jika semua kebutuhan sudah dimengerti dan tidak berubah secara radikal pada tahap pengembangan^[5].

II.2.2 Object Oriented Programming (OOP)

Object Oriented Programming adalah paradigma pemrograman yang memandang perangkat lunak sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem^[2]. Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing objek harus berisikan informasi mengenai dirinya sendiri (*encapsulation*) dan objek yang dapat dikaitkan (*inheritance*)^[3].

Dalam OOP, *Class* merupakan sekumpulan objek yang memiliki atribut-atribut dan *method*^[10]. *Class* merupakan deskripsi dari satu atau lebih objek yang memiliki kesamaan atribut, layanan, metode, hubungan, dan semantik, termasuk deskripsi cara membuat objek baru dalam *class*. Ada juga yang disebut dengan *super class*, sebuah *class* induk yang nantinya mempunyai *class-class* yang terdiri dari *class* dan *subclass*^[10].

Objek dalam OOP adalah sebuah benda atau unit atau sifat kerja yang memiliki atribut-atribut^[10]. Objek adalah sebuah abstraksi dari sesuatu pada domain masalah, menggambarkan kemampuan untuk menyimpan informasi mengenai hal tersebut, berinteraksi dengan hal tersebut atau keduanya.

Abstraksi prosedural dalam OOP disebut dengan operasi, yang menspesifikasi tipe dari perilaku dan terdiri dari fungsi-fungsi^[10]. Istilah lain terdapat *encapsulation*/pengkapsulan, yang merupakan pembatasan ruang lingkup program terhadap data yang diproses supaya data terlindungi oleh prosedur atau objek lain, kecuali prosedur yang berada di objek itu sendiri^[10].

Polymorphism adalah konsep yang menyatakan bahwa sesuatu yang samadapat mempunyai bentuk dan perilaku yang berbeda, bahwa operasi yang sama mungkin memiliki perbedaan dalam *class* yang berbeda^[10].

Pada OOP, terdapat juga yang disebut dengan *inheritance* (pewarisan), yaitu kepemilikan yang bersifat *implicit* dari fitur *subclass* yang didefinisikan dalam *superclass*. Fitur tersebut mencakup *variables* dan *method*^[10].

Konsep dasar dari Pemrograman Berorientasi Objek Pemrograman orientasi-objek menekankan konsep berikut:

- a. Kelas - kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. *Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang*

terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

- b. Objek - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- c. Abstraksi - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
- d. Enkapsulasi - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
- e. Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang

berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

- f. Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

II.3 Unified Modelling Language (UML)

UML adalah bahasa standar pemodelan untuk perangkat lunak dan pengembangan sistem^[5].

Merancang sebuah desain untuk sistem yang besar merupakan hal yang sulit. Dari aplikasi *desktop* yang sederhana sampai sistem *multi-tier* dapat dibangun dari ratusan, bahkan ribuan, komponen perangkat lunak dan perangkat keras. Untuk melacak komponen apa yang dibutuhkan, apa yang dikerjakan, bagaimana mencapai kebutuhan pelanggan, dan bagaimana menjelaskan sistem ke tim kerja lainnya dibutuhkan sebuah pemodelan, ini adalah dimana UML dibutuhkan.

Dalam merancang sebuah sistem, mengatur kompleksitas adalah salah satu alasan utama mengapa harus membuat model. Pemodelan membantu para pengembang untuk dapat fokus, dapat mendokumentasikan, menangkap keseluruhan sistem dan mengkomunikasikan aspek-aspek penting dalam sistem

yang sedang dirancang.

UML tepat digunakan untuk memodelkan sistem dari mulai memodelkan informasi sistem untuk perusahaan hingga aplikasi *web*, bahkan untuk sistem yang rumit sekalipun. UML menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk perancangan aplikasi Berbasis PHP.

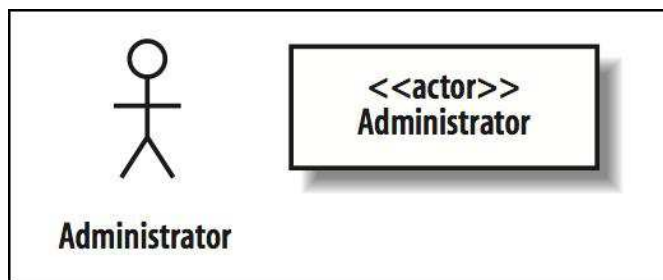
UML mendefinisikan diagram-diagram, antara lain:

1. *Use case diagram*
2. *Activity diagram*
3. *Class diagram*
4. *Sequence diagram*
5. *Collaboration diagram*
6. *Statechart diagram*
7. *Component diagram*
8. *Deployment diagram*
9. *Package diagram*

II.3.1 Use Case Diagram

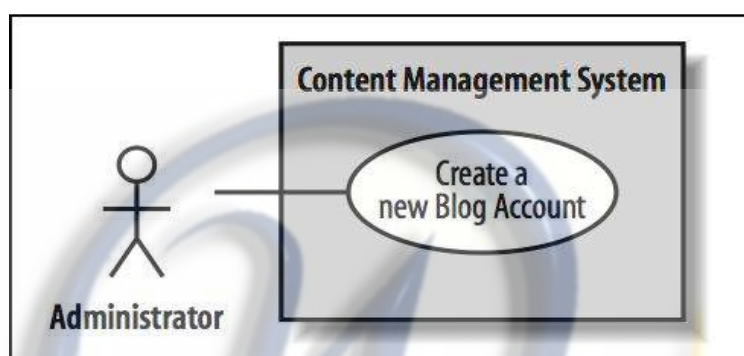
Use Case diagram menunjukkan bagaimana sistem yang sedang dibangun digunakan untuk memenuhi kebutuhan pengguna. Sebuah *usecase* menampilkan sebuah fungsionalitas yang diberikan oleh sistem^[5]. Yang ditekankan pada *use case* adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

Diluar sistem terdapat aktor yang digambarkan dengan istilah “*stick man*” atau sebuah box. Seperti yang terdapat pada Gambar II.5, dimana terdapat dua pilihan untuk menggambarkan seorang aktor.



Gambar II.3 – Notasi Actor Pada Use Case^[5]

Selain aktor, terdapat sebuah *use case*, *communication line*, dan *system boundaries* untuk menggambarkan *use case* diagram secara utuh. Sebuah *use case* menampilkan bagaimana sistem digunakan untuk menyelesaikan pekerjaan aktor. *Communication line* menghubungkan aktor dan *use case* untuk menunjukkan bahwa aktor tersebut berpartisipasi di dalam *use case*. Sedangkan *system boundaries* digunakan untuk menandakan pemisahan antara eksternal sistem (aktor) dan internal sistem (*use cases*). Berikut ini adalah gambar dari gabungan antara aktor, *use case*, *communication line* dan *system boundaries*^[5].



Gambar II.4 - Contoh *Use Case*^[5]

Use case diagram yang menunjukkan *use cases* dan aktor merupakan awalan yang baik, namun diagram tersebut tidak menampilkan rincian yang cukup untuk mengerti sistem. Cara yang paling baik untuk mengekspresikan informasi tersebut adalah dalam bentuk tulisan yang disebut dengan *Use Case Description*. Gambar-gambar berikut mencontohkan sebuah *Use Case Description* berdasarkan contoh *use case* sebelumnya :

Use case name	Create a new Blog Account
Related Requirements	Requirement A.1.
Goal In Context	A new or existing author requests a new blog account from the Administrator.
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.
Successful End Condition	A new blog account is created for the author.
Failed End Condition	The application for a new blog account is rejected.
Primary Actors	Administrator.
Secondary Actors	Author Credentials Database.
Trigger	The Administrator asks the CMS to create a new blog account.

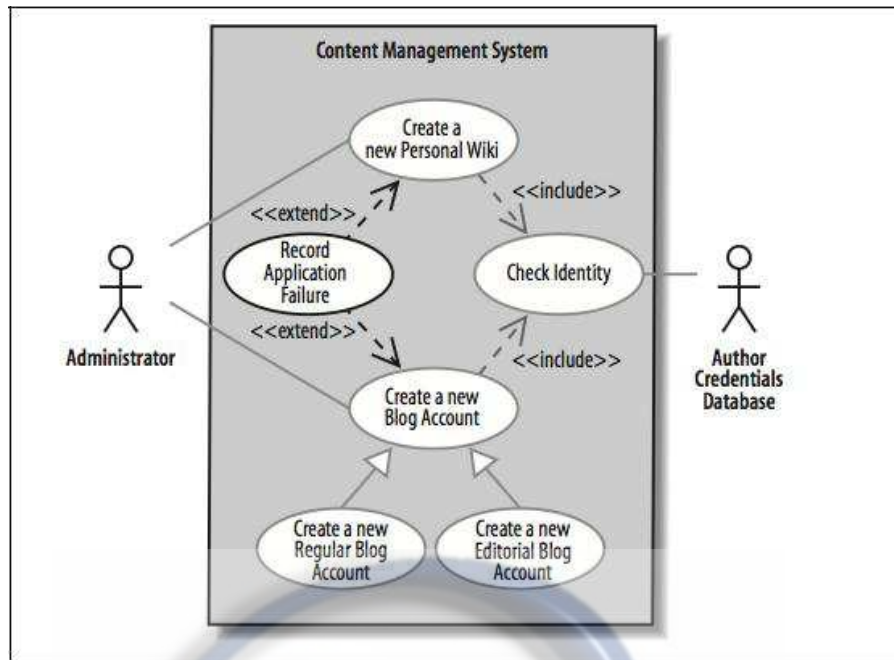
Gambar II.5 - Contoh *Use Case Description*^[5]

Use case name	Create a new Blog Account	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
Extensions	6	A summary of the new blog account's details are emailed to the author.
	Step	Branching Action
	4.1	The Author Credentials Database does not verify the author's details.
	4.2	The author's new blog account application is rejected.

Gambar II.6 - Contoh Use Case Description (Lanjutan)^[5]

Hubungan antar *use case* dapat diekspresikan menggunakan `<<include>>` dan `<<extend>>`. Relasi `<<include>>` digunakan untuk memecahkan sistem menjadi bagian-bagian yang lebih mudah untuk diatur. Sedangkan relasi `<<extend>>` adalah sebuah cara untuk menunjukkan bahwa sebuah *use case* mungkin dapat sepenuhnya menggunakan *behavior* dari *use case* lain. Selain itu *use case diagram* juga dapat menggunakan generalisasi untuk beberapa kasus yang berbeda, misalnya terdapat beberapa tipe dalam membangun sebuah *blog*. *Use case* "Create a new Blog Account" bisa digeneralisasikan menjadi dua tipe, yaitu "Regular Blog Account" atau "Editorial Blog Account"^[5].

Berikut adalah contoh *use case diagram* secara utuh:



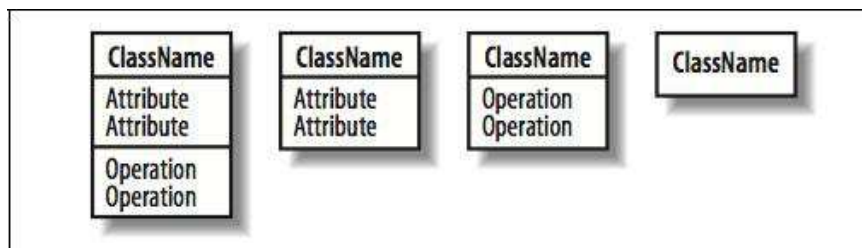
Gambar II.7 - Diagram Use Case^[5]

II.3.2 Class Diagram

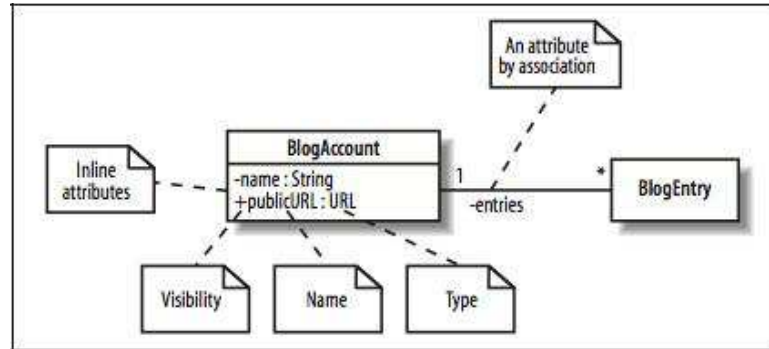
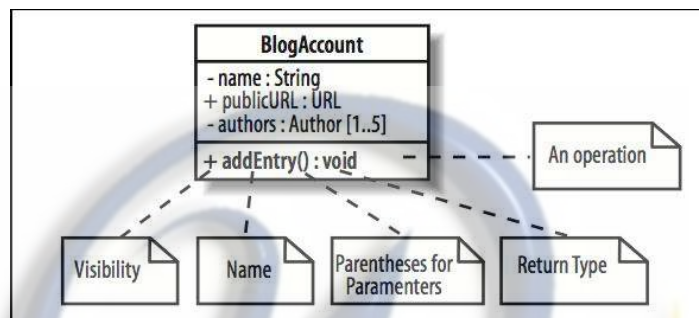
Class bisa disebut sebagai bagan yang digunakan untuk membangun dan mendefinisikan objek-objek. Sehingga setiap hal yang dibangun dari sebuah class disebut sebagai objek atau *instances*.

Deskripsi dari sebuah class mencakup dua bagian: informasi yang akan dimiliki oleh objek dan *behaviour* yang akan didukung objek tersebut. Ini yang membedakan *object-oriented* dengan yang lainnya, bahwa *state information* dan *behaviour* digabungkan dalam satu definisi class, dimana objek-objek dapat terbentuk dari class tersebut^[5].

Class dalam UML digambarkan sebagai persegi panjang dibagikan menjadi tiga bagian. Bagian paling atas berisi nama class, bagian tengah berisi atribut atau informasi yang dimiliki oleh class tersebut, dan bagian akhir berisi operasi yang menunjukkan *behaviour* dari class^[5].



Gambar II.8 - Class Diagram^[5]

Gambar II.9 - Class State : Attribute^[5]Gambar II.10 - Class Behaviour : Operation^[5]

Class mempunyai sifat *visibility*, sebuah sifat yang dapat digunakan untuk mengatur akses ke atribut dan metode atau *operation*, yaitu :

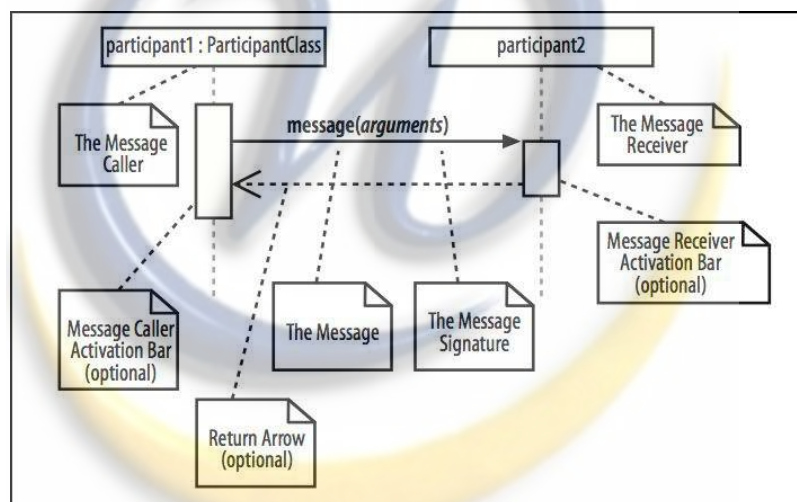
1. *Private*, hanya dapat digunakan oleh metode pada *class* tersebut, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan (metode) dan anak-anak yang mewarisinya (*generalization*).
3. *Public*, dapat dipanggil oleh siapa saja.
4. *Package*, dapat dipanggil oleh *class* yang bersangkutan (metode) dan *class* lain yang berada di dalam *package* yang sama.

II.3.3 Sequence Diagram

Sequence diagram memodelkan bagaimana sistem yang sedang dibangun mendatangi tugasnya. Diagram ini menangkap semua urutan interaksi pada bagian-bagian dari sistem. Dengan menggunakan *sequence diagram*, pengembang bisa menjelaskan interaksi apa yang akan dipanggil ketika sebuah *use case* dieksekusi dan dalam urutan seperti apa interaksi-interaksi tersebut akan terjadi^[5].

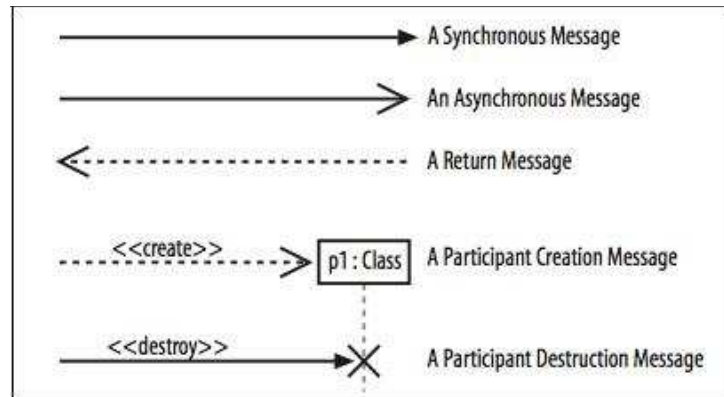
Dalam *sequence diagram* terdapat bagian-bagian yang disebut dengan

participant, *time*, *message*, dan *activation bars*. *Participant* adalah bagian-bagian dari sistem yang berinteraksi dengan satu sama lain selama *sequence* berjalan. *Time* menunjukkan urutan dimana semua interaksi berlangsung sesuai dengan waktu. *Time* pada *sequencediagram* ditunjukkan dengan garis titik-titik vertikal. *Message* dalam *sequence* ditunjukkan menggunakan panah dari *participant* yang ingin mengirimkan pesan ke *participant* lain. *Message Caller* adalah sebutan untuk *participant* yang mengirim *message* dan *Message Receiver* untuk *participant* yang menerima *message*. *Activation bars* menunjukkan bahwa sebuah *participant* sedang melakukan sesuatu untuk jangka waktu tertentu. *Activation bars* dapat ditampilkan pada awal pengiriman *message* dan ujung dari penerimaan *message*. Hal ini menunjukkan bahwa *Message Caller* sibuk pada saat mengirim *message* dan *MessageReceiver* sibuk pada saat *message* sudah diterima^[5].



Gambar II.11 - Contoh Interaksi Pada *Sequence Diagram*^[5]

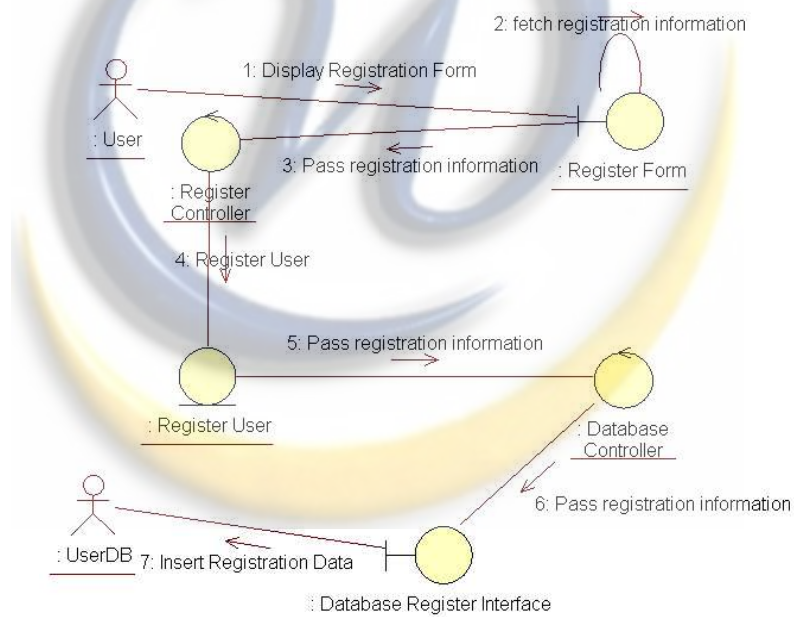
Dan berikut ini adalah bentuk-bentuk dari panah pada *sequencediagram*, setiap bentuk panah mempunyai arti yang berbeda.



Gambar II.12 - Bentuk - Bentuk Panah Pada *Sequence Diagram*^[5]

II.3.4 Collaboration Diagram

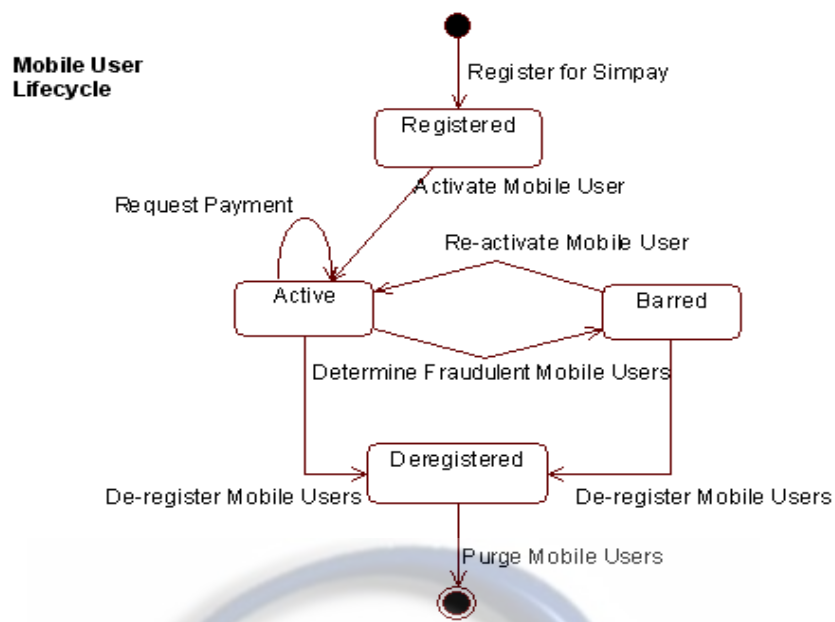
Mendefinisikan peran-peran yang dimainkan ketika satu tugas dilakukan. Peran-peran dimainkan oleh instant-instant yang berinteraksi, dan menyatakan hubungan komunikasi antara objek-objek yang menunjuknya.



Gambar II.13 - Contoh *Collaboration Diagram*^[5]

II.3.5 Statechart Diagram

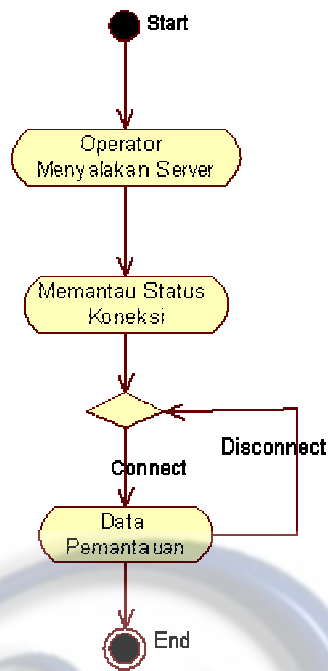
Merupakan perluasan dari *diagram state* yang mendeskripsikan objek berupa *state-state* yang dimiliki, kejadian-kejadian yang berlangsung beserta transisi yang terjadi.



Gambar II.14 - Contoh *Statechart Diagram*^[5]

II.3.6 Activity Diagram

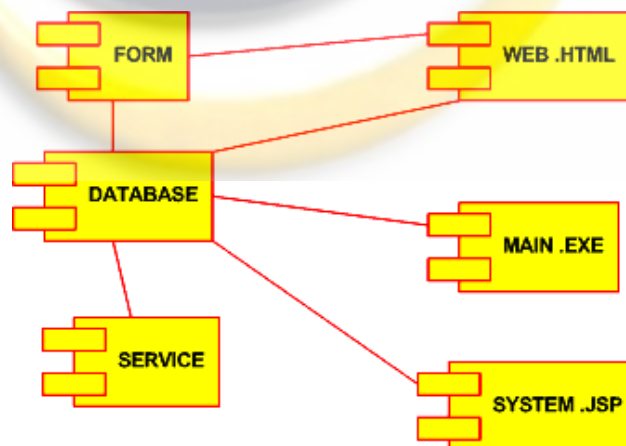
Adalah Representasi secara grafis dari proses dan *control flow* dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks.



Gambar II.15 - Contoh Activity Diagram^[5]

II.3.7 Component Diagram

Adalah *Diagram* yang menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem.

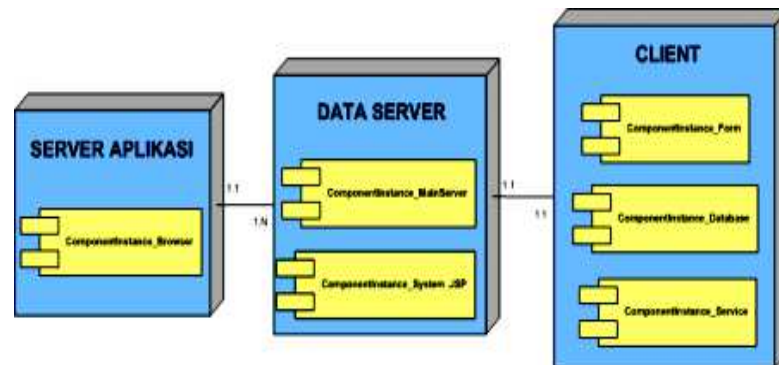


Gambar II.16 - Contoh Component Diagram^[5]

II.3.8 Deployment Diagram

Adalah penggambaran tugas-tugas kongkrit dari setiap *node/software* yang terlibat dalam jaringan sistem, menampilkan keseluruhan *node* dalam jaringan

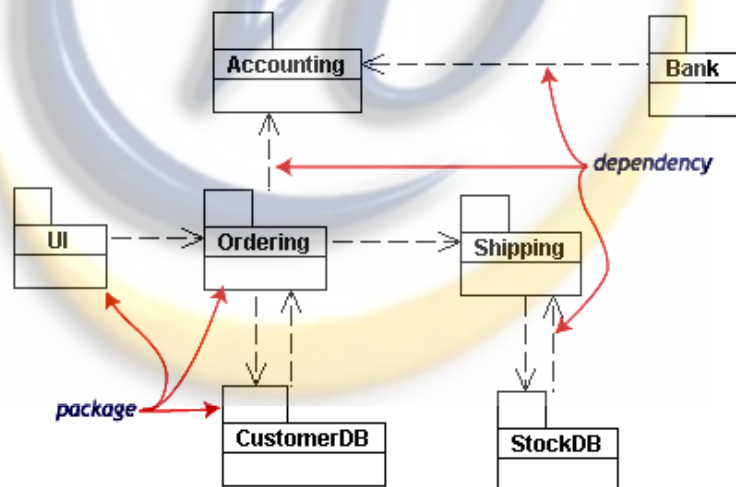
serta hubungan dari *node-node* tersebut termasuk proses-proses yang terlibat di dalamnya.



Gambar II.17 - Contoh *Deployment Diagram*^[5]

II.3.9 Package Diagram

Adalah Penggambaran dan pengelompokan kelas-kelas yang terdapat pada perangkat lunak yang dibangun sesuai dengan fungsi atau subsistem aplikasi yang mempunyai ketergantungan satu sama lainnya.



Gambar II.18 - Contoh *Package Diagram*^[5]

II.4 Internet

II.4.1 Pengertian Internet

Internet merupakan singkatan dari *Interconnection Networking*. Internet berasal dari bahasa latin “*inter*” yang berarti antara. Secara kata perkata INTERNET berarti jaringan antara atau penghubung, sehingga kesimpulan dari definisi internet ialah merupakan hubungan antara berbagai jenis komputer dan

jaringan di dunia yang berbeda sistem operasi maupun aplikasinya dimana hubungan tersebut memanfaatkan kemajuan komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP (*Transmission Control/Internet Protocol*) pengertian tersebut dijelaskan menurut pendapat Supriyanto^[8].

Secara sederhana menurut Harjono:

Internet dapat diartikan kumpulan dari beberapa komputer, bahkan jutaan komputer di seluruh dunia yang saling berhubungan atau terkoneksi satu sama lainnya. Media yang digunakan bisa menggunakan kabel/serat optic, satelit atau melalui sambungan telepon.

Pendapat ini mengartikan bahwa internet merupakan media komunikasi dan informasi modern yang dapat dimanfaatkan secara global oleh pengguna diseluruh dunia dalam interkoneksi antar jaringan komputer yang terbentuk melalui sarana berupa penyedia akses (*provider*) internet, sehingga internet sebagai media informasi dapat menjadi sarana yang efektif dan efisien untuk melakukan pertukaran dan penyebaran informasi tanpa terhalang oleh jarak, perbedaan waktu dan juga faktor geografis bagi seseorang yang ingin mengakses informasi.

Model koneksi internet itu sendiri dapat dilakukan pada komputer pribadi maupun jaringan LAN/WAN. Defenisi LAN/WAN menurut Nugroho^[1], antara lain :

LAN (*Local Area Network*) suatu jaringan yang terbentuk dengan menghubungkan beberapa komputer yang berdekatan yang berada pada suatu ruang atau gedung yang terkoneksi ke internet *gateway*. WAN (*Wide Area Network*) adalah format jaringan dimana suatu komputer dihubungkan dengan yang lainnya melalui sambungan telepon. Data dikirim dan diterima oleh atau dari suatu komputer ke komputer lainnya lewat sambungan telepon. Konektor komputer dengan telepon adalah menggunakan modem (*Modulator, Demodulator*).

Internet merupakan perpustakaan multimedia yang sangat lengkap, bahkan internet dipandang sebagai dunia dalam bentuk lain (maya), dikatakan demikian karena hampirseluruh informasi bidang ilmu pengetahuan disuguhkan internet, seperti bisnis, hiburan, lembaga pemerintahan maupun swasta dari seluruh Negara yang ada di dunia. Khususnya untuk bidang pendidikan, pengguna dapat memperoleh informasi mengenai universitas/institusi/akademik, lembaga-lembaga pendidikan, museum, perpustakaan, data bibilografi, seperti buku, jurnal, disertasi

dan lain sebagainya. Informasi mutakhir seperti surat kabar disetiap Negara untuk setiap harinya, serta seminar dan pertemuan ilmiah sejenis yang sedang berlangsung dan yang akan diselenggarakan, kemudian yang terpenting lagi adalah bahwa informasi di internet setiap hari bertambah terus dengan jumlah yang sangat banyak dan informasi yang disediakan selalu *up to date*.

II.4.2 Sejarah Perkembangan Internet

Sejarah internet bermula pada tahun 1969 ketika dibangun riset APRAnet yang dipelopori oleh DARPA (*Defense Advance Research Projects Agency*). Riset ini dikembangkan oleh Departemen Pertahanan Amerika Serikat yang bekerja sama dengan empat universitas, yaitu UCLA, Stanford Research Institued, UCSB dan University of Utah yang komputer servernya berada di UCLA. Tujuan pertama riset ini difokuskan pada keperluan militer Amerika Serikat. Dan pada tahun 1970, sudah lebih dari 10 komputer yang bergabung dan membentuk jaringan.

Di tahun 1972, Roy Tomlinson menyempurnakan program *e-mail* yang diciptakan untuk APRAnet. Pada tahun yang sama, *icon @* juga diperkenalkan sebagai lambang penting yang menunjukkan “at” atau “pada”. Tahun 1973 APRAnet mulai berkembang diluar AS dan komputer *University College* di London adalah merupakan komputer pertama yang bergabung pada jaringan APRAnet. Di tahun yang sama dua orang ahli komputer Vinton Cerf dan Bob Khan mempresentasikan ide mereka tentang internet di *University Sussex*.

Setahun setelah peristiwa keberhasilan Ratu Inggris mengirimkan *e-mail*, maka lebih dari 100 komputer bergabung di APRAnet membentuk sebuah jaringan atau *network*. Tahun 1979, Tom Triscott, Jim Ellis, dan Steve Bellovin menciptakan *newsgroup* pertama dengan nama *USEnet*. Kemudian tahun 1981, France Telecom berhasil menciptakan telepon televisi pertama, dimana orang bisa saling menelepon sambil berhubungan dengan *video link*.

Seiring perkembangannya, tahun 1982 DCA (*Defense Communication Agency*) bekerja sama dengan DARPA berhasil menciptakan TCP (*Transmission Control Protocol*) dan IP (*Internet Protocol*) untuk digunakan sebagai protokol utama pada ARPAnet. Kemudian pada tahun 1984 diperkenalkan sistem nama *domain* untuk menyeragamkan alamat jaringan komputer. Dan perkembangan

jaringan komputer bertambah menjadi 10.000 lebih. Tahun 1988, Jarko Oikarinen yang berasal dari Finlandia memperkenalkan IRC (*Internet Relay Chat*). Setahun kemudian tepatnya tahun 1987 tak kurang dari 100.000 komputer bergabung dan membentuk jaringan.

Program editor dan *browser* diperkenalkan oleh Tim Bernes Lee pada tahun 1990 program ini disebut dengan *www* atau *world wide web*. Pada tahun 1992 muncullah istilah-istilah lainnya seperti *surfing the internet*, *virtual shopping* atau *e-retail* dan kemudian ditahun yang sama didirikan juga yahoo dan *Netscape Navigator* (Haris, 1996).

Perkembangan teknologi *Internetwork* yang dimulai dari sejarah pendiriannya dan perkembangannya hingga saat ini benar-benar dapat dirasakan sangat bermanfaat dalam setiap aspek kehidupan. Sutedjo^[9] memaparkan sejarah masuknya internet di Indonesia, yaitu:

Di Indonesia, jaringan internet mulai dikembangkan pada tahun 1983 di Universitas Indonesia, berupa UInet oleh Dr. Joseph F.P Luhukay yang ketika itu baru saja menamatkan program dokter filosofi ilmu komputer di Amerika Serikat. Jaringan itu dibangun selama empat tahun. Pada tahun yang sama, Luhukay juga mulai mengembangkan University Network (UNInet) dilingkungan Departemen Pendidikan Dan Kebudayaan yang merupakan jaringan komputer dengan jangkauan yang lebih luas yang meliputi Universitas Indonesia, Institut Teknologi Bandung, Institut Pertanian Bogor, Universitas Gajah Mada, Institut Teknologi Surabaya, Universitas Hasanuddin dan Ditjen.

Jaringan internet telah menjadi pelopor terjadinya revolusi teknologi. Internet semakin diminati oleh banyak kalangan baik perorangan maupun instansi-instansi pemerintah ataupun swasta, termasuk diantaranya perpustakaan. Menurut Prasetyo^[6], hal-hal yang menyebabkan internet menjadi solusi praktis, sehingga diminati banyak kalangan, antara lain:

1. Audien Global.

Informasi yang dipublikasikan dalam internet, dengan segera tersedia ke seluruh pengguna audien secara global. Hal inilah yang membuat internet (*www*) menjadi media yang sangat efektif dari segi pembiayaan untuk

mempublikasikan informasi, kurang lebih mencapai hingga 190 negara di seluruh penjuru dunia.

2. Operasi Non–Stop.

Internet beroperasi non–stop selama 24 jam sehari, hal ini menjadikannya sebagai mesin bisnis yang sangat efisien sekali, sehingga anda tidak perlu lagi menunggu hingga sumber–sumber tersedia untuk menyelenggarakan bisnis anda, karena setiap saat anda dapat menggunakannya.

3. Murah.

Internet relatif lebih murah dibandingkan media lainnya hanya dengan biaya beberapa ratus ribu saja, anda dapat menampilkan informasi ke seluruh pengguna internet di dunia. Jelas sekali kalau internet merupakan media yang praktis dengan biaya yang terjangkau (murah).

4. Penyebaran Informasi.

Ketika informasi ditampilkan kedalam web, saat itu juga siap dinikmati oleh jutaan pengguna yang lain. Misalnya dengan media www, dimana hal ini sangat sulit dilakukan pada dunia nyata.

5. Alat Publikasi.

Tidak terlalu berlebihan jika internet merupakan suatu alat publikasi yang cukup andal. Sebagaimana diketahui, saat ini banyak sekali aplikasi berbasis internet yang telah dikomersialkan dan mudah sekali didapatkan.

Jaringan inetrnet sangat memberikan keuntungan yang beragam dimana dapat digunakan dan dimanfaatkan untuk membantu kegiatan berbagai aspek kehidupan. Keuntungan lain yang diberikan jaringan internet, sehingga membuat internet diminati yaitu internet dapat digunakan sebagai media konfrensi dimana sejumlah orang dapat melakukan diskusi tanpa harus bertatap muka secara langsung satu dengan lainnya.

II.5 Website

II.5.1 Pengertian Website

Dalam mengakses informasi dari dunia internet, pengguna akan menuju sebuah alamat *link* internet yang disebut nama *domain* (*Domain name/URL-Uniform ResourceLocator*) dan akan menemukan informasi berbentuk teks,

gambar, animasi, ataupun suaradalam sebuah media yang disebut dengan *website* atau situs. situs web ini dibuka melalusebuah program penjelajahan (*browser*) yang berada pada sebuah komputer. Program penjelajahan yang dapat digunakan dalam komputer agar dapat melakukan *browser* pada situs web diantaranya ada Internet Eksplorer (IE), Firefox, Mozila, Netscape, Safari dan Opera.

Pengertian Situs Web menurut Wikipedia dalam Zakaria^[11] adalah

A Website is a collection of web pages. A web pages is a document, typically written in HTML, that is almost always accessible via HTTP, a protocol that transfer information from the Situs Web's server to display in the user's web browser.

Pengertian di atas menjelaskan bahwa situs web terdiri dari halaman *web* yaitu sebuah dokumen yang ditulis dalam *Hyper Text Markup Language* (HTML) yang dapat diakses melalui *Protocol Hyper Text Transfer Protocol* (HTTP) yang merupakan protokol untuk menyampaikan informasi dari sebuah pusat situs web untuk ditampilkan dihadapan pengguna program pembaca informasi yang ada pada situs web.

Pendapat lain diutarakan oleh Saputro (2007), situs web adalah

Sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*Hyperlink*)”.

Dari pengertian di atas dapat dilihat bahwa informasi dalam *web* bersifat *multimedia* dan informasi yang didistribusikan melalui *hyperlinks*, dimana pengguna dapat mengakses informasi dengan cara *surfing* yaitu memperoleh informasi dengan meloncat dari satu halaman ke halaman lain tanpa batas ruang dan waktu. Situs Web memiliki sifat statis dan dinamis. Bersifat statis apabila isi informasi situs web tetap atau dikatakan juga jarang berubah dan isi informasinya searah hanya dari pemilik situs web. Sedangkan situs web dikatakan bersifat dinamis apabila isi informasinya selalu berubah-ubah dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna situs web. Situs Web dinamis dapat di *up date* oleh pengguna ataupun pemilik situs web. Contoh situs web statis adalah seperti *profile* perusahaan, instansi atau organisasi baik swasta

maupun pemerintah. Sedangkan contoh situs web dinamis adalah seperti *friendster, facebook, twitter*, dan lain sebagainya.

Untuk menyediakan sebuah situs web, maka kita harus menyediakan unsur-unsur penunjangnya. Unsur-unsur dalam penyedia situs web atau situs menurut Saputro (2007:1-2), antara lain:

1. Nama domain (*Domain name / URL – Uniform Resource Locator*)
2. Rumah tempat *Website (Web hosting)*
3. Bahasa Program (*Scripts Program*)
4. Desain *Website*
5. Program transfer data ke pusat data
6. Publikasi *Website*

1. Nama domain (*Domain name / URL – Uniform Resource Locator*)

Nama domain atau *domain name/URL* adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah situs web. Atau dengan kata lain domain name adalah alamat yang digunakan untuk menemukan sebuah situs web pada dunia internet. Nama domain diperjualbelikan secara bebas di internet dengan status sewa tahunan. Masa sewa dapat diperpanjang, namun apabila pengguna lupa maka nama domain itu akan lepas dari ketersediannya untuk umum.

Nama domain mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan situs web tersebut. Contoh nama domain yang ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Dan contoh nama domain ber-ekstensi lokasi negara indonesia adalah:

- a. .co.id : Untuk badan usaha, berbadan hukum sah
- b. .ac.id : Untuk lembaga pendidikan
- c. .go.id : Untuk lembaga pemerintahan RI
- d. .mil.id : Untuk lembaga militer RI
- e. .or.id : Untuk organisasi
- f. .war.net.id : Untuk industri warung internet di Indonesia
- g. .sch.id : Untuk lembaga pendidikan
- h. .web.id : Untuk badan usaha, organisasi atau yang sifatnya perorangan dalam rangka melakukan kegiatannya.

2. *Web Hosting*

Ini adalah tempat atau ruangan yang berada dalam *hardisk* yang berfungsi untuk menyimpan berbagai data, *file-file*, gambar, video, dan lain sebagainya yang akan diaplikasikan pada situs web. Kapasitas muatan data tergantung dari besarnya *web hosting* yang dipunyai. Biasanya pengguna akan memperoleh *control panel* yang terproteksi dengan *username* dan *password* untuk administrasi situs webnya.

Besarnya *hosting* ditentukan oleh kapasitas ruangan *hardisk* yang tersedia dalam ukuran MB (*Mega Byte*) atau GB (*Giga Byte*). Penyewaan *hosting* dilakukan dari perusahaan yang menyediakan jasa penyewaan *web hosting* yang banyak dijumpai baik di Indonesia maupun luar negeri.

3. Bahasa Program (*Scripts Program*)

Bahasa Program (*Scripts Program*) adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam situs web pada saat diakses. Jenis atau ragam bahasa program sangat menentukan statis, dinamis, atau interaktifnya suatu situs web. Jenis-jenis bahasa program yang banyak dipakai para *designer* situs web antara lain; Hyper Text Markup Language (HTML), Active Server Pages (Asp), Programming Hypertext Processor (PHP), Java Server Pages (JSP), Java Scripts, Java applets, Extensible Markup Language (XML), dan sebagainya.

Bahasa dasar situs web adalah bahasa HTML. Sedangkan bahasa program PHP, Asp, JSP, dan lainnya adalah bahasa pendukung sebagai pengatur dinamis dan interaktifnya suatu Situs Web dan dapat dibuat sendiri. Bahasa ini biasanya digunakan untuk membuat portal berita, artikel, forum diskusi, buku tamu, *email*, *mailing list*, dan sebagainya yang memerlukan *up date* setiap harinya.

4. Desain Website

Desain situs web menentukan kualitas suatu situs web. Desain memiliki pengaruh yang besar untuk kunjungan pengguna. Untuk membuat *design* situs web dapat dilakukan sendiri atau menyewa jasa *web designer*. Banyak program yang tersedia untuk membangun *design* situs web antara lain; Macromedia Firework, Adobe Photoshope, Adobe Dreamweaver, Microsoft FrontPage, dan lain sebagainya.

5. Program Transfer Data ke Pusat

Program transfer data ke pusat data maksudnya adalah *file-file* terpisah seperti bahasa program, data informasi teks, gambar dan lain sebagainya, dapat terpublikasi secara utuh pada situs web setelah melakukan program penjelajah (*browser*). Pengguna akan diberikan akses FTP (*File Transfer Protocol*) setelah memesan sebuah *web hosting* untuk dapat memindahkan *file-file* situs web ke pusat data *web hosting*. Untuk dapat menggunakan FTP diperlukan sebuah program FTP misalnya WS FTP, Smart FTP, Cute FTP, dan lain – lain.

6. Publikasi Website

Keberadaan situs web tidak ada gunanya apabila tanpa ada kunjungan oleh pengguna internet. Efektif tidaknya sebuah situs web tergantung dari besarnya pengunjung dan komentar yang masuk, sehingga dalam hal ini diperlukan publikasi atau promosi. Cara promosi yang efektif dengan tak terbatas ruang dan waktu atau dengan kata lain publikasi atau promosi dilakukan pada jaringan internet melalui *search engine* (mesin pencari, seperti; yahoo, google, MSN, *Search Indonesia*, dan sebagainya).

Keberadaan sebuah situs web didunia internet sangat membantu pengguna dalam menemukan informasi yang efektif dan efisien bagi penggunanya. Namun ternyata penyediaan situs web tidak semuanya bersifat gratis, situs web yang bertujuan bisnis misalnya penyediaannya bersifat komersial.

Untuk mendukung kelanjutan dari situs web diperlukan pemeliharaan setiap waktu yang sesuai dengan yang diinginkan. Pemeliharaan situs web bersifat penambahan informasi, berita, artikel, *link*, gambar, atau lain sebagainya pada *content* sebuah situs web.

Pemeliharaan situs web dapat dilakukan per periode tertentu seperti tiap hari, tiap minggu atau tiap bulan sekali secara rutin atau secara periodik tergantung kebutuhan (tidak rutin). Pemeliharaan rutin biasanya dipakai oleh situs web berita, penyedia artikel, organisasi atau lembaga pemerintah. Sedangkan pemeliharaan periodik biasanya untuk situs web pribadi, penjualan/*e-commerce* dan lain sebagainya.

Situs Web merupakan ruang informasi dalam internet. Pada internet terdapat berjuta situs web dengan banyak kepentingan informasi yang diberikan pada pengguna mulai dari situs web akademis, situs web yang menyajikan

informasi berita-berita seputar dunia, situs web *games* dan lain sebagainya. Berjuta pilihan disajikan sesuai dengan kepentingan pengguna, karena penerbitan sebuah situs web bertujuan untuk mempublikasikan berbagai informasi kegiatan dalam segala aspek kehidupan.

II.5.2 Sejarah Singkat *Website*

Sejarah situs web bermula di European Laboratory For Particle Physics (lebih dikenal dengan nama CERN). Penelitian ini terletak di kota Geneva dekat perbatasan Prancis dan Swiss. CERN adalah Organisasi Eropa untuk riset nuklir. Seorang fisikawan yang bernama Tim Berners Lee menulis proposal untuk manajemen informasi yang menunjukkan bagaimana informasi dapat dipindahkan dengan mudah melalui internet dengan menggunakan *hypertext*.

Perkembangan situs web tidak berhenti, selanjutnya seorang insinyur sistem yang bernama Robert Cailliau bergabung dengan idenya yaitu menghubungkan *hypertext* dengan internet dan komputer pribadi, sehingga mempunyai jaringan tunggal untuk membantu fisikawan CERN berbagi informasi yang tersimpan di laboratorium. *Hypertext* akan memungkinkan pengguna untuk *browsing* dengan mudah antara *text* pada halaman situs web dengan menggunakan *link*. Contoh pertama yang dikembangkan pada NEXT komputer.

Info.Cern.Ch adalah alamat situs web dan *web server* pertama di dunia yang berjalan pada sebuah komputer NEXT di CERN. Halaman (*Home Page*) pertama alamatnya <http://info.cern.ch/hypertext/www/TheProject.html>, yang terpusat pada informasi tentang proyek *www*. Pengunjung bisa belajar lebih banyak tentang *hypertext*, rincian teknis untuk menciptakan halaman *web* mereka sendiri, dan bahkan penjelasan tentang cara untuk mencari *web* untuk mendapatkan informasi.

Proyek pembuatan *web* oleh Tim Berners Lee menggunakan komputer NEXT yang digunakan pada tahun 1990. Komputer NEXT merupakan *server web hypermedia browser* dan *web editor* pertama yang ada dan digunakan untuk proyek pembuatan *web*.

Server mulai berkembang pada tahun 1991 dilembaga lain Eropa, dan Desember 1991 muncul server pertama diluar benua Eropa yang dipasang di AS di SLAC (*Stanford Linear Accelerator*). Pada bulan November 1992 sudah terdapat

26server di dunia dan pada bulan Oktober tahun 1993 angka penambahan server meningkat menjadi lebih dari 200 web server.

Pada Februari 1993, *National Center For Supercomputing Application* (NCSA) di University of Illinois di Urban Champaign menulis versi pertama dari *Mosaic* yang merupakan ketersediaan untuk membuat web bagi orang yang menggunakan PC dan *AppleMacintoshes*.

Konsep awal terciptanya sebuah web dimulai sebagai alat untuk membantu menjawab pertanyaan sulit fisika tentang alam semesta, namun sekarang penggunaannya berlaku untuk berbagai aspek untuk masyarakat global dan mempengaruhi kehidupan sehari-hari.

Di Akhir tahun 1995 jumlah situs web mencapai 300.000 situs web. Dan diperkirakan sekarang jumlah situs web berkisar ± 80 juta situs web dengan ratusan juta pengguna diseluruh penjuru dunia^[1].

Jumlah situs web kian tidak terbatas sampai dengan saat ini. Data dari Netcraft yang merupakan perusahaan internet monitoring dalam Fajriutami (2009:1) "Jumlah situs web di internet sampai dengan juli 2009 mencapai 239.611.111 (239,6 juta)". Pertambahan yang sangat meningkat disebabkan adanya fasilitas *blogger* yang dapat dimanfaatkan pengguna untuk *sharing* komunikasi pada pengguna lainnya dalam internet.

II.5.3 World Wide Web (WWW)

World Wide Web atau Waring Wera Wanua, biasa lebih terkenal disingkat sebagai WWW adalah suatu ruang informasi yang dipakai oleh pengenal global yang disebut Pengidentifikasi Sumber Seragam untuk mengenal pasti sumber daya berguna. WWW sering dianggap sama dengan Internet secara keseluruhan, walaupun sebenarnya ia hanyalah bagian daripada Internet.

WWW merupakan kumpulan peladen web dari seluruh dunia yang mempunyai kegunaan untuk menyediakan data dan informasi untuk dapat digunakan bersama. WWW adalah bagian yang paling menarik dari Internet. Melalui *web*, para pengguna dapat mengakses informasi-informasi yang tidak hanya berupa teks tetapi bisa juga berupa gambar, suara, video dan animasi.

Kegunaan ini tergolong masih baru dibandingkan surat elektronik, sebenarnya WWW merupakan kumpulan dokumen yang tersimpan di peladen *web*, dan yang peladennya tersebar di lima benua termasuk Indonesia yang terhubung menjadi satu melalui jaringan Internet. Dokumen-dokumen informasi ini disimpan atau dibuat dengan format HTML (*Hypertext Markup Language*).

II.5.4 HTTP

HTTP adalah sebuah protokol meminta/menjawab antara client dan server, sebuah client HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu client mengirim kode permintaan (request), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan data tertentu.

Beberapa kepala (header) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan error atau pesan lainnya.

II.5.5 Browser

Browser merupakan suatu program yang dirancang untuk mengambil informasi-informasi dari suatu *server* komputer pada jaringan internet. Tanpa adanya *browser* internet secara otomatis kita tidak bisa mengakses halaman *website*. Informasi ini biasanya dikemas dalam *page-page*, di mana setiap page bisa memiliki beberapa *link* yang menghubungkan *web page* tersebut ke sumber informasi lainnya dengan menggunakan *software* yang banyak tersedia *web browser* yang populer pada saat ini Mozilla Firefox, Chrome, Opera dan masih banyak yang lainnya.

Jika suatu *link* di klik, browser akan melihat alamat dari tujuan *link* tersebut dan kemudian mencarinya di *web server*. Jika *browser* menemukan alamat dari tujuan *link* tersebut, *browser* akan menampilkan informasi yang ada,

dan jika tidak menemukannya *browser* akan memberikan suatu pesan yang menyatakan bahwa alamat dari tujuan *link* tersebut tidak dapat ditemukan.

II.5.6 *Hyper Text Markup Language (HTML)*

HyperText Markup Language (HTML) adalah sebuah bahasa *markah* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan formatting hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi home page dengan perintah-perintah HTML.

HTML merupakan sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

1. Tahun 1980, IBM memikirkan pembuatan suatu dokumen yang akan mengenali setiap elemen dari dokumen dengan suatu tanda tertentu. IBM kemudian mengembangkan suatu jenis bahasa yang menggabungkan teks dengan perintah-perintah pemformatan dokumen. Bahasa ini dinamakan Markup Language, sebuah bahasa yang menggunakan tanda-tanda sebagai basisnya. IBM menamakan sistemnya ini sebagai Generalized Markup Language atau GML.
2. Tahun 1986, ISO menyatakan bahwa IBM memiliki suatu konsep tentang dokumen yang sangat baik, dan kemudian mengeluarkan suatu publikasi (ISO 8879) yang menyatakan markup language sebagai standar untuk pembuatan dokumen-dokumen. ISO membuat bahasa ini dari GML milik IBM, tetapi memberinya nama lain, yaitu SGML (Standard Generalized Markup Language).

ISO dalam publikasinya meyakini bahwa SGML akan sangat berguna untuk pemrosesan informasi teks dan sistem-sistem perkantoran. Tetapi diluar perkiraan ISO, SGML dan terutama subset dari SGML, yaitu HTML juga berguna

untuk menjelajahi internet, saat ini telah berkembang XHTML yang merupakan pengembangan dari HTML.

II.6 Pemrograman

II.6.1 PHP

PHP adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server*. 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* dimana *script* tersebut dijalankan.

Kelebihan PHP dari bahasa pemrograman lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
3. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang cukup banyak.

PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows).

II.6.2 PHP MyAdmin

PHP MyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui jaringan WWW.phpMyAdmin mendukung berbagai operasi MySQL, diantaranya mengelola basis data, tabel-tabel, bidang (fields), relasi (relation), indeks, pengguna (users), perijinan (permissions) dan lain-lain.

Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang

menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

Fasilitas pada PHP Myadmin antara lain :

1. Membuat dan menghapus database
2. Dapat membuat FDF grafik, dan mampu mencari data didalam database.
3. Membuat, menyalin, menghapus, dapat menambah field.
4. Manajemen pengguna dan privilege (hak akses) pada MySQL.

II.6.3 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL juga merupakan perangkat lunak database server atau sebut saja DatabaseSmart. Database ini semakin lama semakin populer. Dengan menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web.

MySQL mempunyai keistimewaan diantaranya adalah :

1. Dapat berjalan stabil pada berbagai sistem operasi seperti: Linux, Windows, FreeBSD, Mac Os X Server, Solaris, dan masih banyak lagi.
2. MySQL didistribusikan secara open source (gratis), dibawah lisensi GNU General Public Licence (GPL).
3. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

4. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
5. MySQL memiliki kecepatan yang baik dalam menangani query (perintah sql). Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.

II.6.4 Apache

Server HTTP Apache atau *Server Web/WWW Apache* adalah *server web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan *HTTP*.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan *server* menjadi mudah.

Apache merupakan perangkat lunak *open source* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

II.7 CodeIgniter

II.7.1 Pengertian CodeIgniter

CodeIgniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal.

Dengan menggunakan *framework*, kita tidak perlu membuat program dari awal, tetapi kita sudah memberikan *library* fungsi-fungsi yang sudah diorganisasi untuk dapat membuat suatu program dengan cepat. Kita hanya perlu memasukan data yang akan diproses dan bagaimana menampilkannya.

CodeIgniter merupakan *framework* yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan

mudah. Pendekatan dari CI sangatlah mudah, dari membuat sekadar tulisan sampai dengan yang kompleks dapat didekati dengan mudah. Tidak seperti *framework* yang lain, untuk mendapatkan tulisan *Hello World* di *browser* saja, kita harus menggunakan beberapa tahap. CI cukup dengan satu file dan satu prosedur atau *method*.

Framework CI telah dibahas juga dalam forum di IBM dan Oracle untuk pengembang aplikasi berbasis *web* dengan menggunakan PHP. Seperti telah diketahui, IBM dan Oracle telah memiliki kerjasama dengan Zend, yang mengembangkan PHP, dengan produk yang dikenal dengan ZendCore for IBM DB2 dan Informix, dan perusahaan tersebut untuk memudahkan pengembang aplikasi Oracle. Pengembang tidak harus direpotkan atau sulit memasang PHP agar bisa mengakses *database-database* tersebut.

II.7.2 *Framework*

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

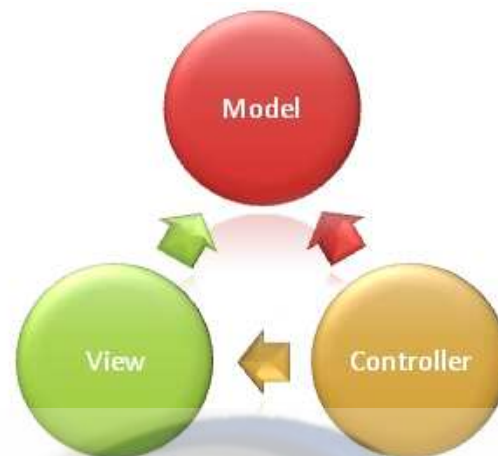
Ada beberapa alasan mengapa menggunakan *Framework*:

- a. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- b. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada).
- c. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya *validasi*, *ORM*, *pagination*, *multiple database*, *scaffolding*, *pengaturan session*, *error handling*, dll).
- d. Lebih bebas dalam pengembangan jika dibandingkan CMS.

II.7.3 *Pemodelan MVC (Model View Controller)*

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama

yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi.



Gambar II.19 - Konsep Pemrograman MVC^[2]

Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu:

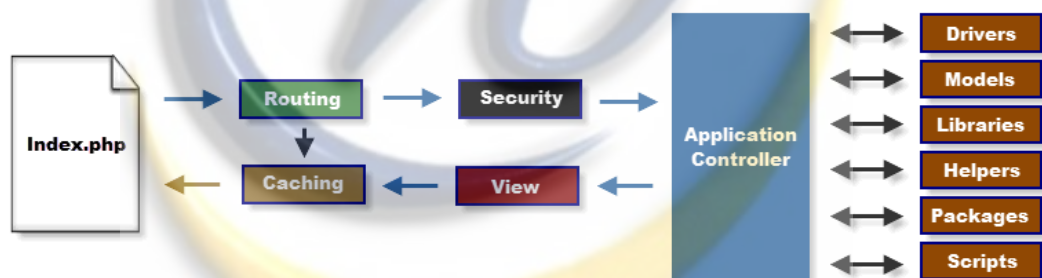
- a. **View**, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
- b. **Model**, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
- c. **Controller**, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*-nya, yaitu programmer yang menangani bagian model dan controller, sedangkan designer yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan maintainability dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara

programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

Dalam aplikasi MVC, kita dapat membuat aplikasi dengan memisahkan aspek dari aplikasi (logika input, bisnis dan antarmuka), ada bagian penampil data (*view*), penyedia atau pemroses data (*model*), dan pengendali program (*controller*). Dengan pemisahan ini, maka pembuatan satu program dapat dibagi-bagi kepada beberapa orang. Setiap bagian atau komponen yang dibuat secara terpisah diharap dapat digunakan tidak hanya oleh satu program saja, tetapi oleh banyak program, terutama bagian model dan view. Teknik pemrograman MVC ini dapat diimplementasi tidak hanya dalam PHP saja, tetapi suatu teknik pemrograman yang sudah dikenal dan digunakan hampir pada semua bahasa pemrograman yang ada.

CI adalah salah satu *framework* pemrograman PHP yang dapat mengimplementasikan teknik pemrograman MVC. Berikut adalah *Flowchart* bagaimana CI mengimplementasi MVC :



Gambar II.20 - *Flowchart* Pengimplementasian MVC^[2]

Berdasarkan *Flowchart* diatas dapat dijabarkan bagaimana peran CodeIgniter sebagai berikut:

1. Index.php berlaku sebagai *controller* di depan, yang meninisialisasi sumber daya yang dibutuhkan untuk menjalankan CodeIgniter.
2. Router akan memeriksa *request* HTTP untuk menentukan apa yang harus dilakukan dengan *request* tersebut.
3. Jika ada *filecache*, maka *file cache* langsung dikirimkan kepada *browser*, dengan memotong eksekusi sistem normal.

4. Sebelum suatu aplikasi *controller* dipanggil, *request* HTTP dan data yang dikirim oleh pengguna disaring (*filter*) terlebih dahulu untuk keamanan.
5. *Controller* akan memuat model, *library* inti, *helper*, dan sumber daya lainnya yang diperlukan untuk memroses *request* tersebut.
6. *View* terakhir di-*render* kemudian dikirimkan kepada *browser* untuk dilihat. Jika *caching* didefinisikan, maka *view* akan di-*cache* terlebih dahulu, setelah itu baru dikirimkan kepada *browser*.

Secara umum, kita harus membuat 3 *file* yang menyatakan 3 komponen, *file model*, *file view*, dan *file controller* untuk bisa membuat aplikasi. Namun dalam pemrograman MVC dengan menggunakan CodeIgniter, kita tidak harus membuat 3 *file*.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan *Framework* PHP lain, diantaranya :

- a. Performa sangat cepat : salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scratch*, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lain.
- b. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa *file* konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan codeigniter dengan setting standard, anda hanya perlu merubah sedikit saja file pada folder *config*.
- c. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
- d. Dokumentasi yang sangat lengkap: Setiap paket instalasi codeigniter sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

II.7.4 URI

Penulisan alamat dari aplikasi atau dokumen *web* yang dihasilkan oleh aplikasi PHP yang menggunakan *framework* CodeIgniter dinyatakan sebagai URI

(*Universal Resource Identifier*), karena dengan CodeIgniter alamat yang ditulis pada kotak URL(*Universal Resource Locator*) browser bukan merupakan bentuk URL yang sebenarnya. URL berbeda dengan URI. Secara praktis, maka perbedaan dari URI dan URL adalah sebagai berikut:

- a. URL, penulisan dapat disebut dengan URL apabila di setiap penulisan alamat selalu diakhiri dengan ekstension dari *file*-nya.

Contoh:

1. <http://24hours7days.biz/page.html> (alamat dari dokumen page.html)
2. <http://24hours7days.biz/coffee.jpg> (alamat yang menuliskan file coffee.jpg)

- b. URI, penulisan akan disebut sebagai URI apabila pada saat menulis alamat kita tidak perlu menuliskan ekstensinya, karena memang kita tidak tahu apa sebenarnya.

Contoh:

1. <http://ibadah.info/shalat>
2. <http://ibadah.info/index.php/hadits>
3. http://localhost/ci20/index.php/c_helloworld/modelmethod

Setiap URL adalah URI, karena URL adalah subser dari URI. Selain URL dan URI, ada juga kita mengenal URN (*Universal Resource Locator*), yaitu penulisan yang diakhiri dengan nama.;

Berikut adalah contohnya

<http://ibadah.info/page.html#DOA>

#DOA digunakan untuk instruksikan *browser* agar menuju bagian dokumen yang memiliki nama DOA; browser akan secara otomatis menscroll dokumen, sehingga bagian dokumen dengan nama DOA akan ditampilkan di awal *browser*. #DOA menunjukan suatu seksi atau bagian dari dokumen HTML yang diberi nama DOA, yang dinyatakan dengan tag `Teks.`.

II.7.5 Helper

Helper atau fungsi *helper* di dalam CI adalah *file-file* yang berisi kumpulan fungsi yang dikelompokkan dalam katagori khusus, untuk membantuk pemrogram menyelesaikan suatu tugas. Fungsi-fungsi yang ada di dalam satu *file*

helper sebenarnya merupakan fungsi yang saling bebas, independen, tidak saling tergantung atau berhubungan.

File-file yang berisi fungsi ini disebut sebagai *helper*. Kita harus menyimpan daftar *file*, atau *helper* yang kita buat ke dalam direktori *helper*, yang ada pada direktori *application*, sejajar dengan direktori *controller*, *model* dan *view* kita.

Kumpulan fungsi atau *helper* yang disediakan oleh CodeIgniter, adalah kumpulan fungsi yang paling sering diperlukan oleh pemrogram, yang belum disediakan oleh PHP. Pemrogram dapat membuat *helper* tambahan, apabila fungsi yang ada di *helper* masih terdapat kekurangan.

II.7.5.1 Daftar *File Helper CodeIgniter*

Berikut ini adalah daftar kategori *helper* yang disediakan oleh CodeIgniter sebagai standar:

Tabel II-1 - Daftar *Helper CodeIgniter*^[2]

Helper	Keterangan
Array	Berisi kumpulan fungsi untuk membantudalam bekerja dengan array tambahan selain yang dimiliki oleh PHP
Captcha	File <i>helper</i> ysng beridi fungsi-fungsi yang digunakan untuk membantu dalam pembuatan gambar-gambar Captcha
Cookies	Kumpulan fungsi untuk bekerja dengan cookies, digunakan untuk membuat dan membaca cookies
Date	Berisi kumpulan fungsi untuk bekerja dengan menggunakan data tanggal
Directory	Kumpulan fungsi untuk membantu dalam bekerja dengan direktori
Download	Kumpulan fungsi untuk mendownload data
Email	Kumpulan fungsi untuk memvalidasi data email dan mengirim email
File	Kumpulan fungsi digunakan untuk membantu kita bekerja dengan file
Form	kumpulan fungsi yang disediakan untuk membantu membuat elemen-elemen form
HTML	Kumpulan fungsi berkaitan dengan pembuatan elemen HTML dasar seperti mengganti baris, heading, image, ling_tag, ol, ul, meta, dan doctype

Infector	Berisi fungsi-fungsi yang memungkinkan kita untuk mengubah kata ke dalam bentuk jamak, tunggal, camel case, dan lain-lain
Language	File yang berisi kumpulan fungsi yang digunakan untuk membantu bekerja dengan file-file bahasa
Number	File yang berisi fungsi-fungsi untuk membantu bekerja dengan data numeric
Path	Berisi kumpulan fungsi yang memungkinkan kita bekerja dengan path file dalam server
Security	Kumpulan fungsi yang berhubungan dengan keamanan
Smiley	Kumpulan fungsi untuk menampilkan icon smiley (emoticon)
String	Kumpulan fungsi untuk membantu bekerja dengan data string
Text	Kumpulan fungsi untuk membantu bekerja dengan string, seperti melakukan pemformatan teks
Typography	Berisi fungsi-fungsi untuk membantu format teks secara sematik dengan cara relevan
URL	Kumpulan fungsi yang disediakan untuk membantu membuat link-link, mendapat data URL dari CodeIgniter
XML	Kumpulan fungsi yang disediakan untuk membantuk bekerja dengan data XML