

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Perancangan**

##### **II.1.1 Pengertian perancangan**

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem.

##### **II.1.2 Tujuan Perancangan Program**

Merancang spesifikasi perangkat lunak atau *software* untuk membantu proses pengolahan data didalam sistem yang akan diusulkan.

##### **II.1.3 Sasaran Perancangan Program**

1. Membuat solusi dari pemecahan kasus atau masalah yang timbul
2. Meningkatkan kualitas & performance dari kinerja kerja
3. Membantu proses pengambilan keputusan

##### **II.1.4 Pengertian Perancangan Sistem**

Ada beberapa pengertian perancangan sistem menurut beberapa ahli antara lain :

1. Verzello / John Reuter III

Tahap setelah analisis dari siklus pengembangan sistem : Pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi : “menggambarkan bagaimana suatu sistem dibentuk .

2. John Burch & Gary Grudnitski

Desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

3. George M. Scott

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem, sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

## **II.2 Alat Bantu Perancangan Sistem**

### **II.2.1 Pengenalan UML**

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang diberbagai Negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak orang untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang. Muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual

untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan metodologi berorientasi objek.

### **II.2.1.2 Sejarah UML**

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemrograman berorientasi objek selanjutnya.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman *smalltalk* pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian saat itu. Setelah lima tahun *smalltalk* berkembang, berkembang pula metode yang pertama diperkenalkan oleh Sally Shlaer dan Stephen Mellor (Shlaer-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Micheal R. Blaha, William Iorensen, Frederick Eddy, William

Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen,1991), dan masih banyak lagi.

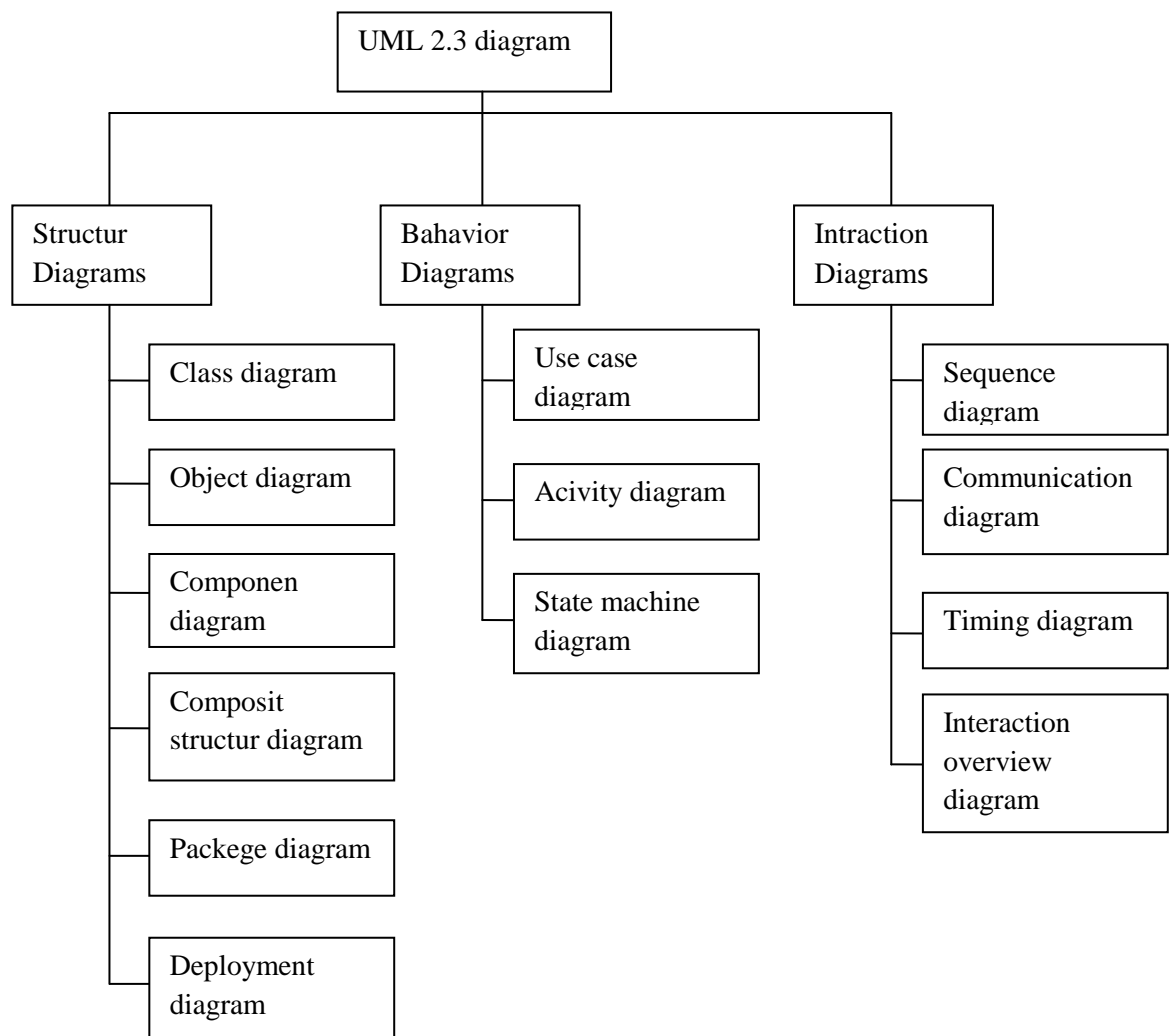
Buku terkenal yang juga terkenal selanjutnya adalah karangan Ivar Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Lima tahun kemudia muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya.

Karena banyaknya metodologi-metodologi yang berkembang pesat maka munculah sebuah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Objek Modelling Technique (OMT)* dari Rumbaugh dan Booch(1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep pemikiran lainya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam perusahaan bernama *Rational Softwer Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standardisasi pemodelan berorientasi objek dan pada bulan september 1997 UML diakomodasi oleh OMG hingga sampai ini UML telah memerikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya.

### II.2.1.3 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Dapat dilihat pada gambar dibawah.



**Gambar: 2.1 Diagram UML**

**Sumber: Modul Pembelajaran Rekayasa Perangkat Luna**

Penjelas dari pembagian kategori:

- a. *Structur diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Bahavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar subsistem pada suatu sistem.
- d. *Class diagram* yaitu diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat membangun sistem.
- e. *Object diagram* yaitu menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.
- f. *Component diagram* yaitu dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.
- g. *Composite structure diagram* yaitu menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat sistem berjalan.
- h. *Package diagram* yaitu menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML.
- i. *Deployment diagram* yaitu menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi sistem

- j. *Use case diagram* yaitu untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.
- k. *Activity diagram* yaitu menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis.
- l. *State machine diagram* yaitu menggambarkan perubahan setatus atau transisi status dari sebuah mesin atau sistem.
- m. *Sequence diagram* yaitu menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.
- n. *Communication diagram* yaitu menggambarkan interaksi antar objek atau bagian dalam bentuk urutan pengiriman pesan.
- o. *Timing diagram* yaitu menggambarkan tingkah laku sistem dalam periode waktu tertentu.
- p. *Interaction overview diagram* yaitu mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. Diagram yang setiap titik mempresentasikan diagram interaksi.

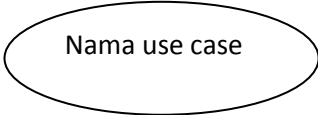

## II.2.2 Diagram Use Case


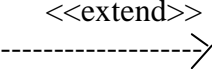
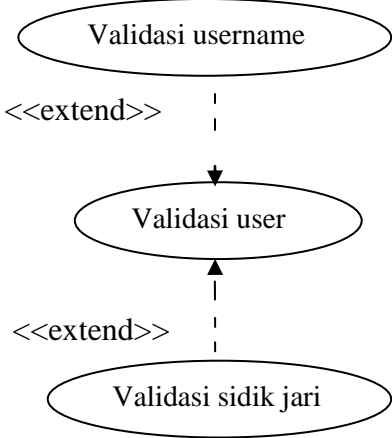
Diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.


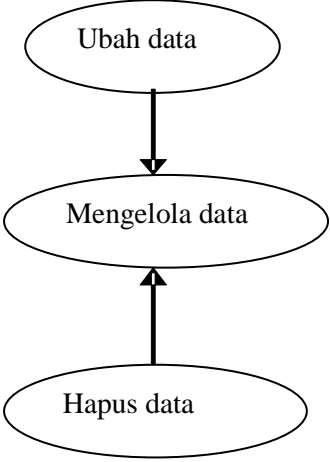
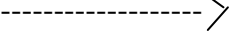

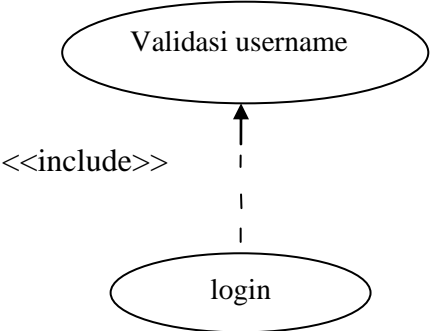
Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu:

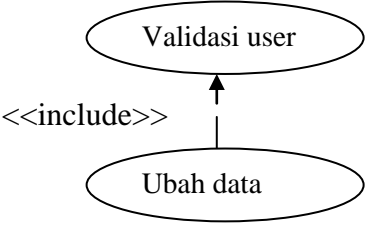
- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah tabel simbol-simbol diagram *Use case*

Simbol	Deskripsi
<p data-bbox="300 1055 419 1084"><i>Use case</i></p> 	<p data-bbox="770 1055 1369 1308">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i></p>
<p data-bbox="300 1357 464 1386">Aktor/ <i>actor</i></p>  <p data-bbox="395 1541 552 1570">Nama aktor</p>	<p data-bbox="770 1357 1369 1899">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walupun simbol dari aktor adalah gambar orang, tapi belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

Simbol	Deskripsi
<p data-bbox="300 338 544 371">Asosiasi/ <i>associati</i></p> 	<p data-bbox="770 338 1369 483">Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case yang memiliki interaksi dengan aktor</p>
<p data-bbox="300 580 507 613">Ekstensi/<i>extend</i></p> 	<p data-bbox="770 580 1369 994">Relasi use case tambahan kesebuah use case dimana use case yang ditambah dapat berdiri sendiri walau tanpa use case tambahan; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal</p> <div data-bbox="842 1055 1235 1491" style="text-align: center;">  <pre> graph TD     A([Validasi username]) -.-&gt; &lt;&lt;extend&gt;&gt;  B([Validasi user])     C([Validasi sidik jari]) -.-&gt; &lt;&lt;extend&gt;&gt;  B </pre> </div> <p data-bbox="770 1536 1342 1570">arah panah pada use case yang ditambahkan</p>
Simbol	Deskripsi

<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada use case yang menjadi generalisasinya (umum)</p>
<p>Menggunakan <i>/include/ uses</i></p> <p>&lt;&lt;extend&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi use case tambahan kesebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.</p> <p>Ada dua sudut pandang mengenai include di use case:</p> <ol style="list-style-type: none"> <li>1. Include berarti use case yang ditambahkan akan selau dipanggil saat use case tambahan dijalankan, misal</li> </ol> 

	<p>2. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambah telah dijalankan sebelum use case tambahan dijalankan, misal</p>  <pre> graph BT     UC1(Validasi user)     UC2(Ubah data)     UC2 -- &lt;&lt;include&gt;&gt; --&gt; UC1   </pre>
--	---

**Tabel 2.1 Simbol-simbol Diagram Use Case**


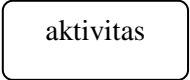
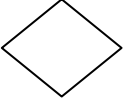


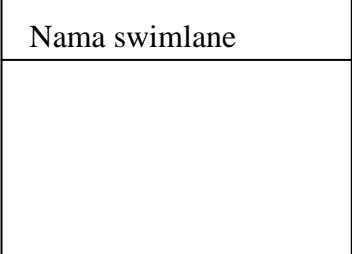
**Sumber: Modul Pembelajaran Rekayasa Perangkat Lunak**

### II.2.3 Diagram Activity

Diagram aktivitas juga banyak digunakan untuk mendefinisikan seperti berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan tampilan antarmuka.
- Rancangan pengujian setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan uji kasusnya.

Berikut simbol-simbol diagram aktivitas:

simbol	Deskripsi
setatus awal 	Sebuah diagram aktivitas memiliki sebuah setatus awal
aktivitas 	Aktivitas biasanya diawali dengan kata kerja
percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Sebuah diagram aktivitas memiliki sebuah setatus akhir.
swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

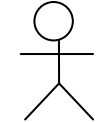
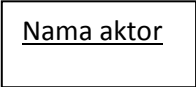

**Tabel 2.2 Simbol-simbol Diagram Activity**

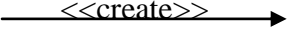
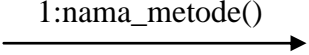
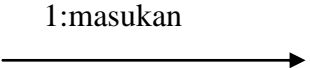
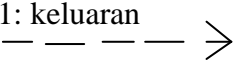
**Sumber: Modul Pembelajaran Rekayasa Perangkat Lunak**

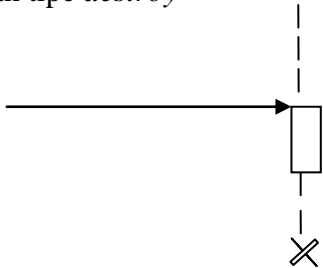
## II.2.4 Diagram Sequence

Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut simbol-simbol diagram sekuen:

simbol	Deskripsi
<p>namaktor</p>  <p>nama aktor</p> <p>atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem, jadi walaupun simbol dari aktor adalah gambar orang,tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup/ lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>

<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>Nama objek : nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 40px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>Pesan tipe create</p> <div style="text-align: center; margin: 10px auto;">  </div>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> <div style="text-align: center; margin: 10px auto;">  </div>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi /metode, karena ini memanggil operasi maka operasi yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe <i>send</i></p> <div style="text-align: center; margin: 10px auto;">  </div>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> <div style="text-align: center; margin: 10px auto;">  </div>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembali</p>

<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri objek yang lain, arah panah mengarah pada objek yang diakhiri, jika ada crate maka ada destroy.</p>
--	--

**Tabel 2.3 Simbol-simbol Diagram Sequence**

**Sumber: Modul Pembelajaran Rekayasa Perangkat Lunak**

### II.3 Pengertian Program

Program adalah unsur pertama yang harus ada demi terciptanya suatu kegiatan. Di dalam program dibuat beberapa aspek, disebutkan bahwa di dalam setiap program dijelaskan mengenai:

1. Tujuan kegiatan yang akan dicapai.
2. Kegiatan yang diambil dalam mencapai tujuan.
3. Aturan yang harus dipegang dan prosedur yang harus dilalui.
4. Perkiraan anggaran yang dibutuhkan.
5. Strategi pelaksanaan.

Melalui program maka segala bentuk rencana akan lebih terorganisir dan lebih mudah untuk dioperationalkan.

Karakteristik program komputer yang terstruktur

1. Menggunakan bahasa pemrograman yang berkualitas atau bahasa pemrograman tingkat tinggi
2. Memiliki portabilitas yang fleksibel

3. Menggunakan struktur organisasi data yang benar
4. Format penulisan program yang benar & proporsional
5. Membuat dokumentasi program

Pemrograman di Internet tidak sama seperti pemrograman pada komputer lokal. Pemrograman di internet berkaitan dengan format dokumen yang digunakan di internet. Secara umum ada dua macam pemrograman di internet, yaitu: *client side programming* dan *server side programming*.

#### ***A. Client Side Programming***

Pada pemrograman jenis ini program didownload oleh pengunjung dan dijalankan pada komputer pengunjung. Program jenis ini memerlukan dukungan browser pengunjung. Selain itu ukuran program harus diperhatikan karena jika program terlalu besar, maka waktu yang diperlukan untuk membuka halaman web melihat dan menyalin *source code program*. Contoh program *client side* adalah : *HTML, CSS, java Script* dan *VBScript*.

#### ***B. Server Side Programming***

Sesuai dengan namanya, jenis pemrograman ini program dijadikan di dalam *webserver*. Hasil pengolahannya dikirimkan pada browser pengunjung sebagai halaman *HTML*. Ukuran program relatif dapat lebih besar karena pengunjung tidak perlu mendownload programnya, tetapi hanya hasil pengolahannya. Keuntungan lainnya yakni *source code* tidak dapat dilihat. Contoh pemrograman *server side* diantaranya: *Perl, PHP, ASP, JSP*. ( Faisal S. Si, 2011: 2-3 ).

#### **II.4. Pengertian Pemberkasan**

Menurut Kamus Umum Bahasa Indonesia, arsip adalah simpanan surat-surat penting. Berdasarkan pengertian ini, tidak semua surat dikatakan arsip. Surat dinyatakan sebagai arsip jika memenuhi persyaratan berikut ini:

- a. Surat tersebut harus masih mempunyai kepentingan bagi organisasi/lembaga baik untuk masa kini dan masa yang akan datang;
- b. Surat yang menyimpan kepentingan tersebut disimpan menurut sistem tertentu sehingga memudahkan temu balik bila diperlukan kembali.

Menurut Kamus Administrasi Perkantoran, arsip adalah kumpulan warkat yang disimpan secara teratur berencana karena mempunyai suatu kegunaan agar setiap kali diperlukan dapat cepat ditemukan kembali. Berdasar pengertian ini, warkat dapat disebut arsip apabila memenuhi 3 syarat, yaitu:

- a. Warkat memiliki kegunaan
- b. Warkat disimpan secara teratur dan berencana, dan
- c. Warkat dapat mudah dan cepat ditemukan jika diperlukan kembali.

Dari dua pengertian di atas, dapat disimpulkan bahwa untuk dapat disebut sebagai arsip, maka surat atau warkat harus memenuhi persyaratan: memiliki nilai guna bagi organisasi/lembaga sehingga surat/warkat tersebut dikelola dengan teratur dan berencana menurut suatu sistem tertentu agar memudahkan penemuan kembali surat/warkat yang disimpan itu jika sewaktu-waktu dibutuhkan kembali nilai informasi yang ada di dalamnya oleh organisasi/lembaga.

Rencana posting berikutnya Pengertian arsip berdasarkan:

- a. Asal Katanya
- b. Undang-Undang Nomor Prp. 19 Tahun 1961
- c. Undang-Undang Nomor 7 Tahun 1971
- d. Seminar Dokumentasi/Arsip Kementerian-kementerian
- e. Lembaga Administrasi Negara

## **II.5 Pengertian calon Anggota**

Anggota tak langsung atau calon anggota adalah warga negara Indonesia yang belum atau tidak terdaftar secara sah menjadi anggota partai pada Dewan Pengurus Cabang setempat dan secara aktif mengikuti kegiatan-kegiatan partai.

Orang-orang yang belum melunasi pembayaran Simpanan Pokok, secara formal belum sepenuhnya melengkapi persyaratan administrasi sebagaimana ditentukan dalam AD/ART, sehingga belum bisa diterima sebagai anggota penuh. Memiliki hak bicara tetapi tidak memiliki hak memilih dan dipilih untuk menjadi pengurus ataupun pengawas. Memperoleh pelayanan yang sama.

### **II.5.1 Tata Cara Pendaftaran Anggota**

Tata cara pendaftaran menjadi anggota partai yaitu:

- a. Mengajukan permintaan menjadi anggota partai kepada Dewan Pengurus Cabang melalui Dewan pengurus ranting setempat, disertai persyaratan persetujuan terhadap Anggaran Dasar, Anggaran Rumah Tangga, Platform Partai, dan membayar uang pangkal;

- b. Apabila permintaan diluluskan, maka yang bersangkutan berstatus sebagai calon anggota selama 3 bulan dengan hak menghadiri kegiatan-kegiatan partai yang dilakukan secara terbuka;
- c. Apabila selama menjadi calon anggota yang bersangkutan menunjukkan hal-hal positif maka diterima menjadi anggota secara penuh dan diberikan Kartu anggota partai yang dikeluarkan oleh Dewan Pengurus Partai.
- d. Permintaan menjadi anggota dapat ditolak apabila terdapat alasan-alasan yang kuat secara organisatoris dan tidak bertentangan dengan Anggaran Dasar dan Anggaran Rumah Tangga.

## **II.6 Pengertian PKB**

PKB ( Partai Kebangkitan Bangsa ) adalah partai yang ada di Indonesia. Partai ini didirikan di Jakarta pada tanggal 23 Juli 1998 (29 Rabi'ul Awal 1419 Hijriyah) yang dideklarasikan oleh para kiai-kiai Nahdlatul Ulama, seperti (Munasir Ali, Ilyas Ruchiyat, Abdurrahman Wahid, A. Mustofa Bisri, dan A. Muhith Muzadi).

### **II.6.I Sejarah Pendirian Partai Politik**

Pada tanggal 21 Mei 1998, Presiden Soeharto lengser keprabon sebagai akibat desakan arus reformasi yang kuat, mulai yang mengalir dari diskusi terbatas, unjuk rasa, unjuk keprihatinan, sampai istighosah dan lain sebagainya. Peristiwa ini menandai lahirnya era baru di Indonesia, yang kemudian disebut era reformasi. Sehari setelah peristiwa bersejarah itu, Pengurus Besar Nahdlatul Ulama (PBNU) mulai kebanjiran usulan dari warga NU di seluruh pelosok tanah

air. Usulan yang masuk ke PBNU sangat beragam, ada yang hanya mengusulkan agar PBNU membentuk parpol, ada yang mengusulkan nama parpol. Tercatat ada 39 nama parpol yang diusulkan. Nama terbanyak yang diusulkan adalah Nahdlatul Ummah, Kebangkitan Umat dan Kebangkitan Bangsa. Ada juga yang mengusulkan lambang parpol. Unsur-unsur yang terbanyak diusulkan untuk lambang parpol adalah gambar bumi, bintang sembilan dan warna hijau. Ada yang mengusulkan bentuk hubungan dengan NU, ada yang mengusulkan visi dan misi parpol, AD/ART parpol, nama-nama untuk menjadi pengurus parpol, ada juga yang mengusulkan semuanya. Di antara yang usulannya paling lengkap adalah Lajnah Sebelas Rembang yang diketuai KH M Cholil Bisri dan PWNU Jawa Barat. Dalam menyikapi usulan yang masuk dari masyarakat Nahdliyin, PBNU menanggapi secara hati-hati. Hal ini didasarkan pada adanya kenyataan bahwa hasil Muktamar NU ke-27 di Situbondo yang menetapkan bahwa secara organisatoris NU tidak terkait dengan partai politik manapun dan tidak melakukan kegiatan politik praktis. Namun demikian, sikap yang ditunjukkan PBNU belum memuaskan keinginan warga NU. Banyak pihak dan kalangan NU dengan tidak sabar bahkan langsung menyatakan berdirinya parpol untuk mewadahi aspirasi politik warga NU setempat. Diantara yang sudah mendeklarasikan sebuar parpol adalah Partai Bintang Sembilan di Purwokerto dan Partai Kebangkitan Umat (Perkanu) di Cirebon.

Akhirnya, PBNU mengadakan Rapat Harian Syuriyah dan Tanfidziyah PBNU tanggal 3 Juni 1998 yang menghasilkan keputusan untuk membentuk Tim Lima yang diberi tugas untuk memenuhi aspirasi warga NU. Tim Lima diketuai oleh KH Ma'ruf Amin Rais (Suriyah/Koordinator Harian PBNU), dengan anggota,

KH M Dawam Anwar (Katib Aam PBNU), Dr KH Said Aqil Siradj, M.A. (Wakil Katib Aam PBNU), HM Rozy Munir, S.E., M.Sc. (Ketua PBNU), dan Ahmad Bagdja (Sekretaris Jenderal PBNU). Untuk mengatasi hambatan organisatoris, Tim Lima itu dibekali Surat Keputusan PBNU.

Selanjutnya, untuk memperkuat posisi dan kemampuan kerja Tim Lima seiring semakin derasnya usulan warga NU untuk menginginkan partai politik, maka pada Rapat Harian Syuriyah dan Tanfidziyah PBNU tanggal 20 Juni 1998 memberi Surat Tugas kepada Tim Lima, selain itu juga dibentuk Tim Asistensi yang diketuai oleh Arifin Djunaedi (Wakil Sekjen PBNU) dengan anggota H Muhyiddin Arubusman, H.M. Fachri Thaha Ma'ruf, Lc., Drs. H Abdul Aziz, M.A., Drs. H Andi Muarli Sunrawa, H.M. Nasihin Hasan, H Lukman Saifuddin, Drs. Amin Said Husni dan Muhaimin Iskandar. Tim Asistensi bertugas membantu Tim Lima dalam menginventarisasi dan merangkum usulan yang ingin membentuk parpol baru, dan membantu warga NU dalam melahirkan parpol baru yang dapat mewadahi aspirasi politik warga NU.

Pada tanggal 22 Juni 1998 Tim Lima dan Tim Asistensi mengadakan rapat untuk mendefinisikan dan mengelaborasi tugas-tugasnya. Tanggal 26 - 28 Juni 1998 Tim Lima dan Tim Asistensi mengadakan konsinyering di Villa La Citra Cipanas untuk menyusun rancangan awal pembentukan parpol.

## **II.7 Pengertian Website**

Web adalah halaman yang ditampilkan di internet yang memuat informasi tertentu (khusus). Web pertama kali diperkenalkan pada tahun 1992. Hal ini

sebagai hasil usaha pengembangan yang dilakukan *CERN* di Swiss. Internet dan web adalah dua hal yang berbeda. Internet yaitu yang dapat menampilkan web-nya, sedangkan web adalah yang ditampilkannya yang berupa susunan dari halaman-halaman yang menggunakan teknologi web dan saling berkaitan satu sama lain.

Suatu standar teknologi web saat ini sudah tersusun, meskipun penerapannya belum didukung oleh seluruh pengembang web. Standar ini disusun oleh suatu badan yaitu *World Wide Web Consortium (W3C)*. Standar ini dibutuhkan karena semakin banyaknya variasi dalam teknologi web sehingga terkadang satu sama lain tidak kompatibel.

Secara umum, teknologi desain web terbagi menjadi beberapa *layer* (lapisan), yaitu *structural layer*, *presentation layer* dan *behavioral layer*.

#### 1. Structural Layer

Layer ini berhubungan dengan struktur dokumen-dokumen web. Bagaimana sebuah dokumen tersusun, format apa yang dipakai, tanda atau mark up apa yang digunakan merupakan bagian dari layer ini. Standar teknologi yang direkomendasikan saat ini adalah *Extensible Hypertext Markup Language (XHTML)* dan *Extensible Markup Language (XML)*. *XHTML* adalah *HTML* yang ditulis ulang dengan aturan-aturan yang lebih ketat yang mengacu pada *XML*. Sedangkan *XML* adalah sekumpulan aturan untuk menyusun bahasa markup.

## 2. Presentation Layer

Layer ini berhubungan dengan bagaimana mengatur tampilan dokumen pada layar, suara yang keluar, atau bagaimana format percetakan dokumen. Pada teknologi web lama, bagian ini menyatu dengan structural layer. Tetapi pada standar baru, layer ini disarankan untuk dipisah. Yang termasuk dalam teknologi ini adalah *Cascading Styles Sheets (CSS)*.

## 3. Behavioral Layer

Layer ini berhubungan dengan masalah penggunaan bahasa skrip dan pemrogramannya untuk tujuan meningkatkan sisi interaktif dan dinamis halaman web. Yang termasuk dalam layer ini adalah *Document Object Model (DOM)* dan *JavaScript*. *DOM* memungkinkan suatu dokumen atau skrip untuk mengakses atau meng-update isi, struktur, dan *style* dari dokumen. *JavaScript* merupakan teknologi yang cukup lama dan tetap digunakan untuk menambah dokumen menjadi lebih hidup. ( Adhi Prasetyo, 2009: 1-2 ).

## II.8 Konsep Dasar Bahasa Pemrograman

### II.8.2 PHP

#### a. Pengertian *PHP*

*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. *PHP* disebut sebagai pemrograman *server side programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. *PHP* adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open*

*source*, yaitu pengguna dapat mengembangkan kode-kode fungsi *PHP* sesuai dengan kebutuhan.

#### **b. Sejarah *PHP***

*PHP* merupakan salah satu bahasa pemrograman *web* yang masih muda namun telah mengalami perkembangan yang cukup signifikan dan telah banyak digunakan oleh banyak *user* dalam membuat aplikasi web baik perseorangan maupun perusahaan.

Pada tahun 1994 seorang pemrogrammer bernama Rasmus Lerdorf awalnya membuat sebuah halaman website pribadi, tujuannya adalah untuk mempertahankan halaman website pribadi tersebut sekaligus membangun halaman web yang dinamis. *PHP* pada awalnya diperkenalkan sebagai singkatan dari *Personal Home Page*. *PHP* pertama ditulis menggunakan *Perl (Perl Script)*, kemudian ditulis ulang menggunakan bahasa pemrograman *C CGI-BIN (Common Gateway Interface-Binary)* yang bertujuan untuk mengembangkan halaman website yang mendukung formulir dan penyimpanan data. Pada tahun 1995 *PHP Tool 1.0* dirilis untuk umum, kemudian pengembangannya dilanjutkan oleh Andi Gutmans dan Zeev Suraski.

Pada November 1997, dirilis *PHP/FI 2.0*. Pada rilis ini, *interpreter PHP* sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP/FI* secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter PHP* menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk *PHP* dan

meresmikan rilis tersebut sebagai *PHP* 3.0 dan singkatan *PHP* diubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter *PHP* baru dan rilis tersebut dikenal dengan *PHP* 4.0. *PHP* 4.0 adalah versi *PHP* yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Perusahaan Zend kemudian melanjutkan pengembangan *PHP* dan merilis *PHP* versi 5 terakhir pada saat ini.

Pemrograman *PHP* dapat ditulis dalam dua bentuk yaitu penulisan baris kode *PHP* pada *file* tunggal dan penulisan kode *PHP* pada halaman html (*embedded* ). Kedua cara penulisan tersebut tidak memiliki perbedaan, hanya menjadi kebiasaan gaya penulisan dari programmer. Berikut contoh penulisan kode program *PHP*:

Singlefile\_php.php

```
<?php
    Echo "<html>";
    Echo "<head>";
    Echo "<title> contoh ini adalah tulisan dari php </title>";
    Echo "</head>";
    Echo "</body>";
    Echo "<p> dibawah ini adalah tulisan dari php</p> ";
    Echo " Teks dari php";
    Echo "</html>";
```

Embedded\_php.html

```
<html>
    <head>
```

```

        <title> contoh php </title>
    </head>
    <body>
        <p> dibawah ini adalah tulisan dari php </p>
        <?php
            Echo “ Teks dari php ”;
        ?>
    </body>
</html>

```

Untuk penulisan kode *PHP* pada halaman *HTML* diperlukan tambahan konfigurasi pada *web server* agar dapat berjalan. Konfigurasi ini bertujuan untuk mendaftarkan ekstensi *.html* agar dapat dikenal *Apache web server* dan dapat diproses seperti halnya *file php* dengan ekstensi *.php*. ( Alexander F. K. Sibero, 2011: 49-50 ).

### II.8.3 Pengertian *MYSQL*

*MySQL* adalah salah satu program yang dapat digunakan sebagai database, dan merupakan salah satu software untuk database server yang banyak digunakan. *MySQL* bersifat *Open Source* dan menggunakan *SQL*. *MySQL* adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* merupakan database server yang bersifat multiuser dan multi-threaded. *SQL* adalah bahasa database standar yang memudahkan penyimpanan, pengubahan akses informasi.

*MySQL* dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*. *MySQL* pertama dikembangkan oleh *MYSQL AB* yang kemudian diakuisisi *Sun Microsystem* dan terakhir dikelola oleh *Oracle Corporation*. ( Alexander F. K. Sibero, 2011: 97 )

*MySQL* memiliki beberapa kelebihan, antara lain :

1. *MySQL* dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
2. *MySQL* memiliki kecepatan yang bagus dalam menangani query sederhana.
3. *MySQL* memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *Where* dalam perintah query.
4. *MySQL* memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan system perijinan yang mendetail serta sandi terenkripsi.
5. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu table serta kurang lebih dari 5 milyar baris. Selain itu bebas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
6. *MySQL* dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix Soket (UNIX), atau Named Pipes (NT).
7. *MySQL* dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa.

8. MySQL dapat berjalan stabil pada berbagai system operasi seperti windows, linux, FreeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi.
9. MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

### II.8.3.1 Melakukan Koneksi ke Databases

Koneksi database digunakan untuk mengakses data-data yang ada dalam database tersebut. Data tidak dapat diakses tanpa ada koneksi terlebih dahulu.

Berikut script untuk koneksi ke database:

```
Mysql_Connect ( nama host, nama user, password );
```

Keterangan :

1. Nama host adalah lokasi tempat MySQL dipublikasikan.
2. Nama user yaitu nama user yang terdaftar dalam MySQL yang digunakan untuk mengakses data yang ada dalam MySQL.
3. Password adalah password yang digunakan untuk membuka database (phpMyAdmin)

Database akan lebih mudah jika langsung membuatnya melalui jendela phpMyAdmin karena langkah ini lebih efektif dan mudah dipelajari.

### Membuat Tabel

Setelah database terbentuk, langkah berikutnya adalah membuat tabel- tabel di dalam database. Tabel yang dibuat dalam sebuah database berfungsi untuk mengatur dan menyimpan informasi. Di dalam tabel tersebut mengandung

field-field data, misalnya field judul yang berisikan daftar judul. ( MADCOMS dan penerbit Andi, 2011: 140-144.

### II.8.3.2 Fungsi-fungsi PHP untuk mengakses MySQL

*PHP* mempunyai fungsi khusus untuk mengakses *MySQL*. Ada sekitar 48 fungsi yang didukung *php* dalam mengakses *MySQL* dalam membuat aplikasi. Adapun yang biasa digunakan diantaranya adalah:

#### a. *mysql\_connect()*

*mysql\_connect()* untuk membuka koneksi *database* ke *MySQL*. Format fungsinya adalah:

*Int mysql\_connect (string host, string user, string password);*

#### b. *mysql\_select\_db()*

*mysql\_select\_db()* digunakan untuk memilih *database* mana yang akan kita pakai dan untuk mengaktifkan *database*. Prototipenya:

*int mysql\_select\_db(string name\_database)*

#### c. *mysql\_query()*

Fungsi *mysql\_query* berguna untuk mengirimkan perintah *sql* pada server. Sebutan untuk mengirim perintah *sql* dinamakan *quary*. *Quary* memberi perintah *server* untuk melakukan apa yang dikehendai.

*Int mysql\_query (string quary [int link\_identifier]);*

#### d. *mysql\_fetch\_array()*

Fungsi *mysql\_fetch\_array()* berguna untuk mengambil data hasil query secara baris perbaris (row). Pengambilan pertama adalah baris data yang paling atas.

Data yang diambil dalam bentuk array, dimana elemen dari array adalah field-field dari tabel data. Prototipenya: *array mysql\_fetch\_row (int hasil)*.

*e. mysql\_num\_row()*

Fungsi *mysql\_num\_row()* digunakan untuk menghitung jumlah record dari hasil query. Prototipenya: *int mysql\_num\_rows ( int hasil)*. ( Faisal S.Si, 2011: 144-149 ).