BAB II
TINJAUAN PUSTAKA

2.1 Profil Tempat kerja Praktek

2.1.1 Sejarah PPET – LIPI

Pusat Penelitian Informatika didirikan dengan surat Keputusan Kepala LIPI No 1151/M/2001 tanggal 5 Juni 2001. Semula bernama Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer didirikan pada tanggal 13 Januari 1986.

Berdirinya Pusat Penelitian Informatika tidak dapat dipisahkan dengan sejarah terbentuknya Lembaga Elektroteknika Nasional (LEN). Lembaga Elektroteknika Nasional sebagai lembaga penelitian dalam bidang elektronika diresmikan pada tanggal 10 Juni 1965. Bermula di kampus ITB dengan ruang seluas 30 meter persegi yang merupakan pinjaman dari ITB. Pada tahun 1967, berdasarkan Keputusan Presiden nomor 128 tahun 1967, Lembaga Elektroteknika Nasional (LEN) bernaung dibawah koordinasi Lembaga Ilmu Pengetahuan Indonesia. Pada tahun 1969, Prusahaan Negara Pertambangan Minyak dan Gas Bumi Nasional (Pertamina), memberikan pinjaman gedung seluas 1000 meter persegi di Jl. Sawunggaling No. 14 Bandung, sehingga ruangan dikomplek ITB ditinggalkan. Pada awal tahun 1978, komplek gedung perkantoran dan laboratorium LEN-LIPI sudah tersebar di 9 lokasi di kota Bandung, dengan luas lantai sekitar 2500 meter persegi.

Tahun 1983 LEN-LIPI pindah ke Jl. Soekarno-Hatta No.407 250A Bandung, serta gedung tersebut diresmikan pemakaiannya oleh Presiden Republik Indonesia, Bapak Soeharto. pada tanggal 31 Mei 1983. Dengan dilakukannya reorganisasi di Lembaga Ilmu Pengetahuan Indonesia (LIPI) berdasarkan Keppres No. l tahun 1986 tanggal 13 Januari 1986, Lembaga Elektroteknika Nasional berkembang menjadi tiga Puslitbang dan satu Unit Pelaksana Teknis, yaitu:

1. Pusat Penelitian dan Pengembangan Telekomunikasi, Elektronika Strategis, Komponen dan Material (Puslitbang Telkoma-LIPI).
2. Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer (Puslitbang Inkom LIPI).
3. Pusat Penelitian dan Pengembangan Tenaga Listrik dan Mekatronika (Puslitbang Telimek-LIPI)
4. UPT Pusat Laboratorium Enjiniring Nasional (UPT Pusat LEN LIPI).

Dalam struktur organisasi LIPI, keempat Puslibang tersebut diatas berada dibawah koordinasi Deputi Bidang Ilmu Pengetahuan Teknik - LIPI.

Pada tanggal 8 Maret 1990 seluruh fasilitas UPT LEN-LIPI diserahkan ke BPIS. Kantor Puslitbang Inkom LIPI pindah ke Jl. Ranggamalela No. 11 Bandung, sedangkan laboratoriumnya menempati salah satu bagian gedung di kompleks LIPI Jl. Cisitu 21/154D Bandung. Pada tahun 2001 dalam Reorganisasi baru Lembaga Ilmu Pengetahuan Indonesia, Pusat Penelitian dan Pengembangan Informatika dan Ilmu Pengetahuan Komputer berubah nama dan struktur organisasinya menjadi Pusat Penelitian Informatika LIPI.

Lembaga Ilmu Pengetahuan Indonesia (LIPI) mempunyai tugas melaksanakan tugas pemerintahan di bidang penelitian ilmu pengetahuan sesuai dengan ketentuan peraturan perundang-undangan yang berlaku. LIPI merupakan Lembaga Pemerintah Non Departemen Republik Indonesia yang dikoordinasikan oleh Kementerian Negara Riset dan Teknologi.

Visi

Terwujudnya kehidupan bangsa yang adil, cerdas, kreatif, integratif dan dinamis yang didukung oleh ilmu pengetahuan dan teknologi yang humanistik

Misi

1. Menguasai ilmu pengetahuan dan teknologi agar menjadi penggerak utama dan acuan dalam meningkatkan kemajuan dan persatuan bangsa, memperkuat daya saing masyarakat
2. Ikut serta dalam usaha mencerdaskan kehidupan bangsa melalui pembangunan berkelanjutan yang berwajah kemanusiaan (sustained humanistic development
3. Memperkuat landasan etika keilmuan

Tugas pokok

LIPI mempunyai tugas melaksanakan tugas pemerintahan di bidang penelitian ilmu pengetahuan sesuai dengan ketentuan peraturean perundang-undangan yang berlaku

Fungsi:

Untuk melaksanakan tugas tersebut, LIPI mempunyai fungsi :

1. Pengkaji dan penyusunan kebijakan nasional di bidang penelitian ilmu pengetahuan
2. Penyelenggaraan riset keilmuan bersifat dasar
3. Penyelenggaraan riset inter dan multi disiplin terfokus
4. Pemantauan, evaluasi kemajuan dan penelaahan kecenderungan ilkmu pengetahuan dan teknologi
5. Koordinasi kegiatan fungsional dalam pelaksanaan tugas LIPI
6. Pelancaran dan pembinaan terhadap kegiatan instansi pemerintah di bidang penelitian ilmu pengetahuan
7. Penyelenggaraan pembinaan dan pelayanan administrasi umum di bidang perencanaan umum
8. Ketatausahaan, organisasi dan tata laksana, kepegawaian, keuangan, kearsipan, hukum, persandian, perlengkapan dan rumah-tangga

Kewenangan:

Untuk menjalankan fungsinya, LIPI mempunyai kewenangan :

1. Penyusunan rencana nasional secara makro di bidangnya
2. Perumusan kebijakan di bidangnya untuk mendukung pembangunan secara makro
3. Penetapan sistem informasi di bidangnya
4. Kewenangan lain yang melekat dan telah dilaksanakan sesuai dengan ketentuan peraturan perundang-undangan yang berlaku, yaitu :
    - Perumusan dan pelaksanaan kebijakan tertentu di bidang penelitian ilmu pengetahuan
    - Penetapan pedoman dan penyelenggaraan riset ilmu pengetahuan dasar
    - Penetapan pedoman etika ilmiah, kedudukan dan kriteria kelembagaan ilmiah
    - Pemberian ijin Peneliti Asing
    - Pemegang kewenangan ilmiah dalam keanekaragaman hayati.

2.1.2 Logo Instansi

Logo berupa tanda yang berfungsi dan memiliki aspek kunci yaitu reputasi. Logo pada jasa dan produk sebagai merek dagang memberikan jaminan kepada pemakainya tentang kualitas yang spesifik dan konsisten dari jasa dan produk tersebut. Reputasi bagi konsumen dan kalangan komersial sangat berarti dan merupakan asset yang berharga.

Yang dimaksud dengan Logo dan nama LIPI harus dijaga dan penggunannya memerlukan kewenangan yang jelas sehingga dapat memberikan imbalan moril dan materil.



Gambar 2.1 Logo Lipi

Logo LIPI berbentuk sebuah lingkaran yang terdiri dari dua bagian yang disatukan yaitu "pohon" dan "wadah", selain itu juga tulisan "LIPI" harus selalu menyertai karena merupakan suatu kesatuan yang tidak terpisahkan. Berwarna *Turqoise Blue* dengan spesifikasi warna.

Warna biru dikenal sebagai warna yang memiliki sifat atau kesan ilusi tenang. Jadi pengertian yang dikandung ialah ketenangan berpikirlah merupakan landasan di dalam pengabdian kepada Ilmu Pengetahuan.

Penampilan logo tidak boleh dimodifikasi, misalnya diberi background, diembos, warnanya dirubah, dan lain-lain. Harus mengikuti ketentuan dan logo yang sudah ada.

2.1.3 Struktur Organisasi PPET-LIPI

Di dalam dunia usaha dan bisnis sudah tidak dapat lagi dipisahkan dari apa yang disebut organisasi. Maka dari itu organisasi mempunyai arti yang sangat besar dalam suatu instansi/lembaga. Hal ini disebabkan karena adanya organisasi tujuan dan sasaran dari instansi dapat ditempuh secara efektif dan efisien melalui tindakan yang dilakukan bersama-sama.

Suatu instansi yang melibatkan banyak orang dalam menangani setiap kegiatan, haruslah memiliki organisasi yang berstruktur dan menggambarkan setiap pekerjaan yang harus dilakukan oleh setiap bagian.

Struktur Organisasi merupakan susunan yang menggambarkan berbagai fungsi sesuai dengan bidangnya, berdasarkan hirarki dan saluran kewenangannya yang berlaku dalam mencapai tujuan tertentu. Dengan adanya struktur organisasi akan terlihat bagaimana pembagian tanggung jawab tugas tersebut dijalankan sesuai dengan fungsinya masing-masing. Kejelasan struktur yang terdapat dalam suatu organisasi akan memberikan kejelasan tentang hubungan kerja secara fungsional antara satu bagian dengan bagian lain, baik hubungan vertikal maupun hubungan horizontal. Struktur organisasi sebuah instansi harus memungkinkan dengan menghadirkan adanya koordinasi usaha diantara semua satuan dan jenjang, untuk kemudian mengambil tindakan-tindakan yang dapat mencapai suatu tujuan umum.

Kepala Pusat PPET – LIPI membawahi seluruh Divisi yang ada di PPET – LIPI, mulai dari Divisi Penelitian yang dibagi berdasarkan Bidang-bidang Penelitian Elektronika dan Telekomunikasi, Divisi Sarana penelitian, dan Divisi Tata Usaha.

Divisi penelitian di PPET – LIPI sendiri ada tiga, yakni Bidang Elektronika, Bidang Telekomunikasi, dan Bidang Bahan dan Komponen Mikro Elektronika. Dimana di setiap Bagian tersebut dikepalai oleh Kepala Bagian dan memiliki Tim atau Kelompok fungsional yang meneliti di bagiannya masing-masing.

Dan yang mempersiapkan dan menyediakan sarana penelitian sendiri di bentuk dalam Divisi/Bidang sarana penelitian tersendiri. Dimana berdasarkan Bidang penelitian yang ada di PPET – LIPI ada tiga, maka Divisi Sarana Penelitian juga dibagi lagi menurut Bidang penelitian yang ada di PPET – LIPI, yakni Sub Bidang Sarana Elektronika, Sub Bidang Sarana Telekomunikasi, Sub Bidang Sarana Bahan dan Komponen Mikroelektronika.

Selain itu untuk mengurus keperluan Perusahaan yang bersifat Tata Usaha atau administrasi. PPET – LIPI memiliki Bagian Tata Usaha yang bertanggung jawab langsung kepada kepala Pusat PPET – LIPI dan

membawahi beberapa Sub Bagian, diantaranya Sub Bagian Kepegawaian, Sub Bagian Umum, dan Sub Bagian Jasa dan Informasi.

Pusat Penelitian Elektronika dan Telekomunikasi mempunyai tugas melaksanakan penyiapan bahan perumusan kebijakan, penyusunan pedoman, pemberian bimbingan teknis, penyusunan rencana dan program, pelaksanaan penelitian bidang elektronika dan telekomunikasi serta evaluasi penyusunan laporan .

2.2 Ruang Lingkup Kerja Praktek

Ruang lingkup kerja praktek kami meliputi teknologi informasi yaitu berupa program aplikasi desktop.

2.3 Landasan Teori

Text mining adalah suatu proses analisis dalam sekumpulan data dalam jumlah besar berdasarkan informasi berkualitas tinggi dari sebuah teks. Analisis yang bisa dilakukan dengan text mining diantaranya: klustering dan klasifikasi.

Secara umum, tahapan yang terjadi dalam melakukan analisis text mining adalah sebagai berikut:

- **Pembacaan data**

Pada tahap ini dilakukan pembacaan terhadap korpus, korpus adalah database besar yang menyimpan text yang akan dianalisis. Korpus bisa didapat dengan cara mengumpulkan data dari internet kemudian menyatukannya kedalam file text. Setelah terbentuk korpus barulah dimulai tahap pembacaan data.

- **Ekstraksi fitur**

Proses ekstraksi fitur bertujuan untuk meng-ekstrak kata-kata kunci dari korpus yang telah dibaca. Kata-kata kunci tersebut disebut dengan fitur atau term. Fitur inilah yang nantinya akan di proses dalam tahap analisis. Jadi proses ekstraksi fitur ini sangat berpengaruh dalam text mining.

Secara umum, proses ekstraksi fitur dapat di-break down kedalam 3 subproses yaitu:

- **Tokenization**

Tokenization bertujuan untuk memecah text menjadi kata tunggal. kata tunggal disini bisa terdiri dari satu kata atau bisa terdiri dari lebih dari satu kata. Misalnya dalam text bahasa indonesia, terdapat frase "Rumah Sakit", frase tersebut bisa dianggap 2 token yaitu "Rumah" dan "Sakit" atau bisa dianggap 1 token yaitu "Rumah Sakit".

- **Stopwords Removing**

Stopwords Removing merupakan proses selanjutnya setelah tokenization yang berfungsi menghilangkan kata tidak penting dalam text. Hal ini dilakukan untuk memperbesar akurasi dari pembobotan *term* nantinya. Untuk proses ini, diperlukan suatu kamus kata-kata yang bisa dihilangkan. Dalam bahasa indonesia, misalnya kata: dan, atau, mungkin, ini, itu, selanjutnya.. adalah kata-kata yang bisa dihilangkan.

- **Stemming**

Proses Stemming bertujuan mengubah sebuah kata turunan menjadi kata dasarnya dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata "diperintahkan", "memerintah", dan "memerintahkan" akan di-*stem* menjadi kata dasarnya yaitu "perintah".

- **Pembobotan term**

Setelah proses stemming kemudian dilakukan Pembobotan *term* yang bertujuan untuk memberikan sebuah nilai untuk sebuah *term* berdasarkan tingkat kepentingan tersebut didalam sekumpulan dokumen masukan.

Metode yang bisa digunakan untuk melakukan pembobotan term adalah metode *Term Frequency * Inverse Document Frequency (TF*IDF)*

Pembobotan ini bertujuan untuk suatu pengukuran statistik untuk mengukur seberapa penting sebuah kata dalam kumpulan dokumen. Tingkat kepentingan meningkat ketika sebuah kata muncul beberapa kali dalam sebuah dokumen tetapi diimbangi dengan frekuensi kemunculan kata tersebut dalam kumpulan dokumen. TF IDF dapat dirumuskan sebagai berikut,

$$TF\ IDF\ (t_k, d_j) = TF\ (t_k, d_j) \bullet IDF(t_k)$$

Dimana sebelumnya dihitung terlebih dahulu *Term Frequency* (TF) yaitu frekuensi kemunculan suatu *term* di tiap dokumen. Kemudian dihitung *Inverse Document Frequency* (IDF) yaitu nilai bobot suatu term dihitung dari seringnya suatu term muncul di beberapa dokumen. Semakin sering suatu term muncul di banyak dokumen, maka nilai IDF nya akan kecil.

- **Seleksi fitur**

Seleksi fitur bertujuan untuk mengurangi jumlah fitur yang akan di analisis. Dengan mengurangi fitur untuk dianalisis, diharapkan proses analisis menjadi lebih cepat, namun perlu dihasilkan akurasi yang cukup tinggi.

Metode termudah dalam mengurangi fitur ini bisa dilakukan sebagai berikut: mengurutkan *term* berdasarkan nilai bobotnya (TF*IDF), kemudian reduksi *term* yang memiliki bobot kecil sesuai dengan keinginan. Misalnya diinginkan fitur 75%, maka akan direduksi 25% *term* dengan bobot terkecil.

2.3.1 Pengertian Sistem Informasi

Menurut Kenneth C. Laudon dan Jane P. Laudon dalam bukunya "Management Information Systems : *New approaches to Organization & Technology"* [6] mengatakan bahwa Sistem informasi merupakan :

> "Komponen – komponen yang saling berhubungan dan bekerjasama untuk mengumpulkan, memproses, menyimpan dan mendistribusikan informasi tersebut untuk mendukung proses pengambilan keputusan, koordinasi dan pengendalian".

Menurut Patrick G. McKeown dalam bukunya *"Management Information Systems : Managing with Computers"* [7] mengatakan bahwa Sistem informasi merupakan :

> "Gabungan dari komputer dan user yang mengelola perubahan data menjadi informasi serta menyimpan data dan informasi tersebut".

Dari definisi – definisi di atas dapat disimpulkan bahwa Sistem informasi adalah kumpulan dari sub – sub sistem baik phisik maupun non phisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna

2.3.2 Sekilas tentang OOP (Object Oriented Programming)

OOP merupakan filosopi desain. OOP menggunakan set bahasa pemrograman yang berbeda dari bahasa pemrograman prosedural (C, Pascal, etc). Semua hal tentang OOP dikelompokan sebagai objek yang berkelanjutan. Oleh karena itu, anda mendapatkan kemampuan menggunakan kembali program dengan empat konsep pemrograman berbasis objek. [8]

Dalam rangka memperjelas pemahanan tentang orientasi objek, mari gunakan "tangan" anda sebagai contoh. Tangan merupakan kelas, tubuh anda memiliki dua buah objek tipe tangan, yaitu tangan kanan, dan tangan kiri. Fungsi utama mereka dikendalikan oleh sinyal kiriman melalui bahu anda (interface). Jadi bahu merupakan interface yang digunakan tubuh anda untuk berinteraksi dengan tangan anda. [8]

Objek dapat dianggap sebagai "sesuatu" yang dapat melakukan serangkaian kegiatan yang terkait. Serangkaian kegiatan objek didefinisikan melalui perilaku objek tersebut. Sebagai contoh, tangan bisa memegang sesuatu atau murid dapat memberi nama atau alamat. Dalam kondisi oop murni, objek adalah sebuah instansi dari kelas. [8]

Sebuah kelas hanyalah representasi dari jenis objek. Kelas merupakan blueprint/rencana/template yang menggambarkan detil objek. Sebuah kelas merupakan sebuah rancangan dari setiap individu yang diciptakan. Kelas terdiri dari tiga hal yaitu, nama, atribut, dan operasi. Sebagai contoh :

```
public class Student
{
}
```

Menurut sempel diatas, dapat dikatakan bahwa objek Student, bernama objectStudent, telah dibuat pada kelas Student.

Dalam dunia nyata, sering ditemukan banyak objek individu yang sejenis. Sebagai contoh, mungkin ada ribuan sepedah yang ada, semua dibuat dengan model yang sama. Setiap sepeda dirancang pada rancangan yang sama(blueprint). Pada kondisi orientasi objek, dapat dibilang bahwa sepada adalah instansi dari sebuah kelas objek yang dinamakan "Sepeda". [8]

Dalam dunia perangkat lunak, terkadang kita tidak sadar bahwa kita telah menggunakan kelas. Sebagai contoh "TextBox", yang selalu digunakan, terbuat dari kelas textbox yang mendefinisikan penampilan dan kemampuannya. Setiap anda menggunakan kontrol textbox maka anda membuat instansi baru dari kelas textbox. [8]

2.3.3 UML (Unified Modeling Language)

The Unified Modeling Language (UML) adalah bahasa pemodelan standar untuk perangkat lunak dan pengembangan sistem. Pernyataan ini sendiri adalah argumen yang cukup meyakinkan untuk membuat UML bagian dari repertoar perangkat lunak Anda, namun ia meninggalkan beberapa pertanyaan yang belum terjawab. Mengapa UML bersatu ? Apa yang dapat dimodelkan ? Bagaimana bahasa UML ? Dan, mungkin yang paling penting, mengapa Anda harus peduli ?. [9]

Sistem desain dalam skala yang besar cukup sulit. Apa pun dari aplikasi desktop sederhana untuk sistem skala perusahaan multi-tier lengkap, dapat terdiri dari ratusan potensi pada ribuan software dan komponen hardware. Bagaimana Anda (dan tim Anda) melacak komponen yang dibutuhkan, apa pekerjaan mereka, dan bagaimana mereka memenuhi kebutuhan pelanggan Anda ? Selain it , bagaimana Anda berbagi desain Anda dengan kolega Anda untuk memastikan potongan bekerja sama ? Ada terlalu banyak detail yang dapat disalahartikan atau dilupakan ketika mengembangkan sebuah sistem yang kompleks tanpa bantuan. Di sinilah modeling and tentu saja UML dibutuhkan. [9]

Dalam desain sistem, model anda untuk satu alasan penting : untuk mengelola kompleksitas. Modeling membantu anda melihat hutan untuk pohon, yang memungkinkan Anda untuk fokus pada, capture, dokumen, dan mengkomunikasikan aspek penting dari desain sistem anda. [9]

Sebuah model adalah abstraksi dari hal yang nyata. Ketika model sistem anda, abstrakan anda meninggalkan rincian yang tidak relevan atau berpotensi membingungkan. Model Anda adalah penyederhanaan sistem nyata, sehingga memungkinkan desain dan kelayakan sistem untuk dipahami, dievaluasi, dan mengkritik lebih cepat daripada jika Anda harus menggali melalui sistem itu sendiri. Bahkan lebih baik, dengan bahasa pemodelan formal, bahasa adalah abstrak namun hanya tepat sebagai bahasa pemrograman. Presisi ini memungkinkan bahasa menjadi mesin yang dapat dibaca, sehingga dapat diartikan, dieksekusi, dan diubah antara sistem. [9]

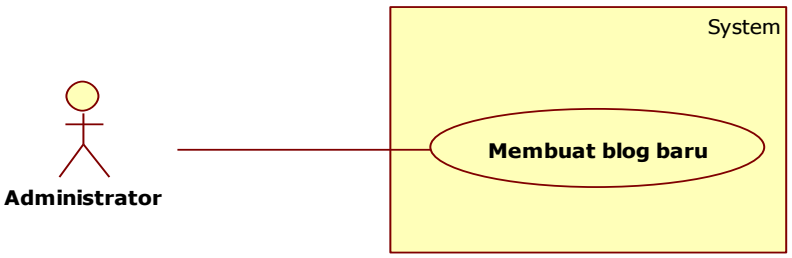
Untuk secara efektif memodelkan sistem, Anda perlu satu hal yang sangat penting : bahasa dengan yang model dapat dijelaskan. Dan di sinilah UML berguna. [9]

Dalam UML terdapat banyak diagram yang digunakan untuk melakukan pemodelan, disini penulis hanya menjelaskan beberapa diagram sesuai yang digunakan pada tahap perancangan. Berikut diagram-diagram yang ada pada UML [9], antara lain :

a. Use Case Diagram

Use case merupakan kasus atau situasi dimana sistem anda digunakan untuk memenuhi satu atau lebih dari kebutuhan konsumen, kasus penggunaan menangkap sebagian fungsionalitas yang tersedia pada sistem. Use case juga bisa dibilang jantung pemodelan anda, karena use case mempengaruhi dan membimbing semua elemen lain dalam sistem anda.

Use case pada pemodelan merupakan interaksi antara sistem dan pengguna atau sistem eksternal lainnya. Juga membantu dalam persyaratan pemetaan dengan sistem anda. Berikut contoh use case diagram :



Gambar 2.2 Contoh Use Case Diagram

Pada gambar tersebut administrator merupakan actor, bisa dibilang user atau pengguna sistem. Kemudian garis tersebut merupakan garis asosiasi yang menghubungkan user dengan fungsional sistem. Gambar yang membentuk seperti persegi yang bertuliskan system disudut kanan atas merupakan sistem anda. Dan yang berbentuk oval bertuliskan "membuat blog baru" merupakan use case atau fungsional atau menu yang terdapat pada sistem anda.

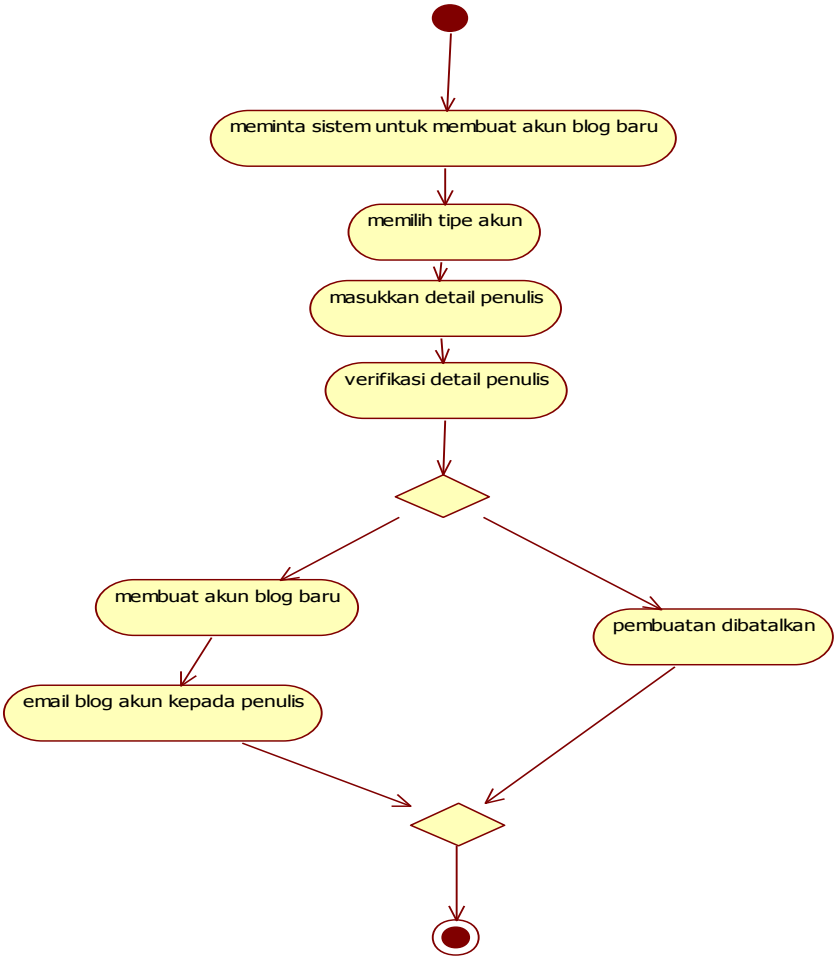
b. Activity Diagram

Diagram aktivitas sangat baik pada proses bisnis modeling. Sebuah proses bisnis adalah seperangkat tugas terkoordinasi yang mencapai tujuan bisnis, seperti pengiriman pesanan pelanggan. Beberapa alat manajemen proses bisnis (BPM) memungkinkan Anda untuk

menentukan proses bisnis menggunakan diagram aktivitas, atau notasi grafis yang sama, dan kemudian mengeksekusi mereka. Hal ini memungkinkan Anda untuk menentukan dan melaksanakannya, misalnya, proses persetujuan pembayaran di mana salah satu langkah memanggil web persetujuan kartu kredit service menggunakan notasi grafis yang mudah seperti diagram aktivitas. Diagram aktivitas adalah satu-satunya diagram UML dalam tampilan proses model sistem Anda.

Diagram aktivitas adalah salah satu yang paling mudah pada UML diagram karena mereka menggunakan simbol-simbol yang mirip dengan notasi flowchart yang dikenal luas, karena itu, mereka berguna untuk menggambarkan proses untuk khalayak luas. Bahkan, diagram aktivitas memiliki akar dalam diagram alur, serta diagram negara UML, diagram aliran data, dan Petri Nets.

Berikut contoh dari diagram aktivitas :

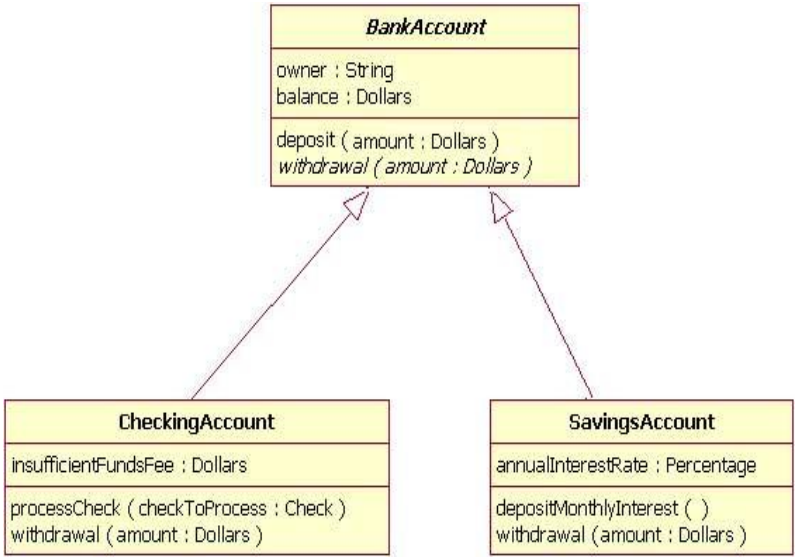


Gambar 2.3 Contoh Diagram Aktivitas

c. Class Diagram

kelas berada di jantung dari setiap sistem berorientasi obyek, karena itu, dapat dikatakan bahwa diagram UML yang paling populer adalah diagram kelas. Struktur Sebuah sistem terdiri dari kumpulan potongan yang sering disebut sebagai objek. Kelas menggambarkan berbagai jenis objek yang Anda miliki pada sistem, dan diagram kelas menunjukkan kelas-kelas dan hubungan mereka.

Use case menggambarkan perilaku sistem anda sebagai satu set yang mesti diperhatikan. Kelas menggambarkan berbagai jenis objek yang diperlukan dalam sistem Anda, untuk memenuhi perhatian tersebut. Kelas merupakan bagian dari pandangan logis model yang anda buat. Berikut :



Gambar 2.4 Contoh Class Diagram

d. Object Diagram

Objek diagram pada UML merupakan diagram yang menunjukkan pandangan lengkap atau sebagian dari struktur sistem dimodelkan pada waktu tertentu.

Dalam Unified Modeling Language (UML), diagram objek berfokus pada beberapa bagian tertentu dari objek dan atribut, dan hubungannya. korelasi pada diagram objek memberikan wawasan tentang bagaimana pandangan sistem ini diharapkan dapat berkembang dari waktu ke waktu. Dalam spesifikasi UML awal diagram objek digambarkan sebagai:

"Sebuah objek diagram adalah grafik dari sebuah kasus, termasuk objek dan nilai-nilai data. Sebuah diagram objek statis adalah turunan dari diagram kelas, hal itu menunjukkan snapshot dari tahap rinci tentang sistem pada suatu titik waktu. Penggunaan diagram objek cukup terbatas, untuk menunjukkan contoh struktur data."

Diagram objek dan diagram kelas terkait erat dan menggunakan notasi yang identik. Kedua diagram dimaksudkan untuk memvisualisasikan struktur statis dari sebuah sistem. Sementara

diagram kelas menunjukkan kelas, diagram objek menampilkan contoh kelas (objek). diagram Objek lebih konkrit daripada diagram kelas. Mereka sering digunakan untuk memberikan contoh atau bertindak sebagai kasus uji untuk diagram kelas. Hanya aspek yang berlaku di model biasanya ditampilkan pada diagram objek. Berikut contoh objek diagram :



Gambar 2.5 Contoh Objek Diagram

e. Sequence Diagram

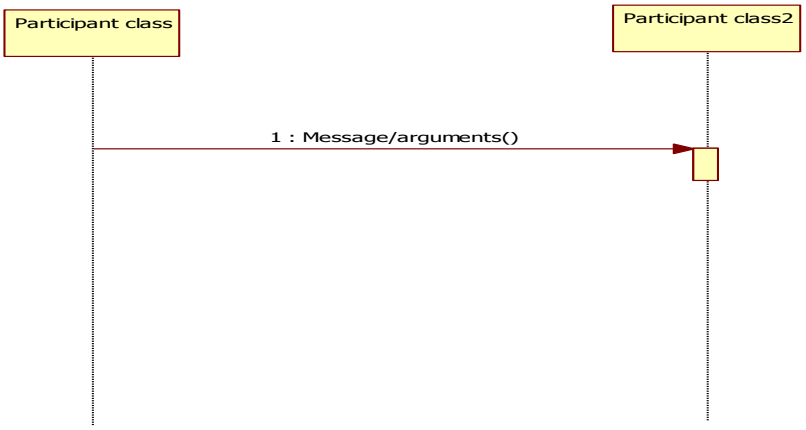
Use case memungkinkan model untuk menggambarkan apa yang harus sistem anda mampu melakukan, kelas memungkinkan model untuk menggambarkan berbagai jenis bagian yang membentuk struktur pada sistem anda. Ada satu bagian besar yang hilang dari penjelasan ini, dengan use case dan kelas saja, Anda belum dapat memodelkan bagaimana sistem anda bekerja sesuai tugasnya. Di sinilah diagram interaksi, dan khususnya diagram sequence, ikut dibutuhkan.

Sequence diagram merupakan anggota penting dari kelompok diagram interaksi. Model diagram interaksi merupakan interaksi penting antara bagian-bagian yang membentuk sistem anda dan merupakan bagian dari pandangan logis dari model.

Sequence diagram paling populer dari tiga tipe diagram interaksi. Ini bisa jadi karena mereka menunjukkan informasi yang tepat atau hanya karena mereka cenderung masuk akal bagi orang-orang baru yang mengenal UML.

Sequence diagram menyangkut hal urutan interaksi antara bagian-bagian dari sistem anda. Menggunakan diagram sequence, Anda bisa menggambarkan interaksi yang akan dipicu ketika use case tertentu dilaksanakan dan dalam rangka apa, interaksi akan terjadi. Sequence

diagram menunjukkan banyak informasi lain tentang interaksi, tapi bentuk mereka sederhana dan efektif di mana mereka berkomunikasi urutan peristiwa dalam suatu interaksi. Berikut contoh diagram sequence :



Gambar 2.6 Contoh Sequence Diagram

2.3.4 Konsep Perangkat Lunak pendukung

2.3.4.1 Microsoft Visual Studio

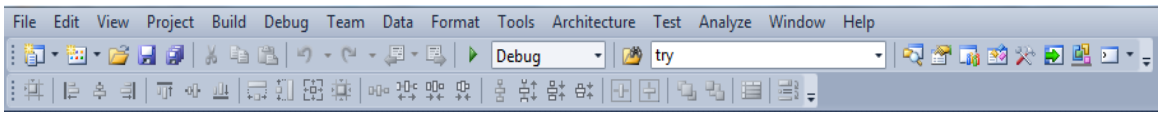
Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

Untuk membuat suatu program aplikasi pada visual studio, maka diperlukan suatu struktur aplikasi atau komponen yang digunakan, berikut beberapa aplikasi yang penulis gunakan pada visual studio antara lain :

a. Toolbar

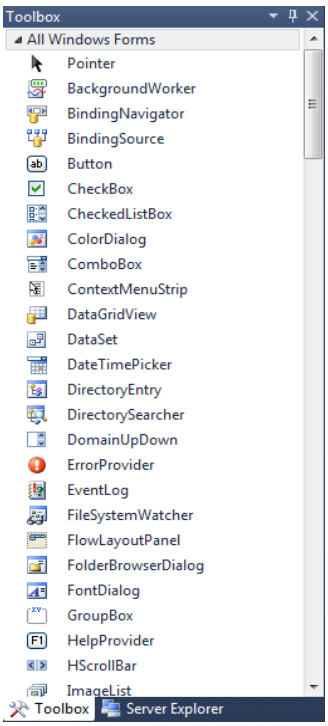
Terdiri atas beberapa komponen yang digunakan untuk membuat objek pada form dan dapat mempercepat pengaksesan perintah – perintah yang ada dalam pemrograman.



Gambar 2.7 ToolBar Visual Studio

b. Toolbox

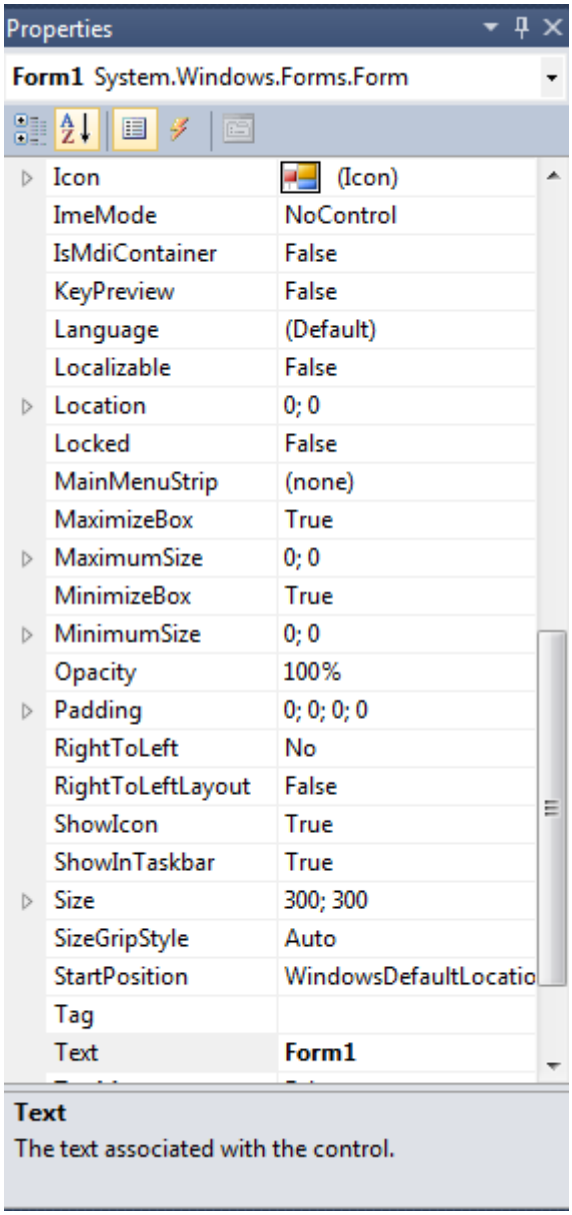
Toolbox Adalah sebuah jendela yang berisi tombol kontrol yang akan user gunakan untuk mendesain atau membangun sebuah form atau report. Toolbox terdiri atas beberapa tombol untuk mengendalikan tampilan, berikut gambar toolbox pada visual studio :



Gambar 2.8 Tampilan Toolbox Visual Studio

c. Jendela Propertis

Jendela propertis adalah jendela yang berfungsi sebagai tempat mengedit properti pada suatu objek yang terpilih pada saat mendesain atau membangun aplikasi bagi user. Berikut gambar jendela propertis :

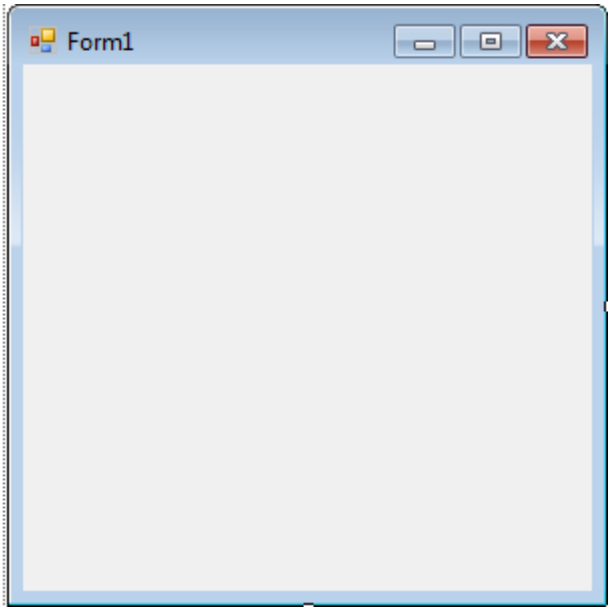


Gambar 2.9 Tampilan Jendela Propertis

d. Form

Form merupakan tempat untuk mendesain atau membangun sebuah program, didalamnya bisa ditanamkan sebuah coding agar form tersebut

dapat menjalankan program yang telah didesain oleh user. Berikut gambar form :



Gambar 2.10 Tampilan Form

2.3.4.2 Mysql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata 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 sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.