BAB II
LANDASAN TEORI

II.1 Bahasa Sunda

Bahasa Sunda adalah sebuah bahasa dari cabang Melayu-Polinesia dalam rumpun bahasa Austronesia. Bahasa ini dituturkan oleh sekitar 27 juta orang dan merupakan bahasa dengan penutur terbanyak kedua di Indonesia setelah Bahasa Jawa [7]. Bahasa Sunda dituturkan di sebagian besar provinsi Jawa Barat (kecuali kawasan pantura yang merupakan daerah tujuan urbanisasi di mana penutur bahasa ini semakin berkurang), melebar hingga batas Kali Pemali (Cipamali) di wilayah Brebes dan Majenang, Cilacap Jawa Tengah, dan di kawasan selatan provinsi Banten. Dari segi linguistik, bersama bahasa Baduy, bahasa Sunda membentuk suatu rumpun bahasa Sunda yang dimasukkan ke dalam rumpun bahasa Melayu-Sumbawa.

Pemakaian bahasa sunda tidak hanya dipergunakan dalam komunikasi lisan saja, tetapi bahasa itu dipergunakan juga dalam komunikasi tulisan. Komunikasi seperti itu terjadi dalam komunikasi pemerintah di wilayah Provinsi Jawa Barat yang dilakukan oleh pamong desa dengan masyarakat dan sebaliknya, baik suasana dinas maupun di luar suasana dinas. Para pamong desa sudah terbiasa menggunakan bahasa Indonesia, karena mereka menyadari bahwa bahasa Indonesia merupakan bahasa resmi yang dipahami oleh semua pihak dari berbagai suku bangsa di Indonesia. Bahasa Indonesia dirasakan lebih tegas pemakaiannya daam menghadapi sebuah pekerjaan, sedangkan bahasa Sunda dirasakan lebih akrab dengan orang yang diajak bicara, mendekatkan rasa kekeluargaan serta dapat dipergunakan dalam hubungan dinas (berdasarkan hasil penelitian yang pernah dilakukan Tim Penelitian Fakultas Sastra, Universitas Padjajaran). Bahasa Sunda terutama dipergunakan dalam suasana tidak resmi, yaitu dalam hubungan kerluarga di rumah, antartetangga sesuku yang disampaikan, baik secara lisan maupun tulisan.

II.1.1 Ragam Hormat Bahasa Sunda

Menurut Kamus Besar Bahasa Indonesia bahasa sopan/hormat adalah ragam bahasa yang dipakai dalam situasi sosial yang mewajibkan adanya norma sopan santun. Menurut hasil Kongres Bahasa Sunda tahun 1986 di Cipayung Bogor, tatakrama bahasa Sunda yang disebut juga Undak Usuk Basa Sunda (UUBS). Menurut penelitian tatakrama bahasa Sunda atau sering disebut juga Undak Usuk Basa Sunda (UUBS) bentuknya ada beberapa ragam (tingkat/jenis) yang biasanya digunakan dalam bahasa Sunda, diantaranya yaitu, ragam bahasa hormat, ragam bahasa loma/kasar [8].

Pada hakekatnya digunakan ragam hormat tidak lain untuk menunjukkan rasa hormat dari pembicara kepada yang diajak bicara dan pada siapa yang menjadi bahan pembicaraan. Ragam bahasa Sunda juga memiliki parameter pemakaiannya dengan melihat usia (tua atau muda), berpendidikan atau tidak, pria atau wanita. Undak Usuk Basa Sunda ada enam jenis diantaranya :

1. Basa Kasar

Basa kasar disebut juga bahasa loma. Digunakan kepada sesama, kepada teman yang sudah akrab. Selain itu (jaman dulu) selalu dipakai juga untuk berbicara kepada orang yang umur dan pangkat dan kedudukannya dibawah si pembicara. Atau bisa juga digunakan untuk membicarakan orang yang umurnya dibawah si pembicara.

2. Basa sedang

Basa sedang sering juga disebut sebagai bahasa lemes keur ka sorangan (halus untuk diri sendiri), yaitu bahasa yang digunakan untuk diri sendiri seperti misalnya berbicara menggunakan bahasa halus atau untuk berbicara kepada orang yang lebih tua. Selain itu bahasa Sedang juga dapat dipakai untuk berbicara kepada orang yang belum dikenal atau akrab apabila yang mengajak berbicara menggunakan bahasa halus.

3. Basa lemes

Basa lemes sering disebut juga sebagai bahasa lemes keur ka batur (bahasa halus untuk orang lain). Bahasa ini digunakan untuk berbicara kepada orang yang umurnya di atas pembicara dan untuk membicarakan orang yang pangkat,

kedudukan dan umurnya di atas kita. Bahasa halus juga dapat dipakai kepada orang yang belum kita kenal.

4. Basa lemes pisan

Ragam bahasa ini dipakai untuk menghormati orang yang kedudukannya lebih tinggi dari pembicara.

5. Basa kasar pisan

Ragam bahasa ini dapat disebut juga sebagai bahasa cohag. Bahasa ini biasanya dipakai oleh orang-orang yang sedang marah atau bertengkar dengan maksud untuk saling menghina. Tetapi umumnya, ragam bahasa ini ditujukkan untuk binatang karena jika ditujukkan pada manusia, bahasa ini akan terasa sangat kasar dan menyinggung.

6. Basa menengah

Ragam bahasa ini dipakai untuk berbicara dengan orang yang pangkat dan kedudukannya di bawah pembicara tetapi umurnya di atas pembicara. Ragam bahasa ini dipakai juga ketika berbicara dengan orang yang menggunakan bahasa halus dan orang yang dibicarakannya itu memiliki pangkat dan kedudukan dibawah mereka, tetapi umurnya di atas mereka. Ragam bahasa ini tingkatannya ada di bawah bahasa halus tetapi di atas bahasa kasar.

II.2 Bahasa Indonesia

Bahasa Indonesia adalah bahasa resmi Republik Indonesia dan bahasa persatuan bangsa Indonesia [9]. Bahasa Indonesia diresmikan penggunaannya setelah Proklamasi Kemerdekaan Indonesia, tepatnya sehari sesudahnya, bersamaan dengan mulai berlakunya konstitusi. Di Timor Leste, bahasa Indonesia adalah bahasa kerja (working language). Dari sudut pandang linguistika, bahasa Indonesia adalah suatu varian bahasa Melayu. Dasar yang dipakai adalah bahasa Melayu Riau dari abad ke-19, namun mengalami perkembangan akibat penggunaanya sebagai bahasa kerja dan proses pembakuan di awal abad ke-20. Hingga saat ini, bahasa Indonesia merupakan bahasa yang hidup, yang terus menghasilkan kata-kata baru, baik melalui penciptaan, maupun penyerapan dari bahasa daerah dan bahasa asing.

Meskipun saat ini dipahami oleh lebih dari 90% warga Indonesia, bahasa Indonesia tidak menduduki posisi sebagai bahasa ibu bagi mayoritas penduduknya. Sebagian besar warga Indonesia berbahasa daerah sebagai bahasa ibu. Penutur bahasa Indonesia kerap kali menggunakan versi sehari-hari (kolokial) dan/atau mencampuradukkan dengan dialek Melayu lainnya atau bahasa ibunya. Namun demikian, bahasa Indonesia digunakan sangat luas di perguruan-perguruan, di surat kabar, media elektronika, perangkat lunak, surat-menyurat resmi, dan berbagai forum publik lainnya sehingga dapatlah dikatakan bahwa bahasa Indonesia digunakan oleh semua warga Indonesia.

Sebagai suatu bangsa yang hidup di tengaah-tengah percaturan politik dan kebudayaan dunia, bangsa Indonesia menerima pengaruh-pengaruh dari luar. Demikian juga berlaku dalam segi bahasa. Kata-kata asing masuk ke dalma bahasa Indonesia seperti bahasa Sansakerta, bahasa Arab, bahasa Portugis, bahasa Tionghoa, bahasa Belanda, bahasa Inggris. Pengaruh ini tidak terbatas pada pemungutan kata-kata, tetapi tampak juga pada struktur kata dan kalimat.

II.3 Pengertian Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Kamus berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus. Biasanya hal ini terdapat dalam kamus bahasa Perancis.

Kata kamus diserap dari bahasa Arab *qamus* (قاموس), dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *Ωκεανός* (*okeanos*) yang berarti 'samudra'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan,khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan khazanah yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya.

II.3.1 Jenis-jenis kamus

Berdasarkan penggunaan bahasa kamus dapat ditulis dalam satu atau lebih dari satu bahasa. Dengan demikian kamus terbagi menjadi beberapa jenis yaitu:

a. Kamus Eka Bahasa

Kamus ini hanya menggunakan satu bahasa. Kata-kata (entri) yang dijelaskan dan penjelasannya adalah terdiri dari bahasa yang sama. Kamus ini memiliki perbedaan yang jelas dengan kamus dwibahasa karena penyusunan dibuat berdasarkan pembuktian data korpus (data yang dipakai sebagai sumber bahan penelitian). Ini bermaksud definisi makna ke atas kata-kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandung kata-kata berhubungan. Contoh bagi kamus ekabahasa adalah *Kamus Besar Bahasa Indonesia* (di indonesia).

b. Kamus Dwibahasa

Kamus ini menggunakan duaa bahasa, yaitu kata masukan dari bahasa yang dikamuskan diberi padanan atau penerjemahan dengan menggunakan bahasa yang lain. Contohnya *Kamus Inggris-Indonesia.*

c. Kamus Aneka Bahasa

kamus aneka bahasa sekurang-kurangnya menggunakan tiga bahasa atau lebih. Misalnya, kata bahasa Melayu, bahasa Inggris, dan bahasa Mandarin secara serentak. Contoh untuk kamus aneka bahasa adalah *Kamus Melayu-Cina-Inggris Pelangi* susunan Yuen Boon Chan pada tahun 2004.

II.4 *Translator*

Penerjemahan literal (*literal translation*) atau disebut juga penerjemahan lurus (*linier translation*) berada diantara penerjemahan kata demi kata dan penerjemahan bebas (*free translationi)*. Dalam proses penerjemahannya, penerjemah mencari konstruksi gramatikan BSu yang sepadan atau dekat dengan Bsa. Penerjemahan literal ini terlepas dari konteks. Penerjemahan ini mula – mula dilakukan seperti penerjemahan kata demi kata, tetapi penerjemah kemudian menyesuaikan susunan kata-katanya sesuai dengan gramatikal BSa.

Penerjemahan literal merupakan salah satu teknik yang digunakan dalam proses terjemah. Teknik ini mencoba menterjemahkan sebuah kata atau ungkapan kata perkata. " *Literal translations is to translate a word or an expression word for word".* Yang dimaksud Molina dan Hurtado Albir dengan kata demi kata pada definisi ini, bukan berarti menerjemahkan satu akta untuk kata yang lainnya, tetapi lebih cenderung kepada menerjemahkan kata demikata berdasarkan fungsi dan maknanya dalam tataran kalimat.

Dilihat dari jauh dekatnya terjemahan dari bahasa sumber dan bahasa sasaran, terjemah dapat diklasifikasikan ke dalam delapan jenis. Kedelapan jenis terjemahan tersebut dapat dikategorisasikan dalam dua bagian besar. *Pertama*, terjemahan yang lebih berorientasi pada bahasa sumber, dalam hal ini penerjemah berupaya mewujudkan kembali dengan setepat-tepatnya makna kontekstual penulis, meskipun dijumpai hambatan sintaksis dan semantik yakni hambatan bentuk dan makna. *Kedua*, terjemahan yang lebih berorientasi pada bahasa sasaran. Dalam hal ini penerjemah berupaya menghasilkan dampak yang relatif sama dengan yang diharapkan oleh penulis asli terhadap pembaca versi bahasa sasaran.

Dilihat dari orientasinya terhadap bahasa sumber, terjemahan dapat diklasifikasikan:

1) Terjemahan kata demi kata (*word for word translation*). Penerjemahan jenis ini dianggap yang paling dekat dengan bahasa sumber. Urutan kata dalam teks bahasa sumber tetap dipertahankan, kata-kata diterjemahkan menurut makna dasarnya diluar konteks. Kata-kata yang bermuatan budaya diterjemahkan secara harfiah. Terjemahan kata demi kata berguna untuk memahami mekanisme bahasa sumber atau untuk menafsirkan teks yang sulit sebagai proses awal penerjemahan.

2) Terjemahan Harfiah (*literal translation*) atau sering juga disebut terjemahan struktural. Dalam terjemahan ini konstruksi gramatikal bahasa sumber dikonversikan ke dalam padanannya dalam bahasa sasaran, sedangkan kata-kata diterjemahkan di luar konteks. Sebagaimana proses penerjemahan awal terjemah harfiah ini dapat membantu melihat masalah yang perlu diatasi.

3) Terjemahan setia (*faithful translation*). Terjemahan ini mencoba menghasilkan kembali makna kontekstual walaupun masih terikat oleh struktur gramatikal bahasa sumber. Ia berpengang teguh pada tujuan dan maksud bahasa sumber sehingga terkesan kaku. Terjemahan ini bermanfaat sebagai proses awal tahap pengalihan.

4) Terjamahan semantis (*semantic teranslation*). Berbeda dengan terjemahan setia. Terjemahan semantis lebih memperhitungkan unsur estetika teks bahasa sumber, sdan kreatif dalam batas kewajaran. Selain itu terjemahan setia sifatnya masih terkait dengan bahasa sumber, sedangkan penerjemahan semantis lebih fleksibel. Apabila ungkapan pasemon (*kinayah*) di atas terjemahan secara semantis, maka hasil terjemahnanya adalah *'dia laki-laki adalah seorang pemberani, terhormat dalam lingkungan keluarga dan masyarakatnya, dan seorang dwermawan'*.

Klasifikasi terjemahan yang berorientasi pada bahasa sasaran yaitu:

1) Terjemahan adaptasi (*adaptation*). Terjemahan inilah yang dianggap paling bebas dan palingdekat kebahasaan sasaran. Terutama untuk jenis terjemahan drama dan puisi, tema, karakter dan alur biasanya dipertahankan. Dalam karangan ilmiah logikanya diutamakan, sedangkan contok dikurangi atau ditiadakan.

2) Terjemahan bebas (*free trantation*). Penerjemahan bebas adalah penulisan kembali tanpa melihat tanpa aslinya. Biasanya merupakan parafrase yang dapat lebih pendek atau lebih panjang dari aslinya.

3) Terjemahan idiomatiuk (*idiomatic translation*). Dalam terjemahan jenis ini pesan bvahasa sumber disampaikan kembali tetapi ada penyimpangan nuansa makan karena mengutamakan kosa kata sehari-hari dan idiom dan tidak ada di dalam bahasa sumber tetapi bisa dipakai dalam bahasa sasaran.

4) Terjemahan komunikatif (*communicative translation*). Terjermahan ini berusaha menyampaikan makna kontekstual dari bahasa sumber sedemikian rupa, sehingga isiu dan bahasanya berterima dan dapat dipahami oleh dunia pembaca bahasa sasaran. Terjemahan ini biasanya dianggap terjemahan yang ideal.

II.5 Stemming

Stemming adalah proses menghilagkan imbuhan dari suatu kata dari suatu dokumen atau *query*. Permasalahan utama dalam stemming adalah bagaimana mendapatkan kata dasar yang benar dari hasil pemenggalan imbuhan dari suatu kata jadian. Efektifitas algoritma stemming dipengaruhi oleh beberapa faktor:

a. Kesalahan dalam proses pemenggalan imbuhan dari kata dasarnya. Kesalahan ini dapat berupa:

1. *Overstemming*: yaitu pemenggalan imbuahan yang melebihi dari yang seharusnya.

Contoh: kata *masalah* menjadi *masa*. Kesalahan ini dapat timbul karena bentuk kata dasar yang menyerupai imbuhan.

2. *Understemming*: yaitu pemenggalan imbuhan yang terlalu sedikit dari yang seharusnya.

Contoh: kata *belajar* menjadi *lajar*. Kesalahan ini dapat timbul karena kekurangan pada aturan pola imbuhan yang didefinisikan.

3. *Unchange*: yaitu kasus khusus dari understemming, dimana tidak terjadi pemenggalan imbuhan sama sekali.

Contoh: kata *telapak*, setelah pemenggalan kata dasar yang didapat tetap *telapak*.

Spelling exception: yaitu huruf pertama kata dasar yang didapat tidak benar yang diakibatkan dari pemenggalan awalan. Contoh: kata *memukul* menjadi *ukul*. Kesalahan ini dapat ditimbulkan karena ada beberapa imbuhan yang berubah bentuk ketika ditempelkan pada suatu kata dasar.

Misalnya awalan *beR-*, *meN-*, *teR-*, *peR-*, akan bergantung pada huruf pertama kata dasar dimana imbuhan tersebut ditempelkan

Contoh: *ber-* + *a*jar = *bel*ajar, *pen-* + *l*ihat = *peng*lihatan, *pen-* + *s*akit = *peny*akit.

Atau sebaliknya ada imbuhan yang mengakibatkan huruf pertama kata dasar yang ditempelinya menjadi luluh. Misalnya *meng- / peng-* meluluhkan huruf '*k*' (Contoh: mengarang dari *meng-* dan *k*arang) atau *men- / pen-* meluluhkan huruf '*p*' Contoh: menuai dari *men-* dan *t*uai.

b. Kekurangan dalam perumusan aturan penambahan imbuhan pada kata dasar. Hal ini dapat terjadi karena morfologi bahasa Indonesia yang kompleks, sehingga sangat sulit atau bahkan tidak mungkin untuk merumuskan aturan yang sempurna.
c. Jumlah total aturan imbuhan yang didapat berhubungan dengan efektifitas proses temu kembali. Dimana semakin banyak pola penambahan imbuhan yang dapat dirumuskan, maka proses temu kembali akan semakin efektif.

II.6 OOP (*Object Oriented Programming*)

OOP (*Object Oriented Programming*) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

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 bagian 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 adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep berikut:

II.6.1 Kelas (*Class*)

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 objek. 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.

II.6.2 Objek (*Object*)

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.

II.6.3 Abstraksi (*Abstract*)

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 23 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. \

II.6.4 Enkapsulasi (*Encapsulation*)

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek

tersebut yang diberi ijin 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.

II.6.5 Polimorfisme (*Polymorphism*)

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 menggerakan 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 24 berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

II.6.6 Inheritas (*Inheritance*)

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

II.7 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk

pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.

UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. 25 UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan system. UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu :

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Conceptual Diagram* untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration Diagram* untuk memodelkan interaksi antar *objects.*
5. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku *Use Cases* dan *objects* di dalam sistem.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur object.
9. *Component Diagram* untuk memodelkan komponen object.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.