

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkati

Yohanes Setyo Prabowo, Kusriani, Andi Sunyoto, 2015 dengan judul “Sistem Pendukung Keputusan Pemilihan Jurusan SNMPTN Bagi Siswa SMAN 7 Purworejo”. Dari penelitian tersebut menghasilkan perhitungan akurasi menunjukkan akurasi model untuk tahun 2013/2014 adalah sebesar 75.7% dan tahun 2014/2015 sebesar 72.6%. Penurunan akurasi dari data tahun 2013/2014 dan 2014/2015 kemungkinan disebabkan karena semakin banyaknya peserta yang mengikuti SNMPTN sehingga menambah ketat persaingan.

Muhammad Taufik Irawan dan Dannya Kriestanto, 2016 yang berjudul “Penerapan *Profile Matching* Untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi” yang menghasilkan sebuah sistem pendukung keputusan menentukan calon penerima beasiswa yang mampu memberikan solusi berupa hasil ranking dari seleksi Metode *Profile Matching* berdasarkan kriteria yang telah di tentukan.

Lili Tanti, 2016 melakukan penelitian yang berjudul “Analisis Promosi Kenaikan Jabatan Berdasarkan Evaluasi Kinerja Pegawai”. Kenaikan jabatan merupakan suatu faktor yang sangat penting bagi perencanaan karir pegawai dan juga untuk meremajakan suatu posisi jabatan agar diduduki oleh seseorang yang mempunyai kriteriakriteria yang cocok. Untuk itu perlu diterapkan salah satu metode didalam sistem pendukung keputusan didalam memilih pegawai yang

cocok sesuai dengan kriteria dilingkungan Universitas Potensi Utama yaitu aspek penilaian umum, aspek penguasaan bidang kerja dan aspek perilaku. Sehingga dengan penerapan metode profile matching didalam system pendukung keputusan dalam proses evaluasi kinerja pegawai diharapkan dapat menentukan urutan prioritas pegawai, yang digunakan untuk membantu pihak manajemen dalam membuat keputusan untuk promosi kenaikan jabatan pegawai yang dinilai berdasarkan evaluasi kinerja pegawai dengan profil jabatan .

Frieyadie, 2016 dengan berjudul “Penggunaan Metode Profile Matching Untuk Sistem Penunjang Keputusan Kenaikan Jabatan Pada Instansi Pemerintah”. Hasil penelitian dari perhitungan Profile Matching menyatakan bahwa alternatif yang terpilih dan paling sesuai dengan kriteria adalah pegawai-5 (P5) unggul 4,15 berbanding dengan pegawai-1 (P1) sebesar 4,06, pegawai-4 (P4) dengan nilai 4.02, pegawai-3 (P3) sebesar 3,88 dan pegawai-2 (P2) sebesar 3,85. Yang memiliki 15 kriteria dan 4 aspek.

Heru Purwanto, 2017 yang berjudul “Penerapan Metode Profile Matching Dalam Sistem Pendukung Keputusan Penilaian Kinerja Karyawan Pada PT.Hyundai Mobil Indonesia Cabang Kalimalang”. Dalam penelitian kali ini metode Profile Matching menjadi pilihan guna memberikan penilaian dan evaluasi kinerja karyawan untuk dipromosikan menjadi Leader. Ada tiga aspek dalam melakukan penilaian dan evaluasi tersebut, yaitu aspek kecerdasan, aspek sikap kerja, aspek perilaku. Setelah dilakukan pembobotan melalui perhitungan dengan metode *Profile Matching* dimana karyawan dengan kode K5 memiliki Rangking

pertama dengan Nilai paling tinggi sebesar 6,55. Sebelumnya perhitungan manual sederhana berada diposisi ke 2.

II.2. Landasan Teori

II.2.1 Profile Matching

Metode *Profile Matching* adalah metode yang sering digunakan sebagai mekanisme dalam pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti. Dalam proses *Profile Matching* secara garis besar merupakan proses membandingkan antara nilai data aktual dari suatu profil yang akan dinilai dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar. (Yohanes Setyo Prabowo , et al : 2015).

Tahapan metode *profile matching* adalah sebagai berikut :

a. Pembobotan Nilai Gap

Pada tahap ini, akan ditentukan bobot nilai masing-masing aspek dengan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Adapun inputan dari proses pembobotan ini adalah selisih dari profil siswa dan profil nilai siswa.

b. Pemetaan Gap.

Gap yang dimaksud adalah perbedaan antara profil siswa dan profil nilai siswa.

GAP = Profil Siswa - Profil Nilai Siswa - - - - -

c. Pencocokan Tabel Bobot Nilai.

Melakukan pencocokan dengan tabel bobot Gap Hasil Gap dari pengurangan profil pegawai dan profil jabatan bila dicocokkan dengan kolom selisih gap pada tabel bobot nilai yang dihasilkan sama yang ditunjukkan pada Tabel 1.

Tabel II.1. (Lili Tanti : 2016)

Tabel II.1. Bobot Nilai

Selisih	Bobot Nilai	Keterangan
0	5	Tidak ada selisih (kompetensi sesuai dengan yg dibutuhkan)
1	4,5	Kompetensi individu kelebihan 1 tingkat
-1	4	Kompetensi individu kekurangan 1 tingkat
2	3,5	Kompetensi individu kelebihan 2 tingkat
-2	3	Kompetensi individu kekurangan 2 tingkat
3	2,5	Kompetensi individu kelebihan 3 tingkat
-3	2	Kompetensi individu kekurangan 3 tingkat
4	1,5	Kompetensi individu kelebihan 4 tingkat
-4	1	Kompetensi individu kekurangan 4 tingkat

d. Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai gap kriteria yang dibutuhkan, kemudian tiap kriteria dikelompokkan lagi menjadi dua kelompok yaitu *core factor* dan *secondary factor*.

1) *Core Factor* (Faktor Utama)

Core Factor merupakan aspek (kompetensi) yang menonjol/paling dibutuhkan oleh suatu jabatan yang diperkirakan dapat menghasilkan kinerja optimal.

Untuk menghitung *Core Factor* digunakan rumus:

$$NCF = \frac{\sum NC(i, s, p)}{\sum IC}$$

Keterangan:

NCF = Nilai rata-rata *core factor*

NC = Jumlah total nilai *core factor*

IC = Jumlah item *core factor*

2) *Secondary Factor* (Faktor Pendukung)

Secondary Factor adalah item-item selain aspek yang ada pada *core factor*.

Untuk menghitung *secondary factor* digunakan rumus:

$$NSF = \frac{\sum NS(i, s, p)}{\sum IS}$$

Keterangan:

NSF = Nilai rata-rata *secondary factor*

NS = Jumlah total nilai *secondary factor*

IS = Jumlah item *secondary factor*

3) Perhitungan Nilai Total

Dari perhitungan *core factor* dan *secondary factor* dari tiap - tiap aspek, kemudian dihitung nilai total dari tiap-tiap aspek yang diperkirakan berpengaruh pada kinerja tiap-tiap profile.

Untuk menghitung nilai total dari masing-masing aspek, digunakan rumus:

$$N = (X)\% NCF + (X)\% NSF$$

Keterangan:

N = Nilai total tiap aspek

NCF = Nilai rata-rata *core factor*

NSF = Nilai rata-rata *secondary factor*

(X)% = Jumlah item *secondary factor*

e. Perangkingan

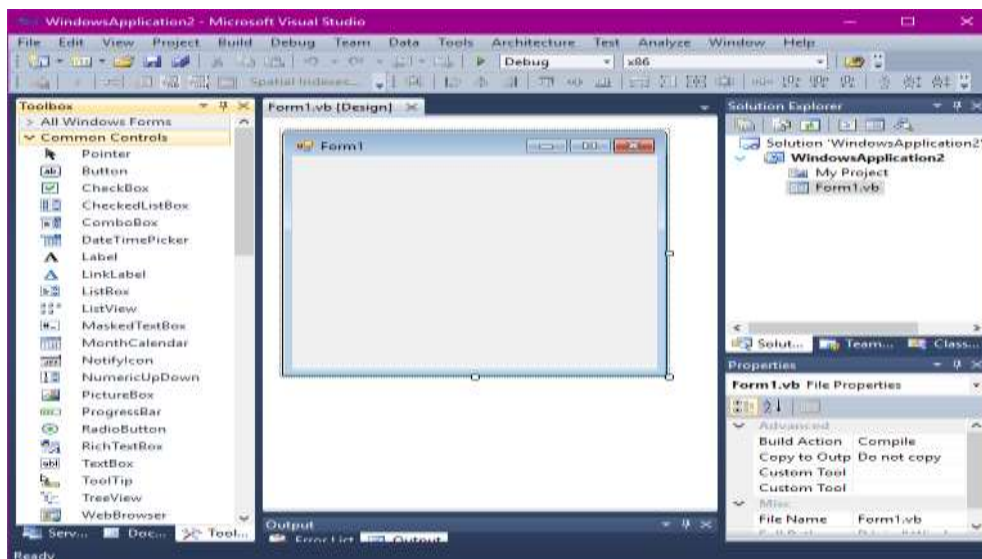
Hasil akhir dari proses *profile mathcing*. (Yohanes Setyo Prabowo , et al : 2015)

II.2.2. Visual Studio 2010

Menurut Edi Winarno, et al (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer di dunia. Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melaksanakan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows.

II.2.2.1. Lingkungan Kerja Visual studio

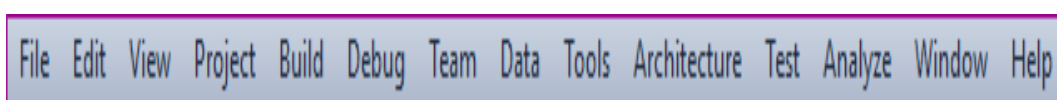
Lingkungan kerja merupakan lingkungan pengembangan program yang terintegrasi yang bersifat visual dan mudah di gunakan untuk menghasilkan program aplikasi seperti berikut. Adapun lingkungan kerjanya dapat dilihat pada gambar II.1.



Gambar II.1 Lingkungan Kerja Micosoft Visual Studio

a. Basis menu

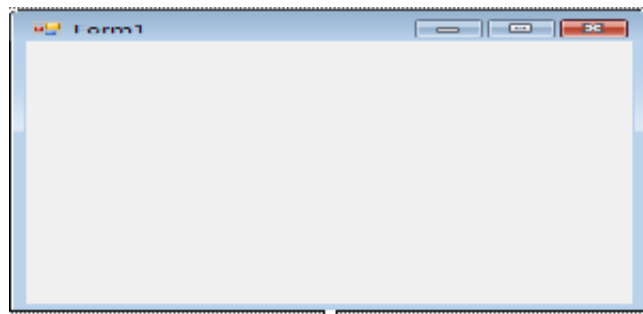
Basis menu terletak pada IDE. Menu merupakan kumpulan dari perintah-perintah yang dikelompokkan dalam kriteria operasi yang dihasilkan. Menu ini berisi semua perintah dalam *Visual basic* yang dapat anda pilih dengan menggunakan mouse atau keyboard. Adapun tampilan menu utama dapat dilihat pada gambar II.2.



Gambar II.2 Menu Utama

b. Form

Form adalah suatu objek yang dipakai sebagai tempat bekerja program aplikasi. *Form* berbentuk jendela yang dapat ditayangkan sebagai kertas atau meja kerja yang dapat dilukis atau diletakkan ke dalam objek-objek lain. Adapun tampilan *form* dapat dilihat pada gambar II.3.



Gambar II.3 Form

c. Toolbar

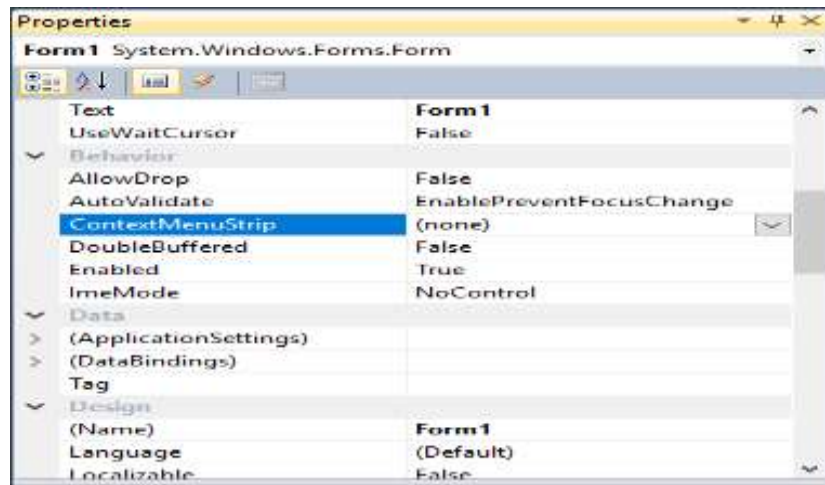
Toolbar adalah tombol-tombol yang mewakili suatu perintah tertentu dari Visual Basic yang sangat membantu dalam mempercepat akses perintah. Adapun tampilan *toolbar* dapat dilihat pada gambar II.4.



Gambar II.4 Toolbar

d. Windows Property

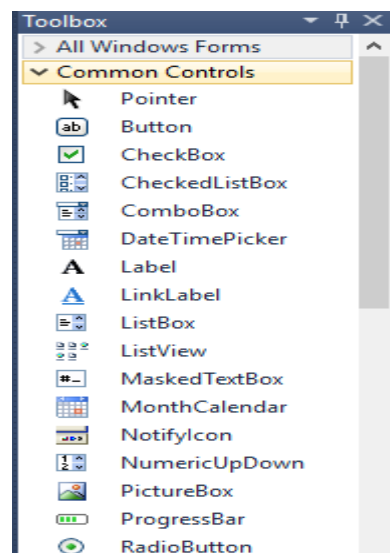
Windows Property adalah jendela yang berisi semua informasi tentang kontrol yang terdapat pada *form*. Adapun tampilan *windows property* dapat dilihat pada gambar II.5.



Gambar II.5 Windows Property

e. Toolbox

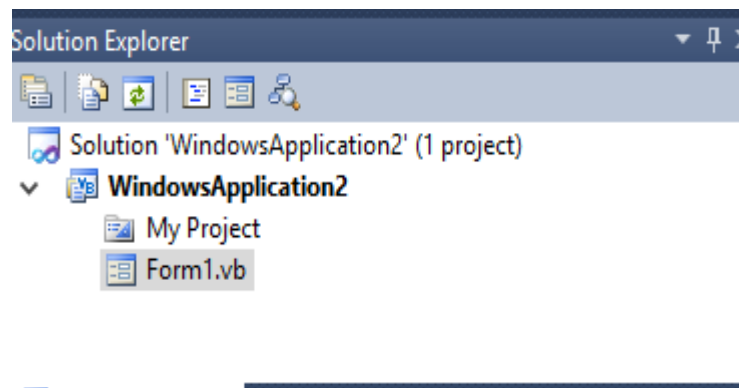
Toolbox merupakan kotak piranti berisi yang semua control yang dapat digunakan untuk merancang interface aplikasi yang diinginkan. Secara default *toolbox* yang terdapat dalam Visual Basic adalah sebagai berikut. Adapun tampilan *toolbox* dapat dilihat pada gambar II.6.



Gambar II.6 Toolbox

f. *Project Explorer*

Project Explorer adalah semua jendela berisi semua file dalam semua aplikasi Visual Basic. Jendela ini berisikan project, form, modul, class, dan beberapa file lainnya. Adapun tampilan *project explorer* dapat dilihat pada gambar II.7.



Gambar II.7 Project Explorer

g. Kode Windows

Kode windows adalah jendela yang berisi kode-kode program yang merupakan intruksi-intruksi program untuk aplikasi Visual Basic. Adapun tampilan kode windows dapat dilihat pada gambar II.8.



Gambar II.8 Kode Windows

II.2.3. SQL Server 2008

SQL SERVER adalah sistem manajemen *Relational Database Management System* (RDBMS) yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak.

- a. **CLIENT** adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya.
- b. **SERVER** adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.
- c. **RDBMS** singkatan *Relational Database Management System*. **RDBMS** adalah dasar untuk SQL, dan untuk semua sistem *database* modern seperti MS SQL Server, *IBM DB2, Oracle, MySQL, dan *Microsoft Access*. Data dalam RDBMS disimpan dalam objek *database* yang disebut tabel.

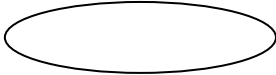
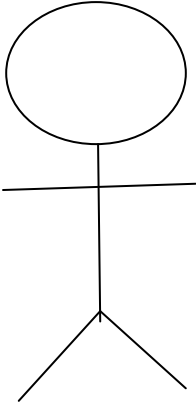


II.2.4. Unified Modeling Language (UML)


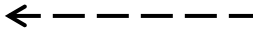
Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

II.2.4.1. Use Case Diagram

Use case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Use Case digunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. (Hamim Tohari, 2014: 47)

Tabel II.2. Simbol-Simbol *Use Case*

Gambar	Keterangan
	<p><i>Uses Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan akhir, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetap tidak memiliki kontrol terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila <i>actor</i> berinteraksi secara pasif dengan sistem.</p>



	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.


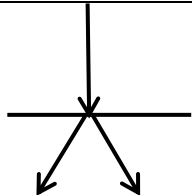
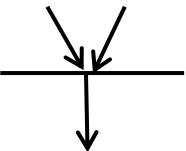
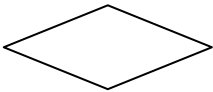

(Sumber : Ade Hendini, Vol 4. 2016. 108-109)

II.2.4.2. Activity Diagram

Activity Diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchat* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity diagram* pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity Diagram* juga bermanfaat untuk menggambarkan *parallel behaviour* atau menggambarkan interaksi antara beberapa *use case*. (Hamim Tohari, 2014: 114)

Tabel II.3. Simbol-Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.

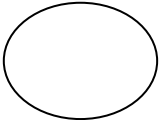
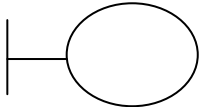
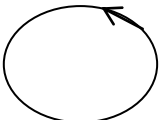

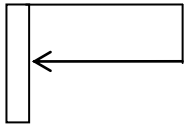
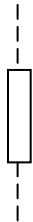

	<p><i>Activities</i>, menggambarkan suatu proses/kegiatan bisnis.</p>
	<p><i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.</p>
	<p><i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision points</i>, menggambarkan pilihan untuk pengambilan keputusan <i>true</i>, <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p>

(Sumber : Ade Hendini, Vol 4.2016: 109-110)

II.2.4.3. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk mewujudkan rangkaian pesan yang dikirim antara objek yang terjadi pada titik tertentu dalam eksekusi sistem. Dalam UML, objek, pada *diagram sequence* digambarkan dalam segi empat, yang berisi nama dari objek yang digaris bawah. (Hamim Tohari, 2014: 101)

Tabel II.4. Simbol-simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan form cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan objek.
	<i>Message</i> , simbol pengiriman antar pesan.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, Vol 4.2016: 109-110)

II.2.4.4. *Class Diagram*

Kelas (*Class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). (Hamim Tohari, 2014: 83)

Dalam pemodelan statis dari sebuah sistem, diagram kelas biasanya digunakan untuk memodelkan salah satu dari tiga hal berikut :

1. Perbendaharaan dari sistem.
2. Kolaborasi.
3. Skema basi data logical.

Kelas memiliki tiga area produk :

1. Nama (dan stereotype).
2. Atribut.
3. Metode atau operasi.

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dari anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja. : (Hamim Tohari, 2014: 84)