BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Instansi

2.1.1 Sejarah Singkat dan Perkembangan Umum Instansi

Penyelenggaraan pendidikan kader pemerintahan di lingkungan Departemen Dalam Negeri yang terbentuk melalui proses perjalanan sejarah yang panjang. Perintisiannya dimulai sejak zaman pemerintahan Hindia Belanda pada tahun 1920, dengan terbentuknya sekolah pendidikan Pamong Praja yang bernama Opleiding School Voor Inlandshe Ambtenaren (OSVIA) dan Middlebare Opleiding School Voor Inlandsche Ambtenaren (MOSVIA). Para lulusannya sangat dibutuhkan dan dimanfaatkan untuk memperkuat penyelenggaraan pemerintahan Hindia Belanda. Dimasa kedudukan pemerintah Hindia Belanda, penyelenggaraan pemerintahan Hindia Belanda dibedakan atas pemerintahan yang langsung dipimpin oleh kaum atau golongan pribumi yaitu Binnenlands Bestuur Corps (BBC) dan pemerintahan yang tidak langsung dipimpin oleh kaum atau golongan dari keturunan Inlands Bestuur Corps (IBC).

Pada masa awal kemerdekaan RI, sejalan dengan penataan system pemerintahan yang diamanatkan oleh Undang Undang Dasar 1945, kebutuhan akan tenaga kader pamong praja untuk melaksnakan tugas-tugas pemerintahan baik pada pemerintah pusat maupun daerah semakin meningkat sejalan dengan tuntutan perkembangan penyelenggaraan pemerintahannya. Untuk memenuhi kebutuhan akan kekurangan tenaga kader pamong praja, maka pada tahun 1948 dibentuklah lembaga pendidikan dalam lingkungan Kementrian Dalam Negeri yaitu Sekolah Menengah Tinggi (SMT) Pangreh Praja yang kemudian berganti nama menjadi Sekolah Menengah Pegawai Pemerintahan Administrasi Atas (SMPAA) di Jakarta dan Makassar.

Pada perkembangan selanjutnya, lulusan APDN dinilai masih perlu ditingkatkan dalam rangka upaya lebih menjamin terbentuknya kader-kader pemerintahan yang ”qualified leadership and manager administrative” terutama

dalam menyelenggarakan tugas-tugas urusan pemerintahan umum. Kebutuhan ini mendorong pemerintah untuk menyelenggarakan pendidikan aparatur di lingkungan Departemen Dalam Negeri setingkat Sarjana, maka dibentuklah Institut Ilmu Pemerintahan (IIP) yang berkedudukan di Kota Malang Jawa Timur berdasarkan Keputusan Bersama Menteri Dalam Negeri dan Menteri Pendidikan dan Kebudayaan No. 8 Tahun 1967, selanjutnya dikukuhkan dengan Keputusan Presiden Nomor 119 Tahun 1967. Peresmian berdirinya IIP di Malang ditandai dengan peresmian oleh Presiden Soekarno pada tanggal 25 Mei 1967.

Pada tahun 1972 Institut Ilmu Pemerintahan (IIP) yang berkedudukan di Malang Jawa Timur dipindahkan ke Jakarta melalui Keputusan Menteri Dalam Negeri Nomor 94 Tahun 1972. Pada tanggal 9 Maret 1972, kampus IIP yang terletak di Jakarta di resmikan oleh Presiden Soeharto yang dinyatakan : "Dengan peresmian kampus Institut Ilmu Pemerintahan, mudah-mudahan akan merupakan kawah candradimukanya Departemen Dalam Negeri untuk menggembleng kader-kader pemerintahan yang tangguh bagi Negara Kesatuan Republik Indonesia".

Seiring dengan pembentukan IIP yang merupakan peningkatan dari APDN Nasional di Malang, maka untuk penyelenggaraan pendidikan kader pada tingkat akademi, Kementrian Dalam Negeri secara bertahap sampai dengan dekade tahun 1970-an membentuk APDN di 20 Provinsi selain yang berkedudukan di Malang, juga di Banda Aceh, Medan, Bukittinggi, Pekanbaru, Jambi, Palembang, Lampung, Bandung, Semarang, Pontianak, Palangkaraya, Banjarmasin, Samarinda, Mataram, Kupang, Makassar, Menado, Ambon dan Jayapura.

Pada tahun 1988, dengan pertimbangan untuk menjamin terbentuknya wawasan nasional dan pengendalian kualitas pendidikan Menteri Dalam Negeri Rudini melalui Keputusan No. 38 Tahun 1988 Tentang Pembentukan Akademi Pemerintahan Dalam Negeri Nasional. APDN Nasional kedua dengan program DIII berkedudukan di Jatinangor, Sumedang Jawa Barat yang peresmiannya dilakukan oleh Mendagri tanggal 18 Agustus 1990. APDN Nasional ditingkatkan statusnya berdasarkan Kepres No. 42 Tahun 1992 tentang Sekolah Tinggi Pemerintahan Dalam Negeri, maka status APDN menjadi STPN dengan program studi D III yang diresmikan oleh Presiden RI pada tanggal 18

Agustus 1992. Sejak tahun 1995, bertititk tolak dari keinginan dan kebutuhan untuk lebih mendorong perkembangan karier sejalan dengan peningkatan eselonering jabatan dalam sistem kepegawaian Republik Indonesia, maka program studi ditingkatkan menjadi program D IV. Keberadaan STPDN dengan pendidikan profesi (program DIV) dan IIP yang menyelenggarakan pendidikan akademik program sarjana (Strata I), menjadikan Departemen Dalam Negeri memiliki dua (2) Pendidikan Pinggi Kedinasan dengan lulusan yang sama dengan golongan III/a.

Kebijakan Nasional mengenai pendidikan tinggi sejak tahun 1999 antara lain yang mengatur bahwa suatu Departemen tidak boleh memiliki dua atau lebih perguruan tinggi dalam menyelenggarakan keilmuan yang sama, maka mendorong Departemen Dalam Negeri untuk mengintegrasikan STPDN ke dalam IIP . Usaha pengintegrasiaan STPDN kedalam IIP secara intensif dan terprogram sejak tahun 2003 sejalan dengan dikeluarkannya UU No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional. Pengintegrasian terwujud dengan ditetapkannya Keputusan Presiden Nomor 87 Tahun 2004 tentang Penggabungan STPDN ke dalam IIP dan sekaligus merubah nama IIP menjadi Institut Ilmu Pemerintahan (IPDN). Tujuan penggabungan STPDN ke dalam IIP tersebu, selain untuk memenuhi kebijakan pendidikan nasional juga untuk meningkatkan efektivitas penyelenggaraan pendidikan kader pamong praja di lingkungan Departemen Dalam Negeri. Kemudian Kepres No. 87 Tahun 2004 ditindak lanjuti dengan Keputusan Mendagri No.892.22-421 tahun 2005 tentang Pelaksanaan Penggabungan dan Operasional Institut Pemerintahan Dalam Negeri, disertai dengan Peraturan Menteri Dalam Negeri No. 29 Tahun 2005 Tentang Organisasi dan Tata Kerja IPDN dan Peraturan Menteri Dalam Negeri 43 Tahun 2005 Tentang Statuta IPDN serta peraturan pelaksanaan lainnya.

Pada Tahun 2007 ini, IPDN dikembalikan lagi menjadi Institut Ilmu Pemerintahan (IIP) yang memiliki beberapa Program D-IV di beberapa Regional, diantaranya Regional Jawa-Bali, Sumatera, Sulawesi, Kalimantan, dan Papua.

2.1.2 Sekilas Mengenai Bagian Pengasuhan IPDN Jatinangor

Bagian Pengasuhan IPDN berdasarkan Peraturan Menteri Dalam Negeri Nomor 60 Tahun 2010 tentang perubahan Atas Peraturan Menteri Dalam Negeri Nomor 39 tahun 2009 Tentang organisasi dan tata kerja institut pemerintahan dalam negeri disebutkan bahwa Bagian Pengasuhan mempunyai tugas melaksanakan urusan pengasuhan dan menyelenggarakan urusan bimbingan dan pengawasan seta urusan pembinaan disiplin praja. Bagian pengasuhan mempunyai dua subbagian yaitu subbagian pengawasan dan subbagian pembinaan disiplin.

Subbagian pengawasan merupakan subbagian dari bagian pengasuhan yang mempunyai tugas melakukan urusan bimbingan dan pengawasan praja. Subbagian ini mempunyai fungsi :

1. Pelaksanaan bimbingan dan pengawasan terhadap Praja
2. Pemantauan dan evaluasi kegiatan Praja

Adapun tugas Subbagian Pengawasan meliputi:

1. Mengkoordinasikan pelaksanaan system, dan materi pengasuhan dalam rangka operasional pengasuhan;
2. Mengkoordiansikan pelaksanaan jadwal kegiatan pengasuhan;
3. Mengkoordinasikan pelaksanaan kegiatan bimbingan dan pengawasan praja;
4. Melaksanakan dan mengawasi pelaksanaan piket pengasuhan;
5. Fasilitasi semua kegiatan yang berhubungan dengan kegiatan Jarlatsuh dan mengkoordiansikan pelaksanaan dan pendampingan kegiatan minat dan bakat;
6. Mengkoordinasikan pelaksanaan kegiatan pemantauan dan monitoring terhadap Praja;
7. Mengkoordianasikan pelaksanaan evaluasi terhadap kinerja pamong pengasuh;
8. Menyelia pelaksanaan tugas pamong pengasuh
9. Bertanggung jawab terhadap masalah-masalah pembinaan disiplin.

Subbagian Pembinaan Disiplin merupakan subbagian dari Bagian Pengasuhan yang mempunyai tugas melakukan pembinaan disiplin praja. Subbagian ini memfunyai fungsi pelaksanaan pembinaan disiplin praja.

Adapun tugas Subbagian Pembinaan Disiplin meliputi:

1. Mengkoordinasikan pelaksanaan program pembinaan disiplin praja;
2. Mengkoordinasikan penyiapan dan pembuatan pedoman dan materi pengasuhan;
3. Menyiapkan bahan pembuatan jadwal kegiatan pengasuhan;
4. Mengkoordinasikan pengelolaan informasi penerapan metode dan materi pengasuhan;
5. Menyiapkan bahan pelaksanaan evaluasi terhadap rencana program kerja bagian pengasuhan;
6. Menyiapkan bahan pembuatan instrument evaluasi kinerja pengasuhan dan evaluasi kepribadian praja;
7. Menyiapkan bahan laporan pelaksanaan pengasuhan;
8. Melaksanakan tugas lain yang diberikan oleh Kepala Bagian Pengasuhan;
9. Bertanggung jawab terhadap masalah-masalah pembinaan disiplin.

2.1.3 Lambang Instansi

Berikut merupakan lambang dari Institut Pemerintahan Dalam Negeri (IPDN)



Gambar 2. 1 Lambang IPDN

1. Bintang warna kuning melambangkan Pancasila
2. Kapas warna putih melambangkan keadilan

3. Daun kapas warna hijau melambangkan kesejukan dan ketentraman
4. Padi warna kuning melambangkan kemakmuran.
5. Kombinasi bunga kapas dan daunnya berjumlah 17 melambangkan tanggal Proklamasi 17 Agustus 1945 berdirinya Negara Kesatuan Republik Indonesia.
6. Padi berjumlah 45 bermakna tahun kemerdekaan Republik Indonesia 1945.
7. Roda kemudi melambangkan pemerintahan.
8. Delapan jari roda kemudi melambangkan bulan lahirnya Proklamasi, dan melambangkan 8 penjuru angin yang dimaknai sebagai ke wilayahan, pemerintahan daerah dan Bhinneka Tunggal Ika.
9. Lambang yang berbentuk kelopak bunga lotus (teratai), bermakna kearifan.
10. Buku melambangkan sumber pengetahuan.
11. Warna biru laut yang mendasari lambang dimaknai sebagai tanggungjawab, ketangguhan, ketenangan dan inovasi yang tinggi.
12. Angka 2004 melambangkan tahun berdirinya IPDN.
13. Among, melaksanakan fungsi pamong yang berarti mengasuh dan mengemong menurut sistem among : Ing Ngarso Sung Tulodho, Ing Madyo Mangun Karso, Tutwuri Handayani.
14. Praja artinya Peserta Didik (termasuk mahasiswa) IPDN.
15. Dharma artinya melaksanakan kewajiban, peraturan, kebenaran.
16. Kata-kata : Among Praja Dharma Nagari secara keseluruhan berarti IPDN mengemong Praja supaya setia pada kewajiban untuk mengabdi kepada Bangsa dan Negara.

2.1.4 Visi dan Misi

2.1.4.1 Visi

Visi yang ditetapkan Institut Pemerintahan Dalam Negeri dalam mewujudkan cita-cita tersebut dalam waktu sepuluh tahun ke depan adalah ***"Menjadi Lembaga Pendidikan Tinggi Kepamongprajaan yang***

terpercaya dalam mengemban tugas pengembangan ilmu, pembentukan perilaku kepamongan dan penyedia kader pemerintahan yang terampil".

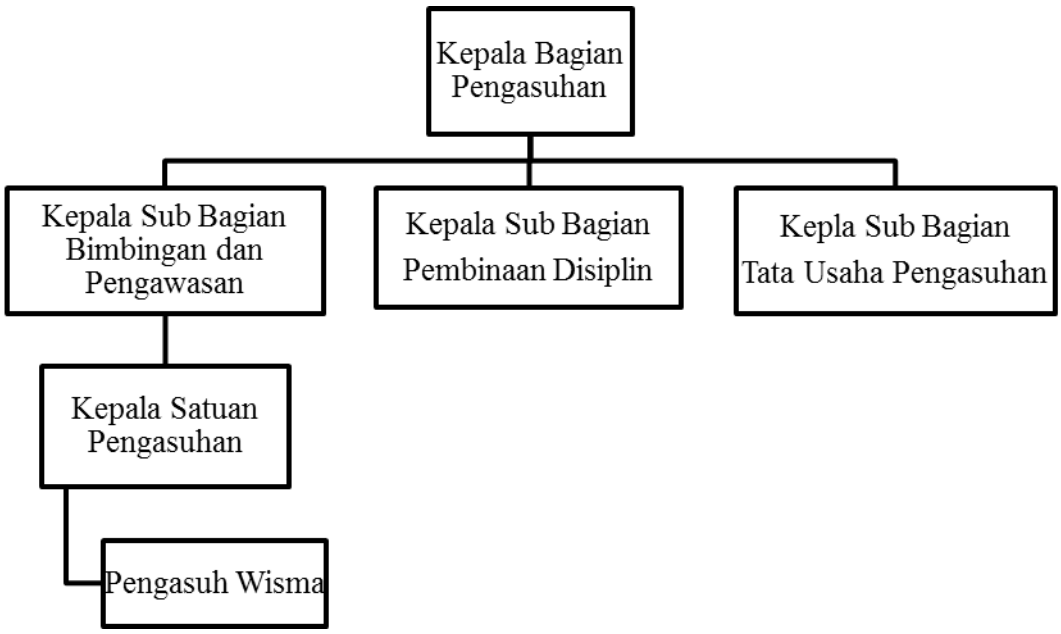
2.1.4.2 Misi

Untuk mencapai visi tersebut, maka misi Institut Pemerintahan Dalam Negeri ditetapkan sebagai berikut:

1. Mensinergikan kekuatan sivitas akademika Institut Pemerintahan Dalam Negeri.
2. Mengembangkan kurikulum berbasis pengajaran, pelatihan, dan pengasuhan (Jar-Lat-Suh).
3. Membangun jaringan kerjasama dengan berbagai kalangan yang mampu mendukung pengembangan kurikulum dan implementasinya.
4. Melaksanakan Tridharma Perguruan Tinggi (Pendidikan, Penelitian, dan Pengabdian Masyarakat).
5. Meningkatkan kualitas sumber daya manusia Institut Pemerintahan Dalam Negeri.
6. Memberdayakan praja sebagai subyek pendidikan dan aset nasional.

2.1.5 Struktur Organisasi

Struktur Organsisasi merupakan kerangka yang mewujudkan pola tetap dari hubungan-hubungan diantara bidang-bidang kerja, maupun orang-orang yang menunjukkan kedudukan dan peranan masing-masing dalam kebulatan kerja sama. Secara umum struktur organisasi Bagian Pengasuhan Institut Pemerintahan Dalam Negeri (IPDN) dapat dilihat pada gambar 2.2.



Gambar 2. 2 Struktur Organisasi IPDN

2.1.6 Uraian Tugas

2.1.6.1 Kepala Bagian Pengasuhan

Tugas dan tanggung jawab kepala bagian pengasuhan adalah:

a. Mengkoordinasikan kegiatan administrasi dan tata usaha pengasuhan.
b. Mengkoordinasikan rencana, program dan keuangan pengasuhan. Menyelia pelaksanaan program kerja pengasuhan.
c. Mengkoordinasikan penyusunan dan pelaksanaan peraturan tata tertib praja dan penegakan disiplin praja.
d. Mengkoordinasikan kegiatan bimbingan dan pengawasan praja.
e. Mengkoordinasikan pelaksanaan kegiatan bimbingan, penyuluhan dan konseling praja.
f. Mengkoordinasikan kegiatan pemantauan dan pengawasan terhadap praja.
g. Mengkoordinasikan kegiatan evaluasi kinerja pengasuh dan kepribadian praja.
h. Memberikan saran dan pertimbangan terhadap hasil bimbingan dan konseling serta hasil penjatuhan hukuman disiplin.
i. Melaksanakan tugas lain yang diberikan oleh pimpinan.

2.1.6.2 Kasubbag Bimbingan dan Pengawasan

Tugas dan tanggung jawab kasubbag bimbingan dan pengawasan adalah:

a. Mengkoordinasikan pelaksanaan sistem, pedoman dan materi pengasuhan dalam rangka operasional pengasuhan.
b. Mengkoordinasikan pelaksanaan jadwal kegiatan pengasuhan.
c. Mengkoordinasikan pelaksanaan kegiatan bimbingan dan Pengawasan Praja.
d. Melaksanakan dan mengawasi pelaksanaan piket pengasuhan.
e. Fasilitasi semua kegiatan yang berhubungan dengan kegiatan Jarlatsuh dan mengkordinasikan pelaksanaan dan pendampingan kegiatan minat dan bakat.
f. Mengkoordinasikan pelaksanaan kegiatan pemantauan dan monitoring terhadap Praja.
g. Mengkoordinasikan pelaksanaan evaluasi terhadap kinerja Pamong Keprajaan dan kegiatan Praja.
h. Menyelia pelaksanaan tugas Pamong Keprajaan Kabupaten, Kecamatan dan Kelurahan.
i. Melaksanakan tugas lain yang diberikan oleh Pamong Keprajaan Propinsi.

2.1.6.3 Kasubbag Pembinaan Disiplin

Tugas dan tanggung jawab kasubbag pembinaan disiplin adalah:

a. Mengkoordinasikan penyiapan bahan peraturan tata tertib dan penegakan disiplin.
b. Mengkoordinasikan evaluasi pelaksanaan tata tertib dan penegakan disiplin praja.
c. Mengkoordinasikan penghimpunan data dan informasi pelaksanaan tata tertib dan penegakan disiplin praja.
d. Menyiapkan dan memberikan saran dan pertimbangan penjatuhan hukuman disiplin.

e. Mengkoordinasikan pelaksanaan administrasi tata usaha unit penegakan disiplin dan penyidikan praja.
f. Melaporkan pelaksanaan pemeriksaan dalam rangka penegakan disiplin dan penyidikan praja.

2.1.6.4 Kasubbag Tata Usaha Pengasuhan

Tugas dan tanggung jawab kasubbag Tata Usaha Pengasuhan adalah:

a. Merencanakan Program Kerja Tahunan Bagian Pengasuhan berdasarkan petunjuk atasan dan ketentuan peraturan perundangan sebagai pedoman dalam pelaksanaan tugas.
b. Membagi tugas kepada bawahan dengan cara lisan atau tertulis agar dapat diproses lebih lanjut.
c. Memberi petunjuk kepada bawahan secara lisan atau tertulis agar bawahan mengerti dan memahami pekerjaannya.
d. Memeriksa pekerjaan bawahan berdasarkan hasil kerja untuk mengetahui adanya kesalahan atau kekeliruan serta upaya penyempurnaannya.
e. Membuat konsep pedoman dan petunjuk teknis pelaksanaan kegiatan ketatausahaan bagian pengasuhan berdasarkan petunjuk atasan sebagai bahan penyusunan pedoman dan petunjuk teknis kegiatan administrasi kepengasuhan IPDN.
f. Melaporkan pelaksanaan tugas kepada atasan berdasarkan hasil kerja sebagai bahan evaluasi bagi atasan

2.1.6.5 Kepala Satuan Pengasuhan

Tugas dan tanggung jawab kepala satuan pengasuhan adalah:

a. Melaksanakan sistem, pedoman dan materi pengasuhan dalam rangka operasional pengasuhan.
b. Melaksanakan jadwal kegiatan pengasuhan.
c. Melaksanakan kegiatan bimbingan dan Pengawasan Praja.
d. Melaksanakan dan mengawasi pelaksanaan piket pengasuhan.

e. Memfasilitasi semua kegiatan yang berhubungan dengan kegiatan Jarlatsuh dan menyiapkan serta mendampingi kegiatan minat dan bakat.
f. Melaksanakan kegiatan pemantauan dan monitoring terhadap Praja.
g. Melaksanakan evaluasi terhadap kinerja Pengasuh Wisma dan kegiatan Praja.
h. Menyelia pelaksanaan tugas Pengasuh Wisma.

2.1.6.6 Pengasuh Wisma

Tugas dan tanggung jawab pengasuh wisma adalah:

a. Menyiapkan dan mengecek pengisian buku jaga wisma, jaga serambi dan buku kendali berobat Praja di Wisma, mencatat serta menyimpan semua data yang berkaitan dengan data medical record Praja yang menjadi tanggung jawabnya.
b. Memberikan pelayanan administratif kepada Praja yang menjadi tanggung jawabnya.
c. Melaksanakan sistem, pedoman dan materi pengasuhan dalam rangka operasional pengasuhan.
d. Melaksanakan kegiatan bimbingan dan Pengawasan Praja.
e. Memfasilitasi semua kegiatan yang berhubungan dengan kegiatan Jarlatsuh dan menyiapkan serta mendampingi kegaitan minat dan bakat.
f. Menyiapkan, merencanakan dan mengawasi pelaksanaan kebersihan wisma dan lingkungan wisma.
g. Melaksanakan kegiatan pemantauan dan monitoring terhadap Praja serta evaluasi kepribadian.

2.2 Landasan Teori

2.2.1 Konsep Dasar Sistem Informasi

Dalam mendefinisikan sistem terdapat dua kelompok pendekatan sistem, yaitu sistem yang lebih menekankan pada prosedur dan elemennya. Prosedur didefinisikan sebagai suatu urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakan, kapan

dikerjakan dan bagaimana mengerjakannya. Berdasarkan pendekatan elemen adalah sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud [2].

Informasi merupakan data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang. Fungsi dari informasi adalah untuk mengurangi ketidakpastian didalam proses pengambilan keputusan tentang suatu keadaan. Informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Informasi digunakan tidak hanya oleh satu pihak didalam organisasi. Nilai sebuah informasi ditentukan dari dua hal yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkan informasi tersebut.

Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer-Based Information Systems* atau CBIS). Dalam prakteknya, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang paling penting.

Ada beragam definisi informasi, sebagaimana tercantum pada tabel 2.1. Dari berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan [3].

Tabel 2. 1 Pengertian Sistem Informasi

Sumber	Definisi
Alter (1992)	Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
Bodnar dan Hopwood (1993)	Sistem informasi adalah kumpulan perangkat keras dan oerangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
Gelinas, Oram, dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada pemakai.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
Turban, McLean, dan Weterbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
Wikinson (1992)	Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sararan perusahaan.

2.2.2 *Monitoring* dan Evaluasi (Monev)

Monev adalah kegiatan monitoring dan evaluasi yang ditujukan pada suatu program yang sedang atau sudah berlangsung. Monitoring atau pemantauan adalah suatu seni mengumpulkan informasi dengan usaha yang minimum untuk membuat suatu keputusan di waktu yang tepat. Informasi yang sudah dikumpulkan dapat digunakan untuk kemudian dilakukan analisis, diskusi, maupun pembuatan laporan. Monitoring memungkinkan kita untuk memantau proses secara rutin dalam jangka waktu tertentu, menentukan apakah suatu proses tersebut sudah berjalan dengan baik atau sudah sesuai dengan tujuan awal [6].

Evaluasi adalah proses untuk mengidentifikasi masalah, mengumpulkan data dan menganalisis data, menyimpulkan hasil yang telah dicapai, menginterpretasikan hasil menjadi rumusan kebijakan, dan menyajikan informasi (rekomendasi) untuk pembuatan keputusan berdasarkan pada aspek kebenaran hasil evaluasi. Perbedaan antara monitoring dan evaluasi adalah monitoring dilakukan pada saat program masih berjalan sedangkan evaluasi dapat dilakukan baik sewaktu program itu masih berjalan ataupun program itu sudah selesai. Atau dapat juga bila dilihat dari pelakunya, monitoring biasanya dilakukan oleh pihak internal sedangkan evaluasi dilakukan oleh pihak internal maupun eksternal. Evaluasi dilaksanakan untuk memperoleh fakta atau kebenaran dari suatu program beserta dampaknya, sedangkan monitoring hanya melihat keterlaksanaan program, faktor pendukung, penghambatnya. Proses *monitoring* dan evaluasi kedisiplinan praja berdasarkan Peraturan Rektor dan dashboard.

2.2.2.1 Parameter Sistem *Monitoring* dan Evaluasi

Parameter yang digunakan sebagai ukuran pada sistem *monitoring* dan evaluasi kedisiplinan praja untuk memonitoring adalah laporan kehadiran praja dan laporan kebersihan wisma. Parameter tersebut berdasarkan:

1. Peraturan Menteri Dalam Negeri Nomor 41 Tahun 2009 Tentang Peraturan Disiplin Praja Institut Pemerintahan Dalam Negeri pada Lampiran D.

2. Peraturan Menteri Dalam Negeri Nomor 46 Tahun 2009 Tentang Kebersihan Keindahan dan Kerapihan Asrama/Wisma dan Lingkungan Sekitar Asrama/Wisma pada Lampiran D.
3. Peraturan Rektor Pemerintahan Dalam Negeri Nomor 3 Tahun 2010 Tentang Kewajiban, Hak, Larangan, dan Jenis Hukuman Atas Pelanggaran Disiplin Praja Institut Pemerintahan Dalam Negeri pada Lampiran D.
4. Peraturan Rektor Pemerintahan Dalam Negeri Nomor 5 Tahun 2010 Tentang Pedoman Penilaian Pengasuhan Praja Institut Pemerintahan Dalam Negeri pada Lampiran D.

2.2.2.2 Dashboard

Dashboard adalah alat bantu kontrol untuk mengidentifikasi jalannya proses bisnis secara mudah dengan menggunakan indikator-indikator tertentu seperti misalnya penggunaan indikator warna-warna khusus yaitu merah, hijau atau kuning yang melambangkan status atau kondisi dari suatu bisnis atau proyek, tanda peringatan, ringkasan-ringkasan, grafik-grafik seperti bar-chart, pie chart, dimana biasanya dibuat satu set dalam suatu portal, dan dapat diatur [7].

2.2.3 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan [4]. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DMBS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

2.2.3.1 DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [4]. Suatu

sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

2.2.3.2 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS [4]. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

2.2.4 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [4]. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan

kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.4.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya) [4]. Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses

yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

   Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat *runtime*

yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat *runtime* (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

   Metode atau operasi dapat berasal dari *event*, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya *Read, Write, Move, Copy,* dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.5 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 di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu,

misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

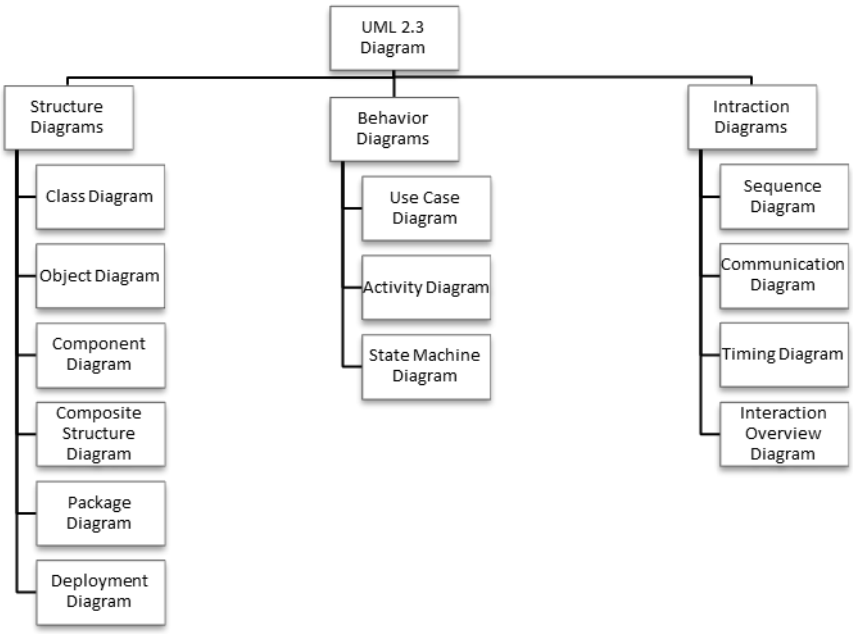
Pada perkembangan teknil pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan voisual untu menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan digaram 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 pada metodologi berorientasi objek.

Seperti yang kita ketahui di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan, lingkunan dan konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang perbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin digambarkan. Secara analogi jika dengan bahasa yang digunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalahhal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi di dalam bidang sistem informasi merupakan hal yang sangat wajar.

2.2.6 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.3 [4].



Gambar 2. 3 Diagram UML[4]

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

2.2.7 *Use Case Diagram*

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam 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 pendefinisian apa yang disebut aktor dan *Use Case.*

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

2.2.8 *Activity Diagram*

Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/*User interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

2.2.9 *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

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

Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.2.10 *Class Diagram*

Diagram kelas atau *Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas Main
2. Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
3. Kelas yang menangani tampilan sistem
4. Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
5. Kelas yang diambil dari pendefinisian *usecase*
6. Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *Use Case*.
7. Kelas yang diambil dari pendefinisian data
8. Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan sat sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan coupling. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi

antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *choesion* yang kuat dan kadar *coupling* yang lemah.

Dalam *Class* diagram terdapat beberapa relasi (hubungan antar *Class*) yaitu;

1. Generalization dan Inheritence

   Diperlukan untuk memperlihatkan hubungan pewarisan (inheritance) antar unsur dalam diagram kelas. Pewarisan memungkinkan suatu kelas mewarisi semua atribut, operasi, relasi, dari kelas yang berada dalam hirarki pewarisannya.
2. Associations

   Hubungan statis antar *Class*. Umumnya menggambarkan *Class* yang memiliki atribut berupa *Class* lain, atau *Class* yang harus mengetahui ekstensi *Class* lain. Dalam notasi UML kita mengenal asosiasi 2 arah (bidirectional) dan 1 arah (undirectional).
3. Aggregation

   Hubungan antar-*Class* dimana *Class* yang satu (part *Class*) adalah bagian dari *Class* lainnya (whole *Class*).
4. Composition

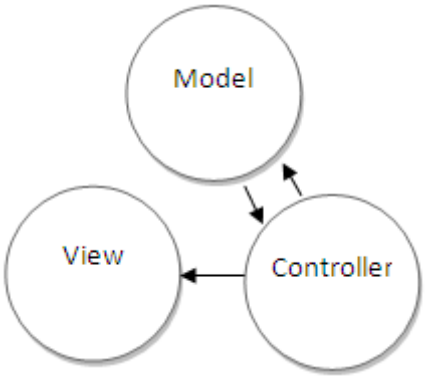
   Aggregation dengan ikatan yang lebih kuat. Di dalam composite aggregation, siklus hidup part *Class* sangat bergantung pada whole *Class* sehingga bila objek instance dari whole *Class* dihapus maka objek instance dari part calss juga akan terhapus.
5. Depedency

   Hubungan antar-*Class* dimana sebuah *Class* memiliki ketergantungan pada *Class* lainnya tetapi tidak sebaliknya.
6. Realization

   Hubungan antar-*Class* dimana sebuah *Class* memiliki keharusan untuk mengikuti aturan yang ditetapkan *Class* lainnya. biasanya realization digunakan untk menspesifikasikan hubungan antara sebuah interface dengan *Class* yang mengimplementasikan interface tersebut.

2.2.11 *Model View Controller*

Teknik pemrograman MVC (Model View Controller) merupakan teknik pemrograman yang populer saat ini, yang mengharapkan pemrogram secara disiplin untuk membagi program menjadi tiga bagian; model, view, dan controller, seperti gambar berikut[5] :



Gambar 2. 4 Arsitektur MVC[5]

1. Model

   Objek model adalah bagian dari aplikasi yang mengimplementasi logika untuk domain data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari database atau menyimpan data ke dalam database.

2. View

   View adalah komponen yang menampilkan antarmuka untuk pengguna (*User* interface, UI) aplikasi. Antarmuka ini dibuat berdasarkan data dari model.

3. Controller

   Controller merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan model, dan memilih view mana yang digunakan untuk merender data.

Dalam aplikasi MVC, view hanya menampilkan informasi, controller menangani dan menanggapi masukan dan interaksi dengan pengguna. Dengan pola MVC, kita membuat aplikasi dengan memisahkan berbagai aspek dari aplikasi (logika input, bisnis, dan antarmuka), ada bagian penampil data (View), penyedia atau pemroses data (model), dan pengendali program (controller).

Dengan pemisahan ini, maka pembuatan satu program dapat dibagi-bagi kepada beberapa orang. Setiap bagian atau komponen yang dibuat secara terpisah diharapkan dapat digunakan tidak hanya oleh satu program saja, tetapi oleh banyak program, terutama bagian model dan view. Teknik pemrograman MVC ini diimplementasi tidak hanya dalam PHP, tetapi suatu teknik pemrograman yang sudah dikenal dan digunakan hampir pada semua bahasa pemrograman yang ada.

2.3 Perangkat Lunak Pendukung

2.3.1 PHP

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun apliaksi *web.* Ketika dipanggil dari *web browser,* program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server,* PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat *User* memilih perintah "*View Source*" pada *web browser* yang mereka gunakan. Selain menggunakan PHP, aplikasi *web* juga dapat dibangun dengan Java (JSP- *JavaServer Pages* dan *Servlet*), Perl, maupun ASP (*Active Server Pages*)[6].

2.3.2 MYSQL

Database MySQL merupakan sistem manajemen basisdata SQL yang sangat terkenal dan bersifat open source. MySQL dibangun, didistribusikan, dan didukung oleh MySQL AB. MySQL AB merupakan perusahaan komersial yang dibiayai oleh pengembang MySQL. MySQL mempunyai dua maca lisensi yaitu lisensi yang bersifat open source dengan menggunakan GNU (*General Public License*) dan lisensi kedua berupa *Standard Commercial License* yang dapat dibeli dari MySQL AB.

Dalam konteks bahasa SQL, informasi disimpan dalam tabel-tabel yang secara logis merupakan struktur dua dimensi yang tersimpan atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column).* Baris pada

tabel disebut *insance* dari data sedangkan kolom sering disebut sebagai *atributes* atau *field.*

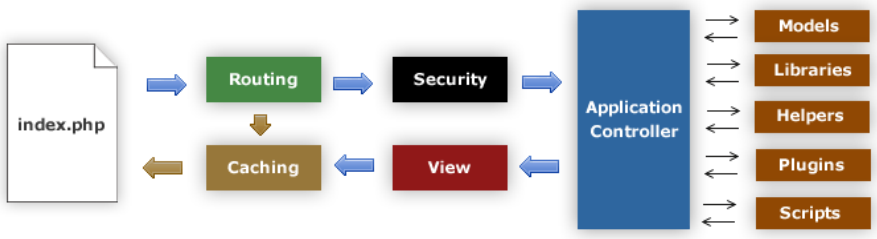
2.3.3 *CodeIgniter*

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis[5]. Pemrogram tidak perlu membuat program dari awal (*from scracth*), karena CI menyediakan sekumpulan librari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengkakses librarynya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan.

Framefork CI merupakan framework yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan mudah. Pendekatan dari CI sangatlah mudah, dari membuat sekadar tulisan sampai dengan yang kompleks dapat didekati dengan mudah. Tidak seperti framework yang lain, untuk mendapatkan tulisan Hello World di browser saja, kita harus menggunakan beberapa tahap. CI cukupd dengan satu file dan satu prosedur atau method.

2.3.4 Pemrograman MVC dengan CodeIgniter

CI adalah satu framework pemrograman PHP yang mengimplementasikan teknik pemrograman MVC. Berikut adalah arsitektur bagaimana CI mengimplementasikan MVC[5].



Gambar 2. 5 Arsitektur CodeIgniter[5]

Index.php berlaku sebagai controller di depan, yang menginisialisasi sumber daya yang dibutuhkan untuk menjalankan CodeIgniter

1. Router akan memeriksa request HTTP untuk menentukan apa yang harus dilakukan dengan request tersebut
2. Jika ada file cache, maka file cache langsung dikirimkan kepada browser, dengan memotong eksekusi sistem normal
3. Keamanan. Sebelum suatu aplikasi controller dipanggil, request HTTP dan data yang dikirimkan oleh pengguna disaring (filter) terlebih dahulu untuk keamanan
4. Controller akan memuat model, libari inti, helper, dan sumberdaya lainnya yang diperlukan untuk memroses request tersebut
5. View terakhir direnderkemudian dikirimkan kepada browser untuk dilihat. Jika caching didefinisikan, maka view akan dicache terlebih dahulu, setelah itu baru dikirimkan kepada browser.

Codeigniter (CI) bisa digunakan untuk belajar melakukan pemrograman dengan teknik pemrograman yang menggunakan konsep MVC (Model View Controller). Ci mengijinkan pemrogram untuk bekerja penuh di dalam controller, sambil memahami konsep MVC dengan benar.