BAB II
TINJAUAN PUSTAKA

2.1 Profil Institusi

Profil Institusi adalah untuk mengetahui keadaan institusi diantaranya adalah sejarah berdirinya, struktur organisasi serta visi dan misi.

2.1.1 Sejarah Singkat Perusahaan

Badan Kepegawaian Negara (BKN) sebagai lembaga pembina kepegawaian memiliki keterlibatan panjang dalam sejarah birokrasi di Indonesia. Pada awal kemerdekaan Indonesia, tepatnya pada tanggal 25 September 1945, pemerintah Indonesia melalui ketua KNPI Kasman Singodimedjo memutuskan untuk menetapkan seluruh pegawai pemerintahan terdahulu dari segala jabatan dan tingkatan menjadi pegawai pemerintahan Indonesia. Inilah langkah awal dari pembentukan birokrasi Indonesia.

Pada tanggal 30 Mei 1948 Pemerintah Republik Indonesia membentuk Kantor Urusan Pegawai (KUP) yang berkedudukan di Yogyakarta, ibu kota pemerintahan kala itu, sebagai instansi pembina pegawai negeri. Situasi pemerintahan yang terbelah dua saat itu mengakibatkan pegawai negeri terbagi dua. Sebagian berada di bawah pemerintah Republik Indonesia yang berada di bawah binaan KUP, sebagian lagi berada di bawah pemerintah Hindia Belanda yang berada di bawah Djawatan Umum Urusan Pegawai (DUUP) yang berkedudukan di Jakarta. Dalam perkembangan selanjutnya KUP inilah yang menjadi cikal bakal perkembangan BKN. Pembubaran RIS dan pembentukan Negara Kesatuan Republik Indonesia pada 15 Agustus 1950 membuat pemerintah merasa perlu untuk memusatkan masalah kepegawaian yang sebelumnya dilaksanaan secara terpisah oleh KUP dan DUUP. Keduanya dilebur dalam wadah KUP yang berkantor di Jalan Kramat Raya, Jakarta.

Meski kedudukan KUP dipindahkan ke Jakarta, dalam pelaksanaan tugasnya masih ada unit kerja yang berkedudukan di daerah yakni Bagian Tata Usaha Kepegawaian (TUK) di Yogyakarta serta Bagian Pensiun dan Tunjangan (Biro P&T) di Bandung.

Situasi politik yang serba memanas dan belum pasti kala itu mengakibatkan terjadinya sejumlah peralihan dalam sistem kenegaraan Indonesia. Situasi tersebut yang menyebabkan pemerintah merasa perlu menetapkan kembali kedudukan, fungsi, dan tugas organisasi KUP. Sesuai dengan perkembangan yang terjadi, peranan aparatur pemerintah semakin dirasakan sehingga pemerintah memandang perlu menetapkan kembali kedudukan, fungsi, tugas, dan organisasi KUP. Pandangan ini sesuai dengan aturan dalam Peraturan Pemerintah Nomor 32 Tahun 1950 beserta peraturan pelaksanaannya yang dimaksud dalam Keputusan Perdana Menteri RI Nomor 30/PM/1951 tanggal 7 April 1951.

KUP yang merupakan institusi yang bertugas melakukan pembinaan kepegawaian diubah menjadi Badan Administrasi Kepegawaian Negara (BAKN) dengan Peraturan Pemerintah Nomor 32 Tahun 1972. Penetapan Peraturan Pemerintah ini adalah juga sebagai pelaksanaan dari Undang-undang Nomor 18 Tahun 1961. Terbentuknya BAKN tidak terlepas dari perkembangan administrasi kepegawaian yang pada saat itu dirasakan semakin meningkat peranannya. Untuk memperkuat hal tersebut maka ditetapkanlah Undang-Undang Nomor 8 tahun 1974 tentang Pokok-Pokok Kepegawaian.

Dalam perkembangannya serta untuk meningkatkan mutu pelayanan kepada PNS, dibentuklah Kantor Wilayah BAKN di tingkat Provinsi berdasarkan Keppres Nomor 53 Tahun 1980. Keppres tersebut kemudian direalisasikan dengan Keputusan Kepala BAKN Nomor 2053/Kep/1981 tanggal 6 Agustus 1981 tentang Organisasi dan Tata Kerja Kanwil BAKN. Seiring dengan pesatnya perkembangan dalam bidang kepegawaian dengan bergesernya paradigma dari masalah administratif ke arah manajemen sumber daya manusia, BAKN juga berkembang dan melakukan reformasi kepegawaian dengan mengubah BAKN menjadi Badan Kepegawaian Negara (BKN) berdasarkan Keppres No. 95 Tahun 1999 Tanggal 11 Agustus 1999. Keberadaan

BKN kemudian diperkuat dengan Undang-Undang Nomor 43 Tahun 1999 tentang perubahan Undang-Undang Nomor 8 Tahun 1974 tentang Pokok-Pokok Kepegawaian. Dengan demikian maka Kantor Wilayah BAKN berubah nama menjadi Kantor Regional BKN.

Kantor Wilayah III BKN pun diubah menjadi Kantor Regional III BKN sesuai dengan Keputusan Kepala BKN no. 3/KEP/2000. Struktur Organisasi dan Tata Kerja Kanreg ditetapkan melalui Keputusan Kepala BKN No. 59 tahun 2001. Seiring dengan otonomi daerah yang merubah kewenangan pembinaan kepegawaian kepada masing-masing daerah maka BKN kini adalah lembaga yang memiliki kewenangan menetapkan Norma, Standar, dan Prosedur Pembinaan Kepegawaian secara nasional sedangkan fungsi administratifnya diserahkan kepada masing-masing daerah/Institusi.

Berdasarkan Kep. 59/2001 Kanreg BKN mempunyai tugas menyelenggarakan sebagian tugas pokok dan fungsi BKN di bidang adiministrasi dan manajemen kepegawaian negara di wilayah kerjanya, yang kewenangannya masih melekat pada pemerintah sesuai dengan ketentuan peraturan perundangan-undangan yang berlaku. Untuk menyelenggarakan tugas tadi, Kanreg BKN melaksanakan fungsi:

1. Koordinasi, bimbingan, pemberian petunjuk teknis, dan pengendalian terhadap pelaksanaan peraturan perundang-undangan di bidang kepegawaian.
2. Pemberian pertimbangan dan atau penetapan mutasi kepegawaian bagi PNS Pusat dan Daerah di wilayah kerjanya sesuai peraturan perundang-undangan yang berlaku
3. Penetapan pensiun PNS Pusat dan penetapan status kepegawaian di wilayah kerjanya.
4. Pemberian pertimbangan pensiun PNS Daerah dan penetapan status kepegawaian di wilayah kerjanya.
5. Penyelenggaraan dan pemeliharaan jaringan informasi daya kepegawaian PNS Pusat dan Daerah di wilayah kerjanya.

6. Penetapan pemindahan PNS antar daerah Provinsi atau antara Daerah Kabupaten/kota dan daerah Kabupaten/Kota lain Provinsi.
7. Tugas-tugas lain yang ditetapkan oleh Kepala BKN.

2.1.2 Visi dan Misi

Dalam menjalankan kegiatan usahanya, BKN memiliki visi dan misi yang tertera di bawah ini.

2.1.2.1 Visi :

1. Menjadi Pembina dan Penyelenggara Manajemen Kepegawaian yang Profesional dan Bermartabat Tahun 2025.

2.1.2.2 Misi :

Visi BKN dalam Renstra 2015-2019 adalah:

1. Mengembangkan dan Mengoptimalkan Sistem Pelayanan Kepegawaian
2. Mengembangkan dan Mengoptimalkan Sistem Pengawasan dan Pengendalian Kepegawaian.
3. Mengembangkan dan Mengoptimalkan Sistem Peraturan Perundang-Undangan, Kinerja dan Kesejahteraan Pegawai.
4. Mengembangkan dan Mengoptimalkan Sistem Informasi Manajemen Kepegawaian.
5. Mengembangkan dan Mengoptimalkan Sistem Manajemen Internal BKN.

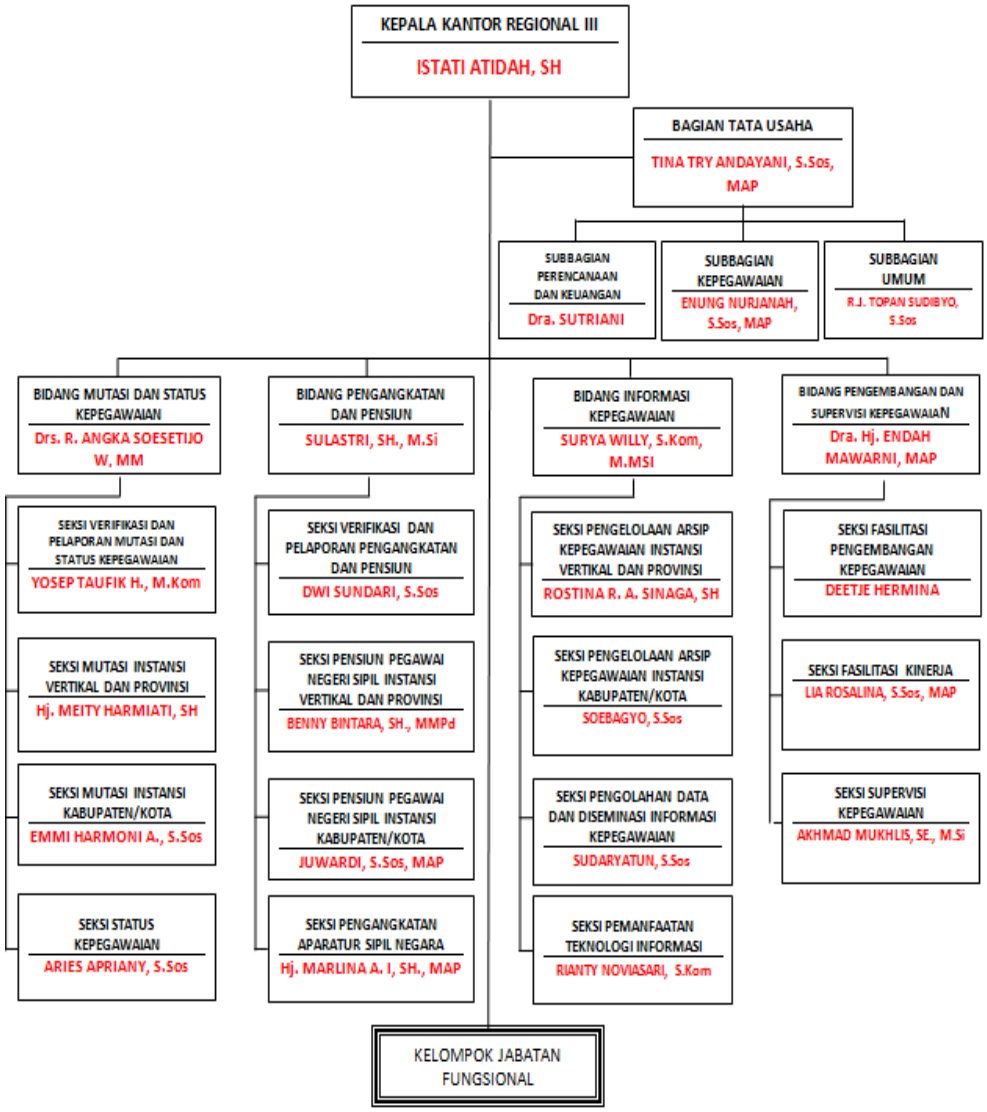
2.1.3 Gedung Kantor Regional III BKN Bandung



Gambar 2.1 Gedung Kantor Regional III BKN Bandung

2.1.4 Stuktur Organisasi

Struktur organisasi mempunyai arti penting karena struktur organisasi merupakan bentuk atau pola formal kegiatan dan hubungan antara berbagai bagian-bagian didalam suatu perusahaan. Dengan mengetahui struktur organisasi dapat diperoleh gambaran tentang bagian-bagian yang ada di dalamnya, apa peranan masing-masing bagian tersebut dan wewenang serta tanggung jawabnya dalam melaksanakan tugasnya.



Gambar 2.2 struktur organisasi kantor bkn regional iii bandung

2.1.4.1 Tugas Pokok dan Fungsi

Kantor Regional Badan Kepegawaian Negara menyelenggarakan fungsi :

1. koordinasi, bimbingan, pemberian petunjuk teknis, dan pengendalian terhadap pelaksanaan peraturan perundang-undangan di bidang kepegawaian;

2. pemberian pertimbangan, persetujuan, dan atau penetapan mutasi kepegawaian bagi Pegawai Negeri Sipil instansi pusat dan instansi daerah di wilayah kerjanya sesuai peraturan perundang-undangan yang berlaku;
3. penetapan pensiun dan status kepegawaian Pegawai Negeri Sipil instansi pusat di wilayah kerjanya;
4. penetapan pensiun dan status kepegawaian Pegawai NegeriSipil instansi daerah di wilayah kertanya;
5. penyelenggaraan dan pemeliharaan jaringan informasi data kepegawaian Pegawai Negeri Sipil instansi pusat dan instansi daerah di wilayah kerjanya;
6. pembinaan, fasilitasi, dan evaluasi penilaian kinerja danpenyusunan standar kompetensi Pegawai Aparatur SipilNegara di wilayah kerjanya;
7. pengelolaan teknologi informasi penilaian kinerja PegawaiAparatur Sipil Negara di wilayah kerjanya; dan
8. pelaksanaan tugas lain yang diberikan oleh Kepala BadanKepegawaian Negara.

Kepala Kantor Regional BKN mempunyai tugas memimpin pelaksanaan tugas dan fungsi Kanreg BKN di wilayah kerjanya.

Untuk melaksanakan tugas sebagaimana dimaksud, Bagian Tata Usaha menyelenggarakan fungsi:

a. penjurusunan rencana dan program;
b. pengelolaan administrasi keuangan;
c. pengelolaan administrasi kepegawaian;
d. pelaksanaan ketatausahaan;
e. pengelolaan barang milik negara dan rumah tangga; dan
f. pelaksanaan dokumentasi dan hubungan masyarakat.

1. Bagian Tata Usaha terdiri atas:

a. Subbagian Perencanaan dan Keuangan

Sub bagian Perencanaan dan Keuangan mempunyai tugas menyiapkan bahan dan penyusunan rencana, program dan anggaran, penyusunan laporan, pe

flgelolaan administrasi keuangan dan pembayaran, serta pembukuan dan verifikasi.

b. Subbagian Kepegawaian

Subbagian Kepegawaian mempunyai tugas menyiapkan bahan pelaksanaan urusan tata usaha kepegawaian, administrasi mutasi dan pengembangan kepegawaian, serta kesejahteraan pegawai.

c. Subbagian Umum

Subbagian Umum mempunyai tugas menyiapkan bahan pelaksanaan urusan persuratan, kearsipan, ekspedisi, penggandaan, perlengkapan, angkutan kendaraan dinas, urusan dalam dan keamanan, menyiapkan bahan pelaksanaan pengelolaan barang milik negara dan pengadaan barang/jasa, serta menyiapkan bahan pelaksanaan dokumentasi dan hubungan masyarakat.

2. Bidang mutasi dan status Kepegawaian

Bidang Mutasi dan Status Kepegawaian mempunyai tugas melaksanakan pemberian persetujuan kenaikan pangkat dan mutasi lainnya, serta pertimbangan teknis mutasi antar kabupaten/kota dalam provinsi, dan penetapan mutasi dari instansi pusat ke instansi daerah, serta penyiapan pertimbangan status kepegawaian di wilayah kerjanya.

Dalam melaksanakan tugasnya, Bidang Mutasi dan Status Kepegawaian menyelenggarakan fungsi:

1. Pemberian persetujuan teknis kepada Pejabat Pembina Kepegawaian Daerah untuk penetapan kenaikan pangkat Pegawai
2. Pemberian persetujuan teknis peninjauan masa kerja;
3. Penetapan mutasi Pegawai Negeri Sipil dari instansi pusat ke instansi daerah;
4. Penyiapan penetapan kartu identitas pegawai dan keluarganya; dan
5. Penyiapan persetujuan pemberian cuti diluar tanggungan negara bagi Pegawai Negeri Sipil instansi pusat dan instansi daerah sampai dengan Pembina Tingkat I golongan ruang IV/ b di wilayah kerjanya.

Bidang Mutasi dan Status Kepegawaian terdiri atas :

1. Seksi Verifikasi dan Pelaporan Mutasi dan Status Kepegawaian

   Seksi Verifikasi dan Pelaporan Mutasi dan Status Kepegawaian mempunyai tugas menyiapkan bahan verifikasi dan pelaporan, administrasi mutasi dan status kepegawaian, serta monitoring dan evaluasi pelaksanaan kegiatan bidang mutasi dan status kepegawaian.

2. Seksi Mutasi Instansi Vertikal dan Provinsi

   Seksi Mutasi Instansi Vertikal dan Provinsi mempunyai tugas menyiapkan bahan pemeriksaan dan pengujian persyaratan pemberian persetujuan kenaikan pangkat dan mutasi lainnya, serta pertimbangan teknis mutasi antar instansi bagi Pegawai Negeri Sipil pada instansi vertikal dan antar provinsi di wilayah kerjanya.

3. Seksi Mutasi Instansi Kabupaten/Kota

   Seksi Mutasi Instansi Kabupaten/Kota mempunyai tugas menyiapkan bahan pemeriksaan dan pengujian persyaratan pemberian persetujuan kenaikan pangkat dan mutasi lainnya serta pertimbangan teknis mutasi antar instansi bagi Pegawai Negeri Sipil pada instansi kabupaten dan kota di wilayah kerjanya.

4. Seksi Status Kepegawaian.

   Seksi Status Kepegawaian mempunyai tugas menyiapkan bahan penetapan kartu identitas pegawai dan keluarganya, serta persetujuan cuti di luar tanggungan negara bagi Pegawai Negeri Sipil instansi pusat dan instansi daerah sampai dengan Pembina Tingkat I golongan ruang IV/b di wilayah kerjanya sesuai dengan ketentuan peraturan yang berlaku.

3. Bidang pengangkatan dan pensiun

Bidang Pengangkatan dan Pensiun mempunyai tugas melaksanakan penyiapan penetapan Nomor Induk Pegawai Negeri Sipil, pengangkatan Calon Pegawai Negeri Sipil/Pegawai Negeri Sipil, penetapan kenaikan pangkat pengabdian, pemberhentian dan pensiun bagi Pegawai Negeri Sipil dan Janda/Dudanya yang telah mencapai batas usia pensiun dan/atau meninggal dunia, Janda/Duda bagi Pensiunan yang meninggal dunia, serta pengelolaan tata naskah pensiun di wilayah kerjanya.

Dalam melaksanakan tugasnya, Bidang Pengangkatan dan Pensiun menyelenggarakan fungsi:

1. Penyiapan penetapan Nomor Induk Pegawai untuk Calon Pegawai Negeri Sipil/Pegawai Negeri Sipil Instansi Daerah di wilayah kerjanya;
2. penyiapan penetapan pemberhentian dan pemberian pensiun Pegawai Negeri Sipil Pada instansi Pusat yang berpangkat Pembina Tingkat I golongan ruang IV/b ke bawah yang mencapai batas usia pensiun dan pensiun janda/dudanya;
3. Penyiapan penetapan pemberhentian dan pemberian pensiun bagi Pegawai Negeri Sipil pada instansi Daerah yang berpangkat Pembina Tingkat I golongan ruang IV/b ke bawah yang mencapai batas usia pensiun dan pensiun Janda/Dudanya;
4. Penyiapan pemberian persetujuan teknis pengangkatan menjadi Pegawai Negeri Sipil bagi Calon Pegawai Negeri Sipil pada instansi Daerah yang menjalani masa percobaan lebih dari 2 (dua) tahun;
5. Penyiapan pemberian persetujuan teknis pengangkatan menjadi Pegawai Negeri Sipil bagi Calon Pegawai Negeri Sipil pada instansi Pusat yang menjalani masa percobaan lebih dari 2 (dua) tahun; dan
6. Penetapan kenaikan pangkat pengabdian Pegawai Negeri Sipil pada instansi Pusat dan instansi daerah.

Bidang Pengangkatan dan Pensiun terdiri atas :

1. Seksi Verifikasi dan Pelaporan Pengangkatan dan Pensiun

   Seksi Verifikasi dan Pelaporan Pengangkatan dan Pensiun mempunyai tugas menyiapkan bahan verifikasi dan pelaporan, administrasi pengangkatan dan pensiun, serta monitoring dan evaluasi pelaksanaan kegiatan bidang pengangkatan dan pensiun.

2. Seksi Pensiun Pegawai Negeri Sipil Instansi Vertikal dan Provinsi

   Seksi Pensiun Pegawai Negeri Sipil Instansi Vertikal dan Provinsi mempunyai tugas menyiapkan bahan pemeriksaan dan pengujian persyaratan penetapan kenaikan pangkat pengabdian, pemberhentian dan pensiun bagi

Pegawai Negeri Sipil dan Janda/Dudanya yang telah mencapai batas usia pensiun atau meninggal dunia, Janda/Duda bagi Pensiunan yang meninggal dunia,
serta pengelolaan tata naskah pensiun pada instansi vertikal dan provinsi di wilayah kerjanya.

3. Seksi Pensiun Pegawai Negeri Sipil Instansi Kabupaten/Kota

   Seksi Pensiun Pegawai Negeri Sipil Instansi Kabupaten/Kota mempunyai tugas melakukan pemeriksaan dan pengujian persyaratan penetapan kenaikan pangkat pengabdian, pemberhentian dan pensiun bagi Pegawai Negeri Sipil dan Janda/Dudanya yang telah mencapai batas usia pensiun atau meninggal dunia, Janda/Duda bagi Pensiunan yang meninggal dunia serta pengelolaan tata naskah pensiun pada instansi kabupaten/kota di wilayah kerjanya.

4. Seksi Pengangkatan Aparatur Sipil Negara

   Seksi Pengangkatan Aparatur Sipil Negara mempunyai tugas menyiapkan bahan penyiapan penetapan Nomor Induk Pegawai untuk Calon Pegawai Negeri Sipil/Pegawai Aparatur Sipil Negara, pemberian penetapan persetujuan pengangkatan menjadi Pegawai Negeri Sipil bagi Calon Pegawai Negeri Sipil yang melebihi masa percobaan lebih dari 2 (dua) tahun di wilayah kerjanya sesuai dengan ketentuan peraturan yang berlaku.

4. Bidang informasi Kepegawaian

Bidang Informasi Kepegawaian mempunyai tugas melaksanakan dan memfasilitasi pengembangan sistem informasi kepegawaian Aparatur Sipil Negara pada instansi di wilayah kerjanya.

Dalam melaksanakan tugasnya, Bidang Informasi Kepegawaian menyelenggarakan fungsi :

1. Pengelolaan dan pemeliharaan basis data kepegawaian.
2. Pengelolaan dan pemeliharaan basis data penilaian kinerja pegawai Aparatur Sipil Negara.
3. Pelaksanaan pengolahan data kepegawaian.
4. Pengelolaan dan pemeliharaan arsip kepegawaian.

5. Pelaksanaan penyuntingan dan penyandian data kepegawaian.
6. Pengelolaan dan pemeliharaan aplikasi sistem penilaian kinerja pegawai Aparatur Sipil Negara.
7. Penyelenggaraan penyajian dan pertukaran informasi.
8. Pelaksanaan fasilitasi pengembangan sistem informasi kepegawaian.

Bidang Informasi Kepegawaian terdiri atas :

1. Seksi Pengelolaan Arsip Kepegawaian Instansi Vertikal dan Provinsi

   Seksi Pengelolaan Arsip Kepegawaian Instansi Vertikal dan Provinsi mempunyai tugas menyiapkan bahan urusan pengelolaan dan pemeliharaan arsip kepegawaian baik dalam bentuk fisik maupun elektronik, serta penyiapan penyusunan laporan di lingkungan instansi vertikal dan provinsi di wilayah kerjanya.
2. Seksi Pengelolaan Arsip Kepegawaian Instansi Kabupaten/Kota

   Seksi Pengelolaan Arsip Kepegawaian Instansi Kabupaten/Kota mempunyai tugas menyiapkan bahan urusan pengelolaan dan pemeliharaan arsip kepegawaian baik dalam bentuk fisik maupun elektronik, serta penyiapan penyusunan laporan di lingkungan instansi kabupaten/kota di wilayah kerjanya.
3. Seksi Pengolahan Data dan Diseminasi Informasi Kepegawaian

   Seksi Pengolahan Data dan Diseminasi Informasi Kepegawaian mempunyai tugas menyiapkan bahan pengelolaan, pengembangan dan fasilitasi implementasi sistem, jaringan dan basis data, serta diseminasi informasi kepegawaian Aparatur Sipil Negara instansi di wilayah kerjanya.
4. Seksi Pemanfaatan Teknologi Informasi

   Seksi Pemanfaatan Teknologi Informasi mempunyai tugas menyiapkan bahan pengelolaan dan fasilitasi pemanfaatan teknologi informasi kepegawaian Aparatur Sipil Negara instansi di wilayah kerjanya.

5. Bidang pengembangan dan supervisi Kepegawaian

Bidang Pengembangan dan Supervisi Kepegawaian mempunyai tugas melaksanakan pengembangan kepegawaian dan supervisi pelaksanaan manajemen Aparatur Sipil Negara instansi di wilayah kerjanya.

Dalam melaksanakan tugasnya, Bidang Pengembangan dan Supervisi Kepegawaian menyelenggarakan fungsi :

1. Pemberian bimbingan dan petunjuk teknis kepegawaian;
2. Penyiapan pengembangan dan pengawasan standar kompetensi jabatan, serta pengendalian pemanfaatan lulusan pendidikan dan pelatihan Pegawai Aparatur Sipil Negara di wilayah kerjanya;
3. Koordinasi dengan aparat pengawasan fungsional bidang kepegawaian;
4. Pelaksanaan supervisi kinerja dan disiplin Aparatur Sipil Negara di lingkungan Kanreg BKN;
5. Asistensi pengukuran standar kinerja pegawai Aparatur Sipil Negara;
6. Asistensi pemantauan dan evaluasi penilaian kinerja pegawai Aparatur Sipil Negara;
7. Pengintegrasian sistem aplikasi kinerja pegawai Aparatur Sipil Negara dengan unit pengguna di wilayah kerjanya; dan
8. Pelaksanaan monitoring penempatan dalam jabatan dan pasca pengembangan kompetensi.

Bidang Pengembangan dan Supervisi Kepegawaian terdiri atas :

1. Seksi Fasilitasi Pengembangan Kepegawaian

   Seksi Fasilitasi Pengembangan Kepegawaian mempunyai tugas menyiapkan bahan fasilitasi pengembangan kepegawaian Aparatur Sipil Negara instansi di wilayah kerjanya.
2. Seksi Fasilitasi Kinerja

   Seksi Fasilitasi Kinerja mempunyai tugas menyiapkan bahan fasilitasi kinerja Aparatur Sipil Negara Instansi di wilayah kerjanya.

3. Seksi Supervisi Kepegawaian

Seksi Supervisi Kepegawaian mempunyai tugas menyiapkan bahan supervisi kepegawaian Aparatur Sipil Negara Instansi di wilayah kerjanya, serta pengawasan dan pengendalian internal.

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan Aplikasi.

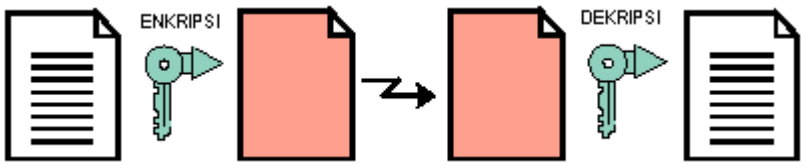
2.2.1 Kriptografi

Secara etimologi kata kriptografi (Cryptography) berasal dari bahasa Yunani, yaitu kryptos yang artinya yang tersembunyi dan graphein yang artinya tulisan (Prayudi, 2005). Awal mula kriptografi dipahami sebagai ilmu tentang menyembunyikan pesan (Sadikin, 2012), tetapi seiring perkembangan zaman hingga saat ini pengertian kriptografi berkembang menjadi ilmu tentang teknik matematis yang digunakan untuk menyelesaikan persoalan keamanan berupa privasi dan otentikasi (Diffie, 1976). Sejarah penulisan rahasia tertua dapat ditemukan pada peradaban Mesir kuno, yakni tahun 3000 SM. Bangsa Mesir menggunakan ukiran rahasia yang disebut dengan hieroglyphics untuk menyampaikan pesan kepada orang-orang yang berhak.

kata kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu “kriptos” dan “graphia”. Kata kriptos digunakan untuk mendeskripsikan sesuatu yang disembunyikan, rahasia atau misterius. Sedangkan kata graphia berarti tulisan. Kriptografi didefinisikan sebagai ilmu dan pelajaran untuk tulisan rahasia dengan pertimbangan bahwa komunikasi dan data dapat dikodekan untuk mencegah dari mata-mata atau orang lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan tertentu dan metode lainnya sehingga hanya orang yang berhak yang dapat mengetahui isi pesan sebenarnya.

Dalam arti lain, cryptography adalah seni dan ilmu dalam mengamankan pesan. Dalam dunia kriptografi, pesan disebut plaintext atau cleartext. Proses untuk

menyamarkan pesan dengan cara sedemikian rupa untuk menyembunyikan isi aslinya disebut enkripsi. Pesan yang telah dienkripsi disebut ciphertext. Proses pengembalian sebuah ciphertext ke plaintext disebut dekripsi.



Gambar 2.3 Konsep Dasar dari Enkripsi dan Dekripsi

Ada beberapa Aspek Keamanan Kriptografi diantaranya :

1. Kerahasiaan (confidentiality), menjamin bahwa data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja. Kerahasiaan bertujuan untuk melindungi suatu informasi dari semua pihak yang tidak berhak atas informasi tersebut.
2. Otentikasi (authentication), merupakan identifikasi yang dilakukan oleh masing – masing pihak yang saling berkomunikasi, maksudnya beberapa pihak yang berkomunikasi harus mengidentifikasi satu sama lainnya. Informasi yang didapat oleh suatu pihak dari pihak lain harus diidentifikasi untuk memastikan keaslian dari informasi yang diterima.
3. Integritas (integrity), menjamin setiap pesan yang dikirim pasti sampai pada penerimanya tanpa ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya, dan ditambahkan. Integritas data bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berhak atas informasi tersebut. Untuk menjamin integritas data ini pengguna harus mempunyai kemampuan untuk mendeteksi terjadinya manipulasi data oleh pihak-

pihak yang tidak berkepentingan. Manipulasi data yang dimaksud di sini meliputi penyisipan, penghapusan, maupun penggantian data.

4. Nirpenyangkalan (Nonrepudiation), mencegah pengirim maupun penerima mengingkari bahwa mereka telah mengirimkan atau menerima suatu pesan. Jika sebuah pesan dikirim, penerima dapat membuktikan bahwa pesan tersebut memang dikirim oleh pengirim yang tertera. Sebaliknya, jika sebuah pesan diterima, pengirim dapat membuktikan bahwa pesannya telah diterima oleh pihak yang ditujunya. (Ariyus, 2008).

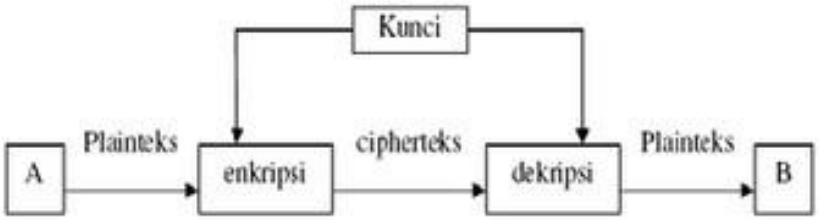
Seiring perkembangan riset kriptografi, telah dikembangkan teknik kriptografi kunci publik yang menandakan revolusi dari perkembangan teknik-teknik kriptografi. Teknik ini berdasarkan pada penggunaan kunci yang berbeda untuk proses enkripsi dan dekripsi pesan. Proses pemilihan dan penjaminan kerahasiaan kunci berdasarkan pada operasi matematika yang sulit dikerjakan. Hingga kini, telah banyak algoritma kriptografi kunci publik yang digunakan. Beberapa diantaranya adalah RSA, El-Gamal, Diffie-Hellman, Elliptic Curve, Rabin, dan Schmidt-Samoa. Algoritma Schmidt-Samoa dipublikasikan pada tahun 2006 dan belum banyak dokumentasi dan implementasi pengamanan data yang dilakukan menggunakan algoritma ini.

Pada dasarnya algoritma ini memanfaatkan kerumitan dalam menghitung perpangkatan modulo dalam melakukan enkripsi dan dekripsi pada pesan. Kekuatan dari algoritma ini terletak pada sulitnya memfaktorkan suatu bilangan yang sangat besar menjadi suatu bilangan prima dan kuadrat dari bilangan prima lainnya. Dengan demikian, dapat dikatakan bahwa algoritma ini memiliki banyak kesamaan dengan algoritma RSA. Jenis Algoritma Kriptografi Berdasarkan jenis kunci, algoritma kriptografi dikelompokan menjadi dua bagian, yaitu : algoritma simetris (algoritma kunci privat) dan algoritma asimetris (algoritma kunci publik)

A. Algoritma Simetris

Algoritma simetris adalah salah satu jrnis kunci pada algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Istilah lain untuk kriptografi kunci simetri adalah kriptografi kunci privat (private-key

cryptography). Sistem kriptografi kunci-simetri diasumsikan sebagai pengirim dan penerima pesan yang sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kuncinya.



Gambar 2.4 Algoritma Simetris

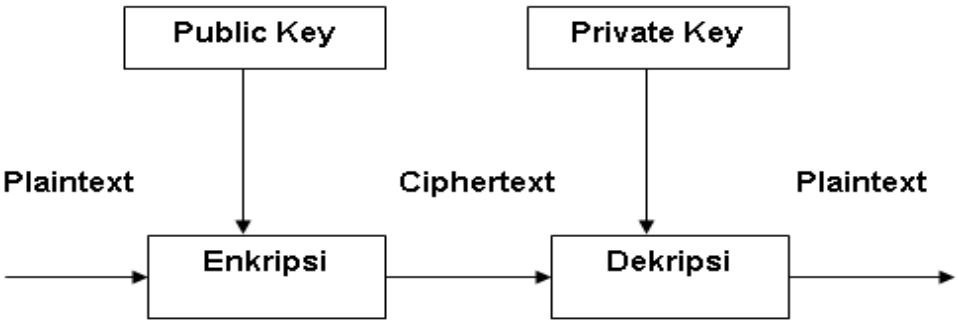
Kriptografi simetri adalah jenis kriptografi yang diketahui masuk ke dalam catatan sejarah hingga tahun 1976. Semua algoritma kriptografi klasik termasuk ke dalam sistem kriptografi simetri. Salah satu kelebihan pada algoritma simetris yaitu proses enkripsi dan deskripsinya jauh lebih cepat dibandingkan dengan algoritma asimetris. Sedangkan kelemahannya yaitu pada permasalahan distribusi kunci (key distribution). Seperti yang telah dibahas sebelumnya, proses enkripsi dan deskripsi pada kriptografi simetri menggunakan kunci yang sama. Sehingga timbul persoalan untuk menjaga kerahasian kunci. Contohnya pada saat pengiriman kunci dilakukan melalui media yang tidak aman seperti internet.

Jika kunci ini hilang atau sudah diketahui oleh orang yang tidak berhak, maka kriptosistem ini dinyatakan tidak aman lagi. Kelemahan lain adalah masalah efisiensi jumlah kunci. Jika terdapat n user, maka diperlukan n(n-1)/2 kunci, sehingga untuk jumlah user yang sangat banyak, sistem ini tidak efisien lagi.

B. Algoritma Asimetris

Algoritma asimetris atau dapat disebut juga dengan algoritma kunci public, didesain sebaik mungkin sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci dekripsinya. Dimana kunci untuk enkripsi tidak rahasia (diumumkan ke

publik), sementara kunci dekripsinya bersifat rahasia (hanya diketahui oleh penerima pesan). Pada kriptografi asimetris, setiap orang yang akan berkomunikasi harus mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim pesan akan mengenkripsi pesan menggunakan kunci publik si penerima pesan dan hanya penerima pesan yang dapat mendekripsi pesan tersebut karena hanya ia yang mengetahui kunci privatnya sendiri.



Gambar 2.5 Algoritma Asimetris

Kriptografi kunci-publik dapat dianalogikan seperti kotak surat yang terkunci dan memiliki lubang untuk memasukkan surat. Setiap orang dapat memasukkan surat ke dalam kotak surat tersebut, tetapi hanya pemilik kotak yang dapat membuka kotak dan membaca surat di dalamnya karena ia yang memiliki kunci. Sistem ini memiliki dua keuntungan. Yang pertama yaitu, tidak ada kebutuhan untuk mendistribusikan kunci privat sebagaimana pada sistem kriptografi simetri. Kunci publik dapat dikirim ke penerima pesan melalui saluran yang sama dengan saluran yang digunakan untuk mengirim pesan. Saluran untuk mengirim pesan umumnya tidak aman. Kedua, jumlah kunci yang digunakan untuk berkomunikasi secara rahasia dengan banyak orang tidak perlu sebanyak jumlah orang tersebut, cukup membuat dua buah kunci, yaitu kunci publik bagi para koresponden untuk mengenkripsi pesan, dan kunci privat untuk mendekripsi pesan. Berbeda dengan kriptografi kunci-simetris yang membuat kunci sebanyak jumlah pihak yang diajak berkorespondensi. Meski masih terbilang

baru (sejak 1976), kriptografi kunci-publik mempunyai kontribusi yang luar biasa dibandingkan dengan sistem kriptografi simetri.

Kriptografi kunci publik dapat dianalogikan seperti pesan yang dimasukkan ke dalam kotak surat yang terkunci. Setiap orang dapat memasukkan surat ke dalam kotak tersebut tetapi hanya pemilik kunci kotak surat yang dapat membuka isinya[2]. Sistem kriptografi ini memiliki dua jenis kunci, yaitu kunci publik yang digunakan untuk enkripsi dan kunci privat yang digunakan untuk dekripsi. Karena dua jenis kunci ynag digunakan berbeda maka teknik kriptografi ini disebut juga kriptografi nirsimetri. Misalkan E adalah fungsi enkripsi dan D adalah fungsi dekripsi. Untuk pasangan kunci publik-privat (e, d) dan pesan-cipherteks (m, c) berlaku persamaan

- Ee(M)=c
- Dd(C)=m

Kriptografi kunci publik didasarkan pada permasalahan- permasalahan matematika yang sulit dipecahkan seperti pemfaktoran dan logaritma diskrit. Contoh masalah pemfaktoran yang sulit adalah mencari faktor prima dari suatu bilangan bulat besar yang merupakan hasil perkalian dari dua bilangan prima besar. Sedangkan contoh masalah logaritma diskrit adalah menentukan nilai x yang memenuhi

- A*=b mod m

jika a, b, dan m sudah diketahui.

Teknik kriptografi kunci publik tidak membutuhkan saluran khusus untuk mendistribusikan kunci privat. Kunci publik dapat dikirimkan melalui saluran yang sama dengan pesan. Selain itu jumlah kunci yang dibuat dapat ditekan, yaitu hanya ada dua kunci yang digunakan pada sistem kriptografi ini.

Beberapa algoritma Asimetris diantaranya :

2.2.2 RSA

Algoritma RSA merupakan salah satu algoritma public key yang populer dipakai dan bahkan masih dipakai hingga saat ini. Kekuatan algoritma ini terletak

pada proses eksponensial, dan pemfaktoran bilangan menjadi 2 bilangan prima yang hingga kini perlu waktu yang lama untuk melakukan pemfaktorannya.

Algoritma ini dinamakan sesuai dengan nama penemunya, Ron Rivest, Adi Shamir dan Adleman(Rivest-Shamir-Adleman) yang dipublikasikan pada tahun 1977 di MIT, menjawab tantangan yang diberikan algoritma pertukaran kunci Diffie Hellman.

Skema RSA sendiri mengadopsi dari skema block cipher, dimana sebelum dilakukan enkripsi, plainteks yang ada dibagi – bagi menjadi blok – blok dengan panjang yang sama, dimana plainteks dan cipherteksnya berupa integer(bilangan bulat) antara 1 hingga n, dimana n berukuran biasanya sebesar 1024 bit, dan panjang bloknya sendiri berukuran lebih kecil atau sama dengan log(n) +1 dengan basis 2. Fungsi enkripsi dan dekripsinya dijabarkan dalam fungsi berikut :

C = Me mod n (fungsi enkripsi)

M = Cd mod n (fungsi dekripsi)

C = Cipherteks

M = Message / Plainteks

e = kunci publik

d= kunci privat

n = modulo pembagi(akan dijelaskan lebih lanjut)

Kedua pihak harus mengetahui nilai e dan nilai n ini, dan salah satu pihak harus memilki d untuk melakukan dekripsi terhadap hasil enkripsi dengan menggunakan public key e. Penggunaan algoritma ini harus memenuhi kriteria berikut :

1. Memungkinkan untuk mencari nilai e, d, n sedemikian rupa sehingga Med mod n = M untuk semua M < n.
2. Relatif mudah untuk menghitung nilai Me mod n dan Cd mod n untuk semua nilai M < n.
3. Tidak memungkinkan mencari nilai d jika diberikan nilai n dan e.

nilai e dan d ini, gcd(d,e)=1

sebelum memulai penggunaan RSA ini, terlebih dahulu kita harus memiliki bahan – bahan dasar sebagai berikut :

1. p, q = 2 bilangan prima yang dirahasiakan
2. n, dari hasil p.q
3. e, dengan ketentuan gcd (Φ(n), e) =1
4. d, e-1 (mod Φ(n))

Contoh :

1. Pilih 2 bilangan prima, misalnya p = 17 dan q = 11.
2. Hitung n = pq = 17 × 11 = 187.
3. Hitung Φ(n) = (p – 1)(q – 1) = 16 × 10 = 160.
4. Pilih nilai e sedemikian sehingga relatif prima terhadap Φ(n) = 160 dan kurang dari Φ(n); kita pilih e = 7.
5. Hitung d sedemikian sehingga de ≡ 1 (mod 160) dan d < 160.Nilai yang didapatkan d = 23,karena 23 × 7 = 161 = (1 × 160) + 1; d dapat dihitung dengan Extended Euclidean Algorithm.

nilai e dan d inilah yang kita sebut sebagai Public Key(e) dan Private Key(d). Pasangan Kunci Publiknya ={7,187} dan Kunci Privatnya = {23, 187}

Misalnya kita punya M 88. Untuk proses enkripsi, kita akan menghitung C = 887 mod 187.

= 887 mod 187

=894,432 mod 187

=11

nilai C =11.

Selanjutnya, nilai C ini dikirimkan kepada penerima untuk didekripsi dengan kunci privat miliknya.

M = Cd mod n

= 1123 mod 187

=79,720,245 mod187

= 88

2.2.3 DSA

Tandatangan digital berbentuk sepasang besar angka yang ditampilkan komputer sebagai string dari digit biner. Tandatangan digital dihitung dengan menggunakan sejumlah aturan dan sejumlah parameter sehingga identitas pemilik dan integritas data dapat diverifikasi. Pembuat tanda tangan menggunakan kunci privat untuk membuat tanda tangan; sedangkan Kunci publik, yang berkorespodensi dengan kunci privat namun tidak sama, digunakan untuk Memverifikasi tanda tangan. Setiap user memiliki sepasang kunci publik dan kunci privat.

Kunci publik diasumsikan diketahui public secara umum, sedangkan kunci privat tidak pernah disebar. DSA dapat dilihat sebagai variasi dari skema tanda tangan ElGamal. Keamanan DSA berdasarkan pada kemampuan logaritma diskrit dalam urutan bilangan prima Z*p. Domain parameter DSA dibangkitkan untuk setiap entitas dalam domain keamanan tertentu :

1. Pilih bilangan prima sepanjang 160 bit dan 1024 bit dengan kondisi : q | p
2. Pilih pembangkit g yang memiliki kelompok putaran yang unik di mana q berada dalam Z*p. Pilih sebuah elemen h ε Z*p dan hitung g = h (p-1)/q mod p. Ulangi hingga g ≠1

3 Parameter domain adalah p, q dan g.

Setiap entitas A dalam domain, dengan domain parameter (p,q,g) melakukan :

1. Pilih bilangan acak x dengan ketentuan $1 \leq x \leq q-1$
2. Hitung y = gx mod p
3. Kunci publik A adalah y, sedangkan kunci privat A adalah x

Untuk menandatangani pesan m, A melakukan :

1. Pilih bilangan acak k denganketentuan $1 \leq k \leq q-1$
2. Hitung X = gk mod p dan r = X mod q. Jika r = 0, lakukan langkah 1
3. Hitung k-1 mod q
4. Hitung e = SHA-1(m)
5. Hitung s = k-1 { e+xr } mod q. Jika s = 0, lakukan langkah 1
6. Tanda tangan A untuk pesan m adalah (r,s)

Untuk memverifikasi tanda tangan A (r,s) pada m, B mendapat salinan sah dari domain
parameter A (p,q,g) dan kunci publik y dan melakukan :

1. Verifikasi bahwa r dan s berada dalam interval [1,q-1]
2. Hitung e = SHA-1(m)
3. Hitung w = s-1 mod q
4. Hitung u1 = ew mod q dan u2 = rw mod q
5. Hitung X = gu1yu2 mod p dan v = X mod q
6. Tanda tangan benar jika dan hanya jika v = r

ALGORITMA TANDA TANGAN DIGITAL

Sepasang kunci :

P: bilangan prima antara panjang 412 dan 1024 bit

q: factor bilangan prima p – 1, panjang 160 bit

g≡ h-(p-1)/q (mod p) > 1, dan h_ < p - 1

(p, q dan g): parameter umum

x <q :kunci rahasia, mempunyai panjang 160 bit

Proses pengiriman :

k < q: bilangan acak

r≡ (gk mod p) (mod q)

s ≡ k-1 (h + xr) (mod q), h = H(m) adalah fungsi Hash satu arah untuk pesan m.

(r, s): tanda tangan digital

Pembuktian tanda tangan digital :

w ≡ s-1 (mod q)

ul ≡ h x w (mod q)

u2 ≡ r x w (mod q)

v ≡ (gu1yu2 (mod p)) (mod q)

If v = r, tanda tangan digital telah dibuktikan keasliannya

algoritmanya

Contoh : Diasumsikan dipilih p = 23 dan q = 11, q adalah factor bilangan prima dari p -1. Dipilih h' =16 < p -1, g ≡ 162(mod 23) ≡ 3 > 1.

Dipilih kunci rahasia x = 7 < q dan kunc iumum y ≡ gx(mod p) ≡ 37(mod 23)≡ 2

1. Pengirim menanda-tangani pesan :

   Dipilih K = 5, k < q = 11 dan untuk tanda tangannya (r,s) seperti di bawah ini:

   r≡ (gk mod p) (mod q)

   ≡ (35 mod 23) (mod 11) ≡ 13 (mod 11) ≡ 2

   Diasumsikan h = H(m) = 10 dan dihitung :

   s ≡ k-1 (h + xr) (mod q)

   ≡ 5-1 (10 + 7 x 2) (mod 11) ≡ (9 x 24) (mod 11) ≡ 216 (mod 11) ≡ 7

   Di mana perkalian inverse k-1adalah:

   k .k-1 ≡ 1 (mod q)

   5k-1 ≡ (mod 11) darik-1 = 9

2. Penerima membuktikan tanda tangan digital dengan perhitungan :

   w ≡s-1 (mod q)

   ≡7-1(mod 11) ≡ 8

   u1≡ (h x w) (mod q)

   ≡ (10 x 8) (mod 11) ≡ 3

   u2 ≡ (r x w) (mod q)

   ≡ (2 x 8) (mod 11) ≡ 5

   v ≡ ((gu1 x yu2) mod p) (mod q)

   ≡ ((33 x 25) mod 23) (mod 11)

   ≡ (864 (mod 23)) (mod 11) ≡ 13 (mod 11) ≡ 2

Karena v = r = 2, tanda tangan digital terbukti keabsahannya.

2.2.4 Elgamal

Algoritma Elgamal merupakan salah satu algoritma kriptografi kunci-publik yang dibuat oleh Taher ElGamal pada tahun 1984. Algoritma in pada umumnya digunakan

untuk digital signature, namun kemudian dimodifikasi sehingga juga bisa digunakan untuk enkripsi dan deskripsi.

ElGamal digunakan dalam perangkat lunak sekuriti yang dikembangkan oleh GNU, program PGP, dan pada sistem sekuriti lainnya. Kekuatan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.

Algoritma Elgamal tidak dipatenkan. Tetapi, algoritma ini didasarkan pada algoritma Diffie – Hellman, sehingga hak paten algoritma Diffie – Hellman juga mencakup algoritma ElGamal. Karena hak paten algoritma Diffie – Hellman berakhir pada bulan April 1997, maka algoritma ElGamal dapat diimplementasikan untuk aplikasi komersil.

Besaran-besaran yang digunakan di dalam algoritma ElGamal:

1. Bilangan prima, p (tidak rahasia)
2. Bilangan acak, g (g < p) (tidak rahasia)
3. Bilangan acak, x (x < p) (rahasia)
4. M (plainteks) (rahasia)
5. a dan b (cipherteks) (tidak rahasia)

Prosedur Membuat Pasangan Kunci

· Pilih sembarang bilangan prima p.

· Pilih dua buah bilangan acak, g dan x, dengan syarat g < p dan $1 \leq x \leq p - 2$.

· Hitung y = gx mod p.

Kunci publik adalah y, kunci rahasia adalah x. Nilai g dan p tidak dirahasiakan dan dapat diumumkan kepada anggota kelompok.

a. Enkripsi

· Plainteks disusun menjadi blok-blok m1, m2, …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam rentang 0 sampai p – 1.

· Pilih bilangan acak k, yang dalam hal ini 0 £ k £ p – 1, sedemikian sehingga k relatif prima dengan p – 1.

· Setiap blok m dienkripsi dengan rumus

a = gk mod p

b = ykm mod p

Pasangan a dan b adalah cipherteks untuk blok pesan m. Jadi, ukuran cipherteks dua kali ukuran plainteksnya.

b. Dekripsi

Untuk mendekripsi a dan b digunakan kunci rahasia, x, dan plainteks m diperoleh kembali dengan persamaan

m = b/ax mod p

Catatlah bahwa karena

ax º gkx (mod p)

Contoh

Siti ingin membangkitkan pasangan kuncinya. Siti memilih p = 2357, g = 2, dan x = 1751. Kemudian menghitung :

y = gx mod p = 21751 mod 2357 = 1185

Jadi kunci publiknya (y = 1185, g = 2, p = 2357) dan kunci privatnya (x = 1751, p = 2357).

Enkripsi

Misalkan Ahmad ingin mengirim palinteks m = 2035 (nilai m masih berada di dalam selang [0, 2357 – 1]). Ahmad memilih bilangan acak k = 1520 (nilai k masih berada di dalam selang [0, 2357 – 1]). Kemudian Ahmad menghitung

a = gk mod p = 21520 mod 2357 = 1430

b = ykm mod p = 11851520 × 2035 mod 2357 = 697

Jadi, cipherteks yang dihasilkan adalah (1430, 697). Ahmad mengirim cipherteks ini ke Siti.

Dekripsi

Siti mendeskripsi cipherteks dari Ahmad dengan melakukan perhitungan sebagai berikut :

1/ax = (ax)– 1 = a p – 1 – x mod p = 1430605 mod 2357 = 872

m = b/ax mod p = 697 × 872 mod 2357 = 2035

Plainteks yang didekripsi, 2035, sama dengan plainteks yang dikirim oleh Ahmad.

b/ax º ykm/ax

º gxkm/gxk

º m (mod p)

2.2.5 Diffie-Hellman

Diffie-Hellman key exchange adalah metode dimana subyek menukar kunci rahasia melalui media yang tidak aman tanpa mengekspos kunci. Metode ini diperlihatkan oleh Dr. W. Diffie dan Dr. M. E. Hellman pada tahun 1976 pada papernya "New Directions in Cryptography". Metode ini memungkinkan dua pengguna untuk bertukar kunci rahasia melalui media yang tidak aman tanpa kunci tambahan. Metode ini memiliki dua parameter sistem, p dan g. Kedua parameter tersebut publik dan dapat digunakan oleh semua pengguna sistem. Parameter p adalah bilangan prima, dan paramater g (sering disebut generator) adalah integer yang lebih kecil dari p yang memiliki properti berikut ini: Untuk setiap bilangan n antara 1 dan p-1 inklusif, ada pemangkatan k pada g sehingga gk = n mod p

Penggunaan Algoritma Diffie-Hellman dalam pertukaran kunci dapat dilakukan secara aman dan efektif dalam pemrosesan jika dibandingkan dengan algoritma RSA yang cenderung lebih lama dalam pemrosesan algoritmanya. Proses pertukaran kunci ini dapat dilakukan lebih dari 2 orang asal memenuhi 2 prinsip yang telah dibahas tadi. Algoritma Diffie-Hellman lebih memfokuskan dalam perubahan nilai kunci dan proses matematis dalam penentuan kunci akhir yang sama.

2.2.5.1 Cara Kerja Protokol Pertukaran Kunci Diffie Hellman

Protocol kesepakatan kunci Diffie-Hellman (Diffie-Hellman Key Agreement) membangun kunci sesi tanpa menggunakan kunci yang dibagikan sebelumnya. Pesan-pesan yang ditukarkan antara Alice dan Bob bisa dibaca oleh siapapun yang bisa menyadap/menguping(eavesdrop) dan si penyadap tidak akan bisa mengetahui kunci sesi yang akhirnya digunakan oleh Alice dan Bob. Di sisi lain, Diffie-Hellman tidak melakukan otentikasi pada pihak-pihak yang terlibat. Salah satu dari kegunaan utama dari DIffie-Hellman adalah dalam protokol Internet Key

Exchange(IKE/ Pertukaran Kunci Internet), sebuah bagian pusat dari arsitektur IP Security(IPsec)

Protokol Diffie-Hellman mempunyai dua parameter, p dan g, yang mana keduanya bersifat publik dan mungkin akan digunakan oleh semua user dalam sistem tertentu. Parameter p harus berupa bilangan prima. Bilangan-bilangan bulat mod p (singkatan dari modulo p) mempunyai jangkauan dari 0 hingga p-1, karena x mod p adalah sisa dari x dibagi p, dan membentuk apa yang para matematikawan sebut sebagai group dibawah perkalian. Parameter g (biasanya disebut sebuah generator) harus berupa sebuah akar primitive(primitive root) dari p: untuk setiap angka n dari 1 sampai p -1 pasti ada nilai k sehingga n = g^k mod p. sebagai contoh jika p adalah bilangan prima 5 (pada sistem yang sebenarnya akan digunakan angka yang jauh lebih besar), maka kita akan memilih 2 sebagai generator karena:

1 = 2^0 mod p

2 = 2^1 mod p

3 = 2^3 mod p

4 = 2^2 mod p

Misalkan Alice dan Bob ingin sepakat pada sebuah kunci simetris yang terbagi. Alice dan Bob dan orang lain sudah mengetahui nilai p dan g. Alice meng-generate sebuah nilai privat acak a dan Bob meng-generate sebuah nilai acak privat b. baik a dan b diambil dari sekumpulan integer {1,….,p-1}. Alice dan Bob menurunkan pasangan nilai publik mereka, nilai yang akan mereka kirimkan satu sama lain tanpa terenkripsi, sebagai berikut. Nilai publik dari Alice adalah :

g^a mod p

dan nilai publik Bob adalah

g^b mod p

Kemudian mereka menukar nilai pubik mereka. Akhirnya, Alice menghitung

g^ab mod p = (g^b mod p)^a mod p

dan bob menghitung

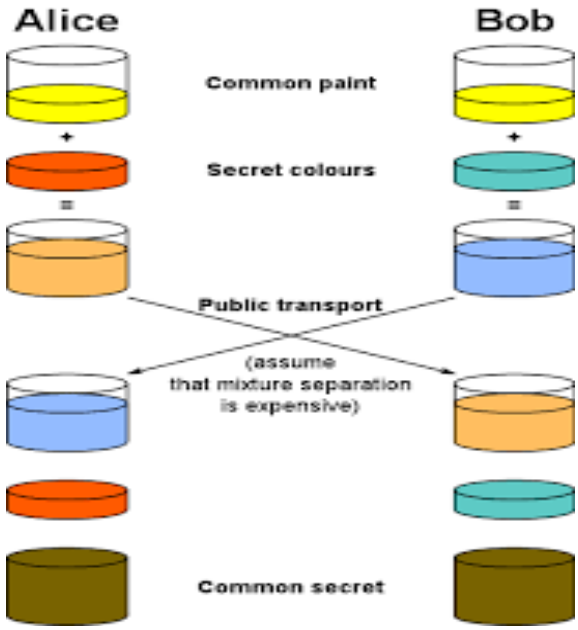
g^ba mod p = (g^a mod p)^b mod p

Alice dan Bob sekarang mempunyai gab mod p = g^ba mod p sebagai kunci simetris mereka yang dibagikan satu sama lain.Sebuah varian dari Diffie-Hellman terkadang disebut Diffie-Hellman yang sudah diperbaiki(fixed) memungkinkan otentikasi dari satu atau kedua pihak.

Mekanisme ini mengandalkan pada sertifikat yang mirip seperti sertifikat kunci publik namun men-sertifikasi parameter publik Diffie-Hellman dari sebuah pihak. Sebagai contoh, sertifikat tersebut akan menyatakan bahwa parameter Diffie-Hellman publik dari Alice adalah p, g, dan ga mod p (sebagai catanan bahwa nilai dari a akan tetap hanya diketahui Alice). Sertifikat tersebut akan meyakinkan Bob bahwa pihak lain dalam Diffie-Hellman adalah Alice atau yang pihak lain tidak akan bisa menghitung kunci rahasia, karena dia tidak tahu nilai a. jika kedua pihak mempunyai sertifikat untuk parameter Diffie-Hellman mereka, mereka bisa melakukan otentikasi satu sama lain.

Jika hanya satu pihak saja yang mempunyai sebuah sertifikat, maka hanya satu yang bisa diotentikasi. Hal ini berguna untuk beberapa situasi, sebagai contoh, ketika satu pihak adalah sebuah server dan lainnya adalah sebuah client sembarang, client bisa melakukan otentikasi web server dan membangun sebuah kunci sesi untuk kerahasiaan sebelum mengirimkan sebuah nomor kartu kredit pada web server.

Secara umum, Diffie-Hellman algoritma digambarkan seperti berikut:



Gambar 2.6 pertukaran kunci Diffie- Hellman

Algoritma pertukaran kunci Diffie- Hellman (protokol Diffie-Hellman) beguna untuk mempertukarkan kunci rahasia pada komunikasi menggunakan kriptografi simetris. Kekuatan algoritma ini adalah pada sulitnya melakukan perhitungan logaritma diskrit. Langkah-langkahnya adalah sebagai berikut,

a. Misalkan dua orang yang berkomunikasi: Alice dan Bob.
b. Mula-mula Alice menyepakati bilangan prima yang lebih besar dari bob , *n* dan *g*, sedemikian sehingga $g < n$.
c. Bilangan *n* dan *g* tidak perlu rahasia. Bahkan, Alice dan Bob dapat membicarakannya melalui saluran yang tidak aman sekalipun.

2.2.5.2 Pembangkitan kunci dengan algoritma Diffie-Hellman

1. Alice membangkitan bilangan bulat acak yang besar *x* dan mengirim hasil perhitungan berikut kepada Bob:

   $X = g^x \bmod n$

2. Bob membangkitkan bilangan bulat acak yang besar *y* dan mengirim hasil perhitungan berikut kepada Alice:

   $Y = g^y \bmod n$

3. Alice menghitung

   $K = Y^x \bmod n$

4. Bob menghitung

   $K' = X^y \bmod n$

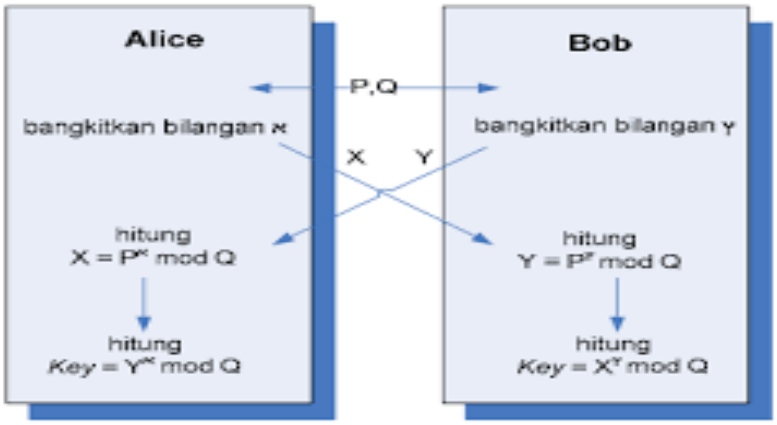
5. Jika perhitungan dilakukan dengan benar, maka

   $K = K'$.

6. Baik *K* dan *K*' sama dengan $g^{xy} \bmod n$.

7. Eve yang menyadap pembicaraan antara Alice dan Bob tidak dapat menghitung *K* . Ia hanya memiliki informasi *n*, *g*, *X* dan *Y*, tetapi ia tidak mempunyai informasi nilai *x* dan *y*.

8. Untuk mengetahui *x* atau *y*, ia perlu melakukan perhitungan logaritma diskrit, yang mana sangat sulit dikerjakan.



Gambar 2.7 Diffie- Hellman

Contoh 1 :

Alice dan Bob menyepakati $n = 97$ dan $g = 5$ ($g < n$)

1. Alice memilih $x = 36$ dan menghitung

   $X = g^x \bmod n = 5^{36} \bmod 97 = 50$

Alice mengirimkan X kepada Bob.

2. Bob memilih y = 58 dan menghitung

   $Y = g^y \bmod n = 5^{58} \bmod 97 = 44$

   Bob mengirimkan Y kepada Alice.

3. Alice menghitung kunci simetri K,

   $K = Y^x \bmod n = 44^{36} \bmod 97 = 75$

4. Bob menghitung kunci simetri K,

   $K = X^y \bmod n = 50^{58} \bmod 97 = 75$

Jadi, Alice dan Bob sekarang sudah mempunyai kunci enkripsi simetri yang sama, yaitu $K = 75$.

Contoh 2:

1. Alice dan Bob setuju untuk menggunakan modulus $p = 23$ dan basis $g = 5$ (yang merupakan akar modulo primitif 23).
2. Alice memilih bilangan bulat rahasia = **6,** kemudian mengirimkan

   Bob $A = g^{\boldsymbol{p}} \bmod$

   $A = 5^{\mathbf{6}} \bmod 23 = 8$
3. Bob memilih bilangan bulat rahasia b=15 kemudian mengirimkan Alice

   $B=g^b \bmod p$

   $B = 5^{15} \bmod 23 = 19$
4. Alice menghitung $\boldsymbol{s} = B^{\boldsymbol{p}} \bmod$

   $\boldsymbol{s} = 19^{\mathbf{6}} \bmod 23 = \mathbf{2}$
5. Bob menghitung $\boldsymbol{s} = A \bmod p^{\boldsymbol{b}}$

   $\boldsymbol{s} = 8^{\mathbf{15}} \bmod 23 = \mathbf{2}$
6. Alice dan Bob sekarang berbagi rahasia (nomor **2).**

Kedua Alice dan Bob telah tiba di nilai yang sama s, karena, di bawah mod p,

- $A^{b \bmod} p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p$
- $(g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p$

1. g = publik (prime) dasar, diketahui Alice, Bob, dan eve(penyadap). g = 5
2. p = publik (prime) modulus, diketahui Alice, Bob, dan eve(penyadap). p = 23
3. **a** = Alice kunci pribadi, yang hanya diketahui Alice. **a** = **6**
4. b=Bob kunci pribadi yang hanya diketahui Bob. b =15
5. A = Alice kunci publik, diketahui Alice, Bob, dan eve(penyadap). $A = g^{p} \bmod = 8$
6. B = Bob kunci publik, diketahui Alice, Bob, dan eve(penyadap). $B = g^{b} \bmod p = 19$

2.2.6 Algoritma Rijndael

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan cipher block. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun Rijndael mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan cipher block. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun Rijndael mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Berikut adalah perbandingan jumlah proses yang harus dilalui untuk masing-masing masukan.

Blok-blok data masukan dan kunci dioperasikan dalam bentuk array. Setiap anggota array sebelum menghasilkan keluaran ciphertext dinamakan dengan state.

Setiap state akan mengalami proses yang secara garis besar terdiri dari empat tahap yaitu, AddRoundKey, SubBytes, ShiftRows, dan MixColumns. Kecuali tahap MixColumns, ketiga tahap lainnya akan diulang pada setiap proses sedangkan tahap MixColumns tidak akan dilakukan pada tahap terakhir. Proses dekripsi adalah kebalikkan dari dekripsi. Karena terjadi beberapa tahap dalam proses enkripsi, maka diperlukan subkeysubkey yang akan dipakai pada tiap tahap.

Pengembangan jumlah kunci yang akan dipakai diperlukan karena kebutuhan subkeysubkey yang akan dipakai dapat mencapai ribuan bit, sedangkan kunci yang disediakan secara default hanya 128-256 bit. Jumla total kunci yang diperlukan sebagai subkey adalah sebanyak Nb(Nr+1), dimana Nb adalah besarnya blok data dalam satuan word. Sedangkan Nr adalah jumlah tahapan yang harus dilalui dalam satuan word. Sebagai contoh, bilamana digunakan 128 bit (4 word) blok data dan 128 bit (4 word) kunci maka akan dilakukan 10 kali proses (lihat Tabel 1). Dengan demikian dari rumus didapatkan 4(10+1)=44 word=1408 bit kunci. Untuk melakukan pengembangan jumlah kunci yang akan dipakai dari kunci utama maka dilakukan key schedule.

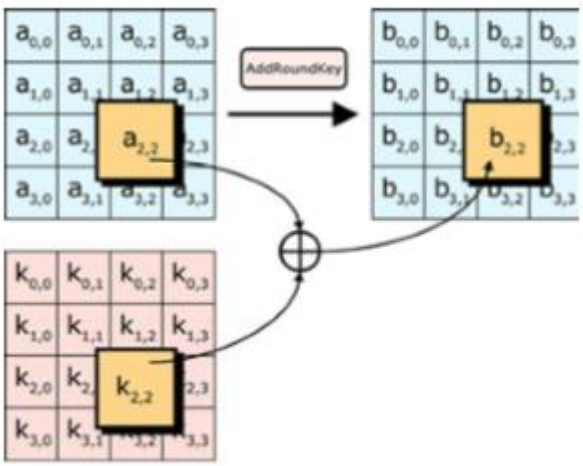
a. **Key Schedule**

Proses key schedule diperlukan untuk mendapatkan subkey-subkey dari kunci utama agar cukup untuk melakukan enkripsi dan dekripsi. Proses ini terdiri dari beberapa operasi, yaitu: • Operasi Rotate, yaitu operasi perputaran 8 bit pada 32 bit dari kunci. • Operasi SubBytes, pada operasi ini 8 bit dari subkey disubstitusikan dengan nilai dari S-Box. Operasi Rcon, operasi ini dapat diterjemahkan sebagai operasi pangkat 2

nilai tertentu dari user. Operasi ini menggunakan nilai-nilai dalam Galois field. Nilai-nilai dari Rcon kemudian akan di-XOR dengan hasil operasi SubBytes. Operasi XOR dengan w[i-Nk] yaitu word yang berada pada Nk sebelumnya.

Add Round Key Pada proses ini subkey digabungkan dengan state. Proses penggabungan ini menggunakan operasi XOR untuk setiap byte dari subkey dengan byte yang bersangkutan dari state. Untuk setiap tahap, subkey dibangkitkan dari

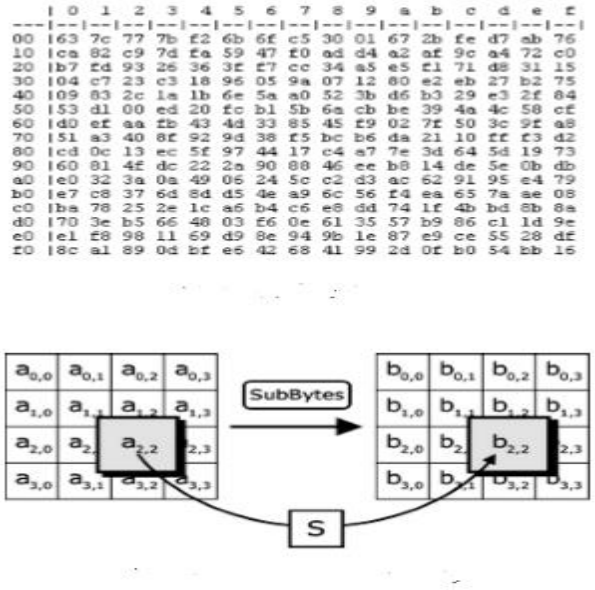
kunci utama dengan menggunakan proses key schedule. Setiap subkey berukuran sama dengan state yang bersangkutan. Proses AddRoundKey diperlihatkan pada Gambar .



Gambar 2.8 Proses add round key

b. SubBytes

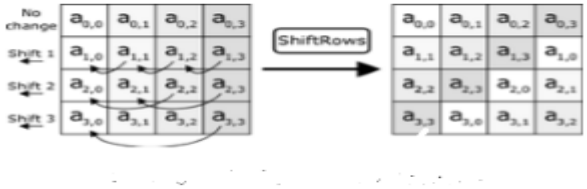
Proses SubBytes adalah operasi yang akan melakukan substitusi tidak linear dengan cara mengganti setiap byte state dengan byte pada sebuah tabel yang dinamakan tabel SBox. Sebuah tabel S-Box terdiri dari 16x16 baris dan kolom dengan masing-masing berukuran 1 byte. Tabel S-Box diperlihatkan pada Gambar 2 sedangkan proses SubBytes diperlihatkan pada Gambar .



Gambar 2.9 Subbytes

c. Shift Rows

Proses Shift Rows akan beroperasi pada tiap baris dari tabel state. Proses iniakan bekerja dengan cara memutar byte-byte pada 3 baris terakhir (baris 1, 2, dan 3) dengan jumlah perputaran yang berbeda-beda. Baris 1 akan diputar sebanyak 1 kali, baris 2 akan diputar sebanyak 2 kali, dan baris 3 akan diputar sebanyak 3 kali. Sedangkan baris 0 tidak akan diputar. Proses ShiftRows diperlihatkan pada Gambar 4.



Gambar 2.10 Shift Rows

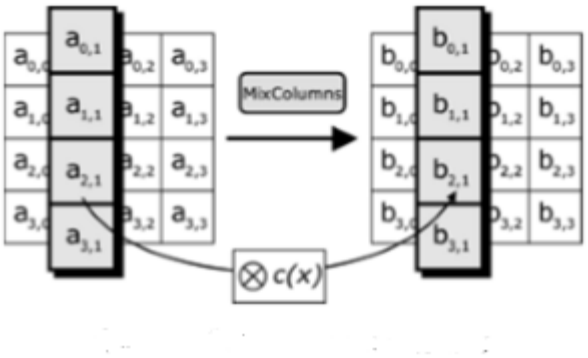
d. MixColumns

Proses MixColumns akan beroperasi pada tiap kolom dari tabel state. Operasi ini menggabungkan 4 bytes dari setiap kolom tabel state dan menggunakan transformasi linier Operasi Mix Columns memperlakukan setiap kolom sebagai polinomial 4 suku dalam Galois field dan kemudian dikalikan dengan c(x) modulo (x4+1), dimana c(x)=3x3+x2+x+2. Kebalikkan dari polinomial ini adalah c(x)=11x3+13x2+9x+14. Operasi MixColumns juga dapat dipandang sebagai perkalian matrix. Langkah MixColumns dapat ditunjukkan dengan mengalikan 4 bilangan di dalam Galois field oleh matrix berikut ini.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Atau bila dijabarkan: r 0=2a0+a3+a2+3a1 r 1=2a1+a0+a3+3a2 r 2=2a2+a1+a0+3a3 a 3=2a3+a2+a1+3a0

Operasi penjumlahan di atas dilakukan dengan operasi XOR, sedangkan operasi perkalian dilakukan dalam Galois field.



Gambar 2.11 Mix Columms

2.2.7 Bahasa Pemrograman C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET

Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis windows (desktop) dan aplikasi berbasis web serta aplikasi berbasis web services.Ada beberapa keunggulan C#, yaitu :

1. C# menghilangkan beberapa hal yang bersifat kompleks yang terdapat dalam beberapa macam bahasa pemrograman seperti Java dan C++, termasuk diantaranya mengilangkan macro, templates, multiple inheritance dan virtual base classes. Hal hal tersebut yang dapat menyebabkan kebingungan pada saat menggunakannya, dan juga berpotensial dapat menjadi masalah bagi para programmer C++. Jika anda pertama kali belajar bahasa C# sebagai bahasa pemrograman, maka hal-hal tersebut diatas tidak akan membuat waktu anda terbuang terlalu banyak untuk mempelajarinya. C# bersifat sederhana, karena bahasa ini didasarkan kepada bahasa C dan C++. Jika anda familiar dengan C dan C++ atau bahkan Java, anda akan menemukan aspekaspek yang begitu familiar, seperti statements, expression,
2. operators, dan beberapa fungsi yang diadopsi langsung dari Java, C dan C++, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana. Perhatikan Gambar 1 yang merupakan komposisi bahasa yang terdapat dalam C# sehingga kita tidak perlu takut lagi untuk mempelajari C#.



Gambar 2.12 Bahasa C#

3. beberapa fitur seperti exception handling, garbage collection, extensible data types, dan code security (keamanan kode/bahasa pemrograman). Dengan adanya fitur-fitur tersebut, menjadikan bahasa C# sebagai bahasa pemrograman yang modern.
4. Kunci dari bahasa pemrograman yang bersifat Object Oriented adalah encapsulation, inheritance, dan polymorphism. Secara sederhana, istilah-istilah tersebut bisa didefinisikan sebagai berikut
5. encapsulation, dimana semua fungsi ditempatkan dalam satu paket (single package).
6. inheritance, adalah suatu cara yang terstruktur dari suatu kode-kode pemrograman dan fungsi untuk menjadi sebuat program baru dan berbentuk suatu paket.
7. Polymorphism, adalah kemampuan untuk mengadaptasi apa yang diperlukan untuk dikerjakan. Sifat-sifat tersebut di atas, telah di miliki oleh C# sehingga bahasa C# merupakan bahasa yang bersifat Object Oriented.
8. C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, spreadsheets, atau bahkan membuat kompiler untuk sebuah bahasa permrograman.
9. C# adalah bahasa pemrograman yang menggunakan jumlah kata-kata yang tidak terlalu banyak. C# hanya berisi kata-kata yang biasa disebut dengan keywords. Keywords ini digunakan untuk menjelaskan berbagai macam informasi.

Tabel 2.1 Daftar keywords yang ada dalam bahasa C#.

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	
goto	if	implicit	in	int
interface	internal	is	lock	long
namespace	new	null	object	operator
out	override	params	private	protected
public	readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc	
static	string	struct	switch	this
throw	true	try	typeof	uint
ulong	unchecked	unsafe	ushort	using
virtual	void	while		

.

10. Kode C# ditulis dengan pembagian masing Class-Class (classes) yang terdiri dari beberapa routines yang disebut sebagai member methods. Class-Class dan metode metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka kita akan dapat membuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (reusable code).
11. Visual C#.NET mendukung pembangunan aplikasi client server, terdistribusi, serta aplikasi yang berbasiskan windows serta web.

2.2.8 OOP (Object Oriented Programming)

A. Pengertian OOP (Object Oriented Programming)

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari.

Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan,

berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

B. Konsep OOP (Object Oriented Programming)

1. Kelas Abstrak (Class Abstraksi)
2. Enkapsulasi (encapsulation)
3. Pewarisan (Inheritance)
4. Polimorfisme (polymorphism)

1. Kelas Abstrak (Class Abstraksi)

Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).

2. Enkapsulasi (encapsulation)

Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.

Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.Contoh: kelas mobil menyediakan

antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

3. Pewarisan (Inheritance)

Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

4. Polimorfisme (polymorphism)

Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.

a. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
b. Method overriding.
c. Method name overloading.
d. Karakteristik OOP (Object Oriented Programming)
e. Semua adalah objek.
f. Komputasi dilakukan dengan komunikasi antar objek. Setiap objek berkomunikasi dengan objek yang lain melalui pengiriman dan penerimaan pesan.
g. Sebuah pesan merupakan permintaan atas sekumpulan aksi dengan semua argumen yang diperlukan untuk menyelesaikan suatu tugas tertentu.
h. Setiap objek memiliki memori sendiri, yang dapat terdiri dari objek-objek lainnya.

i. Setiap objek adalah wakil atau representasi dari suatu kelas. Sebuah kelas dapat mewakili sekelompok objek yang sama.
j. Kelas merupakan kumpulan tingkah laku yang berkaitan dengan suatu objek. Jadi, semua objek yang merupakan wakil dari kelas yang sama dapat melakukan aksi yang sama pula.
k. Kelas-kelas diorganisasikan ke dalam struktur pohon yang berakar tunggal, yang dinamakan dengan jenjang pewarisan (inheritance hierarchy).
l. Setiap objek pada umumnya memiliki tiga sifat, yaitu keadaan, operasi dan identitas objek.
m. Operasi merupakan tindakan yang dapat dilakukan oleh sebuah objek.
n. Keadaan objek merupakan koleksi dari seluruh informasi yang dimiliki oleh objek pada suatu saat.
o. Informasi yang terkandung pada objek tersebut pada akhirnya memberikan identitas khusus yang membedakan suatu objek dengan objek lainnya.

2.2.8.1 Mengenal Object-Oriented Programming dengan Bahasa C#

C# dibaca C-Sharp merupakan suatu bahasa program yang menggunakan konsep Pemrograman Berorientasi Objek atau Object Oriented Programing (OOP). OOP merupakan pemrograman yang memposisikan semua bagian yang terlibat dalam program adalah objek. Program dapat dibentuk dan dijalankan melalui interaksi antara objek yang satu dengan objek yang lainnya.Beberapa hal-hal yang perlu diketahui dalam OOP, yaitu :

1. Class, merupakan pengelompokkan. Misalnya mengelompokkan antara manusia, hewan dan tumbuhan dengan class-class yang berbeda.
2. Object, merupakan isi dari suatu class yang memiliki ciri-ciri unik. Misalnya burung, kucing, sapi temasuk di dalam class hewan. Anggrek, mawar, dan melati termasuk dalam class tumbuhan.

3. Property, merupakan perlengkapan yang dimiliki oleh sebuah object. Misalnya kucing memiliki property berat, tinggi, warna bulu, dll. Fungsi property yaitu untuk menerangkan object secara jelas.
4. Method, merupakan suatu metode atau aksi yang bisa dilakukan oleh suatu object. Misalnya kucing dapat berjalan dan makan.

2.2.8.2 Mengenal NET Framework dan C#

Microsoft .Net Framework merupakan suatu lingkungan kerja untuk memudahkan pengembangan dan eksekusi berbagai macam bahasa pemrograman dan kumpulan pustaka-pustaka (library) agar dapat bekerja sama dalam menjalankankan aplikasi berbasis Windows.

Selama ini, komputer hanya dapat mengerti satu bahasa saja, yaitu bahasa mesin (assembly), namun ketika Anda mendevelop suatu aplikasi dengan bahasa .Net, kemudian dijalankan… maka komputer tidak akan memahami bahasa mesin tersebut sehingga komputer akan menampilkan pesan kesalahan (error message), hal ini disebabkan karena aplikasi yang dibangun dengan bahasa .Net ini membutuhkan suatu komponen yang berguna sebagai "penerjemah".

Hasil dari kompiler .Net sendiri bukanlah bahasa mesin pada umumnya (yang dapat dimengerti komputer). Bahasa mesin yang dihasilkan oleh kompiler .Net dinamakan "Intermediate Language (IL)". IL merupakan mesin virtual yang dikembangkan oleh Microsoft, sehingga mesin virtual inilah yang hanya dapat memahami aplikasi yang dibuat dengan bahasa .Net, dan .Net Framework sendiri yang bertugas sebagai penerjemah perintah-perintah dalam IL menjadi bahasa mesin pada umumnya.

2.2.8.3 Common Language Runtime

Fungsi Common Language Runtime (disingkat menjadi CLR) adalah untuk mengatur eksekusi kode dan menyediakan layanan yang bisa membuat pembuatan kode jauh lebih mudah. Kata "runtime" di sini berarti kode yang sedang dijalankan,

atau dengan kata lain kode yang sedang dieksekusi. Sementara itu kata “Common Language” berarti bahwa runtime ini mengatur eksekusi kode yang ditulis dalam beberapa macam jenis bahasa yang menggunakan layanan yang sama—yakni .NET Framework.

Microsoft mengembangkan C# untuk mengambil keuntungan dari CLR. Fitur C# akan bekerja secara maksimal dengan CLR. Akhirnya, bahasa Visual Basic yang populer pun juga berevolusi karena keberadaan CLR sehingga menjadi Visual Basic .NET.

Para programmer Visual Basic yang klasik memang harus mempelajari beberapa fitur yang baru untuk menggunakan fitur CLR secara keseluruhan dengan menggunakan Visual Basic .NET. Bahasa C++, juga sama-sama berevolusi, menjadi Managed C++, sehingga bahasa tersebut juga bisa dijalankan di atas CLR.

Kode yang dieksekusi oleh CLR disebut juga dengan sebutan managed code. Adalah bagian dari CLR, yang disebut dengan Common Type System (CTS) yang mendefinisikan tipe-tipe data yang harus disediakan oleh setiap bahasa agar bisa berjalan di atas CLR. Selain CTS, CLR juga memiliki bagian yang lain yang bertugas untuk mengatur fitur-fitur yang harus dimiliki oleh setiap bahasa pemrograman untuk mengembangkan program dalam managed code, yang dikenal dengan

Common Language Specification (CLS). Para programmer yang hanya menggunakan fitur-fitur yang terdapat di dalam CLS dapat membangun sebuah aplikasi dengan beberapa macam bahasa. Mereka tahu, jika memang mereka menggunakan sebuah fitur dalam sebuah bahasa pemrograman tertentu dalam program mereka, program di dalam bahasa pemrograman lainnya juga akan dapat menggunakan komponen tersebut.

2.2.8.4 Base Class Library

Komponen kedua di dalam .NET Framework adalah sesuatu yang menyediakan banyak tipe dan sangat membantu proses pengembangan perangkat lunak, yang disebut dengan Base Class Library (BCL). Pustaka tersebut menggabungkan tipe-tipe

tersebut di dalam ruang nama (namespace) yang menggabungkan tipe-tipe yang berkaitan. BCL mengandung lebih dari 100 buah ruang nama.

2.2.9 Microsoft Visual Studio

Microsoft Visual Studio merupakan salah satu Tool besutan Microsoft yang populer dipercaya oleh developer dalam mengembangkan web, aplikasi web, game, dan aplikasi mobile yang berbasis Windows. Kini dengan Visual Studio 2015, developer dapat mengembangkan aplikasi di berbagai platform seperti Windows, Windows Phone iOS, dan Android.

Dengan menggunakan Visual Studio 2015 ini developer dapat membuat aplikasi mobile yang hybrid di berbagai platform seperti iOS, Android, dan Windows hanya dengan menggunakan bahasa pemrograman JavaScript. Dalam hal ini developer dapat menggunakan kode-kode program yang sama untuk membuat aplikasi di berbagai platform sekaligus.

Untuk mencoba mengembangkan aplikasi mobile yang hybrid ini developer dapat mengunduh dan menggunakan Visual Studio Tools for Apache Cordova. Untuk proses instalasinya juga cukup mudah dan hanya tinggal mengikuti instruksi yang disediakan oleh tool ini.

Selanjutnya, developer tinggal membuat kode-kode HTML, CSS, dan JavaScript untuk pengembangan aplikasi mobile hybrid tersebut. Untuk mengetahui hasil sementara dari aplikasi yang dikembangkan pada Visual Studio ini maka developer dapat menggunakan simulator yang tersedia baik untuk Windows, Android, maupun iOS.

Selain memiliki fitur pengembangan aplikasi mobile yang hybrid, Visual Studio 2015 ini juga memiliki fitur untuk pengembangan aplikasi Native pada Windows, Android, dan iOS. Untuk pengembangan aplikasi Native sendiri, developer dapat mengunakan bahasa pemrograman C# atau C++. Tentunya hal ini juga disesuaikan dengan kebiasaan bahasa pemrograman yang digunakan oleh developer.

Apabila developer ingin mengembangkan aplikasi Native multiplatform yang menggunakan bahasa pemrograman C# maka dapat menggunakan Mobile Apps in Visual Studio with Xamarin. Sedangkan bagi developer yang lebih familiar menggunakan C++ untuk mengembangkan aplikasi Native multiplatform maka dapat mencoba Visual C++ Cross-Platform Mobile.Tak ketinggalan, bagi developer yang biasa mengembangkan game multiplatform menggunakan Unity maka Visual Studio 2015 ini telah mendukung untuk pemrograman yang dapat dikombinasikan dengan Unity Game Engine sehingga akan lebih mempermudah developer dalam mengembangkan game yang multiplatform.

Selain itu, Visual Studio 2015 ini juga telah menyediakan Visual Studio Emulator for Android dengan dukungan Hyper-V dan arsitektur x86 sehingga developer dapat mencoba secara cepat aplikasi mobile Android yang sedang dikembangkan dalam emulator android ini.

2.2.10 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal kepopulerannya disebabkan *MySQL* menggunakan *SQL* (*Structured Query Language*) sebagai bahasa dasar untuk mengakses *database*-nya. Selain itu *software* ini bersifat *open source* sehingga tidak perlu membayar untuk menggunakannya. *MySQL* termasuk jenis *RDBMS* (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada *MySQL*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

Untuk mengelola *database MySQL* ada beberapa cara yaitu melalui *prompt DOS* (*tool command line*) dan dapat juga menggunakan program *utility* seperti:

1. PHP MyAdmin
2. MySQL GUI
3. MySQL Manager Java Based
4. MySQL Administrator for windows.

2.2.11 UML

Menurut (Chanoles, 2003 : bab 1) *Unifield Modeling Language* atau UML adalah sebuah bahasa yang memiliki *sintaks* dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan – aturan yang harus di ikuti. Bagaimana elemen pada model – model yang kita buat berhubungan satu dengan yang lainya harus mengikuti stadar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. UML diaplikasikan untuk maksud tertentu, biasanya antar lain untuk :

a. Merancang perangkat lunak.
b. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
c. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
d. Mendokumentasikan sistem yang ada, proses – proses dan organisasinya.

Pada bulan Oktober 1994, Jim Rumbaugh, penemu notasi *Object Modelling Technique* (OMT) dan GradyBooch, penemu *Booch Method* (Metode Booch) bersama–sama menyamakan notasi mereka. Di tahun yang sama Ivan Jacobson (penemu *object method*) ikut bergabung. ketiga orang tersebut, sering disebut "*three amigos*", bersama–sama membangun notasi standar OOP untuk *software Rational Rose.*

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.

   Usecase diagram adalah diagram usecase yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Diagram usecase tidak menjelaskan secara detail tentang penggunaan usecase, namun hanya memberi gambaran singkat hubungan antara usecase, aktor, dan sistem. Melalui diagram usecase dapat diketahui fungsi-fungsi apa saja yang ada pada sistem

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Gambar 2.13 use case

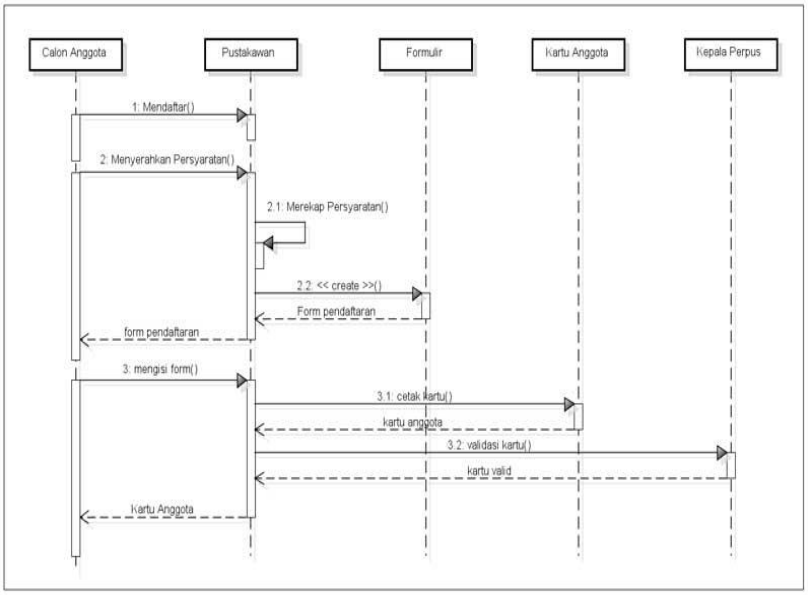
2. *Conceptual Diagram*

   untuk memodelkan konsep-konsep yang ada di dalam aplikasi.

3. *Sequence Diagram*

   sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirimdan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-objek

yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2.14 ***Sequence Diagram***

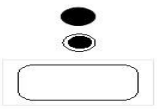
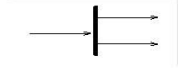
1. *Collaboration Diagram*

   dipakai untuk memodelkan interaksi antar object di dalam sistem.[4] Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh object.

2. *State Diagram*

   Menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu obyek pada sistem sebagai akibat dari stimulans yang diterima, state chart diagram digunakan untuk memodelkan behavior/methode (lifecycle) sebuah kelas atau object, Memperlihatkan urutan kejadian sesaat (state) yang dilalui sebuah object, transisi dari sebuah state ke state lainnya

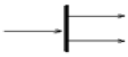
.

Simbol	Keterangan
	Start Point End Point Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Gambar 2.15 State Diagram

3. *Activity Diagram*

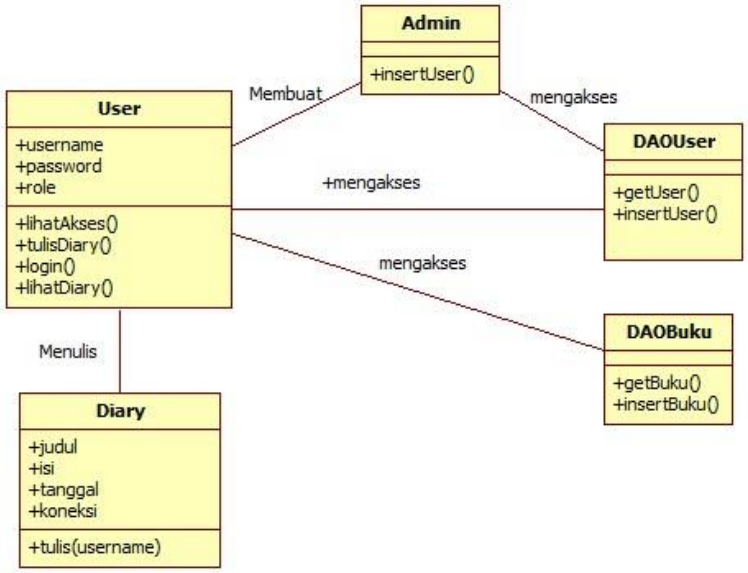
adalah diagram yang menggambarkan worlflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Start State*	Bagaimana objek dibentuk atau diawali
4		*End State*	Bagaimana objek dibentuk dan dihancurkan
5		*State Transitio*	State transition menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
6		*Fork*	Percabangan yang menunjukkan aliran pada *Activity Diagram*
7		*Join*	Penggabungan yang menjadi arah aliran pada *Activity Diagram*
8			Pilihan untuk mengambil keputusan
9		*Flow Final*	Aliran akhir

Gambar 2.16 Activity Diagram

4. *Class Diagram*

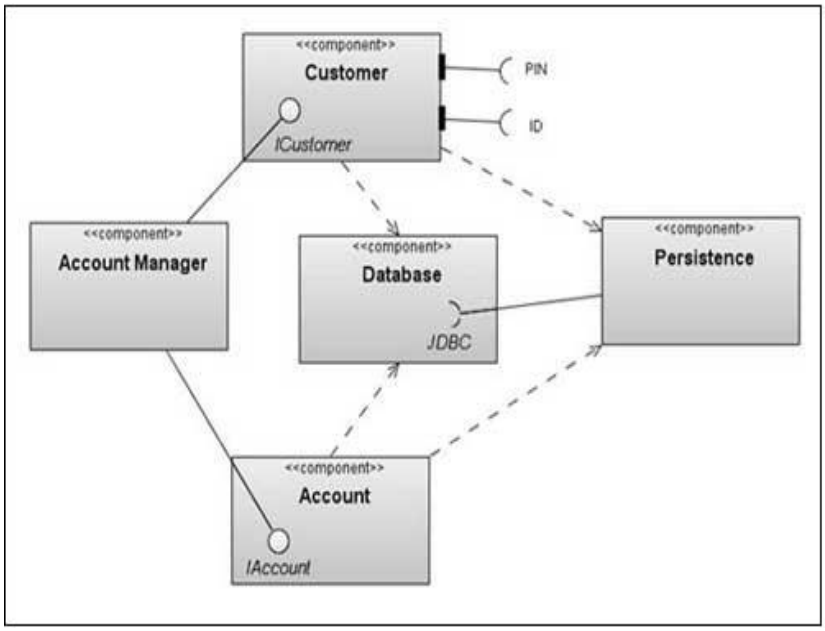
Class diagram adalah model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class. Class diagram mirip ER-Diagram pada perancangan database, bedanya pada ER-diagram tdk terdapat operasi/methode tapi hanya atribut. Class terdiri dari nama kelas, atribut dan operasi/methode.



Gambar 2.17 Class Diagram

5. *Component Diagram*

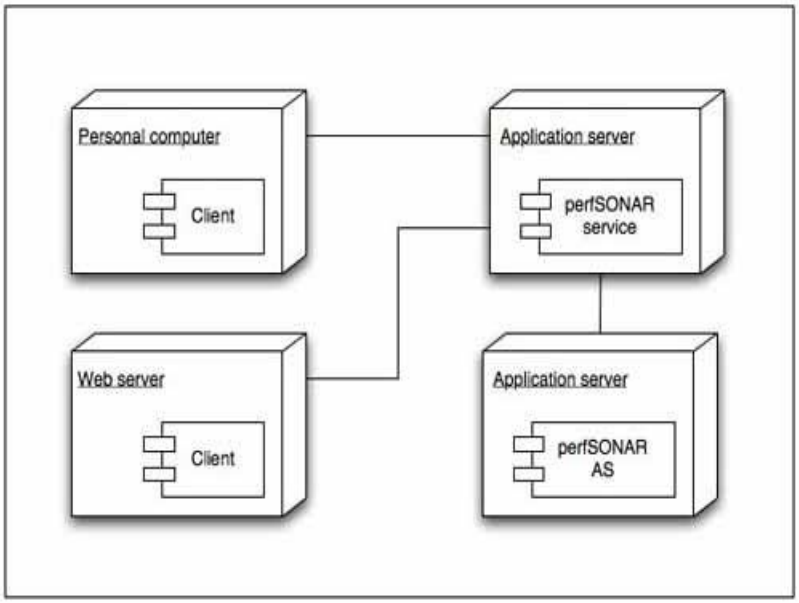
Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan softwere pada suatu sistem. Component diagram merupakan penerapan softwere dari satu ataupun lebih class, dan biasanya berupa file data atau .exe, source kode, table, dokumen dsb.



Gambar 2.18 ***Component Diagram***

6. *Deployment Diagram*

Deployment diagram yaitu salah satu diagram pada UML yang menunjukan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian softwere yang terdapat pada hardwere dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen hardwere. Jadi Deployment diagram intinya untuk menunjukan letak softwere pada hardwere yang digunakan sistem..



Gambar 2.19 ***Deployment Diagram***

2.2.12 Aplikasi

Aplikasi berasal dari kata application yaitu bentuk benda dari kata kerja to apply yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja, dan pemutar media. Kumpulan aplikasi komputer yang digabung menjadi suatu paket biasanya disebut paket atausuite aplikasi (application suite).

Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Umumnya aplikasi-aplikasi tersebut memiliki kemampuan untuk saling berinteraksi sehingga menguntungkan pemakai. Contohnya, suatu lembar kerja dapat dimasukkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan printah-printah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapka

Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Pengertian aplikasi menurut para ahli adalah sebagai berikut :

a. Menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.
b. Menurut Kamus Kamus Besar Bahasa Indonesia (1998 : 52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atauketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.
c. Menurut Rachmad Hakim S, Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur Windows &, permainan (game), dan sebagainya.
d. Menurut Harip Santoso, Aplikasi adalah suatu kelompok file (form, class, rePort) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset, dan lain-lain.

2.2.13 XAMPP

Xampp merupakan sebuah tool yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP, kita tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP, dan mySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikan secara otomatis..

Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkanya XAMPP anda dapat mendownload langsung dari web resminya. Dan berikut beberapa definisi program lainnya yang terdapat dalam XAMPP.

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi seperti (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

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.

PHP: Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memrogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (World Wide Web). phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (fields), relasi (relations), indeks, pengguna (users), perijinan (permissions), dan lain-lain).Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara

mengetikkan baris-baris perintah yang sesuai (command line) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (database), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

phpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Untuk membukanya, buka browser lalu ketikkan alamat http://localhost/phpmyadmin, maka akan muncul halaman phpMyAdmin. Di situ nantinya seseorang bisa membuat (create) basis data baru, dan mengelolanya.