BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan perusahaan yang dijelaskan meliputi profil Badan Geologi, sejarah Badan Geologi, logo Badan Geologi, visi dan misi, struktur organisasi Sekretariat Badan Geologi, deskripsi tugas.

2.1.1 Profil Badan Geologi Bandung

Badan Geologi merupakan salah satu unit teknis di bawah Kementerian Energi dan Sumber Daya Mineral yang berlokasi di Jl. Diponegoro 57, Bandung 40122. Badan Geologi mempunyai tugas yaitu melaksanakan penelitian dan pelayanan di bidang geologi, Badan Geologi juga mempunyai tugas yaitu penyusunan kebijakan teknis, rencana dan program penelitian dan pelayanan di bidang geologi, Pelaksanaan penelitian dan pelayanan di bidang geologi, Pemantauan, evaluasi dan pelaporan pelaksanaan penelitian dan pelayanan di bidang geologi dan pelaksanaan administrasi Badan Geologi. [2]

Badan Geologi terdiri dari unit-unit Eselon II dibawahnya, yaitu:

1. Pusat Survei Geologi
2. Pusat Sumber Daya Geologi
3. Pusat Sumber Daya Air Tanah dan Geologi Lingkungan
4. Pusat Vulkanologi dan Mitigasi Bencana Geologi

Kelembagaan ini terbentuk antara tahun 2005-sekarang bernama Badan Geologi. Badan Geologi menyimpan sebagian besar dokumen hasil penyelidikan mineral dan geologi dari berbagai pelosok wilayah Indonesia, berupa pustaka (laporan, terbitan, peta) dan percontoh (batuan, mineral, fosil). Selain menyimpan dokumen hasil penyelidikan dan pemetaan geologi, juga mewarisi dan merawat semua dokumen hasil penyelidikan dan pemetaan geologi dan bahan tambang yang

dilakukan oleh lembaga-lembaga pendahulunya, mulai dari Dienst van het Mijnwezen (1850-1922) sampai dengan Puslitbang Geologi (1978-2005).

Sesuai dengan Peraturan Menteri ESDM no 18 tahun 2010 Badan Geologi memiliki Tugas dan Fungsi sebagai berikut.

A. TUGAS

Melaksanakan penelitian dan pelayanan di bidang geologi

B. FUNGSI

1. Penyusunan kebijakan teknis, rencana dan program penelitian dan pelayanan di bidang geologi
2. Pelaksanaan penelitian dan pelayanan di bidang geologi
3. Pemantauan, evaluasi dan pelaporan pelaksanaan penelitian dan pelayanan di bidang geologi dan
4. Pelaksanaan administrasi Badan Geologi.

2.1.2 Sejarah Badan Geologi

Sejarah Instansi Kelembagaan Badan Geologi Bandung Departemen Energi dan Sumber Daya Mineral mulai dari tahun 1850-2005. Selengkapnya dapat dilihat pada tabel 2.1.

Tabel 2.1 Sejarah Badan Geologi

Tahun	PEMERINTAHAN/ KEMENTERIAN	LEMBAGA	UNIT-UNIT INSTANSI
1850 - 1922	Pemerintahan Penjajahan Belanda	*Dients van het Mijnwezen*	Mijnwezen
1922 – 1942	Pemerintahan Penjajahan Belanda	*Dients van den Mijnbow*	a. *Opsporingdienst* (Dinas Penyelidikan Geologi) b. *Grondpeilwezen* (Dinas Pendugaan Tanah) c. *Volcanologische Onderzoek* (Dinas Penjagaan Gunungapi) d. D*ienst der Mijnverordeningen* e. Laboratorium Kimia dan Paleontologi

Tahun	PEMERINTAHAN/ KEMENTERIAN	LEMBAGA	UNIT-UNIT INSTANSI
1942 - 1945	Pemerintah Penjajahan Jepang	*Kogyo Zimusho* --> *Chisitsu Chosayo*	a. *Chisitsu Kakari* (Bag. Perpetaan) b. *Kosan Kakari* (Bag. Gunungapi) c. *Seizu Kakari* (Bag. Kartografi)
1945 - 1946	Kemerdekaan RI, Kementerian Pekerjaan Umum	Pusat Jawatan Tambang dan Geologi	a. Bagian Geologi b. Bagian Laboratorium c. Bagian Perusahaan
1946 - 1947	Kementerian Kemakmuran	Pusat Jawatan Tambang dan Geologi	a. Bagian Geologi b. Bagian Geoteknik c. Bagian Laboratorium d. Bagian Hukum dan Inspektur Tambang e. Bagian Perusahaan
1947 - 1949	Kementerian Perekonomian	Kementerian Muda Kemakmuran --> Pusat Jawatan Tambang dan Geologi	a. Bagian Geologi b. Bagian Geologi Teknik c. Bagian Laboratorium d. Bagian Pendidikan e. Bagian Statistik dan Dokumentasi f. Bagian Hukum dan Inspektur Tambang
1949 - 1950	Republik Indonesia Serikat (RIS)	a. Kementerian Kemakmuran --> Pusat Jawatan Tambang b. Kementerian Perdagangan dan Perindustrian --> Jawatan Tambang dan Geologi	Jawatan Tambang dan Geologi
1950 - 1952	Kementerian Perekonomian	Jawatan Pertambangan RI	Pusat Jawatan Pertambangan dan Jawatan Tambang dan Geologi
1952 - 1956	Kementerian Perekonomian	Jawatan Pertambangan RI dipecah menjadi: a. Jawatan Pertambangan b. Jawatan Geologi	Jawatan Geologi
1956 - 1957	Kementerian Perekonomian	Jawatan Geologi diubah menjadi Pusat Jawatan Geologi	Pusat Jawatan Geologi

Tahun	PEMERINTAHAN/ KEMENTERIAN	LEMBAGA	UNIT-UNIT INSTANSI
1957 - 1959	Kementerian Perekonomian dipecah menjadi: a. Kementerian Perindustrian b. Kementerian Perdagangan	Kementerian Perindustrian --> Pusat Jawatan Geologi diubah menjadi Jawatan Geologi	Jawatan Geologi
1959 - 1963	Kementerian Perindustrian dipecah menjadi: a. Departemen Perindustrian Rakyat b. Departemen Perdagangan dan Pertambangan (Depperdatam)	Depperdatam: a. Jawatan Pertambangan --> Biro Urusan Perusahaan Tambang Negara b. Jawatan Geologi	Jawatan Geologi: a. Bagian Geologi Umum b. Bagian Geologi Ekonomi c. Bagian Geologi Teknik dan Pemboran d. Bagian Geohidrologi e. Bagian Geofisika f. Bagian Geokimia g. Bagian Teknik Umum
1963 - 1966	Depperdatam	Jawatan Geologi diubah menjadi Direktorat Geologi	Direktorat Geologi: a. Dinas Perpetaan b. Dinas Geologi Ekonomi c. Dinas Geologi Teknik dan Hidrogeologi d. Dinas Gunung Berapi e. Dinas Geofisika f. Lab. Paleontologi, Petrologi dan Foto Geologi g. Subdit Pemboran h. Subdit Kimia Mineral i. Bagian Publikasi dan Informasi
1966 - 1974	Depperdatam dipecah menjadi: a. Departemen Perindustrian b. Departemen Pertambangan	Departemen Pertambangan --> Ditjen Pertambangan: a. Direktorat Bina Sarana Usaha Tambang b. Direktorat Geologi c. Akademi Geologi dan Pertambangan (AGP)	Direktorat Geologi: a. Dinas Perpetaan b. Dinas Eksplorasi c. Dinas Vulkanologi d. Dinas Geologi Teknik dan Hidogeologi e. Bagian Laboratorium dan Dokumentasi
1974 - 1978	Departemen Pertambangan diubah menjadi Departemen Pertambangan dan Energi (Deptamben)	Organisasi Direktorat Geologi tidak mengalami perubahan	

Tahun	PEMERINTAHAN/ KEMENTERIAN	LEMBAGA	UNIT-UNIT INSTANSI
1978 - 1984	Deptamben	Direktorat Jenderal Pertambangan Umum	a. Direktorat Teknik Pertambangan b. Direktorat Pembinaan Pengusahaan Pertambangan c. Pusbang Teknologi Mineral d. Puslitbang Geologi e. Direktorat Sumber Daya Mineral f. Direktorat Geologi Tata Lingkungan g. Direktorat Vulkanologi
1984 - 1992	Deptamben	Reorganisasi Deptamben dan pembentukan: a. Ditjen Pertambangan Umum b. Ditjen Geologi	Ditjen Geologi: a. Puslitbang Geologi b. Pusat Pengembangan Geologi Kelautan c. Direktorat Sumber Daya Mineral d. Direktorat Geologi Tata Lingkungan e. Direktorat Vulkanologi
1992 - 2001	Deptamben diubah menjadi Departemen Energi dan Sumber Daya Mineral (DESDM)	Ditjen Geologi diubah menjadi Ditjen Geologi dan Sumber Daya Mineral	Ditjen Geologi dan Sumber Daya Mineral: a. Puslitbang Geologi b. Puslitbang Geologi Kelautan c. Direktorat Sumber Daya Mineral d. Direktorat Geologi Tata Lingkungan e. Direktorat Vulkanologi

Tahun	PEMERINTAHAN/ KEMENTERIAN	LEMBAGA	UNIT-UNIT INSTANSI
2001 - 2005	DESDM	Reorganisasi DESDM dengan penggabungan Ditjen Pertambangan Umum ke Ditjen Geologi dan Sumber Daya Mineral, serta pembentukan Badan Litbang ESDM dan Badan Diklat ESDM	Ditjen Geologi dan Sumber Daya Mineral: a. Direktorat Inventarisasi Sumber Daya Mineral b. Direktorat Pengusahaan Mineral dan Batubarahi c. Direktorat Teknik Mineral dan Batubara d. Direktorat Tata Lingkungan Geologi & Kawasan Pertambangan e. Direktorat Vulkanologi dan Mitigasi Bencana Geologi Badan Litbang ESDM: a. Puslitbang Geologi b. Puslitbang Geologi Kelautan c. Puslitbangtek Migas ”Lemigas” d. Puslitbang Tekmira e. Puslitbangtek Energi dan Ketenagalistrikan Badan Diklat ESDM: a. Pusdiklat Geologi b. Pusdiklat Migas c. Pusdiklat Tekmira d. Pusdiklat Energi dan Ketenagalistrikan
2005 - sekarang	DESDM	Reorganisasi DESDM dengan pembentukan Ditjen Mineral, Batubara dan Panas Bumi (khusus menangani pengusahaan pertambangan umum) dan Badan Geologi	Badan Geologi: a. Pusat Survei Geologi b. Pusat Sumber Daya Geologi c. Pusat Lingkungan Geologi d. Pusat Vulkanologi dan Mitigasi Bencana Geologi Balitbang ESDM: a. Puslitbang Geologi Kelautan b. Puslitbangtek Migas ”Lemigas” c. Puslitbang Tekmira d. Puslitbangtek Ketenagalistrikan dan Energi Baru Terbarukan Badan Diklat ESDM: a. Pusdiklat Geologi b. Pusdiklat Tekmira c. Pusdiklat Migas d. Pusdiklat Ketenagalistrikan dan Energi Baru Terbarukan

2.1.3 Logo Badan Geologi

Badan Geologi merupakan salah satu unit eselon I di lingkungan Kementerian Energi dan Sumber Daya Mineral (ESDM) yang bertugas memberikan pelayanan informasi geologi. Kementerian Energi dan Sumber Daya Mineral secara penuh memegang 4 Direktorat, yaitu Direktorat Jenderal Minyak dan Gas Bumi, Direktorat Jenderal Listrik dan Pemanfaatan Energi, Direktorat Jenderal Mineral, Batubara dan Panas Bumi dan Direktorat Jenderal Badan Geologi. Logo dari Direktorat Jenderal Badan Geologi Bandung dapat dilihat pada gambar 2.1. [2]



Gambar 2.1 Logo Badan Geologi

Penjelasan lambang dan arti dari gambar logo di atas dapat dilihat ditabel Logo Badan Geologi dibawah ini.

Tabel 2.2 Logo Badan Geologi

Lambang	Arti
Tower yang berdiri tegak sebagai lambang listrik dan pemanfaatan energi	Mengawasi perkembangan yang terjadi di Direktorat Jenderal Listrik dan Pemanfaatan Energi
Lambang air menunjukan lambang untuk minyak dan gas bumi	Mengawasi perkembangan yang terjadi di Direktorat Jenderal Minyak dan Gas Bumi
Lambang tambang yang bersilang menunjukkan lambang untuk pemanfaatan mineral, batubara dan panas bumi	Mengawasi perkembangan yang terjadi di Direktorat Jenderal Mineral, Batubara dan Panas Bumi

Lambang	Arti
Lambang dua petir diatas lambang lainnya menunjukkan kejadian-kejadian alam yang terjadi, dibawah pengawasan Badan Geologi	Mengawasi perkembangan yang terjadi di Direktorat Jenderal Badan Geologi

2.1.4 Visi dan Misi

Badan Geologi adalah sebuah instansi perusahaan yang bergerak dalam bidang energi dan sumber daya mineral dalam menjalankan organisasinya mempunyai visi dan misi.

a. Visi

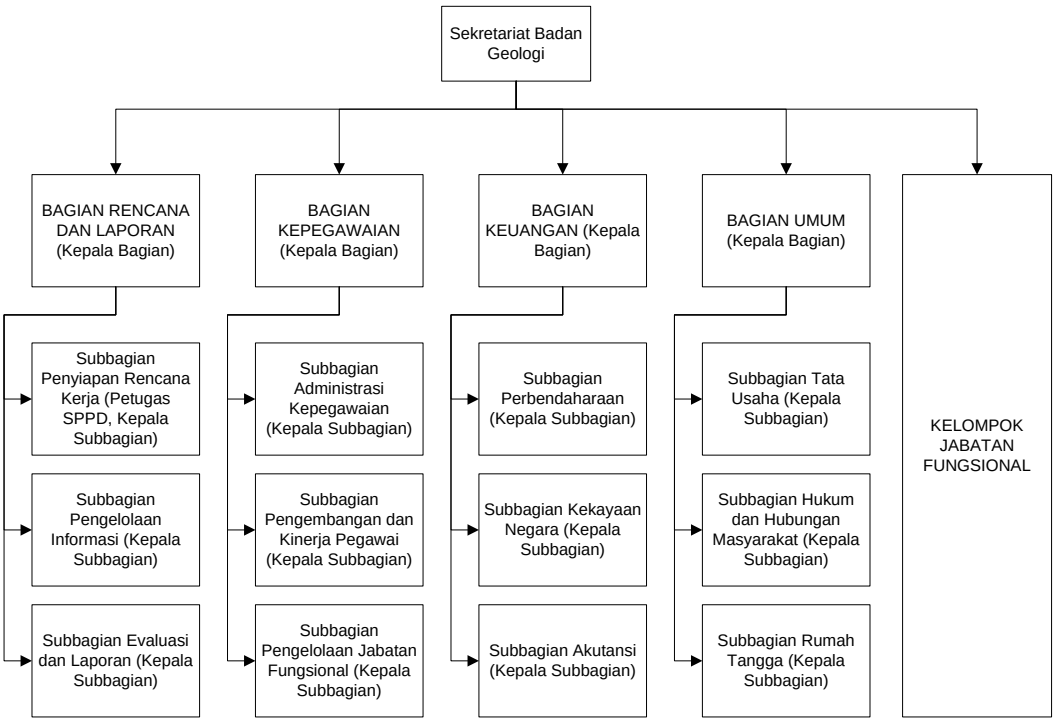
Geologi untuk perlindungan dan kesejahteraan masyarakat

b. Misi

1. Mempromosikan geologi untuk kepentingan perencanaan dan penataan wilayah.
2. Mengungkap potensi geo-resources (sumber daya geologi): migas, panas bumi, batubara, mineral dan air tanah serta potensi geologi lainnya.
3. Mengungkap potensi bencana geologi bagi kepentingan perlindungan manusia dan potensi ekonomi.
4. Mendorong penerapan geo-sciences bagi kepentingan konservasi geo-resources dan potensi geologi lainnya serta perlindungan lingkungan.

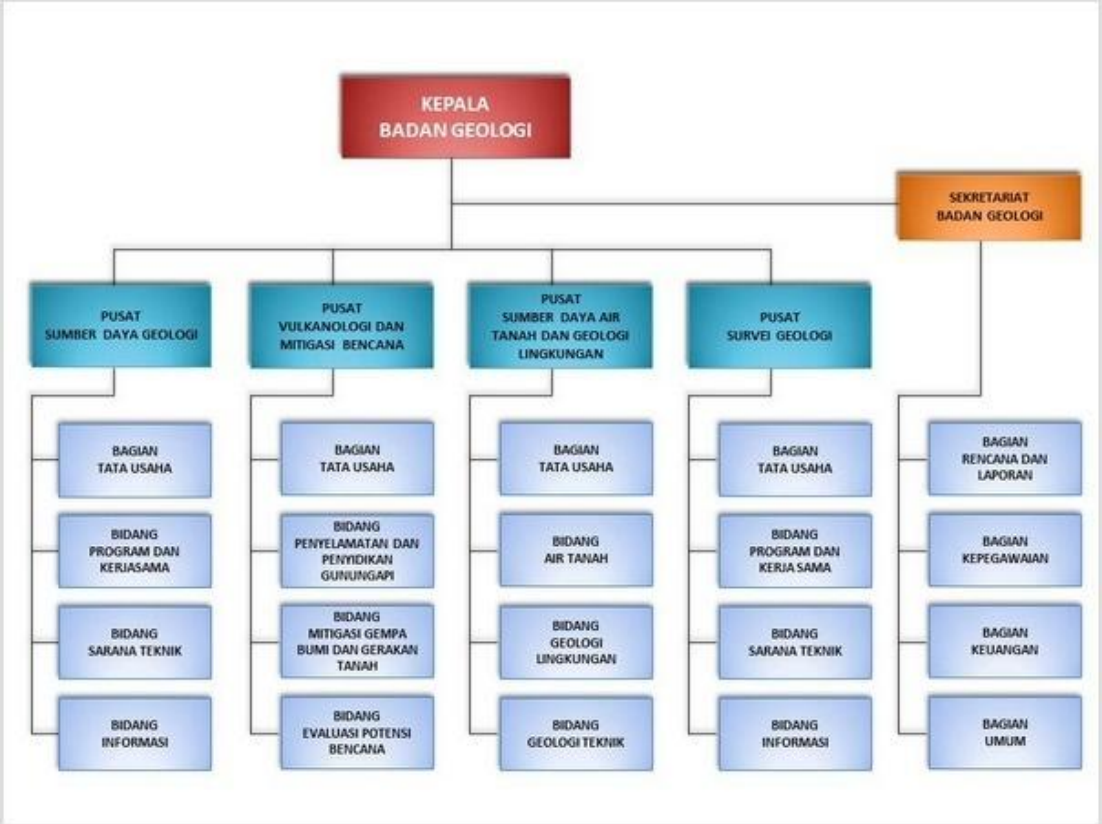
2.1.5 Struktur Organisasi Sekretariat Badan Geologi

Kementerian energi dan Sumber Daya Mineral secara penuh memegang 4 Direktorat, yaitu Direktorat Jenderal Minyak dan Gas Bumi, Direktorat Jenderal Listrik dan Pemanfaatan Energi, Direktorat Jenderal Mineral, Batubara dan Panas Bumi dan Direktorat Jenderal Badan Geologi. Sekretariat Badan Geologi Bandung merupakan bagian dari Direktorat Badan Geologi Bandung. Struktur Organisasi Direktorat Badan Geologi Bandung dapat dilihat pada Gambar 2.2.



Gambar 2.2 Struktur Organisasi Sekretariat Badan Geologi

Adapun struktur organisasi Badan Geologi secara umum adalah sebagai berikut:



Gambar 2.3 Struktur Organisasi Umum Badan Geologi

2.1.6 Deskripsi Tugas

Penjelasan mengenai deskripsi tugas dari masing-masing subbagian di Sekretariat Badan Geologi Bandung diantaranya adalah Bagian Rencana dan Laporan, Bagian Kepegawaian, Bagian Keuangan, dan Bagian Umum.

1. Bagian Perencanaan dan Laporan

Bagian Perencanaan mempunyai tugas melaksanakan koordinasi penyusunan rencana, program dan anggaran, serta evaluasi dan penyusunan laporan departement.

Dalam melaksanakan tugas, Biro Perencanaan menyelenggarakan fungsi:

a. Penyiapan dan penyusunan rencana dan program departemen.
b. Penyiapan dan penyusunan anggaran departemen.

c. Penyiapan dan penyusunan rencana dan program kerjasama lintas sektoral dan daerah.
d. Pelaksanaan monitoring, evaluasi dan penyusunan laporan departemen

Bagian Rencana dan Laporan mempunyai tugas:

a. Menyusun rencana kegiatan Sub Bagian berdasarkan data dan program Sekretariat dan ketentuan perundang-undangan yang berlaku sebagai pedoman kerja.
b. Memimpin dan mengkoordinasikan bawahan agar pelaksanaan tugas berjalan dengan harmonis dan saling mendukung sesuai peraturan perundang-undangan yang berlaku.
c. Memberi petunjuk dan bimbingan kepada bawahan agar dalam pelaksanaan tugas sesuai dengan yang diharapkan.
d. Mengatur dan mendistribusikan tugas kepada bawahan sesuai dengan bidang tugas dan permasalahannya.
e. Memeriksa dan mengevaluasi hasil kerja bawahan dengan cara mencocokkan dengan petunjuk kerja yang diberikan dan ketentuaan yang berlaku agar tercapai kesesuaian dan kebenaran hasil kerja.
f. Menilai prestasi kerja bawahan berdasarkan hasil yang dicapai sebagai bahan pertimbangan dalam peningkatan kerier.
g. Menyusun program/rencana kerja, mengumpulkan, mengolah dan membuat laporan kegiatan badan.
h. Menyusun rencana anggaran Badan yang meliputi pembuatan DPA dan RKA.
i. Melaksanakan kegiatan kehumasan, keprotokolan, dokumentasi dan perpustakaan.
j. Menghimpun dan menyusun Rencana Strategis (RENSTRA) dan RENDJA Badan Kepegawaian Daerah.
k. Membuat laporan bulanan,triwulan semester.

l. Menghimpun dan menyusun bahan pembuatan laporan akutabilitas instansi pemerintah (LAKIP)
m. Mengevaluasi pelaksanaan kegiatan Sub Bagian Perencanaan secara keseluruhan.
n. Membuat laporan kegiatan di bidang tugasnya sebagai bahan informasi dan pertanggung jawaban kepada atasan.
o. Melaksanakan tugas kedinasan lain yang diberikan atasan.

Sub Bagian Perencaaan dimimpin oleh seorang kepala Sub Bagian yang berada dibawah dan bertanggung jawab kepada sekertaris badan.

2. Bagian Kepegawaian

Bagian Kepegawaian mempunyai tugas melaksanakan urusan kepegawaian di lingkungan Badan.

Bagian Kepegawaian menyelenggarakan fungsi:

a. Pelaksanaan pengembangan kepegawaian, analisis kebutuhan, dan penyaringan pegawai dalam rangka pendidikan dan pelatihan serta ujian jabatan
b. Pelaksanaan administrasi dan koordinasi pengembangan jabatan fungsional
c. Penyiapan bahan penyusunan formasi, tata usaha, dokumentasi, statistik dan kesejahteraan pegawai
d. Pemantauan dan evaluasi implementasi kepatuhan internal pegawai di lingkungan Badan.

Bagian Kepegawaian terdiri dari:

a. Subbagian Administrasi Kepegawaian mempunyai tugas melakukan pengadministrasian dan koordinasi pengembangan pegawai.
b. Subbagian Pengembangan dan Kinerja Pegawai mempunyai tugas melakukan urusan mutasi kepegawaian, analisis kebutuhan, penyaringan dan pengusulan pegawai dalam rangka pendidikan dan pelatihan serta ujian jabatan.
c. Subbagian Pengelolaan Jabatan Fungsional mempunyai tugas melakukan pengadministrasian dan koordinasi pengembangan jabatan fungsional

3. Bagian Keuangan

Bagian Keuangan mempunyai tugas melaksanakan urusan keungan di lingkungan Badan.

Bagian Keuangan mempunyai tugas :

a. Menyusun rencana kegiatan Bagian Keuangan, berdasarkan data dan program yang ditetapkan oleh Asisten , Administrasi Umum serta ketentuan peraturan perundang-undangan yang berlaku sebagai pedoman kerja
b. Memimpin dan mengkoordinasikan bawahan agar pelaksanaan tugas berjalan dengan harmonis dan saling mendunkung sesuai peraturan perundang-undangan yang berlaku
c. Memberi petunjuk dan bimbingan kepada bawahan agar dalam pelaksanaan tugas sesuai dengan yang diharapkan
d. Mengatur dan mendistribusikan tugas kcpada bawahan sesuai dengan bidang tugas dan permasalahannya
e. Memeriksa dan mengevaluasi hasil kerja bawahan dengan cara mencocokkan dengan petunjuk kerja yang diberikan dan kctentuan yang berlaku agar tercapai kesesuaian dan kebenaran hasil kerja
f. Menilai prestasi kerja berdasarkan hasil yang dicapai sebagai bahan pertimbangan dalam peningkatan karier
g. Menyusun rancangan APBD dsn rancangan perubahan APBD
h. Mengendalikan pelaksanaan APBD
i. Memberikan petunjuk teknis pelaksanaan sistim penerimaan dan pengeluaran kas daerah
j. Melaksanakan pengolahan keuangan daerah sesuai dengan kebijakan yang ditetapkan oleh bupati serta ketentuan perundang – undangan yang berlaku
k. Melaksanakan pembayaran berdasarkan permintaan pengguna anggran sesuai dengan ketentuan dan perundang – undangan yang berlaku
l. Melaksakan sistim akutansi dan pelaporan keuangan daerah

m. Melakukan pembayaran gaji pegawai administrasi perjalanan dinas dan mengontrol pengawasan keuangan bagi belanja pegawai, serta pembelian alat-alat tulis kantor
n. Memeriksa penyelenggaran administrasi keuangan tentang belanja pegawai agar sesuai dengan ketentuan dan perundang-undangan yang berlaku
o. Mengoreksi SPJ pembayaran gaji dan SKPP pegawai yang pindah agar tertib administrasi dan kelancaran pelaksanaannya selalu terpelihara dengan baik
p. Mengoreksi realisasi pelaksanaan anggaran perjalanan dinas pegawai sesuai dengan ketentuan yang berlaku sehingga perjalanan dinas berlangsung efektif dan efesien
q. Meneliti bukti tagihan pihak swasta kepada pemerintah kabupaten dan menyetujui pembayaran bila telah memenuhi ketentuan yang berlaku
r. Mengevaluasi pelaksanaan kegiatan bagian keuangan, serta keselururhan
s. Membuat laporan kegiatan di bidang tugasnya, sebagai bahan informasi dan pertanggungjawaban kepada atasan
t. Melaksanakan tugas kedinasan lain yang diberikan kepada atasan

4. Bagian Umum (Kelompok Jabatan Fungsional)

Bagian umum mempunyai tugas untuk melaksanakan urusan di lingkungan Badan.

Bagian Umum (Kelompok Jabatan Fungsional) memiliki tugas :

a. Kepala Badan membentuk kelompok program penelitian dan pelayanan berdasarkan usulan Kepala Pusat yang bersangkutan
b. Kelompok Jabatan Fungsional berada di bawah dan bertanggungjawab kepada Sekretaris Badan atau Kepala Pusat yang bersangkutan
c. Kelompok Jabatan Fungsional di lingkungan Badan Geologi mempunyai tugas melaksanakan dan memberikan jasa penelitian dan pelayanan di bidang geologi, serta melaksanakan tugas lainnya yang didasarkan pada keahlian dan atau keterampilan tertentu sesuai ketentuan peraturan perundang-undangan.

d. Kelompok Jabatan Fungsional terdiri dari Jabatan Fungsional Peneliti, Perekayasa, Penyelidik Bumi, Teknisi Litkayasa, serta sejumlah jabatan fungsional tertentu lainnya yang terbagi dalam berbagai kelompok jabatan fungsional sesuai dengan bidang keahliannya yang diangkat dan diatur berdasarkan ketentuan perundang-undangan.
e. Kelompok Jabatan Fungsional dikoordinasikan oleh seorang Tenaga Fungsional Senior yang diangkat oleh Kepala Badan Geologi.
f. Jumlah Tenaga Fungsional ditentukan berdasarkan kebutuhan dan beban kerja.

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dengan penerapan teknologi *mobile* Android untuk menunjang system informasi *repository* di Badan Geologi Bandung. Beberapa teori yang terkait dengan penerapan teknologi *mobile* Android untuk menunjang sistem informasi repository di Badan Geologi Bandung adalah sistem informasi, Android, *PHP, Java, MySQL, Wamp Server,* UML *(Unified Modeling Language), Web service, Eclips, Software Devolepmen Kit* (SDK), *Java Devolepmen Kit* (JDK), *Android Devolepment Tool* (ADT), JSON, CSS *(Cascade Style Sheet), JQuery, JavaScript.*

2.2.1 Repository

Repository adalah alat yang secara otomatis memfasilitasi penampakan, pemeliharaan, pemanfaatan informasi tentang perusahaan. Repository merupakan tempat standar penyimpanan data yang direkam dan digunakan untuk menyelusuri tingkat pemanfaatan data diseluruh perusahaan.

Knowledge repository merupakan kumpulan dari informasi atau pengetahuan yang disimpan dalam database yang berguna dan menarik untuk perusahaan dan dapat diakses oleh jumlah populasi yang besar.

Jadi kesimpalan bahwa knowledge repository sistem merupakan suatu sistem yang dapat menyimpan, memelihara dan memanfaatkan informasi atau pengetahuan perusahaan secara terus menerus yang memungkinkan pengetahuan untuk dicari dan diakses kembali dalam jumlah yang besar dengan tingkat efisiensi dan akurasi yang tinggi untuk mendukung kerjasama dalam perusahaan. [4]

2.2.2 Sistem Informasi

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri. [5]

1. **Pengertian Sistem**

Pengertian sistem bisa dilihat dari dua pendekatan, yaitu berdasarkan prosedur dan komponennya. Sistem dilihat dari pendekatan prosedur berarti kumpulan prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu sedangkan sistem dilihat dari pendekatan komponennya berarti sistem terdiri dari komponen-komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu.

2. **Pengertian Data**

Data adalah kenyataan yang menggambarkan suatu kejadian–kejadian dan kesatuan yang nyata. Atau data adalah : representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pelanggan), hewan, peristiwa, konsep, keadaan dll, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

3. **Pengertian Informasi**

Informasi merupakan data yang memiliki nilai lebih bagi penerimanya karena telah melewati proses pengolahan tertentu sehingga bisa dimanfaatkan untuk

mengambil keputusan baik itu untuk saat ini ataupun untuk masa yang akan datang atau informasi didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

4. Pengertian Sistem Informasi

Berdasarkan pemaparan sebelumnya mengenai sistem, data, dan informasi maka dapat diambil kesimpulan bahwa pengertian dari sistem informasi adalah sekumpulan komponen-komponen dan prosedur-prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu. Berikut pengertian-pengertian lainnya dari sistem informasi:

1. *A set of people, data, procedures, hardware and software that work together to achieve the common goal of information management.*
2. Keterkaitan komponen-komponen dalam pengumpulan (atau menerima), proses, menyimpan dan mendistribusikan informasi untuk mendukung pembuatan keputusan (*decision making*), mengkoordinasi serta mengkontrol
3. Sistem informasi juga membantu, manajer dan para pegawai (pekerja) menganalisa masalah, menggambarkan subyek yang kompleks dan membuat (menghasilkan) produk-produk baru
4. Dasar (pondasi) dari bisnis pada umumnya.

Ada tiga aktifitas pada sistem informasi menghasilkan informasi pada organisasi yang diperlukan untuk membuat suatu keputusan, pengkontrolan, analisa masalah-masalah dan membuat produk baru atau layanan. Aktifitas itu adalah *input, process dan output.*

Input berarti menangkap atau pengumpulkan data mentah (*raw data*) didalam organisasi atau dari luar organisasi – *environment* / lingkungan. Proses diartikan dengan menterjemahkan data *input* kedalam bentuk yang lebih berarti. Sedangkan *output* adalah mentransfer informasi yang telah diproses kepada manusia dan aktivitasnya.

2.2.3 Android

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang di-*release* oleh *Google*. Sedangkan Android SDK (*Software Development Kit*) menyediakan *Tools* dan *API* yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman *Java*.

Dikembangkan bersama antara *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam* OHA *(Open Handset Alliance)* dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*). [1]

2.2.3.1 Sejarah Android

Android merupakan sistem operasi yang dikembngkan untuk perangkat mobile berbasis *Linux*. Pada awalnya sistem operasi ini dikembangkan oleh Android *Inc*. yang kemudian dibeli oleh *Google* pada tahun 2005.

Dalam usaha mengembangkan Android, pada tahun2007 dibentuklah *Open Handset Alliance* (OHA), sebeuah konsorsium dari beberapa perusahaan, yaitu *Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronis, Sprint Nextel,* dan *T-Mobile* dengan tjuan untuk mengembangkan standar terbuka untuk perangkat mobile. Pada tanggal 9 Desember 2008, ia diumumkan bahwa 14 anggota baru akan bergabung proyek Android, termasuk *PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.*

Sistem operasi Andorid dirilis sebagai berikut:

Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

Android versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; *CDMA / EVDO, 802.1x, VPN, Gestures,* dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi *VWGA*.

Android versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan hardware,

peningkatan *Google Maps* 3.1.2, perubahan *UI* dengan browser baru dan dukungan *HTML5*, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital *Zoom*, dan *Bluetooth* 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, *Google* melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah *Shazam*, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh *MySpace* dan *Facebook*.

Android versi 2.2 (*Froyo*)

Pada 20 Mei 2010 kembali diluncurkan ponsel android dengan versi 2.2 (*Froyo*) perubahan yang dilakukan meliputi optimasi kecepatan, memori, dan kinerja sistem operasi secara keseluruhan, dukungan untuk menginstal aplikasi pada memori eksternal, dukungan *Adobe Flash* 10.1 serta fungsi *USB tethering* maupun *Wi-Fi hotspot*.

Android versi 2.3 (*Gingerbread*)

Pada 1 Desember 2010 *Google* kembali meluncurkan versi terbaru yaitu Android versi 2.3. Pada versi ini terdapat peningkatan manajemen daya, control melalui aplikasi, penggunaan *multiple* kamera peningkatan performa serta penambahan sensor seperti *gyroscope*.

Android versi 3.0/3.1 (*Honeycomb*)

Pada versi ini berbeda dengan versi-versi sebelumnya. Versi ini dirancang khusus untuk *PC Tablet* sehingga memiliki *User Interface* yang berbeda dan dukungan ukuran layar yang lebih besar. Selain itu, pada versi ini memungkinkan penggunaan multiprosesor dan akselerasi perangkat keras untuk grafis. SDK versi pertama diluncurkan Februari 2011.

Android versi 4.0 (*Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline,* dan berbagi informasi dengan menggunakan *NFC.* Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galxy Nexus.

Android versi 4. 1 (*Jelly Bean*)

Android *Jelly Bean* yaang diluncurkan pada acara *Google I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan *input keyboard*, desain baru fitur pencarian, *UI* yang baru dan pencarian melalui *Voice Search* yang lebih cepat.

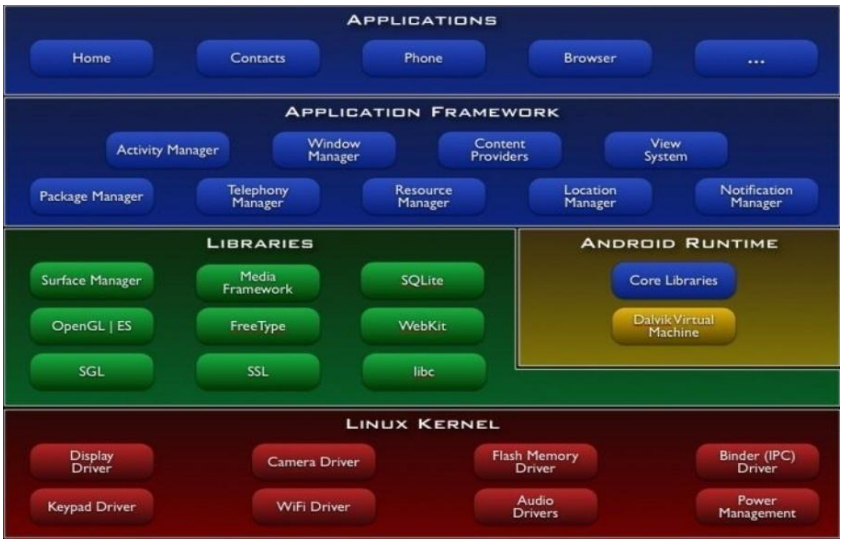
Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android *Jelly Bean* 4.1 muncul pertama kali dalam produk tablet Asus, yakni *Google Nexus* 7.

2.2.3.2 Fitur dan Arsitektur Android

Fitur yang tersedia pada Android adalah:

1. *Framework* aplikasi : memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.
2. *Dalvik virtual machine* : *virtual machine* yang dioptimalkan untuk perangkat mobile.
3. Grafik : grafik 2D dan grafik 3D yang didasarkan pada *library OpenGL*.
4. *SQLite* : untuk penyimpanan data.
5. Mendukung media : audio, video, dan berbagai format gambar (*MPEG4*, *H.264*, *MP3*, *AAC*, *AMR*, *JPG*, *PNG*, *GIF*)
6. *GSM*, *Bluetooth*, *EDGE*, *3G*, and *Wifi* (tergantung hardware)
7. *Camera*, *Global Positioning System* (GPS), *compass*, *dan accelerometer* (tergantung *hardware*)
8. Lingkungan pengembangan yang kaya, termasuk emulator, peralatan *debugging*, dan *plugin* untuk *Eclipse IDE*

Sistem operasi Android dibangun berdasarkan kernel *Linux* dan memiliki arsitektur sesuai dengan gambar.



Gambar 2.4 Arsitektur Android

Arsitektur Android dapat dijelaskan sebagai berikut:

a. Applications

Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi inti yang telah terdapat pada Android termasuk kalender, kontak, SMS, dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan bahasa pemrograman *Java*.

b. Application Framework

Pengembang aplikasi memiliki akses penuih ke Android sama dengan aplikasi inti yang telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm, menambahkan pemberitahuan ke status bar dan lain sebagainya. Arsitektur aplikasi ini dirancang untuk menyederhanakan penggunaan kembali komponen, aplikasi apa pun dapat memublikasikan kemampuan dan aplikasi lain dapat menggunakan kemampuan mereka sesuai dengan batasan keamanan. Dasar dari aplikasi adalah seperangkat layanan dan sistem, yaitu berbagai *View* yang digunakanuntuk membangun *UI*, *Content Provider* yang memungkinkan aplikasi berbagi data, *ResourceManager* menyediakan akses bukan kode seperti grafik, *string*, dan *layout*, *NotificationManager* yang akan membuat aplikasi dapat menampilkantanda pada status bar dan *ActivityManager* yang berguna mengatur daur hidup dari aplikasi.

c. Libraries

Satu set *libraries* dalam bahasa *C/C++* yang digunakan oleh berbagai komponen pada sistem Android.

d. Android Runtime

Satu set *libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman Java. Setiap aplikasi akan berjalan sebagai proses sendiri pada *Dalvik Virtual Machine* (*VM*).

e. Linux Kernel

Android tergantung pada *Linux* versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan model *driver*. *Kernel* juga bertindak sebagai lapisan antara *hardwar*e dan seluruh *software*.

2.2.4 PHP

PHP singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *Web* yang disisipkan pada dokumen *HTML*.

PHP merupakan *software Open-Soure* yang disebarkan dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya. *PHP* ditulis menggunakan bahasa *C*. [6]

A. Sejarah PHP

PHP diciptakan pertama kali oleh *Rasmus Lerdorf* pada tahun 1994. Awalnya, *PHP* digunakan untuk mencatat jumlah serta untuk mengetahui bsiapa saja pengunjung pada homepagenya. *Rasmus Lerdorf* adalah salah seorang pendukung *open source*. Oleh karena itu, ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis kemudian menambah kemampuan *PHP* 1.0 dan meluncurkan *PHP* 2.0.

Pada tahun 1996, *PHP* telah banyak digunakan dalam website di dunia. Sebuah kelompok pengembang *software* yang terdiri dari *Rasmus, Zeew Suraki, Amdi Gutman, Stig Bakken, Shane Caraveo,* dan *Jim Winsteadbekerja* sama untuk penyempurnakan *PHP* 2.0. akhirnya, pada tahun 1998, *PHP* versi 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan *PHP* 4.0.

tidak berhenti sampai di situ, kemampuan *PHP* terus ditambah, dan saat ini versi terbaru diluncurakan adalah *PHP* versi 5.0.x.

B. Kelebihan PHP

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa script sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bias melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman *Web* dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih daripada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain *Linux*, *Unix* (termasuk variannya *HP-UX*, *Solaris*, dan *OpenBSD*), *Microsoft windows*, *Mac OS* X, *RISC OS*. *PHP* juga mendukung banyak *Web server*, seperti *Apache*, *Microsoft* Internet *Information Server* (MIIS), *Personal Web Server* (PWS), *Netscape and iPlanet servers*, dan masih banyak lagi lainnya, bahkan *PHP* dapat bekerja sebagai suatu *CGI processor*.

PHP tidak terbatas pada hasil keluaran HTML (*Hypertext Markup Languages*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file *PDF*, dan *movues Flash*. *PHP* juga dapat menghasilkan teks seperti *XHTML* dan file *XML* lainnya.

2.2.5 Java

Java adalah salah satu bahasa pemrograman berorientasi objek OO (Object Oriented Programming). Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek.

Pemrograman Berorientasi Obyek (OOP)**,** Pemisalan Objek dalam OOP Objek-objek dalam dunia nyata, mempunyai 2 karakteristik khusus : Status dan Perilaku. Contohnya, sepeda punya status (jumlah gir, jumlah pedal, dua buah ban) dan perilaku (mengerem, mempercepat, ubah gir). Bahasa yang berorientasi pada objek pun mempunyai karakteristik yang sama dengan objek-objek di dunia nyata.

Yaitu status yang dalam bahasa pemrograman biasanya disimpan sebagai Variabel dan perilaku yang diimplementasikan sebagai *Method*. [7]

A. Sejarah Java

Dimulai pada tahun 1991 dibentuknya suatu tim yang diberi nama green Tim ini dipimpin oleh *Patrick Naughton* dan *James Gosling* (Sekelompok insinyur Sun). Awalnya mereka ingin membuat suatu bahasa komputer yang dapat digunakan oleh TV kabel (*Cable TV Box*) yang memiliki memori kecil dan setiap perusahaan memiliki tipe yang berbeda. Untuk mengaplikasikan hal tersebut mereka menggunakan hal yang pernah dicoba oleh bahasa pascal.

Sebelumnya *Niclaus Wirth* telah membuat sebuah bahasa yang portable yang akan digunakan dalam mesin bayangan selanjutnya ini disebut sebagai Virtual Mesin namun saat itu belum diumumkan. Akhirnya tim green mengunakan ide ini akhirnya mereka menciptakan *Java* Virtual Mesin. Vitual mesin inilah yang menbuat *Java* dapat dijalankan diberbagai *platform*.

Pada tahun 1992 tim green membuat produknya yang diberi nama *7 (*Star Seven*) Namun produk ini gagal dipasaran. Setelah itu dibuat produk yang baru. Inilah *Java*, pada awalnya bahasa pemrograman yang dibuat tersebut diberi nama *oak* (Kemungkinan nama ini diambil dari nama pohon yang ada didepan jendela *James Gosling*) Namun setelah diperiksa ternyata nama oak sudah pernah digunakan. Maka lahirlah nama *Java* .

Pada tahun 1993 sampai pertengahan 1994 tim *green* yang mempunyai nama baru *First person*. *Inc* hanya menghabiskan waktu dengan memperhatikan penjualan dari produknya. Ternyata tidak satupun orang ditemukan. Akhirnya pada tahun 1994 *First Person* dibubarkan. Pada tahun itu *Web* berkembang sangat pesat saat itu *browser* yang banyak digunakan adalah *Mosaic* suatu *broser* yang gratis.

Dalam suatu wawancara pada pertengahan tahun 94 James Gosling mengatakan "*We could build a real cool browser. It was one of the few things in the client/serve mainstream that needed some of the weird things we'd done: architecture*

neutral, real-time,reliable, secure—issues that weren't terribly important in the workstation world. So we built a browser." Akhirnya dibuatlah suatu *browser* oleh *Patrick Naughton* dan *Jonathan Payne*. *Browser* itu diberi nama *Hot Java*. *Browser* ini dibuat dengan menggunakan bahasa *Java* untuk menunjukkan kemapuan dari *Java*.

Yang tidak kalah hebatnya mereka juga membuat teknologi yang disebut *Aplet* sehingga *browser* dapat menjalankan kode program didalam *browser*. Inilah bukti dari teknologi itu yang ditunjukkan java pada tanggal 25 mei 1995 Sejak saat itu *browser Java* memberikan lisensinya kepada *browser* lain untuk bisa menjalankan *Applet* dan pada tahun 1996 diberikan pada *Internet Explorer*. Akhirnya pada tahun 1996 dikeluarkan versi pertama dari java dengan mengeluarkan java 1.02

Pada tahun 1998 dikeluarkan *Java* 1.2 dan java mulai mengeluarkan slogannya *"Write once, Run anywhre "*

B. Edisi Java

Java adalah bahasa yang dapat dijalankan dimanapun dan di sembarang *platform* apapun, di beragam lingkungan : *internet, intranets,consumer electronic products*, dan computer applications.The java 2 platform tersedia dalam 3 edisi untuk keperluan berbeda. Untuk beragam aplikasi yang dibuat dengan bahasa *Java*, *Java* dipaketkan dalam edis – edisi berikut :

1. *Java 2 Standard Edition* (J2SE)
2. *Java 2 Enterprise dition* (J2EE)
3. *Java 2 Micro Edition* (J2ME)

Masing – masing edisi berisi *Java* 2 *Software Development Kit* (J2SDK) untuk mengembangkan aplikasi dan *Java* 2 *Runtime Environent* (J2RE) untuk menjalankan aplikasi.

C. Kelebihan Java

Beberapa kelebihan *Java* adalah sebagai berikut :

a. *Multiplatform*. Kelebihan utama dari *Java* ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis *Java* dikerjakan diatas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah *Microsoft Windows, Linux, Mac OS* dan *Sun Solaris*. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs *Java*) untuk meninterpretasikan *bytecode* tersebut.

b. *Objek Oriented Programming* (OOP) yang artinya semua aspek yang terdapat di *Java* adalah objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut objek. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan *Java* sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang *advance* seperti komunikasi antara komputer sekalipun.

c. Perpustakaan kelas Yang Lengkap. *Java* terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program-program yang disertakan dalam pemrograman *Java*) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Bergaya *C++*, memiliki sintaks seperti bahasa pemrograman *C++* sehingga menarik banyak pemrogram *C++* untuk pindah ke *Java*. Saat ini pengguna *Java* sangat banyak, sebagian besar adalah pemrogram *C++* yang pindah ke *Java*. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan *Java* kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
e. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa *C++* yang dipakai secara luas).
f.

2.2.6 MySQL

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

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang software dan konsultan database bernama *MYSQL AB* yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama *TcX DataKonsult AB*, dan tujuan awal dikembangkannya *MySQL* adalah untuk mengembangkan aplikasi berbasis *web* pada *client*. Awalnya *Michael* "*Monty*" Widenius, pengembang satu-satunya di *TcX* memiliki sebuah aplikasi *UNIREG* dan rutin *ISAM* buatannya sendiri dan sedang mencari antarmuka *SQL* yang cocok untuk diimplementasikan ke dalamnya. Mula-mula *Monty* memakai *miniSQL* (*mSQL*) pada eksperimennya itu, namun *SQL* dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan query. Akhirnya *Monty* menghubungi *David Hughes*, pembuat *mSQL* yang sedang merilis versi kedua dari *mSQL*. Kemudian *Monty* mencoba membuat sendiri mesin *SQL* yang

memiliki antarmuka mirip dengan *SQL*, tetapi dengan kemampuan yang lebih sesuai sehingga lahirlah *MySQL*. Tentang pengambilan nama *MySQL*, sampai saat ini masih belum jelas asal usulnya. Ada yang berpendapat nama My diambil dari huruf depan dan belakang *Monty*, tetapi versi lain mengatakan nama itu diambil dari putri *Monty* yang kebetulan juga bernama *My*. [8]

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh *MySQL*:

1. *Potability*

   MySQL dapat berjalan stabil pada berbagai *OS* seperti *Windows*, *Linux*, *Unix*, *Mac OS, Solaris, Unix, Amiga, HP-UX, Symbian*.

2. *Open Source "limited"*

   Dahulu *MySQL* didistribusikan secara *open source* (gratis), dibawah lisensi *GPL* sehingga kita dapat menggunakannya secara cuma-cuma tanpa dipungut biaya. Namun, saat ini karena *MySQL* telah dibeli oleh *SUN*, maka kita tidak dapat lagi menikmati fitur-fitur baru yang ada di *MySQL*, karena *SUN* akan membatasi fitur-fitur baru ini hanya untuk user yang membeli lisensinya. Sehingga *MySQL* tidak lagi sebuah *opensource* yang benar-benar gratis lagi. *MySQL* sekarang hanya menyediakan fitur-fitur dasar saja yang saat ini sudah menggunakan versi 5.1.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik. Hal ini memungkinkan sebuah *database server MySQL* dapat diakses klien secara bersamaan.

4. *Performance Tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.

5. *Column Types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set* serta *enum*.

6. *Command* dan *Functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scalability* dan *Limits*

   MySQL ammpu menangani *database* dalam skala besar dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas *index* yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan klien menggunakan *TCP/IP*, *Unix* soket (*Unix*), atau *Named Pipes* (*NT).*

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari dua puluh bahasa.

11. *Interface*

    MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients* dan *Tools*

    MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur Tabel

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE* dibandingkan *database* lainnya.

2.2.7 UML (Unified Modeling Language) [9]

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Chonoles mengatakan sebagai bahasa, *berate* UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. UML bukan hanya sekedar diagram, tetapi juga menceritakan konsepnya. Ketika pelanggan memesan suatu dari sistem, bagaimana transaksinya? bagaimana sistem mengatasi *error* yang terjadi? Bagaimana keamanan terdahap sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak
2. Sarana komunikasi antara perankat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam bidang investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales dan suplier.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis timing diagram) dan lainnya ada bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini.

Sebagai perancang sistem, mau tidak mau pasti akan menjumpai UML, baik kita sendiri yang membuat atau sekedar membaca diagram UML buatan orang lain.

1. Evolusi UML

Evolusi UML menjelaskan bahwa sebelum ada UML, para pengembang bahasa pemrograman berorientasi objek sulit untuk berkomunikasi satu ama lain. Ada kira-kira 50 jenis notasi dan grafik yang meggambarkan bahasa pemrograman berorientasi objek pada waktu itu.

Para pengguna notasi yang berlainan ini saling berebut pengaruh agar notasi yang digunakan menjadi notasi standar. Walaupun dijumpai beberapa notasu yang cukup jelas dan sangat cocok untuk menyelesaikan masalah pembuatan perangkat lunak *Object Oriented Program* (OOP), tetapi belum diakui oleh pengembang sistem yang lain.

Pada bulan Oktober 1994, Jim Rumbaugh, penemu notasi *Object Modelling Technique* (OMT) dan *Grady Booch*, penemu *Booch Method* (*Metode Booch*) bersama-sama menyamakan notasi mereka. Ditahun yang sama, Ivar Jacobson (penemu *Objectory Method*) ikut bergabung. Ketiga orang itu, sering disebut three amigos, bersama-sama membangun notasi standar OOP untuk *software Rational IBM*. *Jim Rumbaugh* banyak member masukan dalam hal pembuatan notasi dan analisa UML. *Grady Booch* merancang secara detil kapabilitas UML sedangkan Ivan Jacobson berusaha membuat UML cocok dengan model bisnis dan mencoba mengembangkan use case diagram lebih lanjut.

Ketiga pengembang tersebut mengalami kesulitan akibat kompleksnya permasalahan OOP yang ada, namun mereka dibantu oleh *Object Management Group* (OMG). OMG adalah gabungan dari kurang lebih 800 perusahaan pengembang perangkat lunak berorientasi objek. Setelah perdebatan sengit yang cukup lama, consensus tentang notasi berasil dicapai dengan sangat sukses pada bulan November 1997 berkat bantuan OMG.

Sejak tahun 1997, divisi *Revision Task Force* (RTF) milik OMG beberapa kali merevisi UML. Revisi dimaksudkan untuk memperkuat konsistensi notasi, meningkatkan kekompakan antara user dan pengembang perangkat lunak. Akan tetapi UML terpaksa mengikuti perkembangn *software-software* berbasis objek yang ada (misalnya *Java*) dari sisi pendekatan komponen (*Componen - based development*) dan kemampuan *tool software-software* tersebut. Setelah dilakukan perubahan secara sistematik, akhirnya dihasilkan UML 2.0 pada tahun 2003.

2. Diagram-diagram UML

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkandelepan karena beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram kelas

Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.

2. Diagram paket

Diagram paket (*package diagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. Diagram Use-Case

Diagram *Use-Case* bersifat statis. Diagram ini memperlihatkan *use-case* dan actor-aktor (suatu jenis khusus dari kelas).

4. Diagram interaksi dan sequence (urutan)

Diagram interaksi dan *sequence* (urutan) berifat dinamis. Diagram urutan adalah diagram interaksi yang menenkankan pada pengiriman pesan dalam suatu waktu tertentu.

5. Diagram komunikasi (*Communication Diagram*)

Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kaloborasi UML 1.4 yang menenkankan organisasi structural dari objek-objek yang menerima serta mengirim pesan.

6. Diagram Statechart (*Statechart Diagram*)

Diagram Statechart (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan pada 43ystem, memuat status (*state*), transisi, kejadian serta aktifitas.

7. Diagram Aktivitas (*Activity Diagram*)

Diagram Aktivitas (*Activity Diagram)* bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.

8. Diagram Komponen (*Component Diagram*)

Diagram Komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

9. Diagram Deployment (*Deployment Diagram*)

Diagram Deployment (*Deployment Diagram*) bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya (misalnya Data Flow Diagram, Entitu Relationship Diagram dan sebagainya).

2.2.8 Web Service

Web Service adalah sekumpulan application logic beserta objek-objek dan metode-metode yang dimilikinya yang terletak di suatu server yang terhubung ke *internet* sehingga dapat diakses menggunakan *protocol* HTTP dan SOAP (*Simple*

Object Access Protocol). Dalam penggunaannya, web *service* dapat digunakan dari hanya untuk memeriksa data *user* yang login ke sebuah *website* ataupun untuk digunakan pada transaksi perbankan *online* yang rumit.

Tujuan dari teknologi ini adalah untuk memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat platform (platform-neutral) dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*). Hal tersebut dapat terjadi karena penggunaan *XML* standar yang didukung oleh banyak perusahaan besar di dunia, yang digunakan untuk bertukar data. Selain daripada itu, penggunaan SOAP menjadikan metode-metode dari objek-objek yang ada dalam sebuah web service dapat di akses dari aplikasi lain seperti halnya aplikasi tersebut mengakses metode lokal.

1. **Web Services Messaging**

W3C mendefinisikan web service sebagai "sebuah *software* aplikasi yang dapat teridentifikasi oleh *URI* dan memiliki *interface* yang didefiniskan, dideskripsikan, dan dimengerti oleh *XML* dan juga mendukung interaksi langsung dengan *software* aplikasi yang lain dengan menggunakan message berbasis *XML* melalui protokol *internet*" *Web service* adalah sebuah *sofware* aplikasi yang tidak terpengaruh oleh platform, ia akan menyediakan *method-method* yang dapat diakses oleh *network*. Ia juga akan menggunakan *XML* untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda.

Beberapa karakteristik dari *web service* adalah:

a. *Message-based*
b. *Standards-based*
c. *Programming language independent*
d. *Platform-neutral*

Beberapa key *standard* didalam *web service* adalah:

a. *XML*.
b. *SOAP*.
c. *WSDL*.
d. *UDDI*.

SOAP (*Simple Object Access Protocol*) adalah sebuah *XML*-based mark-up language untuk pergantian pesan diantara aplikasi-aplikasi. *SOAP* berguna seperti sebuah amplop yang digunakan untuk pertukaran data object didalam *network*. *SOAP* mendefinisikan empat aspek didalam komunikasi: *Message envelope, Encoding, RPC call convention,* dan bagaimana menyatukan sebuah message didalam protokol *transport*.

Sebuah *SOAP message* terdiri dari *SOAP Envelop* dan bisa terdiri dari attachments atau tidak memiliki *attachment*. *SOAP envelop* tersusun dari *SOAP* header dan *SOAP body*, sedangkan *SOAP* attachment membolehkan non-XML data untuk dimasukkan kedalam *SOAP message*, *di-encoded,* dan diletakkan kedalam *SOAP message* dengan menggunakan *MIME-multipart*.

2.2.9 Eclips

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independ*ent). Berikut ini adalah sifat dari *Eclipse*:

Multi-platform: Target sistem operasi *Eclipse* adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*.

Mulit-language: *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP*, dan lain sebagainya.

Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

1. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak *IBM Visual Age for Java* 4.0. Produk ini diluncurkan oleh *IBM* pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan *Eclipse* lebih lanjut dan pengaturan organisasinya.

2. Arsitektur Eclipse

Sejak versi 3.0, *Eclipse* pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan didalam *Eclipse* sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari *Eclipse* yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk *RCP*:

a. *Core platform*
b. *OSGi*
c. *SWT* (*Standard Widget Toolkit*)
d. *JFace*
e. *Eclipse Workbench*

Secara standar *Eclipse* selalu dilengkapi dengan JDT (*Java Development Tools*), plug-in yang membuat *Eclipse* kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan plug-in

baru. *Eclipse* beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman *Java*.

Konsep *Eclipse* adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program *Java*, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program *C/C++* terdapat *plug-in CDT (*C/C++ *Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh *Eclipse*, *plug-in UML2* tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* secara gratis seperti *Eclipse downloads by project*.

3. Kontroversi Eclipse

Hadirnya *Eclipse* di tengah-tengah persaingan IDE, terutama *Java* IDE, menimbulkan banyak kontroversi. Salah satunya adalah penggunaan user interface *Eclipse*. *Sun Microsystems* sebagai perusahaan yang membuat Java, selama ini mengeluarkan dua library untuk pengembangan *Graphical User Interface* (GUI), yaitu *Abstract Windowing Toolkit* (AWT) dan Swing. Akan tetapi dalam pembuatan Eclipse, dua library ini ditinggalkan karena berbagai alasan, yang salah satunya adalah kedua library ini tidak memberikan look and feel yang baik. Sehingga untuk hal ini dikembangkanlah *GUI-library* yang baru, *SWT*. Selain itu, *Sun* yang mempunyai *IDE* sendiri untuk pengembangan *Java* bernama *NetBeans,* menjadi gerah atas kehadiran *Eclipse* yang sampai saat ini komunitasnya cukup besar. Hal ini terlihat dengan menolaknya *Sun* untuk bekerja sama dalam *Eclipse Foundation*, walau telah diundang.

Nama *Eclipse* seolah-olah memberi pesan *explisit*: cahaya *Sun* (matahari) yang pancarannya dihalangi oleh *Eclipse* (gerhana).

2.2.10 Software Development Kit (SDK)

*Software Development Kit (*SDK*)* adalah suatu kit atau *library* dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya *SDK* terdiri dari kumpulan tools yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai *SDK* yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis *java*.

2.2.11 Java Development Kit (JDK)

*Java Development Kit (*JDK*)* adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak yang berbasis *Java*, Sedangkan *JRE* adalah sebuah implementasi dari *Java Virtual Machine* yang benar-benar digunakan untuk menjalankan program *Java*.Biasanya, setiap *JDK* berisi satu atau lebih *JRE* dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya. Perbedaan *JDK* dengan *SDK* (*Software Development Kit*) yaitu *JDK* adalah sebuah *SDK* tetapi sebuah *SDK* tidak harus menjadi sebuah *JDK*.

2.2.12 Android Development Tool (ADT)

*Android Development Tools (*ADT*)* adalah *plugin* untuk *Eclipse Intergrated Development Environment* (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android. ADT memperluas kemampuan *Eclipse* untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi *UI*, menambahkan komponen berdasarkan Android *Framework* API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi. Mengembangkan aplikasi di *Eclipse* dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom *XML* editor, dan *debug panel ouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.2.13 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*, Standar *ECMA*-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga *C* termasuk *C, C++, C#, Java, JavaScript, Perl, Python* dll. Oleh karena sifat-sifat tersebut, menjadikan JSON *ideal* sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

1. **Objek**

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. **Larik**

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

3. **Nilai**

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.

4. **String**

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.

5. **Angka**

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.

2.2.14 JQuery

JQuery merupakan suatu *framewor**k*** (*library*) *Javascript* yang menekankan bagaimana interaksi antara *Javascript* dan HTML.

JQuery pertama kali dirilis pada tahun 2006 oleh John Resig. Pada perkembangannya JQuery tidak sekedar sebagai *framework Javascript*, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak *developer web* menggunakannya. JQuery memiliki slogan "*Write less, do **more***" yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan code, tapi dengan hasil yang lebih banyak.

JQuery merupakan library open source dengan lisensi *GNU General Public License* dan *MIT License*. Dari sisi ukurannya, *framework* JQuery sungguh ramping, hanya sekitar 20 KB dan hanya terdiri dari satu file. Namun demikian, bagi yang menginginkan fungsi lebih, JQuery memungkinkan penambahan fungsionalitas dalam bentuk *plugin*. Saat ini tersedia ribuan *plugin* yang dapat diperoleh secara gratis di internet. Di bawah ini hal-al yang bisa dilakukan oleh JQuery :

1. Mengakses bagian halaman tertentu dengan mudah

Tanpa adanya *library Javascript* khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan *Document Object Model* (DOM) dan pengaksesan harus secara spesifik menyesuaikan dengan struktur HTML. Dengan kata lain, pengaksesan bagian tertentu dari halaman sangat tergantung pada struktur dari HTML. JQuery menawarkan cara yang mudah (bahkan sangat mudah) dalam mengakses bagian tertentu dari halaman. Pengaksesan juga tidak terlalu bergantung pada struktur HTML.

2. Mengubah tampilan bagian halaman tertentu

CSS *(Cascading Style Sheet)* menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman web. Namun terkadang CSS punya kelemahan yang cukup mengganggu, yaitu beberapa perintah CSS tidak didukung oleh semua browser. Cukup merepotkan jika kita harus mendesign halaman *web* dengan beberapa CSS sekaligus. Sekali lagi JQuery menawarkan solusi untuk mengatasi hal tersebut. Dengan JQuery, “kesenjangan” yang terjadi antara browser dalam urusan CSS akan tertutup dengan baik.

3. Mengubah isi dari halaman

Jaman dulu sebelum JQuery lahir cukup sulit jika kita akan mengubah sebagian isi dari halaman. Mengubah disini dapat berarti mengganti teks, menambahkan teks atau gambar, mengurutkan suatu daftar (list), menghapus baris tabel dan sebagainya. Dengan JQuery, hal tersebut dapat dilakukan dengan hanya beberapa baris perintah.

4. Merespond interaksi user dalam halaman

Website yang baik tidak cukup digambarkan dengan user-interface dan tampilan yang memukau. Namun lebih dari itu, bagaimana pengunjung dapat berinteraksi dengan *website* dan dapat mengatur tampilannya sendiri. Interaktivitas sangat bergantung bagaimana pemrograman yang dipakai dalam menangani *event-handling*. *Javascript* sendiri memiliki beberapa event-handling seperti onclick untuk menangani event saat terjadi *click*. Namun demikian, *event handling* pada Javascript terbatas pada *object-object* tertentu, dan jenisnya pun terbatas. JQuery melengkapi semuanya dengan tambahan penanganan *event-handling* yang semakin mudah.

5. Menambahkan animasi ke halaman

Animasi seringkali disertakan dalam suatu halaman *web* untuk menambah kecantikannya. Saat ini animasi masih cukup digemari oleh para peselancar situs. Animasi dapat dibuat dalam berbagai gaya, ada yang menggunakan *Flash*, gambar bergerak (*GIF*), video, dan sebagainya. Masing-masing tentu memiliki kelebihan dan kekurangan masing-masing. *JQuery* sendiri menawarkan konsep animasi (walaupun masih sederhana) yang cukup apik namun ramah bandwidth alias ringan. Salah satu animasi yang bisa dibuat dengan JQuery adalah fading jika terdapat suatu bagian dari halaman ditambahkan atau dihilangkan.

6. Mengambil informasi dari server tanpa me-refresh seluruh halaman

Mengambil informasi dari server tanpa refresh halaman merupakan salah satu konsep dasar dari yang namanya AJAX *(Asynchronous Javascript and XML).* Pada penerapannya, cukup ribet jika harus membangun website dengan konsep AJAX, saat ini banyak library khusus yang berusaha mempermudahnya. JQuery merupakan salah satunya.

7. Menyederhanakan penulisan Javascript biasa

Semboyan JQuery adalah **"***Write less, do more***"** atau dengan kata lain kesederhanaan dalam penulisan code, tetapi menghasilnya tampilan yang lebih. Sebenarnya inilah yang menjadi daya tarik tersendiri buat para pengembang web untuk menggunakan *JQuery*.

2.2.15 JavaScript

JavaScript adalah bahasa pemrograman berbasis prototipe yang berjalan disisi klien. Jika kita berbicara dalam konteks *web*, sederhananya, kita dapat memahami *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk di *browser* atau halaman *web* agar halaman *web* menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu *Java* dan *Script*. Java adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program.

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan *JavaScript* adalah membuka halaman *pop up*, fungsi validasi pada *form* sebelum data dikirimkan ke *server*, merubah image kursor ketika melewati objek tertentu, dan lain lain.

JavaScript pertama kali diperkenalkan oleh *Netscape* pada tahun 1995. Pada awalnya bahasa yang sekarang disebut JavaScript ini dulunya dinamai *LiveScript* yang berfungsi sebagai bahasa sederhana untuk *browser* Netscape *Navigator* 2 yang

sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan Sun (pengembang bahasa pemrograman *Java*) pada masa itu, maka *Netscape* memberikan nama *JavaScript* kepada bahasa tersebut pada tanggal 4 desember 1995.

Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai *Jscript* di *browser* milik mereka yaitu Internet *Explorer* 3. *JavaScript* sendiri merupakan modifikasi dari bahasa pemrograman *C++* dengan pola penulisan yang lebih sederhana dari bahasa pemrograman *C++*.

Yang harus diperhatikan dalam pengelolaan pemrograman *JavaScript*, diantaranya *JavaScript* adalah *case sensitive*, yang artinya *JavaScript* membedakan huruf besar dan huruf kecil. Jika Anda pernah belajar bahasa pemrograman seperti Turbo *C* atau *C++*, maka sama seperti bahasa pemrograman tersebut, dimana huruf T tidak sama dengan huruf t.

Dalam bahasa pemrograman *JavaScript* juga, sebagai contoh fungsi perintah var tidak boleh ditulis Var dan juga tidak boleh ditulis VAR (huruf besar semua), yang benar adalah var (huruf kecil semua). Perintah lain adalah new Date tidak boleh ditulis new date (huruf kecil semua), dsb.

Javascript tidak memerlukan compiler untuk menjalankannya (pada kenyataanya *compiler* javascript sendiri sudah termasuk dalam *webbrowser*). *Javascript* berbeda dengan *Java*. *Java* merupakan bahasa yang didesain untuk mengeksekusi cepat dan keamanan penulisan sedangkan *Javascript* memiliki kemampuan yang tinggi meskipun hanya dituliskan dalam beberapa baris perintah.

2.2.16 Metode Pencarian Hashing Untuk Pencarian File dan Surat

Hashing adalah transformasi aritmetika sebuah string dari karakter menjadi nilai yang merepresentasikan string aslinya. Menurut bahasanya, hash berarti memenggal dan kemudian menggabungkan. *Hashing* digunakan sebagai metode untuk menyimpan data dalam sebuah larik (*array*) agar penyimpanan data, pencarian

data, penambahan data, dan penghapusan data dapat dilakukan dengan cepat. Ide dasarnya adalah menghitung posisi record yang dicari dalam larik, bukan membandingkan *record* dengan isi pada larik. Fungsi yang mengembalikan nilai atau kunci disebut fungsi *hash* dan larik yang digunakan disebut tabel hash. Secara teori, kompleksitas waktu (T(n)) dari fungsi hash yang ideal adalah O(1). Untuk mencapai itu setiap record membutuhkan suatu kunci yang unik.

a. Fungsi Hash

Fungsi Hash (dilambangkan dengan h(k)) bertugas untuk mengubah k (key) menjadi suatu nilai dalam interval [0....X], dimana "X" adalah jumlah maksimum dari record-record yang dapat ditampung dalam tabel. Jumlah maksimum ini bergantung pada ruang memori yang tersedia. Fungsi Hash yang ideal adalah mudah dihitung dan bersifat acak, agar dapat menyebarkan semua key. Dengan key yang tersebar, berarti data dapat terdistribusi secara seragam tabrakan dapat dicegah. Sehingga kompleksitas waktu model Hash dapat mencapai O(1), di mana kompleksitas tersebut tidak ditemukan pada struktur model lain. Ada beberapa macam fungsi hash yang relative sederhana yang dapat digunakan dalam penyimpanan database salah satunya:

b. Metode Pembagian Bersisa (*division-remainder method*)

Jumlah lokasi memori yang tersedia dihitung, kemudian jumlah tersebut digunakan sebagai pembagi untuk membagi nilai yang asli dan menghasilkan sisa. Sisa tersebut adalah nilai hashnya. Secara umum, rumusnya h(k) = k mod m. Dalam hal ini m adalah jumlah lokasi memori yang tersedia pada larik. Fungsi hash tersebut menempatkan record dengan kunci k pada suatu lokasi memori yang beralamat h(k). Metode ini sering menghasilkan nilai hash yang sama dari dua atau lebih nilai aslinya atau disebut dengan tabrakan. Karena itu, dibutuhkan mekanisme khusus untuk menangani tabrakan yang disebut kebijakan resolusi tabrakan.

c. Resolusi Collision (tabrakan)

Dalam memasukan nilai kunci ke alamat dapat timbul kemungkinan bahwa record dengan nilai kunci yang berbeda di masukan pada satu alamat record yang sama. Fungsi Hash bertujuan untuk menentukan alamat yang berbeda untuk dua kunci atau lebih pada kunci yang sama pada record. Jika dua buah kunci atau lebih yang mempunyai kunci sama menempati alamat yang sama menyebabkan terjadinya tabrakan (*collision*) yaitu suatu kondisi untuk dua data yang tidak sama setelah dilakukan perhitungan menghasilkan alamat yang sama, dimana terdapat lebih dari satu kunci (key) yang menempati slot alamat yang sama yang akan mengakibatkan penurunan efisiensi pencarian data, meskipun sudah menentukan fungsi Hash yang cukup baik. Dengan demikian, harus mempunyai suatu cara untuk mengatasi tabrakan yang mungkin terjadi, yang disebut dengan *resolusi* kolisi (*collision resolution*) Satu pendekatan sederhana untuk mengatasi terjadinya tabrakan (*collision*) adalah dengan cara menyimpan kunci yang bertabrakan dalam ruang berikutnya yang tersedia.

d. Double Hashing

Double hashing adalah satu metoda yang terbaik yang tersedia untuk pengalamatan terbuka sebab permutasi yang dihasilkan mempunyai banyak karakteristik permutasi terpilih secara acak. fungsi H1 untuk menentukan *home address* dan fungsi H2 untuk menentukan increment jika terjadi tabrakan. *Double hashing* menggunakan fungsi *Hash* dengan format

$$h(k,i) = (h1(k)+ih2\ (k)) \bmod N$$

dengan N adalah ukuran tabel dan i = (0,1,...N-1)

h1 adalah :

$$h1(k) = k \bmod N$$

h2 adalah :

$$h2(k) = 1 + k \bmod N'$$

N’ dipilih yang kurang dari N (N-1 atau N-2)