

BAB 2

TINJAUAN PUSTAKA

2.1. Tinjauan Intansi

Penelitian ini bekerja sama dengan dua Intansi yaitu Dinas Pariwisata dan Kebudayaan Kabupaten Jember dan Dinas Perhubungan Kabupaten Jember untuk keterkaitan data yang digunakan dalam aplikasi.

2.1.1. Profil Dinas Pariwisata dan Kebudayaan Kabupaten Jember

Dinas Pariwisata dan Kebudayaan Kabupaten Jember adalah salah satu dinas pemerintahan yang mempunyai tugas dalam mengelolah destinasi wisata dan budaya untuk wilayah Kabupaten Jember. Dinas ini mengelolah, mengembangkan dan meningkatkan destinasi pariwisata, sarana dan prasarana apreatur, kekayaan budaya dan SDM aparatur. Dinas Pariwisata dan Kebudayaan juga bertugas dalam pelaporan capaian kinerja dan keuangan pariwisata.

2.1.1.1. Visi dan Misi

Visi

Terwujudnya kabupaten jember sebagai daerah tujuan wisata yang bertumpu pada nilai agama, budaya dan berwawasan lingkungan demi peningkatan ekonomi masyarakat.

Misi

1. Merencanakan, mengembangkan, memasarkan dan memperkenalkan potensi kepariwisataan jember di tingkatlokal, nasional dan internasional.
2. Mempersiapkan, mengembangkan dan meningkatkan sarana prasarana dan infrastuktur untuk mendukung keberadaan potensi dan daya tarik wisata di Kabupaten Jember.
3. Meningkatkan kualitas SDM baik internal maupun eksternal dalam bidang kepariwisataan untuk meningkatkan kepercayaan wisata dan stake holder.

2.1.1.2. Logo

Berikut adalah logo dari dinas pariwisata dan kebudayaan di Kabupaten Jember yang dapat dilihat pada Gambar 2 :



Sumber gambar: <https://jemberkab.go.id/>

Gambar 1 Logo Dinas Pariwisata dan Kebudayaan

2.1.2. Profil Dinas Perhubungan Kabupaten Jember

Dinas Perhubungan Kabupaten Jember adalah salah satu dinas pemerintahan yang mempunyai tugas dalam mengelolah sektor lalu lintas, angkutan, sarana dan keselamatan perjalanan. Dinas Perhubungan memiliki tugas untuk mengelolah rekayasa lalu lintas, mengelolah transportasi angkutan umum dan angkutan perintis atau pedesaan di Kabupaten Jember.

2.1.2.1. Visi dan Misi

Visi

Menjadi dinas yang unggul dalam menciptakan lalu lintas yang tertib, lancar, aman, dan nyaman

Misi

1. Meningkatkan kemampuan sumber daya manusia, baik aparatur maupun masyarakat
2. Meningkatkan sarana prasarana penunjang ketertiban, kelancaran keamanan dan kenyamanan lalu lintas
3. Meningkatkan peran dalam penggalian sumber pendapatan asli daerah dan sektor perhubungan

2.1.2.2. Logo

Berikut ini adalah logo dari Dinas Perhubungan di Kabupaten Jember yang dapat dilihat pada Gambar 3 :



Sumber gambar: <https://jemberkab.go.id/>

Gambar 2 Logo Dinas Perhubungan

2.2. Pariwisata

Pariwisata adalah perjalanan yang dilakukan untuk singgah atau beraktivitas sementara waktu ke suatu tempat di luar tempat tinggal atau pun di luar tempat kerjanya yang biasa. Pariwisata merupakan serangkaian aktivitas berupa aktivitas perpindahan orang yang terjadi untuk sementara waktu ke satu tujuan di luar tempat tinggalnya atau pun di luar tempat kerjanya yang biasa, aktivitas yang dilaksanakan selama berada di lokasi tujuan, dan berbagai macam kemudahan yang disediakan untuk memenuhi kebutuhannya baik selama perjalanan menuju lokasi tujuan atau pun ketika berada pada tujuan tersebut [4]. Pariwisata memiliki beberapa komponen pokok secara umum disepakati, yaitu:

1. *Traveller*, yaitu orang yang melakukan perjalanan antar dua atau lebih lokalitas.
2. *Visitor*, yaitu orang yang melakukan perjalanan ke daerah yang bukan merupakan tempat tinggalnya, kurang dari 12 bulan, dan tujuan perjalanannya bukanlah untuk terlibat dalam kegiatan untuk mencari nafkah, pendapatan, dan penghidupan di suatu tempat tujuan.
3. *Tourist*, yaitu bagian dari visitor yang menghabiskan waktu paling tidak satu malam (24 jam) di daerah yang dikunjungi [5].

2.3. Tempat Wisata

Tempat Wisata merupakan letak dari suatu keadaan alam yang memiliki daya wisata dan dibangun atau dikembangkan sehingga memiliki daya tarik untuk dikunjungi oleh wisatawan. Suatu daerah untuk menjadi suatu daerah tujuan wisata yang baik, harus mengembangkan tiga hal agar daerah tersebut menarik untuk dikunjungi [6], yaitu :

- a. Adanya sesuatu yang dapat dilihat (*something to see*), maksudnya adalah adanya sesuatu yang menarik untuk dilihat, dalam hal ini obyek wisata yang berbeda dengan tempat-tempat lain (mempunyai keunikan tersendiri).
- b. Adanya sesuatu yang dapat dibeli (*something to buy*), yaitu terdapat sesuatu yang menarik yang khas untuk dibeli dalam hal ini dijadikan cendramata untuk dibawa pulang ke tempat masing-masing sehingga di daerah tersebut harus ada fasilitas untuk dapat berbelanja yang menyediakan souvenir maupun kerajinan tangan lainnya dan harus didukung pula oleh fasilitas lainnya seperti money changer dan bank.
- c. Adanya sesuatu yang dapat dilakukan (*something to do*), yaitu suatu aktivitas yang dapat dilakukan di tempat itu yang bisa membuat orang yang berkunjung merasa betah di tempat tersebut.

Dari keterangan di atas, dapat disimpulkan bahwa suatu objek wisata yang baik dan menarik untuk dikunjungi harus mempunyai keindahan alam dan juga harus memiliki keunikan dan daya tarik untuk dikunjungi dan juga didukung oleh fasilitas pada saat menikmatinya. Ada berapa jenis pariwisata yang sudah dikenal antara lain :

1. Wisata Budaya

Wisata budaya yaitu perjalanan yang dilakukan atas dasar keinginan untuk memperluas pandangan hidup seseorang dengan cara mengadakan kunjungan ketempat lain atau keluar negeri, mempelajari keadaan rakyat, kebiasaan dan adat istiadat mereka, cara hidup mereka, kebudayaan dan seni mereka.

2. Wisata Kesehatan

Wisata kesehatan yaitu wisatawan-wisatawan yang melakukan perjalanan dengan tujuan berolahraga atau memang sengaja bermaksud mengambil bagian aktif dalam pesta olahraga di suatu tempat atau negara.

3. Wisata Olahraga

Wisata olahraga yaitu wisatawan-wisatawan yang melakukan perjalanan dengan tujuan berolahraga atau memang sengaja bermaksud mengambil bagian aktif dalam pesta olahraga di suatu tempat atau negara.

4. Wisata Komersial

Wisata kesehatan yaitu termasuk perjalanan untuk mengunjungi pameran-pameran dan pekan raya yang bersifat komersial, seperti pameran industri, pameran dagang dan sebagainya.

5. Wisata Industri

Wisata kesehatan yaitu perjalanan yang dilakukan oleh rombongan pelajar atau mahasiswa, atau orang-orang awam ke suatu kompleks atau daerah perindustrian dengan maksud dan tujuan untuk mengadakan peninjauan atau penelitian.

6. Wisata Bahari

Wisata kesehatan yaitu wisata yang banyak dikaitkan dengan danau, pantai atau laut.

7. Wisata Cagar Alam

Wisata kesehatan yaitu jenis wisata yang biasanya diselenggarakan oleh agen atau biro perjalanan yang mengkhususkan usaha-usaha dengan mengatur wisata ke tempat atau daerah cagar alam, taman lindung, hutan daerah pengunungan dan sebagainya yang kelestariannya dilindungi oleh undang-undang.

8. Wisata Bulan Madu

Wisata kesehatan yaitu suatu penyelenggaraan perjalanan bagi pasangan-pasangan pengantin baru yang sedang berbulan madu dengan fasilitas khusus dan tersendiri demi kenikmatan perjalanan [7].

Berikut adalah tempat wisata yang ada di Jember :

Tabel 1 Tempat Wisata di Kabupaten Jember

No	Nama	Jenis	Alamat
1	Pantai Payangan	Bahari	Sumberrejo, Ambulu, Jember Regency, East <i>Java</i> 68172
2	Pantai Papuma	Bahari	Jl. Pantai, Lojejer, Wuluhan, Kabupaten Jember, Jawa Timur 68162
3	Pantai Watu Ulo	Bahari	Jl. Watu Ulo, Sumberrejo, Jember Regency, East <i>Java</i>
4	Pantai Bandalit	Bahari	Andongrejo, Tempurejo, Jember Regency, East <i>Java</i> 68173
5	Rembangan	Buatan	Jl. Rembangan, Kemuning Lor, Arjasa, Kemuning Lor, Arjasa, Kabupaten Jember, Jawa Timur 68191
6	Tiara Jember Park	Buatan	Jl. Kaliurang, Sumbersari, Kabupaten Jember, Jawa Timur 68124
7	Botani Sukorambi	Argowisata	Jl. Mujahir, Sukorambi, Kabupaten Jember, Jawa Timur 68151
8	Museum Tembakau	Edukasi	Jl. Kalimantan No.1, Sumbersari, Kabupaten Jember, Jawa Timur 68121
9	Puslit Kopi dan Kakao	Edukasi	Jl. PB. Sudirman No. 90, Nogosari, Rambipuji, Nogosari, Rambipuji, Kabupaten Jember, Jawa Timur 68175
10	Taman Galaxy	Taman	Tempurejo, Jember Regency, East <i>Java</i> 68173
11	Air Terjun Tancak	Cagar Alam	Suci, Panti, Suci, Panti, Kabupaten Jember, Jawa Timur 68153
12	Glantangan	Argowisata	Pondokrejo, Tempurejo, Jember Regency, East <i>Java</i> 68173

2.4. Angkutan Umum

Angkutan adalah suatu sarana untuk memindahkan orang atau barang dari satu tempat ke tempat lain. Dengan tujuan untuk membanu orang atau kelompok orang dalam menjangkau berbagai tempat yang dikehendaki, atau mengirimkan barang dari tempat asalnya ketempat tujuannya. Prosesnya dapat dilakukan menggunakan sarana angkutan berupa kendaraan.

Angkutan Umum adalah angkutan penumpang yang dilakukan dengan sistem sewa atau berbayar. Angkutan umum penumpang diantaranya adalah angkutan kota seperti bus, mini bus, kereta api, angkutan air dan angkutan udara [8].

Berdasarkan Peraturan Pemerintah No. 41 Tahun 1993 tentang Angkutan Jalan dijelaskan angkutan adalah pemindahan orang atau barang dari satu tempat ke tempat lain dengan menggunakan kendaraan. Sedangkan kendaraan umum adalah setiap kendaraan bermotor yang disediakan untuk dipergunakan oleh umum dengan dipungut bayaran. Pengangkutan orang dengan kendaraan umum dilakukan dengan menggunakan mobil bus atau mobil penumpang dilayani dengan trayek tetap atau teratur dan tidak dalam trayek. Tujuan utama keberadaan angkutan umum penumpang adalah menyelenggarakan pelayanan angkutan yang baik dan layak bagi masyarakat.

Di Kabupaten Jember dua tipe angkutan yaitu : angkutan pedesaan (angkutan perintis) dan angkutan kota (line). Angkutan Pedesaan merupakan angkutan penumpang yang ditetapkan melayani trayek terminal tipe C. Ciri utama lain yang membedakannya angkutan pedesaan dengan yang lainnya adalah pelayanan lambat, tetapi jarak pelayanan tidak ditentukan.

Berikut adalah angkutan kota dan pedesaan yang ada di Kabupaten Jember :

Tabel 2 Angkutan Umum di Kabupaten Jember

No	Nama	Rute
1	Line A	<p>Berangkat : Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Sultan Agung - Jl. Sudirman - Jl. Wijaya Kusuma - Jl. PB. Sudirman - Jl. M Seruji - Jl. Slamet Riyadi - Jl. Supriadi - Jl. S Agung - Terminal Arjasa</p> <p>Kembali : Terminal Arjasa - Jl. S Agung - Jl. Supriadi - Jl. Slamet Riadi - Jl. M. Seruji - Jl. PB Sudirman - Jl. A Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Mangli - Terminal Tawang Alun</p>
2	Line B	<p>Berangkat: Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Mawar - Jl. Kenanga - Jl. St. Agung - Jl. Sudarman - Jl. PB. Sudirman - Jl. M. Seruji - Jl. Slamet Riyadi - Jl. Supriyadi - Jl. S Agung - Terminal Arjasa</p>

No	Nama	Rute
		<p>Kembali: Terminal Arjasa - Jl. S Agung - Jl. Supriyadi - Jl. Slamet Riyadi - Jl. M. Seruji - Jl. PB Sudirman - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
3	Line D	<p>Berangkat: Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Sultan Agung - Jl. Sudirman - Jl. Wijaya Kusuma - Jl. PB. Sudirman - Jl. M Seruji - Jl. Mastrip - Jl. Kalimantan - Jl. Jawa - Jl. Karimata - Jl. Letjen. S. Parman - Jl. Letjen Haryono - Jl. Letjen Katamso - Terminal Pakusari</p> <p>Kembali: Terminal Pakusari - Jl. Letjen Katamso - Jl. Letjen Haryono - Jl. Letjen S. Parman - Jl. Karimata - Jl. Jawa - Jl. Kalimantan - Jl. Mastrip - Jl. PB Sudirman - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
4	Line E	<p>Berangkat: Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Sultan Agung - Jl. Sudirman - Jl. A. Yani - Jl. Letjen Panjaitan - Jl. Letjen. S. Parman - Jl. Letjen Haryono - Jl. Letjen Katamso - Terminal Pakusari</p> <p>Kembali: Terminal Pakusari - Jl. Letjen Katamso - Jl. Letjen Haryono - Jl. Letjen S. Parman - Jl. Letjen Panjaitan - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
5	Line G	<p>Berangkat: Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Sultan Agung - Jl. Sudirman - Jl. A. Yani - Jl. Letjen Suprpto - Jl. Jend. Basuki Rahmat - Jl. Gladak Pakem - Jl. MH. Thamrin - Terminal Ajung</p> <p>Kembali: Terminal Ajung - Jl. MH Thamrin - Jl. Gladak Pakem - Jl. Jend. Basuki Rahmat. - Jl Letjen Suprpto - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
6	Line H	<p>Berangkat : Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada</p>

No	Nama	Rute
		<p>Kembali : Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
7	Line K	<p>Berangkat : Terminal Arjasa - Jl. S Agung - Jl. Supriadi - Jl. Slamet Riadi - Jl. M. Seruji - Jl. Mastrip - Jl. Kalimantan - Jl. Jawa - Jl. Karimata - Jl. Letjen. S. Parman - Jl. Letjen Haryono - Jl. Letjen Katamso - Terminal Pakusari</p> <p>Kembali : Terminal Pakusari - Jl. Letjen Katamso - Jl. Letjen Haryono - Jl. Letjen S. Parman - Jl. Karimata - Jl. Jawa - Jl. Kalimantan - Jl. Mastrip - Jl. M. Seruji - Jl. Slamet Riyadi - Jl. Supriyadi - Jl. S Agung - Terminal Arjasa</p>
8	Line L	<p>Berangkat : Terminal Tawang Alun - Jl. Brawijaya - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Melati - Jl. Manggar - Jl. Cinderawasih - Jl. dr. Soebandi - Jl. Slamet Riyadi - Jl. Supriyadi - Jl. S Agung - Terminal Arjasa</p> <p>Kembali : Terminal Arjasa - Jl. S Agung - Jl. Supriyadi - Jl. Slamet Riyadi - Jl. dr. Soebandi - Jl. Cinderawasih - Jl. Manggar - Jl. Melati - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Brawijaya - Terminal Tawang Alun</p>
9	Line N	<p>Berangkat : Terminal Ajung - Jl. MH. Thamrin - Jl Otisata - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. St. Agung - Jl. Sudarman - Jl. Wijaya Kusuma - Jl. PB. Sudirman - Jl. M Seruji - Jl. Slamet Riyadi - Jl. Supriyadi - Jl. S. Agung - Terminal Arjasa</p> <p>Kembali : Terminal Arjasa - Jl. S. Agung - Jl. Supriyadi - Jl. Slamet Riyadi - Jl. M Seruji - Jl. PB. Sudirman - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Otista - Jl. MH. Thamrin - Terminal Ajung</p>
10	Line O	<p>Berangkat : Terminal Ajung - Jl. MH Thamrin - Jl. Otista - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. Melati - Jl. Kenanga - Jl. St. Agung - Jl. Sudarman - Jl. PB. Sudirman - Jl. M. Seruji - Jl. Slamet Riyadi - Jl. Supriyadi - Jl. S. Agung - Terminal Arjasa</p>

No	Nama	Rute
		Kembali : Terminal Arjasa - Jl. S.Agung - Jl. Supriyadi - Jl. Slamet Riyadi - Jl. M Seruji - Jl. PB. Sudirman - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokrp Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Otista - Jl. MH. Thamrin - Terminal Ajung
12	Line Q	Berangkat : Terminal Ajung - Jl. MH Thamrin - Jl. Otista - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. St. Agung - Jl. Wijaya Kusuma - Jl. PB Sudirman - Jl. Moch. Seruji - Jl. Mastrip - Jl. Kalimantan - Jl. Jawa - Jl. Karimata - Jl. S. Parman - Jl. MT Haryono - Jl. Katamso - Terminal Pakusari Kembali : Terminal Pakusari - Jl. Katamso - Jl. MT. Hariyono - Jl. S. Parman - Jl. Karimata - Jl. Jawa - Jl. Mastrip - Jl. Moch Seruji - Jl. PB. Sudirman - Jl. A. Yani - Jl. Trunojoyo - Jl. Cokro Aminoto - Jl. Gajah Mada - Jl. Hayam Wuruk - Jl. Otisata - Jl. MH. Thamrin - Terminal Ajung
13	Line R	Berangkat : Terminal Ajung- Jl. MH Thamrin - Jl. Otista - Jl. Hayam Wuruk - Jl. Gajah Mada - Jl. St. Agung - Jl. A. Yani - Jl. Panjaitan - Jl. S. Parman - Jl. MT Haryono - Jl. Katamso - Terminal Pakusari Kembali : Terminal Pakusari - Jl. Katamso - Jl. MT. Hariyono - Jl. S. Parman - Jl. Panjaitan Jl. A. Yani Jl. Trunojoyo Jl. Cokro Aminoto Jl. Gajah Mada Jl. Hayam Wuruk Jl. Otista Jl. MH. Thamrin Terminal Ajung
14	Perintis 1	Terminal Tawang Alun - Jubung - Ajung - Pancakarya - Klompangan - Sukamakmur - Jenggawa - Wonojati - Karanganyar - Ambulu - Andongsari - Sidodadi - Wonosari - Curah - Nongko – Andongrejo
15	Perintis 2	Terminal Tawang Alun - Ajung - Ambulu - Pantai Payangan - Pantai Watu Ulo - Pantai Papuma PP

2.5. Aplikasi

Aplikasi adalah sebuah perangkat lunak yang dibuat dengan tujuan untuk dapat membantu pengguna dalam menyelesaikan atau mempercepat penyelesaian masalah. Aplikasi adalah sebuah perangkat lunak yang menjadi front end dalam sebuah sistem yang digunakan untuk mengolah data menjadi suatu informasi yang

berguna untuk orang-orang dan sistem yang bersangkutan [9]. Contoh aplikasi yang dapat membantu pekerjaan yang biasa dilakukan adalah pengolah kata, sebelum adanya aplikasi pengolah kata, manusia mengolah kata dengan cara ditulis seperti pembuatan surat yang di mana apabila ada kesalahan maka harus menulis ulang, lalu mulai muncul mesin ketik yang digunakan untuk membantu proses pengolahan data, namun ketika ada kesalahan maka harus mengulang proses pengetikan berbeda dengan aplikasi pengolah kata, ketika ada salah pengetikan maka hanya perlu menghapus bagian yang akan dirubah. Contoh lain adalah aplikasi pencarian tempat yang akan ditujuh berbasis android, aplikasi ini berguna untuk mempermudah pencarian lokasi dengan dibantu dengan penunjuk arah yang akan mengarahkan pengguna ke lokasi yang dituju.

2.5.1. Aplikasi Berdasarkan Fungsi

Aplikasi dapat dibagi menjadi beberapa berdasarkan fungsinya, yaitu :

1. Word Processing

Word Processing adalah aplikasi yang berfungsi untuk pengolahan kata seperti menulis dokumen atau surat.

Contoh : *Microsoft Word, Google Docs.*

2. Desktop Publishing

Desktop Publishing adalah aplikasi yang berfungsi untuk membatu penulisan naskah.

Contoh: *Final Draft, Trelby.*

3. Program Speadsheet

Program *Speadsheet* adalah aplikasi yang berfungsi untuk perhitungan matematika dengan layout baris dan kolom.

Contoh: *Microsoft Excel, Google Sheets.*

4. Database Management System

Database Management System adalah aplikasi yang berfungsi untuk pengolah basis data. Pengolahan ini dapat berupa penambahan, pengubahan, pembacaan dan penghapusan data.

Contoh: *Microsoft Access, phpMyAdmin.*

5. *Graphics*

Graphics adalah aplikasi yang berfungsi untuk pengolahan citra.

Contoh : *Adobe Photoshop, Paint.*

6. Program Akuntansi

Aplikasi yang berfungsi untuk mengolah data akuntansi dan keuangan.

Contoh : *ZipBooks, Zoho Books.*

7. Program Statistik

Aplikasi yang berfungsi untuk mengolah data yang mampu menghasilkan informasi yang berguna.

Contoh : *SPSS, Sttata, GraphPad.*

8. *Communication*

Communication adalah aplikasi yang berfungsi membantu dalam hubungan komunikasi jarak jauh antar dua atau lebih orang dalam waktu yang bersamaan atau berbeda.

Contoh: *Skype, Whatsapp.*

9. Multi media

Multi media adalah aplikasi yang berfungsi untuk pekerjaan multimedia seperti suara, video, teks dan gambar.

Contoh : *Adobe Premiere, Blender.*

10. *Game*

Game adalah aplikasi yang berfungsi untuk melakukan permainan yang menghibur dengan media video.

Contoh : *Dota 2, World of Warcraft.*

11. Anti Virus

Anti Virus adalah aplikasi yang berfungsi untuk mendeteksi dan membersihkan virus pada komputer.

Contoh : *Smadav, Windows Defender.*

2.6. Android

Android adalah sistem operasi untuk mobile yang ditujukan untuk perangkat layar sentuh. Android menyediakan platform terbuka bagi para pengembang untuk

membuat aplikasi. Aplikasi-aplikasi yang dibangun menggunakan Bahasa pemrograman *java*. Pada awalnya android dikembangkan oleh sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk mobile yaitu Android Inc yang kemudian dibeli oleh *Google Inc*. Saat ini android sudah mengeluarkan versi ke 8.0 dan terus berkembang. [10]

2.6.1. Versi Android

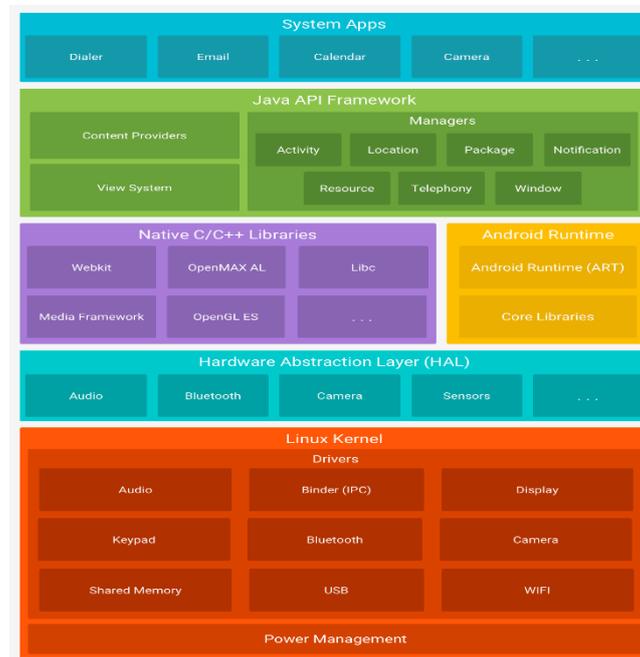
Android mengeluarkan versi 1.0 pada tanggal 23 September 2008 yang merupakan versi pertama dari android dan hingga saat ini adalah versi 8.0 yang keluar pada 21 Agustus 2017.

Berikut daftar versi android dapat dilihat pada tabel 3 :

Tabel 3 Daftar Versi Android

Nama Versi	Versi	Tanggal Rilis
<i>(No Codename)</i>	1.0	23 September 2008
<i>(Internally known as "Petit Four")</i>	1.1	9 February 2009
<i>Cupcake</i>	1.5	27 April 2009
<i>Donut</i>	1.6	15 September 2009
<i>Éclair</i>	2.0	26 Oktober 2009
<i>Froyo</i>	2.2	20 Mei 2010
<i>Gingerbread</i>	2.3	6 Desember 2010
<i>Honeycomb</i>	3.0	22 Februari 2011
<i>Ice Cream Sandwich</i>	4.0	18 Oktober 2011
<i>Jelly Bean</i>	4.1	9 Juli 2012
<i>Kitkat</i>	4.4	31 Oktober 2013
<i>Lollipop</i>	5.0	12 November 2014
<i>Marshmallow</i>	6.0	5 Oktober 2015
<i>Nougat</i>	7.0	22 Agustus 2016
<i>Oreo</i>	8.0	21 Agustus 2017

Android merupakan tumpukan perangkat lunak berbasis Linux yang dibuat untuk berbagai perangkat dan factor bentuk. Diagram berikut menunjukkan komponen besar dari platform Android :



Sumber : <https://developer.android.com/guide/platform/>

Gambar 3 Tumpukan perangkat lunak android

Berikut adanjelasan dari tumpukan perangkat lunak android :

1. ***Linux Kernel***

Linux Kernel adalah fondasi platform android yang berfungsi threading dan manajemen memori tingkat rendah. Menggunakan kernel linux memungkinkan android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen untuk mengembangkan perangkat keras.

2. ***Hardware Abstraction Layer (HAL)***

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja *Java API*. HAL terdiri dari beberapa modul yang masing-masing mengimplementasikan antarmuka untuk komponen perangkat tertentu.

3. ***Android Runtime***

Untuk perangkat yang menjalankan Android versi 5.0 (*API level 21*) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan

tahap *Android Runtime* (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi *file* DEX, format *bytecode* yang didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal.

4. **Pustaka C/C+ Asli**

Komponen dan layanan sistem android banyak dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja *Java API* mengekspos fungsionalitas beberapa pustaka asli pada aplikasi.

5. **Kerangka Kerja Java API**

Keseluruhan fitur Android OS tersedia melalui API yang ditulis dalam Bahasa *Java*. API ini membentuk elemen dasar yang diperlukan untuk membuat aplikasi Android.

6. **Aplikasi Sistem**

Android dilengkapi dengan aplikasi inti untuk email, sms, kalender, internet, kontak, dan lainnya. Aplikasi tidak memiliki status khusus yang ingin dipasang pengguna, sehingga beberapa aplikasi pihak ketiga dapat menjadi aplikasi utama.

2.7. *Global Positioning System (GPS)*

GPS (*Global Positioning System*) adalah sistem navigasi berbasis satelit yang bisa saling berhubungan yang berada diorbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Departemen of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *reciver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi diubah menjadi titik yang dikenal dengan nama *Way-point* nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian dilayar pada peta elektronik.

Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer sekarang mulai terbuka untuk publik. Satelit ini mengorbit pada ketinggian sekitar

12.000 mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau area *coverage* yang lebih luas. Satelit ini akan selalu berada diposisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya *blank spot* (area yang tidak terjangkau oleh satelit). Ada beberapa kategori GPS untuk pemakaiannya [11]:

1. **Lokasi**

Digunakan untuk menentukan dimana lokasi suatu titik dipermukaan bumi berada.

2. **Navigasi**

Membantu dalam mencari lokasi suatu titik di bumi.

3. **Tracking**

Membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat.

4. **Timing**

Dapat dijadikan dasar penentuan jam seluruh dunia, karena memakai jam atom yang jauh lebih presisi dibanding dengan jam biasa.

Berikut contoh sintak teknologi gps dalam bahasa pemrograman *Java* :

```

1. public void onMapReady(GoogleMap googleMap) {
2.     if (ActivityCompat.checkSelfPermission(this,
3.         Manifest.permission.ACCESS_FINE_LOCATION)
4.         != PackageManager.PERMISSION_GRANTED
5.         && ActivityCompat.checkSelfPermission(this,
6.         Manifest.permission.ACCESS_COARSE_LOCATION)
7.         != PackageManager.PERMISSION_GRANTED) {
8.         ActivityCompat.requestPermissions(this,
9.         new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
10.        LOCATION_REQUEST);
11.        return;
12.    }
13.    googleMap.setMyLocationEnabled(true);
14. }
15.
16. @SuppressWarnings("MissingPermission")
17. @Override
18. public void onRequestPermissionsResult(int requestCode, @NonNull String[]
19. permissions, @NonNull int[] grantResults) {
20.     switch (requestCode) {
21.     case LOCATION_REQUEST:
22.         if (grantResults.length > 0 && grantResults[0]
23.             == PackageManager.PERMISSION_GRANTED) {
24.             googleMap.setMyLocationEnabled(true);
25.         }
26.         break;
27.     }
28. }

```

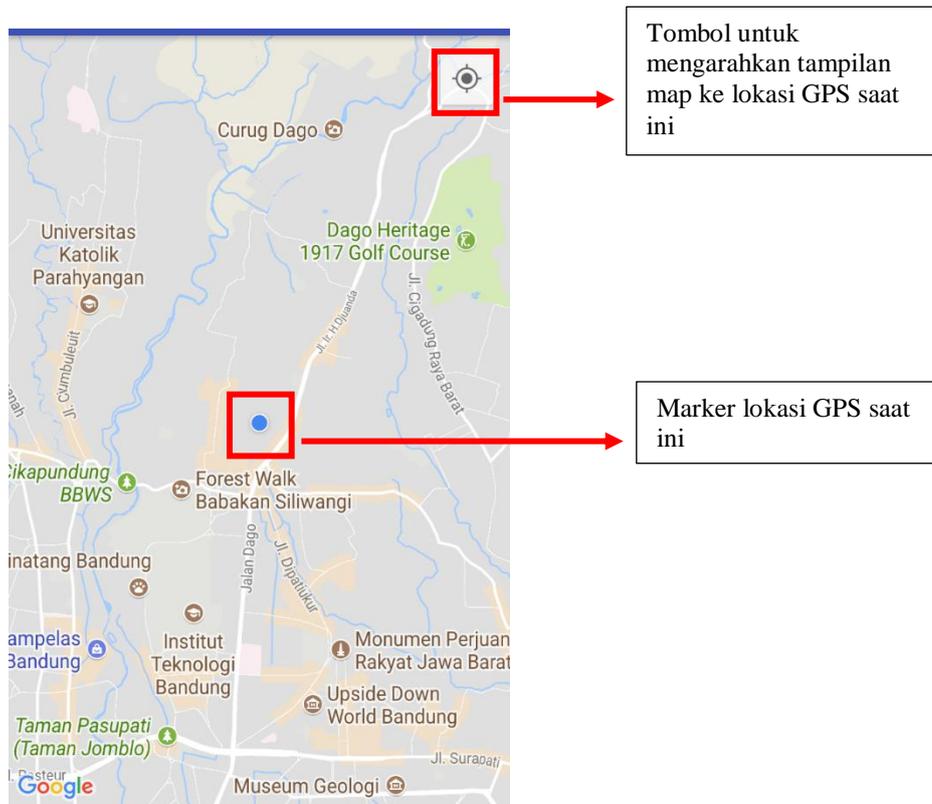
Gambar 4 Request current location use GPS Android

Untuk penggunaan gps pada smartphone membutuhkan permission penggunaan contoh permission pada android :

```
1. <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Gambar 5 Permission request pada android bahasa pemrograman java

Contoh hasil tampilan request *current location* dengan GPS Android



Gambar 6 Hasil request lokasi GPS Smartphone

2.8. Web Services

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service*

menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

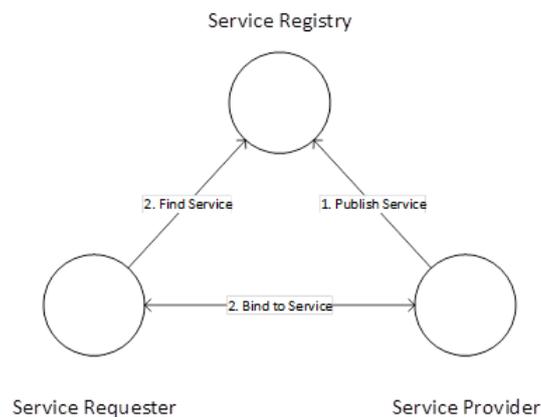
Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya. Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut :

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi *firewall*.

2.8.1. Arsitektur *Web Service*

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. *Service Requester* (peminta layanan)
2. *Service Provider* (penyedia layanan)
3. *Service Registry* (daftar layanan)



Sumber: Teknik Pemrograman Web Service PHP

Gambar 7 Arsitektur *Web Services*

- Service Provider* berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
- Service Registry* berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-*register*.
- Service Requestor* meminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut. [12]

2.9. Java

Java menurut definisi Sun merupakan sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer stand alone ataupun lingkungan jaringan. Bahasa pemrograman *java* ini bersifat *case sensitive*, sehingga harus lebih memperhatikan bentuk tertentu, sehingga dalam menuliskan semua baris *code* tersebut dalam satu baris tersendiri [13].

Pada bulan Agustus 1991, *java* dikembangkan dengan nama “*Java 2*” yang merupakan generasi kedua dari *java* platform. Pergantian nama baru tersebut diterapkan di setiap semua edisi *Java* diantaranya : *Standard Edition* (J2SE), *Enterprise Edition* (J2EE), dan *Micro Edition* (J2ME) [13].

Java memiliki tiga *platform* yang masing-masing mengarahkan sebagai tujuan tertentu dan sebagai lingkungan komputasi yang berbeda-beda :

1. ***Standard Edition (J2SE)***

J2SE merupakan inti dari sebuah bahasa pemrograman *Java*. J2SE didesain sebagai jalan pada komputer desktop dan komputer *workstations*.

2. ***Enterprise Edition (J2EE)***

J2EE berfungsi sebagai built-in dan mendukung untuk *servelts*. JSP dan XML, edisi ini bertujuan untuk aplikasi yang berbasis *server*.

3. ***Micro Edition (J2ME)***

J2ME didesain untuk piranti dengan keterbatasan memori, layar *display* terbatas dan power pemrosesannya terbatas juga.

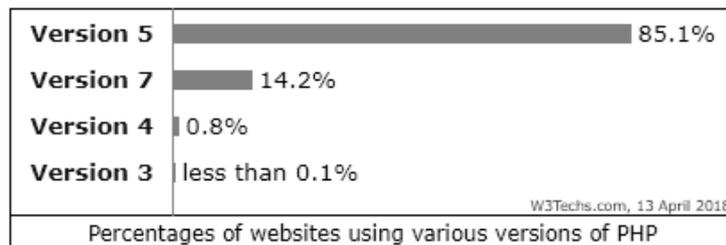
Java dikenal sebagai bahasa pemrograman yang bersifat *strongly*, yang harus mendeklarasikan tipe data dari semua *variable* yang apabila salah dalam mengikuti aturan pendeklarasian *variable*, maka akan terjadi *error* pada saat proses kompilasi. Tipe data yang terdapat pada bahasa pemrograman *Java* pada umumnya yakni terdiri dari *integer*, *floating point*, *char*, dan *boolean* [13].

Dan untuk aturan penulisan pada bahasa pemrograman *Java* tidak jauh berbeda dengan bahasa pemrograman yang lain yakni harus diawali dengan huruf atau abjad, karakter mata uang, dan *underscore* (*_*) dan terdiri dari huruf atau abjad, angka, dan *underscore* (*_*). Dan tidak boleh menggunakan kata-kata yang dikenal oleh bahasa pemrograman *Java* (*keyword*), seperti *byte*, *case*, *int*, dan lainnya [13]. Penggunaan bahasa pemrograman *Java* pada pembangunan aplikasi perencanaan dan panduan wisata di Kabupaten Jember, digunakan untuk membangun aplikasi untuk bagian *backend*.

2.10. PHP

PHP adalah sebuah bahasa pemrograman sumber terbuka yang ditujukan untuk pemrograman web dan dapat diaplikasikan ke *HTML*. *PHP* merupakan bahasa skrip yang ditanam dalam *HTML*. Ini berarti bahwa anda dapat menggabungkan kode *PHP* dan *HTML* dalam berkas yang sama [14]. *PHP* merupakan bahasa pemrograman yang akan menjalankan program di *server* dan hasilnya diintegrasikan ke dalam kode *HTML*. *PHP* diciptakan untuk membuat halaman *web* yang interaktif.

PHP cukup dikenal oleh banyak kalangan pemrogram *web*. Menurut W3Techs lembaga survei teknologi *web*, menyatakan 83.3% *website* menggunakan bahasa *PHP* [15]. *PHP* versi 5 adalah yang paling banyak digunakan saat ini, diikuti dengan versi 7 lalu versi 4 dan 3. Berikut adalah perbandingan penggunaan versi *PHP* yang dapat dilihat pada Gambar 4 :



Sumber gambar: <http://w3techs.com>

Gambar 8 Perbandingan pengguna versi *PHP*

Pemrograman *PHP* diapit dengan sintak `<?php ?>` dan ekstensi format *file php* adalah *.php*. *PHP* dapat digabung dengan *HTML*, namun hanya akan membaca kode yang berada di antara sintak `<?php ?>`, hasil pemrosesan akan ditampilkan dalam kode *HTML*. Contoh sebuah program *PHP* :

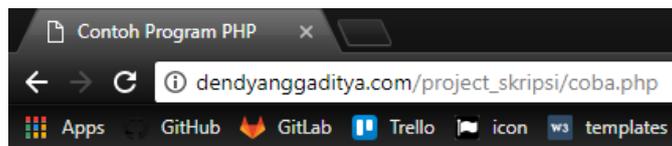
```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>Contoh Program PHP</title>
5. </head>
6. <body>
7. <br>
8.     <?php
9.         echo 'Sekarang tanggal : ' . date('j F Y') . ' .';
10.    ?>
11. <br>
12. </body>
13. </html>

```

Gambar 9 Contoh bahasa pemrograman *PHP*

Program diatas, setelah dibuka di *browser* akan menghasilkan :



Sekarang tanggal : 7 April 2018.

Gambar 10 Hasil dari contoh pemrograman PHP

2.11. MySQL

MySQL adalah aplikasi basis data yang dijalankan di *server*, MySQL menggunakan sintak SQL dalam penggunaannya. MySQL bersifat relasional yang memungkinkan satu tabel dapat berelasi dengan yang lain, sehingga memungkinkan dilakukan normalisasi dengan tujuan untuk mengurangi redundansi. MySQL menggunakan *InnoDB* mampu menampung sebanyak 1017 kolom dan besar data mencapai 256TB [16].

Dalam penggunaan database MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL terdapat 3 sub perintah, yaitu *DDL (Data Definition Language)*, *DML (Data Manipulation Language)* dan *DCL (Data Control Language)*.

1. *DDL (Data Definition Language)*

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri merupakan kependekan dari apa yang dikenal dengan nama Data Definition Language. DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur database. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. *CREATE* berfungsi untuk membuat database baru, tabel baru, *view* baru dan kolom.
2. *ALTER* berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. *DROP* berfungsi untuk menghapus database dan tabel.
4. *TRUNCATE* berfungsi untuk Menghapus semua catatan dari tabel.

5. *COMMENT* berfungsi untuk Menambahkan komentar pada data.

6. *RENAME* berfungsi untuk mengubah nama obyek.

2. *DML (Data Manipulation Language)*

Data Manipulation Language (DML) adalah perintah yang digunakan untuk mengelolah data dalam database. Terdapat beberapa perintah DML pada MySQL sebagai berikut :

1. *SELECT* untuk menganampikan data dari database.

2. *INSERT* untuk menambah data pada database.

3. *UPDATE* untuk merubah data dalam tabel.

4. *DELETEI* untuk menghapus data dari tabel.

5. *CALL* untuk memanggil subprogram SQL atau *Java*

6. *EXPLAIN PLAN* untuk menjelaskan jalur akses ke data.

7. *LOCK TABEL* untuk mengunci tabel.

3. *DCL (Data Control Language)*

Data Control Language (DCL) ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah DCL pada MySQL sebagai berikut :

1. *GRANT* berfungsi untuk memberikan hak akses pengguna ke database.

2. *REVOKE* berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah *GRANT* [17].

Berikut adalah contoh sintak dasar yang perlu diketahui untuk mengelola basis data MySQL:

4. Membuat database :

```
1. create database db_mahasiswa;
```

Gambar 11 Membuat Database di MySQL

5. Membuat tabel :

```
1. create tabel mahasiswa(
2. id_mhs int(4) auto_increment primary key,
3. nim varchar(12),
4. nama_depan varchar(25) not null,
5. nama_belakang varchar(25) not null,
7. tanggal datetime default CURRENT_TIMESTAMP
8. )
```

Gambar 12 Membuat Tabel di MySQL

6. Menambah data :

```
1. insert into mahasiswa(nim,nama_depan,nama_belakang)
2. value('10113064', 'Dendy', 'Anggaditya');
```

Gambar 13 Menambah data di MySQL

7. Mengambil data :

```
1. select * from mahasiswa;
```

Gambar 14 Mengambil data di MySQL

8. Mengubah data :

```
1. update mahasiswa set nama_belakang = 'Yogaswara'
2. where nim = '10113064';
```

Gambar 15 Mengubah data di MySQL

9. Menghapus data :

```
1. delete from mahasiswa where nim = '10113064';
```

Gambar 16 Menghapus data di MySQL

10. Menghapus tabel :

```
1. drop tabel mahasiswa;
```

Gambar 17 Menghapus tabel di MySQL

2.12. JSON

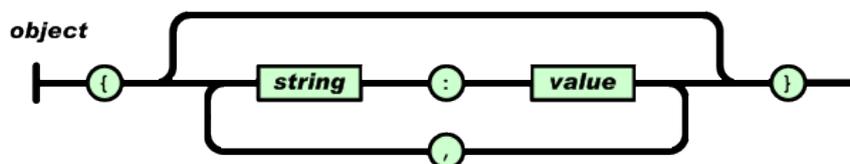
JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan. Sangat mudah dibaca dan menulisnya, serta mudah bagi komputer untuk mem-parsing dan menghasilkan. *JSON* format teks yang bebas Bahasa tetapi menggunakan konversi seperti C, C++, C#, *Java*, *JavaScript*, Perl, Python, dan banyak lainnya. *JSON* kebanyakan juga sebagai Bahasa pertukaran data. *JSON* di bangun dengan dua struktur :

1. Kumpulan nama atau nilai. Dalam berbagai bahasa, ini direalisasikan sebagai *object*, *record*, *struct*, *dictionary*, *hash tabel*, *keyed list* atau *associative array*.
2. Daftar nilai yang terurut. Sebagian besar bahasa, ini direalisasikan sebagai *array*, *vector*, *list* atau *sequence*.

Struktur tersebut merupakan struktur data universal. Hampir semua Bahasa pemrograman yang modern sudah mendukungnya dalam satu bentuk atau lainnya. Masuk akal bahwa format data yang dapat dipertukarkan dengan bahasa pemrograman juga didasarkan pada struktur ini. *JSON* memiliki format – format tipe data seperti berikut [18] :

1. *object*

Berikut adalah format *JSON object* yang dapat dilihat pada Gambar 19 :



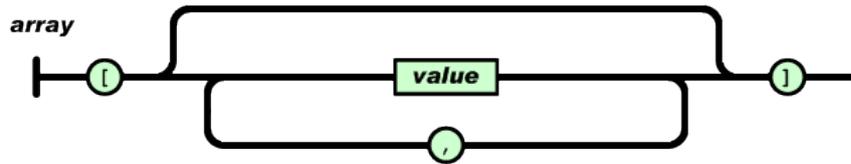
Sumber Gamber : <http://json.org/>

Gambar 18 Format *JSON object*

Object adalah kumpulan dari pasangan nama atau nilai tak berurutan. *Object* dimulai dengan “{” (kurung kurawal kiri) dan diakhiri dengan “}” (kurung kurawal kanan). Setiap nama diikuti oleh “:” (titik dua) dan pasangan nama atau nilai dipasangkan oleh “,” koma.

2. *array*

Berikut adalah format *JSON array* yang dapat dilihat pada Gambar 20 :



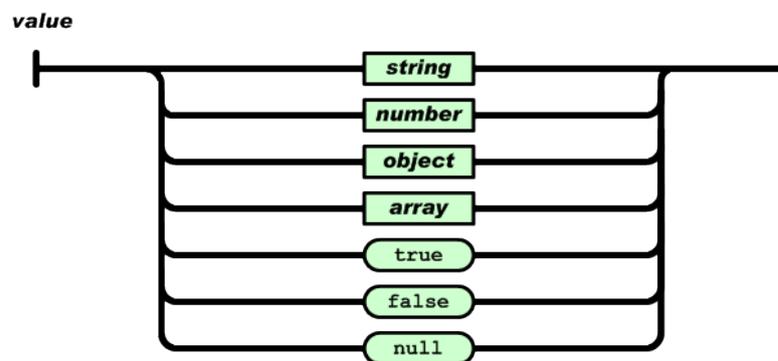
Sumber Gamber : <http://json.org/>

Gambar 19 Format *JSON array*

Array adalah kumpulan nilai yang diurutkan. *Array* dimulai dengan “[“ (kurung siku kiri) dan diakhiri dengan “]” (kurung siku kanan). Nilai dipisahkan oleh “,” (koma).

3. *value*

Berikut adalah format *JSON value* yang dapat dilihat pada Gambar 21 :



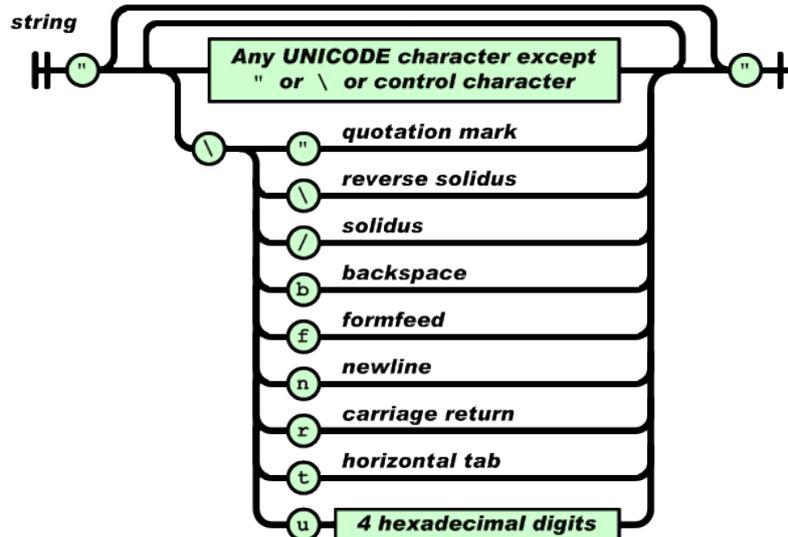
Sumber Gamber : <http://json.org/>

Gambar 20 Format *JSON value*

Value (nilai) dapat berupa *string* dalam tanda kutip ganda, *number*, *true* atau *false* atau *null*, *object* atau *array*. Struktur-struktur ini dapat diulang.

4. *string*

Berikut adalah format *JSON string* yang dapat dilihat pada Gambar 22 :



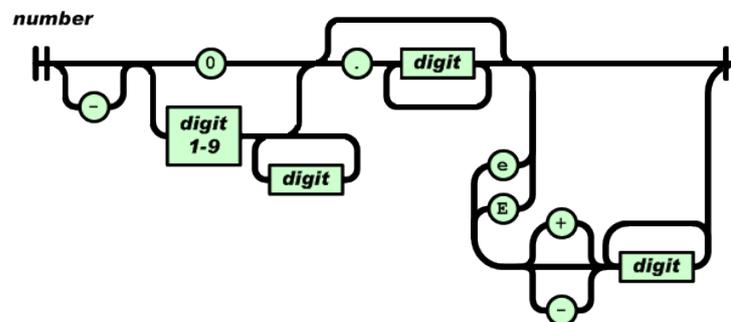
Sumber Gambar : <http://json.org/>

Gambar 21 Format *JSON string*

String adalah urutan karakter kosong atau karakter *unicode* yang diapit oleh tanda kutip dua atau karakter khusus diawali dengan karakter “\” (garis miring terbalik). Karakter direpresentasikan sebagai *string* karakter tunggal. *String* sangat mirip dengan *string C* atau *Java*.

5. *number*

Berikut adalah format *JSON number* yang dapat dilihat pada Gambar 23 :



Sumber Gambar : <http://json.org/>

Gambar 22 Format *JSON number*

Number sangat mirip dengan angka *C* atau *Java*, kecuali format *octal* dan *heksadesimal* tidak digunakan.

2.13. Google Maps API

Google Maps API adalah sebuah *library* yang berasal layanan dari *Google Maps* yang berbentuk *JavaScript*, yang berisi fungsi-fungsi pemrograman yang bisa diintegrasikan kedalam web atau aplikasi yang sedang dibuat [19]. Hanya dengan menggunakan *Key Google Maps API*, aplikasi yang dibuat dapat mengakses *maps* pada *Google Maps*.

Google Maps API dapat digunakan secara gratis dan tidak perlu untuk mengeluarkan biaya sebagai lisensi. Tetapi *request* maksimal pada sebuah peta hanya diperbolehkan sebanyak 2500 *request/* hari. Jika lebih pengguna harus membeli lisensi *Google Maps API* untuk bisnis. Karena *Google Maps API* berbasis Web, maka untuk pembuatan aplikasinya hanya perlu menggunakan *tool Text Editor*, dan perangkat lain yang harus disiapkan yaitu *Browser*, dan koneksi internet.

Aplikasi perencanaan dan panduan wisata ini pun membutuhkan sebuah tampilan peta sebagai media informasi sebuah lokasi yang akurat maka dari itu dengan menerapkan teknologi *Google Maps API* pada aplikasi yang akan dibangun, selain bisa menampilkan lokasi geografis pada peta *Google*. Pada teknologi *Googel Maps API* ini pun bisa mengidentifikasi keberadaan si pengguna aplikasi itu berada.

Berikut adalah contoh cara untuk mendapatkan *Google Map API key* yang dapat di lihat pada Gambar 24 :

API key

AIzaSyC02xKA_hb1mvWdYcGj2JXcBy0ChaD_xT4

Name

Project

Application restrictions

None

HTTP referrers (web sites)

IP addresses (web servers, cron jobs, etc.)

Android apps

iOS apps

Restrict usage to your Android apps (Optional)

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps

Get the package name from your AndroidManifest.xml file. Then use the following command to get the fingerprint:

```
$ keytool -list -v -keystore mystore.keystore
```

Package name	SHA-1 certificate fingerprint
com.dendyanggaditya.jemberguide	15:CD:42:F9:8A:F3:72:A6:C7:DE:48:21:F1:11:6A:F1:C8:6C:EC:0F
com.dendyanggaditya.jembertrip	15:CD:42:F9:8A:F3:72:A6:C7:DE:48:21:F1:11:6A:F1:C8:6C:EC:0F

+ Add package name and fingerprint

Gambar 23 Pembuatan *Google Map API key*

Berikut adalah penjelasan dari pembuatan *Google Map API key* :

1. *API key* yang di dapat dari pembuatan *API key* pada halaman <https://console.developers.google.com/apis/credentials/key/> yang akan di gunakan pada android.
2. Pemilihan tipe penggunaan API pada sistem *platform* yang akan di bangun.
3. Menambahkan nama *package* dari aplikasi yang sedang atau akan kita buat.

Contoh sintak *Google Map* pada *platform* android dapat dilakukan dengan membuat tampilan *fragment map* yang ditulis dalam bahasa pemrograman *Java* pada Gambar 25 :

```

1.  @Override
2.  public View onCreateView(LayoutInflater inflater, ViewGroup
3.  container, Bundle savedInstanceState) {
4.
5.      View v = inflater.inflate(R.layout.fragment_location,
6.  container, false);
7.      SupportMapFragment mapFragment = (SupportMapFragment)
8.  getChildFragmentManager().findFragmentById(R.id.mapViewLokasi);
9.      mapFragment.getMapAsync(this);
10.     return v;
11. }
12.
13.
14. @Override
15. public void onMapReady(GoogleMap googleMap) {
16.     LatLng openmap = new LatLng(-8.1732, 113.7008);
17.
18.     // For zooming automatically to the location of the marker
19.     CameraPosition cameraPosition = new
20. CameraPosition.Builder().target(openmap).zoom(13.2f).build();
21.
22. googleMap.animateCamera(CameraUpdateFactory
23.     .newCameraPosition(cameraPosition));
24. }

```

Gambar 24 Contoh pembuatan map pada bahasa pemrograman *Java*

2.14. UML

UML (*Unified Modelling Language*) adalah bahasa pemodelan secara visual yang menjadi sarana perancangan sistem berorientasi objek, atau definisi UML yaitu bahasa yang sudah menjadi standar visualisasi, perancangan dan dokumentasi sistem perangkat lunak.

Beberapa tujuan dan fungsi dari penggunaan UML, diantaranya :

1. Dapat memberikan bahasa pemodelan visual kepada pengguna dari berbagai macam pemron proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam pemodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa pemodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.

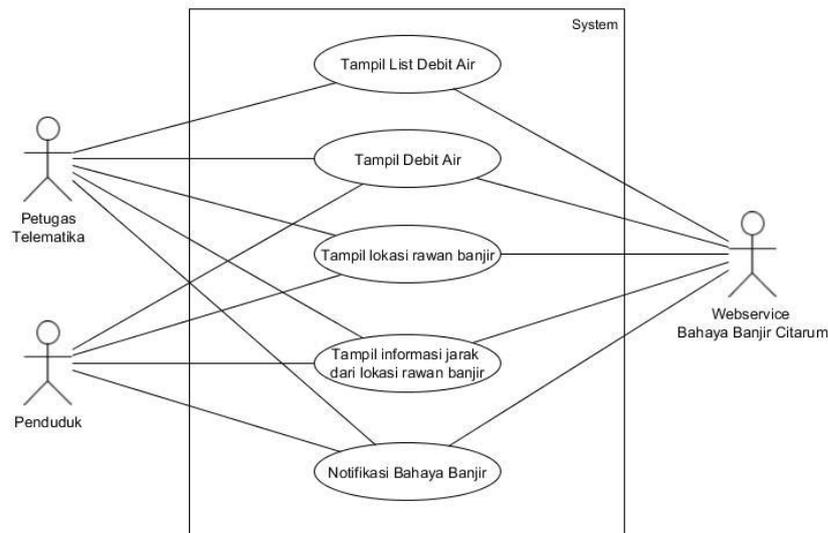
4. Dapat berguna sebagai blueprint, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak saja.
6. Dapat menciptakan suatu bahasa pemodelan yang nantinya dapat dipergunakan oleh manusia maupun mesin.

2.14.2. Use case Diagram

Use case diagram merupakan gambaran fungsionalitas dari beberapa atau semua *actor*, *use case*, dan interaksi untuk memperkenalkan suatu sistem. *Use case* diagram tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberikan gambaran singkat tentang hubungan *use case*, *actor*, dan sistem. Didalam *use case* hanya diketahui fungsi-fungsi yang berkaitan dengan pembangunan sistem yang akan dibuat.

Sebuah *use case* dapat di *include* oleh lebih dari suatu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan menarik keluar fungsionalitas yang *common*. *Use case* juga dapat meng *extend* *use case* lain dengan *behavior*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialis dari yang lain. Diagram *use case* memiliki beberapa *element* yang digunakan untuk menggambarkan rancangan dan penghubung antar komponen, baik *actor* dengan *use case* atau *use case* dengan *use case* yang lainnya.

Berikut contoh perancangan diagram *Use case* :



Gambar 25 Contoh Diagram *Use Case*

2.14.3. Skenario *Use case*

Skenario *Use case* merupakan hasil instansiasi dan penjelasan dari setiap *Use case*. Skenario *Use case* menceritakan detail yang terjadi. Format skenario *Use case* adalah berbentuk tabel.

Komponen-komponen Skenario *Use case* :

1. *Name*: Memberikan penjelasan singkat tentang nama dari *use case*.
2. *Actors*: Daftar aktor yang dapat mengakses *use case*.
3. *Goals*: Menjelaskan apa yang aktor coba dapatkan dari *use case*.
4. *Preconditions*: Kondisi sistem sebelum *use case* dijalankan.
5. *Summary*: Memberikan penjelasan singkat tentang deskripsi informal dari *use case*.
6. *Steps*: Menjelaskan setiap langkah yang dijalankan pada *use case*.
7. *Post conditions*: Kondisi sistem setelah *use case* dijalankan.

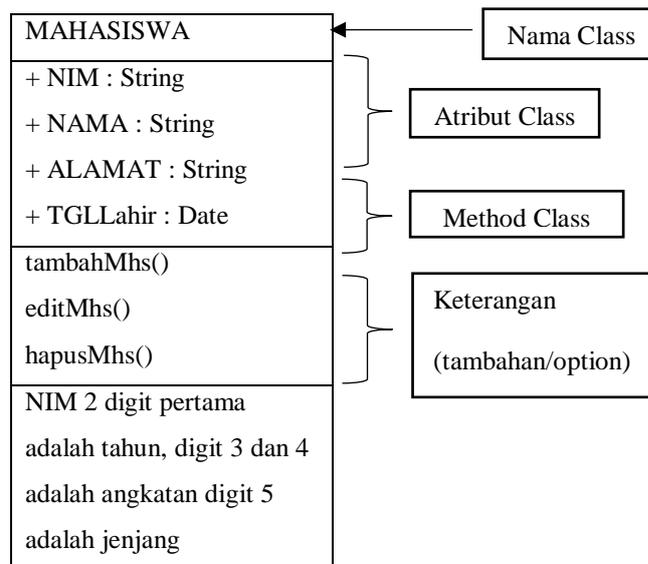
Berikut merupakan contoh tabel dari skenario *use case*:

Tabel 4 Contoh Skenario *Use case* Tambah Data Debit Air

<i>Use case Name</i>	Tambah Data Debit Air	
<i>Goal In Context</i>	Petugas BPBD dapat menambah data debit air.	
<i>Description</i>	Fungsionalitas ini digunakan oleh Petugas BPBD untuk menambah data debit air pada webservice Bahaya Banjir Citarum.	
<i>Related Use case</i>	Request Tambah Data Debit Air	
<i>Successful End Condition</i>	Aktor dapat menambah data debit air ke webservice.	
<i>Failed End Condition</i>	Aktor tidak dapat menambah data debit air ke webservice.	
<i>Actors</i>	Petugas BPBD	
<i>Trigger</i>	Membuka halaman web Tambah Data Debit Air.	
<i>Main Flow</i>	<i>Step</i>	<i>Action</i>
	1.	Petugas BPBD membuka halaman web Tambah Data Debit Air.
	2.	Sistem menampilkan halaman web Tambah Data Debit Air
	3.	Petugas BPBD memilih lokasi rawan banjir.
	4.	Petugas BPBD memasukkan jumlah debit air.
	5.	Petugas BPBD memilih proses kirim.
	6.	Sistem melakukan permintaan penambahan data ke webservice.
<i>Extension</i>	<i>Step</i>	<i>Branching Action</i>
	6.1.	Sistem gagal menambah data debit air ke webservice.

2.14.4. *Class diagram*

Class diagram merupakan elemen terpenting dalam sistem berorientasi objek, kelas mendeskripsikan satu blok pembangun sistem. *Class diagram* memiliki fitur yang memodelkan multiplisitas, ketampakan, penanda, *polymorphism*, dan karakter-karakter lainnya. UML menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas Terdapat empat bagian masing-masing bagan kelas, yaitu : nama, atribut dan perilaku. Gambar berikut merupakan contoh perancangan *class diagram* kelas :



Gambar 26 Contoh *Class diagram*

2.14.5. *Activity diagram*

Activity diagram digunakan untuk menganalisis proses, *activity diagram* biasanya dibuat untuk sebuah *use case* dan menampilkan beberapa skenario [20].

Activity diagram bukan sebuah alat yang sempurna untuk menganalisis masalah dari system. Setiap *activity diagram* mempunyai satu *initial state*. *Activity diagram* dapat diakhiri dengan memberikan *activity final node* yang digambarkan dengan lingkaran padat dengan mempunyai cincin dibagian luarnya.

a. Activity Node

Activity node adalah sesuatu yang dilakukan atau sedang terjadi dalam *Activity diagram*. *Activity node* mempunyai simbol yang hampir mirip dengan *use case* namun mempunyai bentuk yang lebih ramping dan menyerupain bujur sangkar.

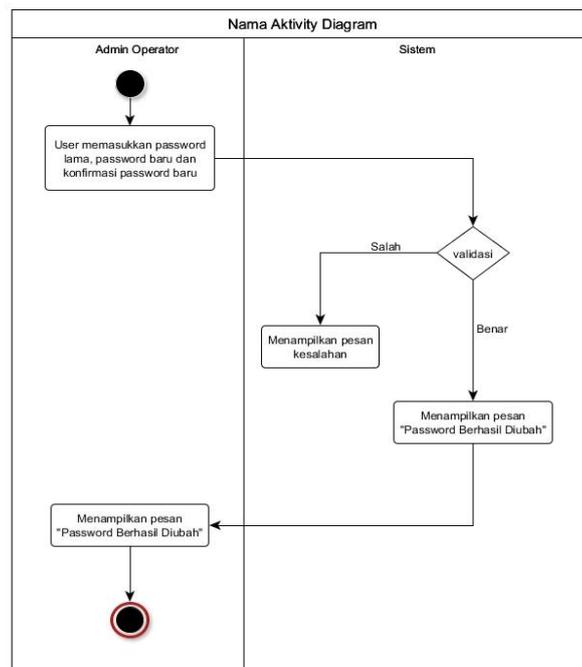
b. Decision Node

Decision node disebut *decision diamonds* di dalam flowchart. Simbol diamond adalah satu elemen yang membuat *activity diagram* mengingat akan *flowchart*, yang berguna untuk memberika kondisi percabangan.

c. Transition Fork

Transition fork untuk menggambarkan tingkah laku yang paralel atau bercabang. Sebuah *transition join* untuk mempertemukan tingkah laku yang paralel.

Berikut merupakan contoh gambar dari *activity diagram*:

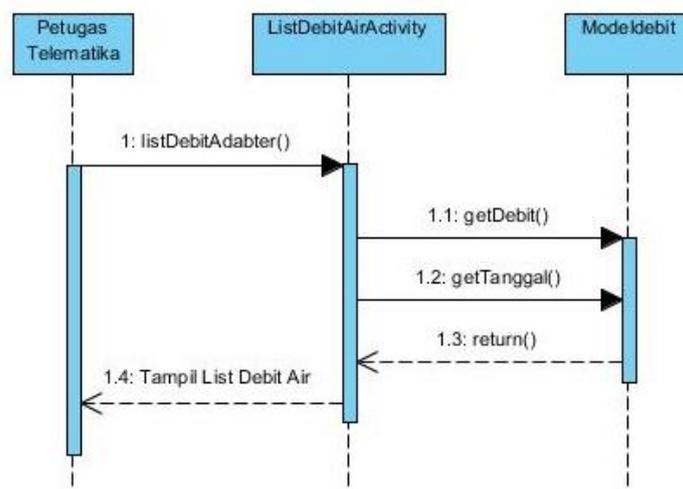


Gambar 27 Contoh Activity diagram

2.14.6. Sequence diagram

Diagram Sequence digunakan untuk memberikan gambaran interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* dapat diturunkan dari usecase dan untuk menurunkan interaksi, relasi, dan metode dari objek sebuah sistem [20]. Diawali dengan men-*trigger* aktivitas, proses dan perubahan yang terjadi secara internal dan output apa yang dihasilkan. Setiap objek termasuk aktor memiliki *lifeline* vertikal. Pesan digambarkan sebagai garis panah dari satu objek ke objek yang lainnya.

Berikut merupakan contoh perancangan *sequence diagram* :



Gambar 28 Contoh *Sequence diagram*