

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan pustaka

Pada tinjauan pustaka peneliti akan memaparkan tinjauan peneliti atas beberapa penelitian dan kajian ilmiah terdahulu serta beberapa konsep yang memiliki keterkaitan dengan penelitian yang akan dilakukan peneliti.

(Utomo, 2008), dalam penelitian yang berjudul Penanganan Pengaduan Masyarakat Menganai Pelayanan Publik (Studi Pada Pusat Penanganan Pengaduan Pelayanan Publik Kota Semarang). Penelitian menggunakan metode pendekatan kualitatif menitik beratkan penelitian pada bagaimana proses pengaduan dilaksanakan dan bagaimana keterlibatan masyarakat di dalamnya. Banyak keluhan melalui kotak saran atau saluran lainnya belum banyak ditanggapi oleh penyedia pelayanan publik di Kota Semarang. Maka dari itu Pemerintah Kota Semarang membentuk unit penanganan pengaduan yang disebut Pusat Penanganan Pengaduan Pelayanan Publik Kota Semarang atau disingkat P5 sebagai instrumen partisipasi masyarakat yang diandalkan dan dapat digunakan untuk memperbaiki pelayanan publik.

Hasil penelitian yang dilakukan Sad Dian Utomo memaparkan bahwa sebagian besar *stake holders* memandang positif dan menilai P5 cukup efektif. Sementara itu dengan membandingkan instrumen partisipasi itu dengan teori *ladder of citizen participation* dari Arnstein dan teori *ladder of citizen empowerment* dari Burn, Hambleton & Hogget dapat disimpulkan bahwa tingkat partisipasi warga masyarakat belum mencapai titik ideal yaitu kontrol masyarakat. Pencapaian tingkat partisipasi dan efektivitas P5 itu dipengaruhi oleh faktor-faktor seperti kepemimpinan, regulasi atau peraturan, kewenangan P5, peran *civil society*, informasi dan momentum partisipasi. Kelebihan dari penelitian ini adalah Sad menggunakan pendekatan kualitatif dalam memaparkan suatu gejala sosial dimana partisipasi masyarakat dibutuhkan dalam upaya memperbaiki kualitas pelayanan publik. Karena suatu gejala sosial sulit untuk diukur dengan pendekatan kuantitatif.

Adapun kekurangannya adalah pemaparan teori yang kurang proporsional dimana lebih banyak teori partisipasi masyarakat sedangkan teori pengaduan masih belum banyak dideskripsikan.

(Nurbaity, 2010) dalam penelitian yang berjudul Pengembangan Sistem informasi kepegawaian (SIMPEG) Berbasis *Web* (Studi Kasis: Subbag Administrasi Kepegawaian Pusat UIN Syarif Hidayatullah Jakarta), UIN Syarif Hidayatulloh Jakarta merupakan institusi perguruan tinggi yang memiliki pegawai yang besar. Subbag administrasi kepegawaian UIN Jakarta memiliki tugas melakukan administrasi mutasi kepegawaian yang meliputi kenaikan pangkat jabatan, kenaikan gaji berkala dan pensiun pegawai. Data-data mengenai kepegawaian sudah berada dalam suatu database dengan program aplikasi Sistem Pegawai (SIMPEG) berbasis *desktop*. Namun, aplikasi tersebut masih memiliki kekurangan diantaranya fitur pembuatan Surat Keputusan (SK), kenaikan pangkat dan kenaikan gaji berkala sehingga pembuatannya dilakukan menggunakan *Excel* dan *Word*. Untuk mengatasi permasalahan tersebut dibuat konsep pengelolaan data pegawai dengan fitur pengelolaan data pegawai, data riwayat pangkat dan jabatan, data kenaikan pangkat pegawai dan kenaikan gaji berkala, data pensiunan dan rekapitulasi pegawai per unit kerja serta laporan yang disediakan dalam bentuk website. Pengembangan Sistem informasi Kepegawaian (SIMPEG) Berbasis *Web* ini dirancang menggunakan *PHP*, *HTML* dan *MYSQL*

Dalam penelitian yang berjudul Pengembangan Aplikasi Administrasi Personalia Berbasis Android (Studi Kasus : PT Indocement Tunggul Prakasa Tbk.), ini dibangun untuk digunakan dalam proses pengaduan keluhan absensi, keluhan cuti, keluhan *overtime* kerja, pendaftaran beasiswa anak, pendaftaran BPJS, pengajuan perubahan *id card*, dan pengumpulan surat keterangan pada PT Indocement Tunggul Prakasa Tbk. tujuannya adalah untuk memberikan kemudahan pada pegawai dalam memenuhi kebutuhan administrasi dan dalam proses pengolahannya.

Sistem dibangun menggunakan *Platform Android* yang memudahkan karyawan dalam melakukan kegiatan administrasi dari manapun pegawai tersebut

berada. *MySql* digunakan untuk pengolahan *database*. Pegawai akan menggunakan akses NIK (Nomor Induk Karyawan) sebagai validasi pengguna Aplikasi Administrasi Personalia adalah pegawai di PT. Indocement Tungal Prakasa Tbk.

Tabel 2.1 Tabel komparasi antar tinjauan pustaka

Varian pembandingan	Sad Dian Utomo (P5)	Nurbaity (SIMPEG)
<i>User interface</i>	Tatap muka	<i>Website</i>
<i>Database</i>	-	MYSQL
Fitur aplikasi	1. Pengaduan masyarakat	1. Pengolahan data pegawai 2. Riwayat kenaikan pangkat dan jabatan 3. Data kenaikan pangkat dan kenaikan gaji berkala 4. Data pensiunan 5. Rekapitulasi pegawai per unit kerja 6. Laporan dalam bentuk <i>website</i>

2.2 Landasan teori

A. Teknologi pengembangan aplikasi

Unified Modelling Language (UML)

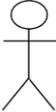
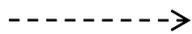
Menurut (Booch, 2005) UML adalah Bahasa standar untuk membuat rancangan *software*. UML biasanya digunakan untuk menggambar dan membangun dokumen artifak dari *software –intensive system*. Model UML yang dipakai dalam pengembangan aplikasi administrasi personalia di PT Indocement Tbk berbasis android antara lain *use case diagram*, *activity diagram*, *class diagram* dan *ER diagram*.

Use case diagram

Menurut (Dittman, 2004) *Use Case Modelling* adalah sebuah pendekatan yang memfasilitasi pengembangan yang berpusat pada penggunaan. *Use Case*

diagram digunakan untuk memodelkan proses berdasarkan perspektif pengguna sistem. *Use Case Diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* mempresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada tabel 2.1.

Tabel 2.2 *Use Case Diagram*

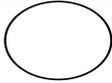
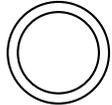
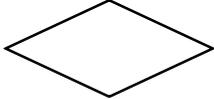
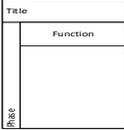
No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan saat berinteraksi dengan <i>use case</i>
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang tidak bergantung pada elemen yang tidak mandiri (<i>independent</i>)
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i>
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6		<i>Association</i>	Apa yang menghubungkan antara suatu objek dengan objek yang lainnya

Activity Diagram

Menurut (Dittman, 2004), *activity diagram* digunakan untuk menggambarkan alur dari proses bisnis atau langkah – langkah usecase secara berurutan. Diagram ini juga digunakan untuk menggambar *action* (tindakan) yang

akan dieksekusikan ketika suatu proses sedang berjalan dan beserta hasil dari proses eksekusi tersebut. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2

Tabel 2.3 Activity Diagram

No	Gambar	Nama Gambar	Keterangan
1		<i>Start Point</i>	Merupakan awal dalam aktifitas
2		<i>End Point</i>	Akhir dalam aktivitas
3		<i>Activities</i>	Menggambarkan proses kegiatan bisnis
4		<i>Decision Point</i>	Menggambarkan pilihan pengambilan keputusan aktifitas.
5		<i>Swimlane</i>	Berfungsi sebagai pembagi <i>activity diagram</i> yang menunjuki siapa yang melakukan aktifitas

Class Diagram

Menurut (Dittman, 2004), *class diagram* menggambarkan struktur objek yang terdapat pada sebuah sistem. Diagram ini menunjukkan objek – objek yang terdapat pada suatu sistem dan relasi antar objek – objek tersebut. *Class* memiliki tiga area pokok yaitu: Nama, Atribut dan Metode / *Operation*. Atribut dan Metoda dapat mempunyai sifat *Private* (tidak dapat dipanggil dari luar class), *Protected* (Hanya dapat dipanggil oleh class yang bersangkutan dan anak yang mewarisinya)

dan *Public* (dapat dipanggil siapa saja). Berikut adalah contoh dari sebuah kelas. *Class Diagram* secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

Tabel 2.4 Class Diagram

Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi / spesialisasi (umum / khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole – part</i>)

B. Aplikasi

Menurut (Supriyanto, 2005) Aplikasi adalah program yang memiliki aktifitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk

mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

C. Android

(wikipedia, 2008) Android adalah sistem operasi *mobile* yang dikembangkan oleh *Google*. Sistem ini digunakan oleh beberapa *smartphone*.

Sistem operasi Android didasarkan pada *kernel Linux* terbuka. Berbeda dengan *OS iPhone*, Android adalah *open source*, yang berarti pengembang dapat memodifikasi dan menyesuaikan *OS* untuk setiap telepon. Oleh karena itu, ponsel berbasis Android yang berbeda mungkin memiliki berbagai antarmuka pengguna grafis *UI* yang beragam meskipun mereka menggunakan *OS* yang sama.

Saat ini android mempunyai beberapa, yaitu:

1. Android v1.0 *Astro (Alpha)*

Sebenarnya sebelum mereka memberikan nama-nama kudapan sebagai nama untuk versi *OS* nya, Android sempat memiliki 2 versi awal dengan nama Android *Alpha* dan *Beta*. Nama untuk versi pertama ini sendiri sebenarnya adalah Android *Astro*, namun karena alasan hak cipta (*trademark*), nama ini tidak jadi digunakan. Di versi awal ini belum ada perangkat dengan sistem operasi Android yang dijual secara komersil.

2. Android v1.1 *Bender (Beta)*

Versi ini dirilis pada tanggal 5 November 2007 yang merupakan versi lanjutan dari Android *Astro (Alpha)*. Sama seperti versi awalnya, nama *Bender* juga juga tak jadi digunakan karena alasan hak cipta (*trademark*). Kemudian lahirlah telepon seluler pertama dengan sistem operasi Android yang dijual secara komersil yakni *HTC Dream*.

3. Android v1.5 *Cupcake*

Ini merupakan versi pertama yang menggunakan nama makanan manis sebagai kode nama untuk tiap versi Android yang kemudian

tradisi untuk menamai versi Android dengan nama makanan manis masih diteruskan hingga saat ini. Android *Cupcake* dirilis pada tanggal 30 April 2009.

4. Android v1.6 *Donut*

Dirilis tidak sampai setahun setelah perilisan Android *Cupcake*, yakni pada tanggal 15 September 2009. Versi ini dihadirkan untuk menutupi *bug* pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

5. Android v2.0-v2.1 *Éclair*

Sistem operasi ini juga dirilis tidak sampai setahun setelah perilisan dua versi sebelumnya yakni pada tanggal 26 Oktober 2009. Mereka masih berfokus untuk menutupi *bug* yang ada dan juga menambahkan beberapa fitur seperti *Bluetooth*, *flash* pada kamera, fitur digital *zoom* pada kamera, *multi-touch*, *live wallpaper*, dan lainnya. Hadirnya perangkat seri *Nexus* dari *Google* yang pertama kali muncul yakni *HTC Nexus One* juga menggunakan versi OS Android *Éclair*.

6. Android v2.2 *Frozen Yoghurt (Froyo)*

Dirilis pada tanggal 20 Mei 2010. Perangkat dengan OS Android semakin banyak dan kehadirannya mulai dilirik oleh pasar meski masih jauh dibawah kepopuleran OS lain seperti *Symbian* dan *Windows Mobile*.

7. Android v2.3 *Gingerbread*

Dirilis pada tanggal 6 Desember 2010 bersamaan dengan dihidirkannya *Nexus S* yang merupakan perangkat *smartphone* seri *Nexus* yang diproduksi oleh *Samsung*. Versi OS ini juga mengawali kesuksesan Android di jagad *smartphone* meski masih kalah populer dengan *BlackBerry OS*. Beberapa vendor mulai serius untuk menggarap perangkat dengan OS Android.

8. Android v3.0-3.2 *Honeycomb*

Versi ini dirilis pada tanggal 10 Mei 2011 dan dirancang khusus untuk perangkat *tablet*, yang kala itu mulai populer di pasaran salah satunya berkat promosi *Samsung* dan juga kepopuleran *Apple iPad*.

9. Android v4.0 *Ice Cream Sandwich*

Dirilis pada 16 Desember 2011. Bisa dibilang merupakan Android *Honeycomb* yang disempurnakan, dan dioptimalkan untuk penggunaan baik *smartphone* maupun *tablet*. Perubahan yang paling terlihat dari versi ini dibanding dengan versi sebelumnya adalah dari segi *User interface* yang nampak lebih bersih dan elegan. Versi ini juga lebih dioptimalkan untuk urusan *multitasking*.

10. Android v4.1-v.3 *Jelly Bean*

Dirilis pada 9 Juli 2012. Bersamaan dengan diperkenalkannya versi OS 4.1 pada 27 Juni 2012, *Google* juga memperkenalkan *Nexus 7* yang diproduksi oleh *ASUS*. *Nexus 7* (generasi 1) merupakan seri *Nexus* pertama yang merupakan perangkat *tablet*. *Jelly Bean* mengalami 3x *update* versi yakni 4.1, 4.2 hingga 4.3. Selanjutnya mereka memperkenalkan Android v4.2 bersamaan dengan dihadirkannya *Nexus 4*, *smartphone* yang diproduksi oleh *LG plus Nexus 10*, perangkat *tablet* yang diproduksi oleh *Samsung*.

11. Android v4.4 *Kitkat*

Nama *Kitkat* diambil dari sebuah produk cemilan wafer berlapis coklat yang dimiliki oleh *Nestle*. Sebelumnya Android versi “K” ini disebut-sebut sebagai *Key Lime Pie*, namun atas beberapa pertimbangan akhirnya *Google* lebih memilih untuk memberi nama *Kitkat*.

12. Android v5.0-5.1 *Lollipop*

Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih *flat* dengan konsep *material design*. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk penggunaan RAM diatas 3 GB pada *hardware* perangkat. Penggunaan prosesor 64-bit pun makin

banyak diadopsi oleh para *vendor*, mulai dari penerapan pada perangkat *flagship* hingga perangkat kelas menengah kebawah.

13. Android v6.0 *Marshmallow*

Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya *Google* juga memperkenalkan 2 perangkat *smartphone Nexus* sekaligus yang diproduksi oleh 2 *vendor* yang berbeda.

14. Android v7.0 *Nougat*

Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi “N” ini adalah *Nutella*. Namun *Google* menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung *icon* Android yang berdiri diatas potongan *Nougat*.

D. *Volley library*

(Developers) *Volley library* adalah perpustakaan *HTTP* yang membuat jaringan untuk aplikasi android lebih mudah dan cepat. *Volley* tersedia di *GitHub*.

Volley menawarkan keuntungan sebagai berikut:

1. Penjadwalan otomatis dalam permintaan jaringan.
2. *Multiple concurrent* dalam jaringan.
3. Mendukung prioritas *request* dalam jaringan.
4. Mendukung *Cancellation request API*.
5. Mudah dalam penggunaan. Contoh pada *retrofit* dan *backoff*.
6. Alat *debuging* dan *tracing*.

Volley unggul pada operasi tipe *RPC* yang digunakan untuk mengisi *UI*, seperti mengambil halaman hasil pencarian sebagai data terstruktur. Ini terintegrasi dengan mudah dengan protokol apapun dan keluar dari kotak dengan dukungan untuk data mentah (*raw strings*), gambar, dan *JSON*. Dengan menyediakan dukungan *built-in* untuk fitur yang dibutuhkan, *Volley* membebaskan penulisan

kode *boilerplate* dan memungkinkan berkonsentrasi pada logika yang spesifik untuk aplikasi.

Volley tidak cocok untuk operasi *download* atau *streaming* besar, karena *Volley* memegang semua tanggapan dalam memori selama penguraian. Untuk operasi unduhan yang besar, pertimbangkan untuk menggunakan alternatif seperti *DownloadManager*.

Perpustakaan *Volley* ini dikembangkan di *GitHub* dan berisi pipa pengiriman permintaan utama serta seperangkat utilitas yang umum berlaku, tersedia di kotak peralatan *Volley*.

E. Java

(Wikipedia) Pemrograman *Java* adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer atau *smartphone*. Bahasa pemrograman ini banyak mengadopsi sintaks yang terdapat pada bahasa *C* dan *C++* namun dengan sintaks model yang lebih. Aplikasi berbasis *Java* pada umumnya dikompil ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Mesin *Virtual Java* (*JVM*).

Program *Java* merupakan bahasa yang bersifat umum (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda, *Java* juga dikenal dengan slogannya “Tulis sekali, jalankan dimanapun”. Bidang ini merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam berbagai jenis perangkat lunak aplikasi *mobile* ataupun aplikasi berbasis *web*.

Berikut adalah beberapa penjelasan tentang Kelebihan yang dimiliki oleh *Java* yaitu:

a. *Multiplatform*

Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda.

b. *OOP* atau *Object Oriented Programming*:

Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam *Java*. Hal ini akan memudahkan pemrograman menjadi lebih mudah karena kelengkapan *library* semakin beragam jika ditambah dengan komunitas *Java*.

c. *Multithread*

Kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. *Thread* dalam *Java* memiliki kemampuan untuk memanfaatkan kelebihan multi prosessor apabila sistem operasi yang digunakan mendukung multi prosessor.

d. Dapat didistribusi dengan mudah

Java memiliki *library* rutin yang lengkap untuk dirangkai pada *protocol TCP/IP* (seperti *HTTP* dan *FTP*) dengan mudah. Kemampuan *networking Java* lebih kuat dan lebih mudah digunakan. *Java* memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah socket koneksi. *Java* juga memudahkan pembuatan *CGI (Common Gateway Interface)*.

e. Bersifat dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang, *Java* bersifat dinamis dalam tahap *linking*. *Class* yang ada dapat di *link* sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di *link* dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

Berikut adalah beberapa penjelasan tentang kekurangan yang dimiliki oleh *java* yaitu:

a. Pada satu slogannya.

“Tulis sekali dan jalankan dimana saja” ternyata tidak sepenuhnya benar. Beberapa hal harus disesuaikan jika dijalankan pada *platform* yang berbeda. Misalnya *J2SE* dengan *platform SWT-AWT bridge* tidak dapat berfungsi di *Mac OS X*.

b. Kemudahan aplikasi *java* tidak didekompilasi

Dekompilasi adalah suatu proses membalikkan sebuah aplikasi menjadi kode sumbernya. Hal ini memungkinkan terjadi pada *Java* karena berupa *bytecode* yang menyimpan bahasa tingkat tinggi. Hal ini terjadi pula pada *platform .NET* dari *Microsoft* sehingga program yang dihasilkan mudah dibajak kodenya karena sulit untuk disembunyikan.

c. Penggunaan *memory* yang banyak

Penggunaan memori untuk program berbasis *Java* jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti *C/C++* dan *Pascal*. Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berlutut dengan mesin komputer berumur lebih dari 4 tahun.

F. Web service

Menurut (W3C, 2004) *Web service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* di deskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya *WDSL*). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan *SOAP*. Yang biasanya disampaikan dengan *HTTP* dan *XML* sehingga mempunyai korelasi dengan standar *web*.

Pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara *client* dan *server* dijembatani oleh *file web service* dalam format tertentu. Sehingga akses terhadap *database* akan ditangani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service*. Peran dari *web service* akan mempermudah distribusi sekaligus integrasi *database* yang tersebar di beberapa *server* sekaligus.

G. JAX-RS

Menurut (Wikipedia, 2017) *JAX-RS: Java API for RESTful Web Services* (*JAX-RS*) adalah *API* bahasa pemrograman *Java* yang memberikan dukungan dalam membuat layanan web sesuai dengan pola arsitektur *Representational State*

Transfer (REST). *JAX-RS* menggunakan anotasi, diperkenalkan di *Java SE 5*, untuk mempermudah pengembangan dan penyebaran klien layanan *web* dan titik akhir.

Dari versi 1.1, *JAX-RS* adalah bagian resmi *Java EE 6*. Fitur penting untuk menjadi bagian resmi *Java EE* adalah bahwa tidak diperlukan konfigurasi untuk mulai menggunakan *JAX-RS*. Untuk lingkungan *non-Java EE 6*, sebuah entri kecil dalam deskriptor penyebaran *web.xml* diperlukan.

JAX-RS menyediakan beberapa anotasi untuk membantu memetakan kelas sumber daya (*POJO*) sebagai sumber *web*. Anotasi meliputi:

1. *@Path* menentukan *path* relatif untuk sumber kelas atau metode.
2. *@GET*, *@PUT*, *@POST*, *@DELETE* dan *@HEAD* untuk menentukan jenis permintaan *HTTP*.
3. *@Produces* menentukan *response* jenis media Internet (digunakan untuk negosiasi konten).
4. *@Consumes* menentukan jenis media permintaan internet yang diterima.

Selain itu, ia menyediakan anotasi lebih lanjut untuk parameter metode untuk menarik informasi dari permintaan. Semua anotasi *@ * Param* mengambil kunci dari beberapa bentuk yang digunakan untuk mencari nilai yang dibutuhkan.

1. *@PathParam* mengikat *parameter* metode ke segmen jalan.
2. *@QueryParam* mengikat *parameter* metode ke nilai parameter kueri *HTTP*.
3. *@MatrixParam* mengikat *parameter* metode ke nilai parameter matriks *HTTP*.
4. *@HeaderParam* mengikat *parameter* metode ke nilai *header HTTP*.
5. *@CookieParam* mengikat *parameter* metode ke nilai *cookie*.
6. *@FormParam* mengikat *parameter* metode ke nilai bentuk.
7. *@DefaultValue* menentukan nilai default untuk *bindings* di atas saat tombol tidak ditemukan.
8. *@Context* mengembalikan keseluruhan konteks objek (misalnya *@Context HttpServletRequest request*).

Pada bulan Januari 2011 *JCP* membentuk kelompok ahli *JSR 339* untuk mengerjakan *JAX-RS 2.0*. Sasaran utama adalah *API* klien yang umum dan dukungan untuk *Hypermedia* mengikuti prinsip *HATEOAS REST*. Pada bulan Mei 2013, mencapai tahap Final Release.

H. *MySql*

Menurut (Wikipedia, 2017) *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang multi alur, multi pengguna, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di bawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan *GPL*.

Tidak sama dengan proyek-proyek seperti *Apache*, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia *MySQL AB*, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL AB* adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas.

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka.

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi *GPL* sehingga dapat digunakan secara gratis.

3. *Multi-User*.

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *'Performance tuning'*.

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

5. Ragam tipe data.

MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. Perintah dan Fungsi.

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

7. Keamanan.

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *User* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan.

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

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

10. Lokalisasi.

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka.

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

12. Struktur tabel.

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.

I. Metode pengujian sistem

Pengujian sistem menyajikan anomali yang menarik bagi rekayasa perangkat lunak pada proses perangkat lunak, perancang berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

Menurut (Pressman, 2002) *Black Box Testing* atau Pengujian Kotak Hitam, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black Box Testing* memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program. *Black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program.