BAB 2
TINJAUAN PUSTAKA

2.1 Profil Bumi Perkemahan Rancaupas

Wana Wisata Ranca Upas terletak di kawasan wisata Ciwidey. Secara administratif, Wana Wisata Ranca Upas terletak di Kabupaten Bandung. Lokasi spesifik dari wana wisata tersebut yaitu di jalan Ciwidey-Patengan KM.11, Patengan, rancabali, Bandung, Jawa Barat dengan kode pos 40973. Secara geografis, Wana Wisata Ranca Upas terletak pada koordinat 100°-110°26° BT dan 1°-1°26° LS. Wana Wisata tersebut terletak pada ketinggian 1700mdpl dengan konfigurasi lapangan datar sampai bergelombang dan memiliki suhu rata-rata 17°C - 20°C.

Wana Wisata Ranca Upas merupakan dataran tinggi yang memiliki luas wilayah 72,78 Ha. Luas tersebut merupakan bagian dari luas total keseluruhan di Wana Wisata Ranca Upas. Luas pengelolaan kawasan wisata yang efektif 20 Ha. Luas efektif pengelolaan tersebut akan ditambahkan sesua dengan rencana Tahunan (RT) dan Rencana Jangka Panjang (RJP). Luas kawasan non-efektif dari Wana Wisata tersebut masih tertutupi oleh vegetasi rapat yang mengelilingi obyek utama.

Wana Wisata Ranca Upas adalah tempat wisata yang berlokasi di Ciwidey, Bandung Selatan. Daya tarik utama Ranca Upas adalah lahan untuk berkemah, penangkaran rusa dan kolam renang air hangat (Kampung Cai). Daya tarik tambahan pada kawasan Ranca Upas terdiri dari penangkaran rusa dengan aktivitas memberi makan rusa dan dapat dijadikan sebagai lokasi untuk *pre-wedding.*

Alasan kawasan tersebut diberi nama Ranca Upas diambil dari bahasa sunda, yaitu "*Ranca*" yang berarti "Rawa" karena di kawasan tersebut terdapat banyak rawa dan *"Upas"* yaitu sebutan petugas hutan yang berasal dari Belanda. Petugas tersebut disebut Upas oleh masyarakat sekitar. Cerita lain mempercayai kisah yang berbeda dari nama kawasan tersebut. Ranca Upas diambil dari kata bahasa Sunda yaitu "*Ranca*" yang berarti "Rawa" karena di kawasan tersebut terdapat banyak rawa dan *"Upas"* yaitu berarti nama seorang petugas perhutani melegenda di kawasan Gunung Patuha.. Diceritakan bahwa Upas merupakan seorang petugas berkebangsaan Belanda yang berbadan kekar dengan tinggi 198 cm. Dia meninggal

saat melaksanakan tugas lapangan untuk menjelajahi rawa di kawasan yang sekarang menjadi Ranca Upas. Sampai saat ini mayatnya tidak pernah ditemukan. Masyarakat setempat percaya bahwa arwah Upas masih ada di alam dunia ini. Pada tahun 1960 – 1980, masih banyak masyarakat setempat yang bertemu dengan arwah Upas saat mencari kayu bakar di hutan. Konon, dikatakan bahwa Upas selalu memakai topi laken (koboi) dengan diameter yang lebar, lengkap dengan seragam Perhutani yang pada masa itu berwarna coklat dengan kepala menunduk ditutupi topi lebarnya. Pada saat dibangun sebuah penangkaran rusa di Ranca Upas oleh pihak Perhutani, hanya seorang juru kunci yang dapat melakukan pemanggilan terhadap rusa-rusa tersebut dan memberikan makanan secara langsung. Dia adalah Abah Taji. Namun setiap orang dapat berinteraksi dan menikmati keindahan dari makhluk yang bertanduk indah ini.

2.1.1 Logo Bumi Perkemahan Rancaupas

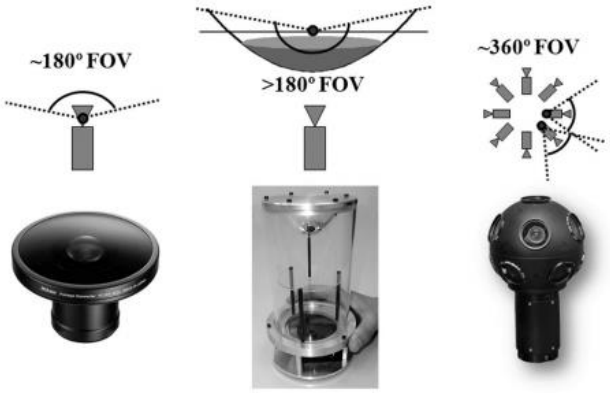
Gambar 2. 1 Logo Bumi Perkemahan Rancaupas

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang akan dibangun. Landasan teori yang digunakan dalam penyusunan Aplikasi Pintar Bumi Perkemahan Rancaupas pada *Smartphone* Android meliputi profil Bumi Perkemahan Rancaupas , Pengertian Aplikasi Mobile, Android, Gyroscopye, Pengertian JAVA, Web Service, JSON, UML, MySQL, PHP, API Accu Weather.

2.2.1 Omnidirectional Camera

Omnidirectional Camera atau sering kita kenal sebagai kamera 360 derajat. Kamera ini memiliki kelebihan dengan sudut pandang 360 derajat sehingga memungkinkan dalan satu kali potret semua sudut dapat direkam. Kamera 360 adalah kamera yang mencakup bidang horizontal dan menyerupai seluruh sudut bidang bola [3].



Gambar 1 (a) Dioptric camera ; (b) catadioptric camera; (c) omnidirectional Camera *[3]*

Gambar diatas merupakan proses perkembangan sehingga terciptanya kamera 360 dari mula kamera yang hanya 180 derajat dan bisa lebih dari 180 derajat sehingga terciptalah kamera 360 derajat.

2.2.2 Aplikasi Mobile

Aplikasi mobile adalah sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti telepon genggam (hanphone), PDA, atau smartphone. Aplikasi mobile dapat mengakses dan menggunakan sebuah aplikasi web secara nirkabel dengan menggunakan perangkat mobile, dimana data yang diperoleh hanyalah berupa teks sehingga tidak perlu membutuhkan bandwidth yang terlalu besar [4]. Aplikasi Mobile juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah fungsionalitas dari perangkat mobile itu sendiri. Untuk mendapatkan mobile application yang diinginkan, user dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. Google Play dan iTunes merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan. (Mobile Marketing Association, 2015) [5].

2.2.3 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis linux yang mencangkup sistem operasi, middleware dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone.* Kemudian untuk mengembangkan Android , dibentuklah *Open Handset Alliance,* konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasukl Google, HTC, Intel, Motorola, Qualcom, T-Mobile, dan Nvidia [6].

Penggunaan android hingga saat ini makin bertambah, ponsel produk terbaru hampir seluruhnya menggunakan sistem operasi android. Dalam pembuatan aplikasi android sendiri pada dasarnya menggunakan Bahasa pemrograman JAVA namun seiring perkembangan muncul Bahasa pemrograman baru yang di naungi oleh Google yaitu Kotline. Pada *smartphone* android sendiri sudah dibekali berbagai macam sensor untuk membantu dalam pengembangan pembuatan aplikasi pintar. Salahsatu sensor yang digunakan dalam pembangunan

aplikasi pintar Bumi Perkemahan Rancaupas adalah sensor Gyroscope yang berfungsi memeberikan data kemiringan pada ponsel Android.

2.2.3.1 Versi-Versi Android

Sejak pertama kali dikembangkan pada tahun 2007, Android telah banyak mengalami perkembangan yang begitu pesat, ditandai dengan banyaknya system operasi Android yang ada hingga saat ini. Tabel 2.1 menunjukan beberapa versi yang ada beserta tanggal rilisnya.

Tabel 2. 1Versi-Versi Android *[7]*

VERSI ANDROID	API Level	TANGGAL RILIS	CODENAME
1.0	1	23 September 2008	
1.1	2	9 Februari 2009	
1.5	3	30 April 2009	Cupcake
1.6	4	15 September 2009	Donut
2.0/2.2.1/2.1	5/6/7	26 Oktober 2009	Éclair
2.2-2.2.3	8	20 Mei 2010	Froyo
2.3-2.3.2/2.3.3-2.3.7	9/10	6 Desember 2010	Gingerbread
3.0/3.1/3.2-3.2.6	11/12/13	22 Februari 2011	Honeycomb
4.0-4.0.2/4.0.3-4.0.4	14/15	19 Oktober 2011	Ice Cream Sandwich
4.1-4.1.2/4.2-4.2.2/4.3-4.3.1	16/17/18	27 Juni 2012	Jelly Bean
4.4-4.4.4/4.4w	19/20	31 Oktober 2013	KitKat
5.0-5.0.2/5.1-5.1.1	21/22	12 November 2014	Lollipop
6.0-6.0.1	23	5 Oktober 2015	Marshmallow

2.2.3.2 Fitur-Fitur Android

Sejak pertama kali muncul android sudah dilengkapi dengan beberapa fitur utama yang canggih, dengan seiring perkembangan zaman fitur-fitur android pun semakin canggih lagi. Berikut adalah beberapa fitur android yang dapat dimanfaatkan oleh pengembang untuk aplikasi yang akan dibangun :

a) Media penyimpanan *(storage)* – untuk media penyimpanan internal, Android menggunakan SQLite. Selain itu, Android juga mendukung fasilitas penyimpanan data di *external storage*, seperti SD *card.*

b) Koneksi jaringan *(connectivity)* – Android mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, dan WiMAX.
c) *Messaging* – Android mendukung SMS dan MMS.
d) *Web browser* – *Web browser* bawaan Android dibangun berbasis pada *open source* Blink (sebelumnya Webkit) dan Chrome's V8 JavaScript *engine.*
e) Dukungan media *(media support)* – Android mendukung berangam jenis media seperti H.263, H.264, MPEG-4, AMR, AAC, HE-AAC, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF dan BMP.
f) Dukungan perangkat keras *(hardware support)* – Android mendukung beragam perangkat keras yang dapat digunakan oleh pngembang aplikasi, seperti *gyroscope, accelerometer, barometers, magnetometers,* GPS dll.
g) *Multitouch* – mendukung *multitouch screens.*
h) *Multitasking* – mendukung *multitasking applications.*
i) *Tethering* – mendukung *sharing* koneksi internet sebagai *wired/wireless hotspot.*
j) Dukungan Flash – sejak android 2.3 mendukung Flash 10.1.
k) Fitur-fitur berbasis suara *(voice-based feature)* – Android menyediakan beragam aplikasi berbasis suara, seperti google search, Google's Knowledge Graph, dan lain sebagainya.
l) Tangkapan layar *(screen capture)* – sejak Android 4.0, pengguna Android dapat mengambil tangkapan layar dengan menekan tombol **Power + Volume Down** pada perangkatnya.
m) Aksesbibilitas *(accessibility)* – Android menyediakan *built-in text-to-speech* dari TalkBack dan berbagai fasilitas lain untuk memeberikan kemudahan bagi semua pengguna android, terutama dari kalangan difabel.
n) *Handset layout* – aplikasi Android dapat berjalan pada beragam ukuran layar dan dapat dikoneksikan ke layar eksternal melalui HDMI atau Miracast *(Wirelessly).*

Itulah fitur-fitur utama yang biasa ada di dalam perangkat Android yang dapat kita gunakan untuk pembangunan aplikasi Android [7].

2.2.3.3 Android Market

Pada Agustus 2008, Google mengumumkan Android Market yang pada waktu itu dapat diakses melalui tautan http://android-market.co/ , sebagai suatu took aplikasi *online* yang khusus diperuntukan bagi perngkat Android [7]. Untuk sekarang android market telah berubah menjadi satu kesatuan dari beberapa market yang ada yaitu Google Play Store.

2.2.3.4 The Dalvik Virtual Machine (DVM)

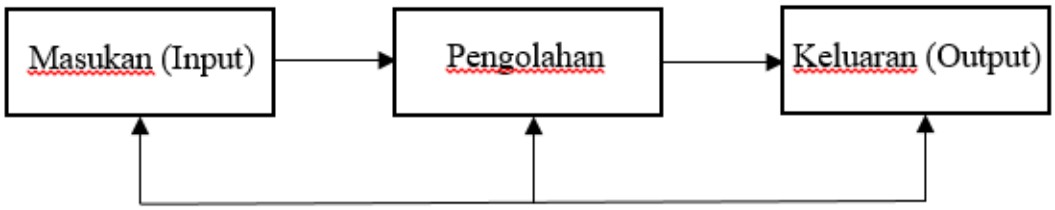
Dalvik Virtual Machine (DVM) adalah "register bases" , DVM didesain dan ditulis oleh Dan Bornsten dan beberapa *enginers* Google lainnya. Dalvik *Virtual Machine* menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading,* dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C/ C + sama halnya pada OS Linux kebanyakan. Meskipun dalam kenyataan kita harus banyak memeahami Arsitektur dan proses sistem dari kernel Linux yang digunakan dalam Android Tersebut [6].

Dalvik Virtual Machine mengeksekusi *executable file,* sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan di kompilasi menggunakan *tools* yang disediakan dalam SDK Android [6].

2.2.4 Sistem Informasi

2.2.4.1 Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. *Mudrick dan Ross (1993)* mendifinisikan system sebagai perangkat elemen yang digabungkan satu dengan yang lainnya untuk suatu tujuan bersama. Sementara, definisi sistem dalam kamus *Webster's Unbriged* adalah elemen-elemen yang saling berhubungan dan membentuk satu kesatuan atau organisasi [8].



Gambar 2. 2 Model Sistem

Gambar diatas menunjukan bahwa sistem atau pendekatan sisitem minimal mempunyai empat komponen yakni masukan, pengolahan, keluaran dan balikan atau kontrol [8].

2.2.4.2 Informasi

2.2.4.3 Sistem Informasi

Untuk memahami sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting membentuk sistem informasi. Data merupakan nilai, keadaaan,, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang (Davis, 1995). Mc Leod (1995) mengatakan bahwa informasi adalah data yang telah diproses, atau data yang memiliki arti [8]. Maka sistem informasi dapat diartikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya *(Kertahadi, 1995).*

Gambar 2. 3 Konsep Sistem Informasi

2.2.5 Webserver

2.2.5.1 MySQL

MySQL adalah sebuah sistem manajemen database relasi (*relational database manajement system*) yang bersifat terbuka (*open source*). Sejarah MySQL yang merupakan hasil buah pikiran dari Michael "Monty" Widenius, David Axmark, dan Allan Larson dimulai tahun 1995. Mereka bertiga kemudian mendirikan perusahaan bernama *MySQL AB* di Swedia. Tujuan awal ditulisnya program MySQL adalah mengembangkan aplikasi web yang akan digunakan oleh salahsatu klien MySQL AB. MySQL memiliki kinerja, kecepatan proses, dan ketangguhan yang tidak kalah disbanding database-database besar lainnya yang komersil seperti ORACLE, Sysbase, Unify, dan sebagainya. Ber-semangatkan *Open Source*, MySQL berkembang menjadi salah satu sistem manajemen database relasi terkemuka yang diakui secara internasional [9]. Beberapa penghargaanpun diperoleh MySQL sebagai salah satu bukti pengharggan atas prestasinya, seperti :

- *Linux Magazine's Editors Choice Award* pada tahun 2000 dan 2001.
- *Linux New Media Award 2001* sebagai *The Best Comercial Software.*
- *Linux Jurnal Readers' Choice Award* pada tahun 1998, 1999, 2000 dan 2001.
- *Linux Journal's Editors' Choice Award* pada tahun 2002.

2.2.5.2 PHP (Personal Home Page)

PHP adalah salahsatu Bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan diparsing di dalam web server oleh *interpreter* PHP dan diterjemahkan kedalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program PHP dilakukan di lingkungan *web server*, PHP dikatakan sebagai Bahasa sisi server (*server-side*) [10].

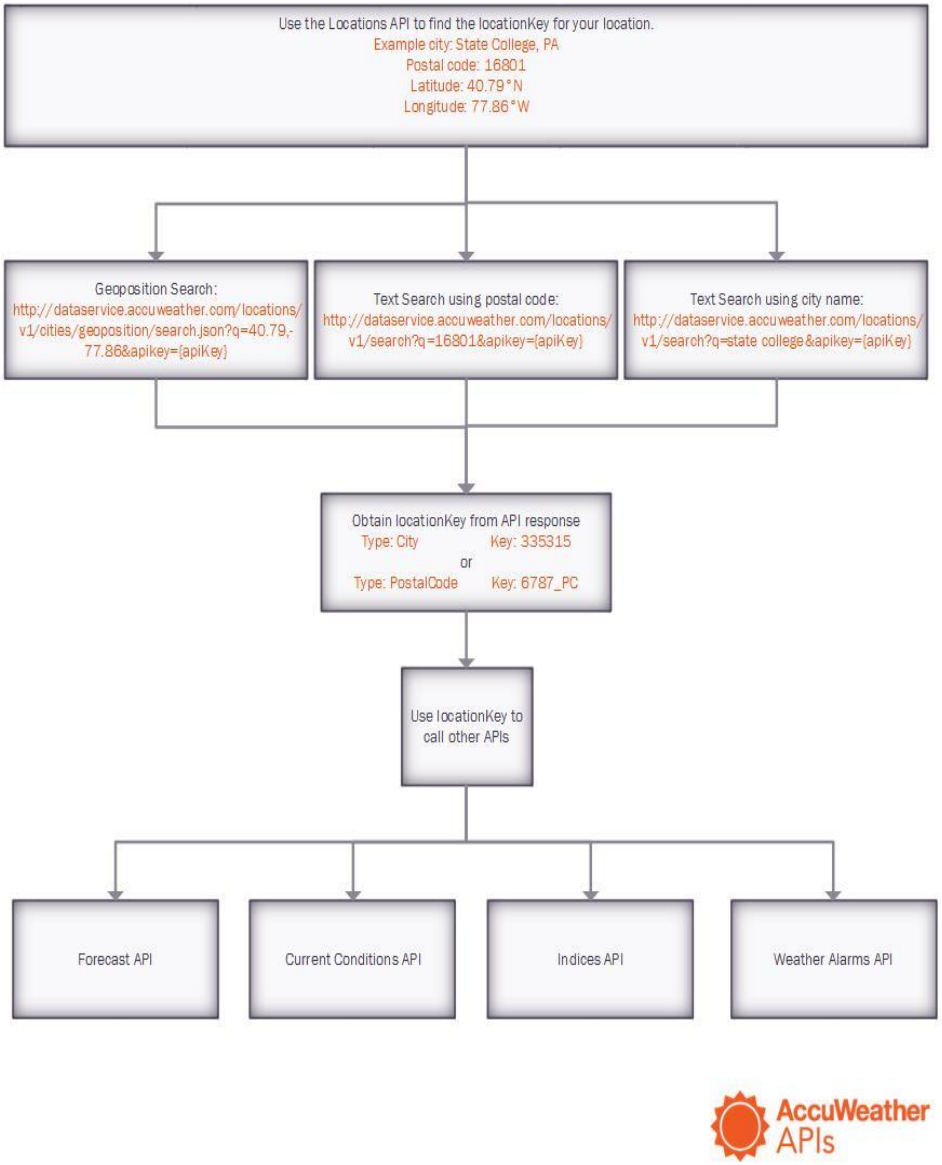
2.2.6 API Accu Wheater

API Accu Weather adalah API yang memberikan data cuaca sesuai lokasi yang kita inginkan. Data yang diberikan oleh Accu Weather Adalah sebagai berikut :

Tabel 1 Tabel data yang disediakan oleh API Accu Weather *[11]***.**

Element	Imperial	metric
Cloud Ceiling	######0	######0
Hourly Preception	##0.00	###0
Preasure	##0.00	#000.0
Temperature	##0	##0.0
Visibility	#0.0##	#0.0
Wind Gust	##0.0	##0.0
Wind Speed	##0.0	##0.0

Dari data diatas kita dapat memanfaatkannya sebagai prediksi cuaca yang akan datang untuk membangun suatu aplikasi. Untuk lebih lengkapnya kita bisa lihat flow Diagram Accu Weather.



Gambar 2 Flow Diagram Accu Weather [12]

2.2.7 SDLC (*Software Development Life Cycle*)

SDLC atau *Software Development Life Cycle* adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) [13].

Tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

- Inisiasi

Tahap ini biasanya ditandai dengan pembuatan proposal pembuatan proyek perangkat lunak.

- Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

- Perencanaan (*planning)*

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber dara (*resource*) yang dibutuhkan untuk memperoleh solusi.

- Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user.* Membuat dokumen fungsional

- Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

- Pengembangan (*development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan; membuat basisdata dan mempersiapkan prosedure pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program; peninjauan pengujian.

- Integrasi dan pengujian

Mendemontrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional.

- Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan teredintifikasi dari fase integrasi dan pengujian.

- Operasi dan pemeliharaan (*operation and maintenance*)

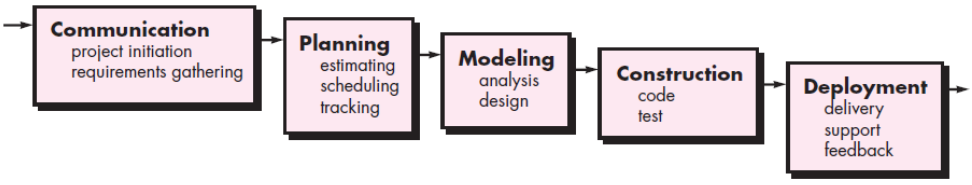
Mendeskripsikan pekerjaan untuk mengoprasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada pemrosesan peninjauan.

- Disposisi (*dispotition*)

Mendeskripsikan aktivitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktivitas *user.*

Ada beberapa model SDLC yang dapat digunakan semuanya memeliki kekurangan dan kelebihan pada setiap modelnya. Dalam penelitian ini kita akan hanya akan membahas satu metode yaitu *waterfall.*

2.2.7.1 Model *Waterfall*



Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linear (*sequential linear*) atau alur hidup klasik (*clasic life cycle*). Model airterjun menyediakan pendekatan alur hidup perangkat lunak secara sekuaensial atau terurut dimulai dari analisis, desain pengod

1) *Communication*

Tahap *communication* merupakan tahapan untuk mengadakan pengumpulan data dengan pertemuan bersama konsumen, maupun pengumpulan data-data tambahan, dan tahapan analisis terhadap keperluan software

2) *Plaining*

Tahapan *plaining* adalah lanjutan dari proses *communication*. Tahapan ini nantinya menghasilkan dokumen yang berikatan dengan keinginan pengguna atau bisa disebut sebagai *user requirement* dalam pembuatan perangkat lunak.

3) *Modeling*

Dalam proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus kepada perancangan struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) *procedural*. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4) *Construction*

Tahapan *Construction* merupakan proses dalam pembuatan kode. *Coding* atau perkodean adalah penerjemah desain di dalam bahasa yang mampu dikenali oleh *computer*. Programmer akan menerjemahkan transaksi yang diinginkan oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu perangkat lunak, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan terhadap sistem tersebut untuk kemudian nantinya bisa diperbaiki.

5) *Deployment*

Tahapan *deployment* bisa disebut sebagai hasil akhir dari pembuatan software atau sistem. Setelah melakukan analisis, desain, serta pengkodean, maka sistem yang telah dibuat akan digunakan oleh pengguna. Kemudian perangkat lunak yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Dari kenyataan yang terjadi sangat jarang model airterjun dapat dilakukan sesuai dengan alurnya karena sebab berikut :

- Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan
- Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan.
- Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir pengembangan.

Model air terjun sangat cocok digunakan kebutuhan pelanggan sudah sangat dipahami dan kemingkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari model air terjun adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan disetiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan (tidak ada tumpang tindih pelaksanaan tahap) [13].

2.2.8 Metode SAW

Menurut Fishburn (1967) dalam **Invalid source specified.** menjelaskan bahwa “Metode SAW juga sering dikenal istilah metode penjumlahan terbobot karena dalam perhitungannya memperhitungkan bobot pada kriteria yang digunakan. Konsep dasar metode ini adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut”.

Metode SAW membutuhkan proses normalisasi matrik keputusan ( ) kedalam skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating dan bobot tiap atribut **Invalid source specified.**.

Langkah penyelesaian metode SAW menurut **Invalid source specified.** adalah sebagai berikut:

1. Menentukan alternatif yaitu $A_{i.}$
2. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu $C_{j.}$
3. Menentukan bobot preferensi atau tingkat kepentingan (W) setiap kriteria.

   $W = [W1,W_2,W_3,\ldots..,W_j]$……..(2.1)

4. Membuat tabel ranking kecocokan dari setiap alternatif pada setiap kriteria.
5. Membuat matriks keputusan (X) yang dibentuk dari tabel ranting kecocokan dari setiap alternatif pada setiap kriteria. Nilai X setiap alternatif (A_i) pada setiap kriteria (C_j) yang sudah ditentukan, dimana, i=1,2,…m dan j=1,2,…n.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} \\ . & & & . \\ . & & & . \\ x_{i1} & x_{i2} & \dots & x_{ij} \end{bmatrix}$$

6. Melakukan normalisasi matriks keputusan dengan cara menghitung nilai rating kinerja ternormalisasi (r_{ij}) dari alternatif A_i pada kriteria C_j.

$$r_{ij} = \begin{Bmatrix} \dfrac{x_{ij}}{min\ _i x_{ij}} \dots\dots (2.2) \\ \\ \dfrac{x_{ij}}{min\ _i x_{ij}} \dots\dots (2.3) \end{Bmatrix}$$

Keterangan :

a. Kriteria keuntungan dilakukan apabila nilai memeberikan keuntungan bagi pengambil keputusan. Sebaliknya, kriteria biaya dilakukan apabila timbulnya biaaya bagi pengambil keputusan.
b. Apabila berupa kriteria keuntungan maka nilai dibagi dengan nilai dari setiap kolom. Sedangkan untuk kriteria biaya, nilai dari setiap kolom dibagi dengan nilai.

7. Hasil dari nilai rating kinerja ternormalisasi (r_{ij}) membentuk matriks ternormalisasi (R).

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1j} \\ . & & & . \\ . & & & . \\ r_{i1} & r_{i2} & \dots & r_{ij} \end{bmatrix}$$

8. Hasil akhir dari preferensi (V_i) diperoleh dari penjumlahan dan perkalian elemen baris matriks ternormalisasi (R) dengan bobot preferensi (W) yang bersesuaian elemen kolom matriks (W)

$$V_i = \sum_{j=1}^{n} w_j r_{ij} \quad (2.4)$$

Hasil perhitungan nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i merupakan alternatif terbaik. [14]

2.2.9 Pengujian Perangkat Lunak

Sebuah perangkat lunak perlu dijaga kualitasnya bahwa kualitas bergantung kepada kepuasan pelanggan (*customer*) [15].kualitas perangkat lunak perlu dijaga untuk keperluan sebagai berikut :

- Agar dapat "survive" bertahan hidup di dunia bisnis perangkat lunak
- Dapat bersaing edngan perangkat lunak yang lain
- Mengefektifkan biaya agar tidak banyak membuang perangkat lunak karena kegagalan pemasaran atau kegaagalan produksi
- Memepertahankan pelanggan dan meningkatkan keuntungan

Pengujian adalah satu set aktivitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan [15]. Secara umum pola pengujian pada perangkat lunak adalah sebagai berikut :

- Pengujian dimulai dari level komponen sehingga intregrasi antar komponen menjadi sebuah sistem
- Teknik pengujian berbeda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada pengujian pada bagian mana yang dibutuhkan
- Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak
- Pengujian dan penirkutuan (*debugging*) merupakan aktivitas yang berbeda, tapi penirkutuan (*debugging*) harus diakomodasi pada berbagai strategi pengujian.

Pengujian peranhkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas sering dikaitkan dengan verifikasi dan validasi (V&V) [15]. Dapat juga dikatakan sebagai berikut

- **Verifikasi :** “Apakah Produk dibangun dengan benar ?” (lebih kearah apakah proses pengembangan produk sudah benar dan telah berhasil mengimplementasikan fungsi yang benar
- **Validasi :** “apakah sudah membangun produk yang benar ? “ (lebih ke arah hasil produk apakah sudah sesuai dengan yang diinginkan