

BAB II LANDASAN TEORI

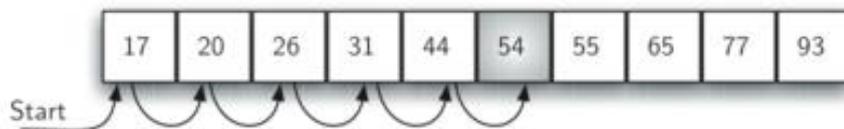
2.1. Rancang Bangun

Perancangan adalah proses yang mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaan.

Bangun sistem adalah kegiatan menciptakan sistem baru mau pun pengganti atau memperbaiki sistem telah ada secara keseluruhan (Roger S.Pressman, 2002). Rancang bangun berarti mengatur segala sesuatu (sebelum bertindak, mengerjakan, atau melakukan sesuatu), merencanakan (Purwanto, 2008).

2.2. Metode *Sequential Search*

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu persatu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan. Konsepnya yaitu dengan melakukan perbandingan data satu persatu secara berurutan sampai data tersebut ditemukan ataupun tidak ditemukan.



Gambar 2.1 Konsep *Sequential Search*

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen

data harus dilakukan pencarian sebanyak N kali pula. (Sembiring & Pranata, 2013). Berikut Adalah Gambar dari Algoritma Sequential Search

```
i ← 0
Ketemu ← False
Jika (Data [i] = Key) Maka
    Ketemu ← True
Jika Tidak
    i ← i+1
Jika (Ketemu) Maka
    i adalah indeks dari data yang dicari
Jika tidak
    Data tidak ditemukan
```

Gambar 2.2 Algoritma Sequential Search

2.3. Ilmu Gizi

Ilmu gizi merupakan ilmu dan seni tentang pangan dalam hubungan dengan kesehatan. Sejarah keilmuan gizi menunjukkan bahwa penyimpangan dari keadaan gizi normal menyebabkan menurunnya derajat kesehatan dan meningkatkan resiko berbagai penyakit infeksi maupun degeneratif. Berbagai aspek ilmu gizi menyangkut disiplin ilmu kesehatan, kesehatan masyarakat, kedokteran, dietetika, kimia, biokimia, ilmu pangan, teknologi pangan, bioteknologi, gastronomi, kuliner, sampai dengan ilmu sosial yang mempengaruhi pola konsumsi makanan dalam keluarga. (Sandjaja, et al., 2009)

WHO mengartikan ilmu gizi adalah sebagai ilmu yang mempelajari proses-proses yang terjadi pada hidup organisme hidup. Proses tersebut dapat mencakup pengambilan dan pengolahan antara zat padat dan cair yang berasal dari makanan yang diperlukan untuk memelihara kehidupan, pertumbuhan, serta berfungsinya organ-organ tubuh dan menghasilkan energi.

Ilmu gizi merupakan salah satu ilmu terapan yang berkaitan dengan berbagai ilmu dasar misalnya ilmu kimia, biokimia, fisiologi, biologi, pathologi, ilmu pangan dan lain sebagainya. Oleh karena itu, untuk menguasai ilmu gizi maka seseorang harus menguasai ilmu-ilmu yang relevan dengan kebutuhan ilmu gizi.

Ilmu kimia dan Biokimia berkembang melahirkan ilmu gizi. Antoine Lavoiser seorang pakar kimia dari Perancis dijuluki sebagai Bapak ilmu kimia modern berhasil meletakkan dasar ilmu gizi berupa fungsi kimia dan biokimia dalam tubuh manusia, sehingga beliau menyandang predikat sebagai bapak ilmu gizi. Lahirnya ilmu gizi diawali dengan penemuan tentang hal yang berkaitan dengan penggunaan energi makanan, antara lain meliputi proses pernapasan, kalorimetri dan oksidasi. Penelitian tersebut menggunakan hewan percobaan yaitu sejenis kelinci yang biasa digunakan dalam penelitian biologi. Disimpulkan bahwa pernapasan merupakan proses pembakaran yang sama dengan pembakaran yang terjadi di luar tubuh.

2.4. Kamus Gizi

Kamus gizi adalah buku acuan yang memuat kata dan ungkapan yang biasanya disusun menurut abjad berikut keterangan tentang maknanya, dan terjemahannya. Kamus gizi dapat juga digunakan sebagai buku rujukan yang menerangkan makna kata-kata yang berfungsi untuk membantu seseorang mengenal perkataan baru yang berkaitan dengan istilah pada gizi. (Sandjaja, et al., 2009)

2.5. Android

Android merupakan sistem operasi untuk perangkat mobile yang berbasis Linux dan bersifat terbuka atau opensource dengan lisensi GNU yang dimiliki Google (Wahana, 2013). Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*

konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi.

2.6. Tinjauan Perangkat Lunak

2.6.1. Eclipse



Gambar 2.3 Logo Eclipse

2.6.1.1. Definisi Eclipse

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

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

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

2.6.1.2. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta^[1] untuk pengembangannya. Semenjak itu konsorsium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.6.1.3. Arsitektur Eclipse

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

1. *Core platform*
2. OSGi
3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel

untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik^[2]. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* secara gratis seperti *Eclipse downloads by project*.

2.6.1.4. Versi Peluncuran Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama *Simultaneous Release*. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.

2.6.1.5. The Dalvik Virtual Machine (DVM)

Salah satu element kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di *Dalvik Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaan dengan *Java Virtual Machine* (JVM) seperti Java ME (*Java Mobile Edition*), tetapi Android menggunakan *virtual machine* sendiri yang dirancang untuk memastikan beberapa *featur-featur* berjalan lebih efisien pada perangkat *mobile*.

2.6.1.6. Android SDK (Software Development Kit)

Android SDK adalah *tools API* (*Application Programing Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

2.6.1.7. ADT (Android Development Tools)

Android Development Tools adalah *plugin* yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. Berikut Adalah Tabel Versi Android :

Versi	Nama	Rilis	Catatan
1.0	-	23-Sep-08	Android pertama, hanya untuk <i>smartphone</i>
1.1	-	09-Feb-09	
1.5	Cupcake	30-Apr-09	Mulai pakai kode nama
1.6	Donut	15-Sep-09	
2.0-2.1	Eclair	26-Okt-09 (2.0)	
		12-Jan-10 (2.1)	
2.2	Frozen Yogurt	20-Mei-10	
2.3	Gingerbread	06-Des-10	Digunakan <i>smartphone jenis lama</i>
3.0-3.2	Honeycomb	22-Feb-11 (3.0)	Hanya untuk tablet
		10-Mei-11 (3.1)	
		15-Jul-11 (3.2)	
4.0	ICS	19-Okt-11	<i>Smartphone & Tablet</i>
4.1-4.3	Jelly Bean	09-Jul-12 (4.1)	Update fitur-fitur ICS
		13-Nov-12 (4.2)	
		24-Jul-13 (4.3)	
4.4	Kit Kat	Okt-13	
5.0	Lollipop	2014	Peningkatan system keamanan dan tampilan
6.0	Marshmallow	2015	

Tabel 2.1 Tabel Versi Android

2.7. DataBase SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional berbasis ACID-compliant dan memiliki ukuran library yang relatif kecil. Software ini ditulis menggunakan bahasa C.

SQLite merupakan proyek yang bersifat publik domain yang dikerjakan oleh D, Richard Hipp. Tidak seperti paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan.

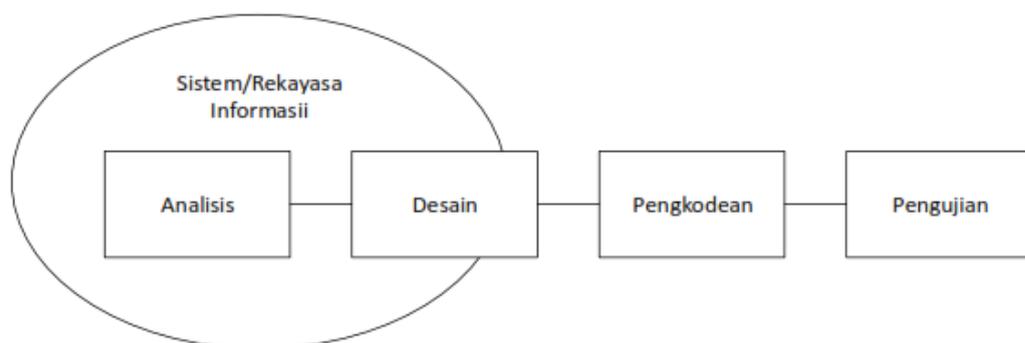
Protokol komunikasi utama yang digunakan adalah melalui pemanggilan API langsung melalui bahasa pemrograman. Mekanisme seperti ini tentu membawa keuntungan karena dapat mereduksi overhead dan latency time; intinya lebih sederhana.

Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan pada sebuah file. Kesederhanaan dari sisi desain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai. Salah satu keunggulan adalah hanya memakan sedikit memori saat dijalankan, hanya sekitar 250Kb sehingga cocok untuk piranti mobile. (Wahana, 2013).

SQLite mendukung tipe data TEXT (yang mirip dengan String di Java), INTEGER (mirip dengan long di Java), serta REAL (mirip dengan double di Java).

2.8. Metode *waterfall*

Metode *waterfall* (sekuensial linier) mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan (Roger S.Pressman, 2002)



Gambar 2.4 Metode Waterfall

Proses pada metode *waterfall* dapat dijelaskan sebagai berikut :

2.8.1. Analisis

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2.8.2. Desain

Desain perangkat lunak adalah proses multi langkah yang *focus* pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean.

2.8.3. Pengkodean

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

2.8.4. Pengujian

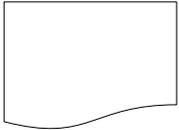
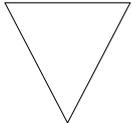
Pengujian fokus pada perangkat lunak dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

2.9. Diagram *Flowchart*

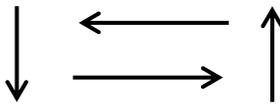
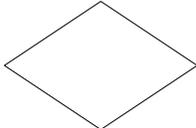
Bagin Alir merupakan alat pengembangan sistem yang akan membantu menggambarkan sistem yang sedang berjalan pada perusahaan. Menurut (Jogianto H.M, 2005) bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Terdapat beberapa jenis bagan alir yang biasa digunakan, yaitu sebagai berikut:

2.9.1. Bagan Alir Dokumen (*document flowchart*)

Bagan alir dokumen (*document flowchart*) atau disebut dengan bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan tembusannya. Adapun simbol bagan alir dokumen dapat dilihat pada tabel 2.2 sebagai berikut.

SIMBOL	KETERANGAN
Dokumen 	Menunjukkan dokumen yang digunakan untuk input dan output baik secara manual maupun komputerisasi
Proses Manual 	Menunjukkan pekerjaan yang dilakukan secara manual.
Proses komputerisasi 	Menunjukkan proses dari operasi program komputer.
Simpanan 	Menunjukkan arsip atau Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
Terminator 	Menyatakan permulaan atau akhir suatu Program

Tabel 2.2 document flowchart

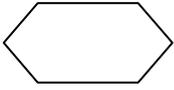
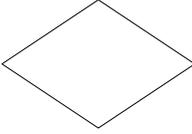
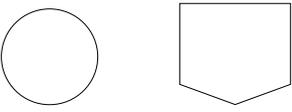
SIMBOL	KETERANGAN
	Menyatakan jalannya arus suatu proses
Decision 	Digunakan untuk suatu penyeleksian kondisi di dalam program.

Tabel 2.2 lanjutan

2.9.2. Bagan Alir Program (*Program Flowchart*)

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terperinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika, sedangkan bagan alir program komputer terperinci digunakan untuk menggambarkan instruksi-instruksi program komputer secara terperinci.

Bagan alir logika program dipersiapkan oleh analis sistem. sedangkan bagan alir program komputer terperinci dipersiapkan oleh pemrogram. Berdasarkan penjelasan diatas penulis dapat menarik kesimpulan bahwa bagan alir (*flowchart*) adalah suatu gambaran umum tentang sistem yang berjalan yang berfungsi sebagai alat bantu komunikasi dan untuk dokumentasi serta menyajikan kegiatan baik secara manual maupun komputerisasi. Adapun symbol bagan alir program dapat dilihat pada tabel 2.3 berikut.

Simbol	Keterangan
Terminator 	Digunakan untuk memberikan awal dan akhir suatu proses.
Proses 	Menunjukkan proses dari operasi program komputer.
<i>Preparation</i> 	Proses inialisasi/pemberian harga awal.
<i>Input/Output Data</i> 	Proses <i>input/output</i> data, parameter, informasi.
Garis alir 	Digunakan untuk menunjukkan arus dari proses.
<i>Decision</i> 	Digunakan untuk suatu penyeleksian kondisi di dalam program.
Proses terdefinisi 	Simbol yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
Penghubung 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda

Tabel 2.3 Program Flowchart

2.10. Data Flow Diagram

(Rosa A.S & Shalahuddin, 2015), Data *Flow* Diagram adalah representasi grafik yang menggambarkan aliran informasi dan tranformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepsentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Notasi-notasi pada DFD dapat dilihat pada tabel 2.4 berikut:

Notasi	Keterangan
<p data-bbox="475 972 584 1010">Proses</p> 	<p data-bbox="751 972 1378 1279">Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harus menjadi fungsi atau prosedur didalam kode program</p>
<p data-bbox="384 1330 679 1368"><i>File</i> atau basis data</p> 	<p data-bbox="751 1330 1378 1749"><i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat mejadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (ERD)</p>

Tabel 2.4 Notasi Data Flow Diagram (DFD)

Notasi	Keterangan
<p data-bbox="518 309 683 342">Entitas luar</p> 	<p data-bbox="821 309 1452 645">Entitas luar (<i>external entity</i>) atau masukan atau keluaran atau orang-orang yang memakai atau berinteraksi dengan perangkat lunak yang dimodelkan atau <i>system</i> lain yang terkait dengan aliran data dari <i>system</i> yang dimodelkan</p>
<p data-bbox="526 674 675 707">Aliran data</p> 	<p data-bbox="837 674 1444 907">Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses kemasukan (<i>input</i>) atau keluaran (<i>ouput</i>)</p>

Tabel 2.4 Lanjutan

2.11. Black-Box Testing (Pengujian Kotak Hitam)

(Rosa A.S & Shalahuddin, 2015), *Black-box testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.

