



BUPATI TANGERANG
PROVINSI BANTEN

PERATURAN BUPATI TANGERANG
NOMOR 56 TAHUN 2018

TENTANG

STANDARISASI PEMBANGUNAN PERANGKAT LUNAK
DAN DOKUMENTASI SISTEM

DENGAN RAHMAT TUHAN YANG MAHA ESA

BUPATI TANGERANG,

Menimbang : bahwa untuk peningkatan pengembangan perangkat lunak, perlu adanya suatu standarisasi yang dapat digunakan sebagai pengukuran kelayakan dari perangkat lunak yang dikembangkan yang bertujuan untuk menjamin perangkat lunak yang dikembangkan telah mengikuti aturan baku yang telah ditetapkan, maka perlu menetapkan Peraturan Bupati tentang Standarisasi Pembangunan Perangkat Lunak Dan Standarisasi Dokumentasi Sistem;

Mengingat : 1. Pasal 18 ayat (6) Undang-Undang Dasar Negara Republik Indonesia Tahun 1945;

2. Undang-Undang Nomor 14 Tahun 1950 tentang Pembentukan Daerah-Daerah Kabupaten Dalam Lingkungan Propinsi Djawa Barat (Berita Negara Tahun 1950) sebagaimana telah diubah dengan Undang-Undang Nomor 4 Tahun 1968 tentang Pembentukan Kabupaten Purwakarta dan Kabupaten Subang dengan mengubah Undang-Undang Nomor 14 Tahun 1950 tentang Pembentukan Daerah-Daerah Kabupaten Dalam Lingkungan Propinsi Djawa Barat (Lembaran Negara Republik Indonesia Tahun 1968 Nomor 31, Tambahan Lembaran Negara Republik Indonesia Nomor 2851);

3. Undang-Undang Nomor 23 Tahun 2000 tentang Pembentukan Propinsi Banten (Lembaran Negara Republik Indonesia Tahun 2000 Nomor 182, Tambahan Lembaran Negara Republik Indonesia Nomor 4010);

4. Undang-Undang Nomor 11 Tahun 2008 tentang Informasi Dan Transaksi Elektronik (Lembaran Negara Republik Indonesia Tahun 2008 Nomor 58, Tambahan Lembaran Negara Republik Indonesia Nomor 4843) sebagaimana telah diubah dengan Undang-Undang Nomor 19 Tahun 2016 tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 tentang Informasi Dan Transaksi Elektronik (Lembaran Negara Republik Indonesia Tahun 2016 Nomor 251, Tambahan Lembaran Negara Republik Indonesia Nomor 5952);

5. Undang-Undang...

5. Undang-Undang Nomor 25 Tahun 2009 tentang Pelayanan Publik (Lembaran Negara Republik Indonesia Tahun 2009 Nomor 112 dan Tambahan Lembaran Negara Republik Indonesia Nomor 5038);
6. Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintahan Daerah (Lembaran Negara Republik Indonesia Tahun 2014 Nomor 244, Tambahan Lembaran Negara Republik Indonesia Nomor 5587) sebagaimana telah diubah beberapa kali terakhir dengan Undang-Undang Nomor 9 Tahun 2015 tentang Perubahan Kedua Atas Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintahan Daerah (Lembaran Negara Republik Indonesia Tahun 2015 Nomor 58, Tambahan Lembaran Negara Republik Indonesia Nomor 5679);
7. Peraturan Presiden Nomor 95 Tahun 2018 tentang Sistem Pemerintahan Berbasis Elektronik;
8. Peraturan Daerah Nomor 1 Tahun 2008 tentang Urusan Pemerintah Kabupaten Tangerang (Lembaran Daerah Kabupaten Tangerang Tahun 2008 Nomor 01, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 0108);
9. Peraturan Daerah Kabupaten Tangerang Nomor 11 Tahun 2016 tentang Pembentukan dan Susunan Perangkat Daerah Kabupaten Tangerang (Lembaran Daerah Kabupaten Tangerang Tahun 2016 Nomor 11, Tambahan Lembaran Daerah Kabupaten Tangerang Nomor 1116);

MEMUTUSKAN:

Menetapkan : PERATURAN BUPATI TENTANG STANDARISASI PEMBANGUNAN PERANGKAT LUNAK DAN DOKUMENTASI SISTEM.

BAB I
KETENTUAN UMUM

Pasal 1

Dalam Peraturan Bupati ini yang dimaksud dengan:

1. Daerah adalah Kabupaten Tangerang.
2. Daerah adalah Kabupaten Tangerang;
3. Pemerintah Daerah adalah Bupati dan Perangkat Daerah sebagai unsur penyelenggara Pemerintahan Daerah Kabupaten Tangerang;
4. Pemerintahan Daerah adalah penyelenggaraan urusan pemerintahan oleh pemerintah daerah dan dewan perwakilan rakyat daerah menurut asas otonomi dan tugas pembantuan dengan prinsip otonomi seluas-luasnya dalam sistem dan prinsip Negara Kesatuan Republik Indonesia sebagaimana dimaksud dalam Undang-Undang Dasar Negara Republik Indonesia Tahun 1945.
5. Bupati adalah Bupati Tangerang.

6. Perangkat...

6. Perangkat Daerah yang selanjutnya disingkat PD adalah unsur pembantu bupati dan Dewan Perwakilan Rakyat Daerah Kabupaten Tangerang dalam penyelenggaraan Urusan Pemerintahan yang menjadi kewenangan Daerah.
7. Perangkat lunak adalah Program komputer, prosedur, dan dokumentasi yang menyertai serta data yang digunakan untuk mengoperasikan sistem komputer.
8. Standarisasi Pengembangan Perangkat Lunak adalah proses penentuan spesifikasi pengembangan perangkat lunak di Lingkungan Kabupaten Tangerang.

BAB II JENIS STANDARISASI PENGEMBANGAN PERANGKAT LUNAK

Pasal 2

Standarisasi akan dibagi menjadi 2 (dua) jenis, yang meliputi:

- a. Standarisasi Teknis Pemrograman Sistem
Dalam standarisasi teknis pemrograman sistem, akan dibahas terkait dengan beberapa hal teknis yang berkenaan dengan pemilihan bahasa pemrograman, framework dan basis data yang digunakan, serta hal lain yang terkait dengan teknis penulisan kode program seperti penamaan variable, modul, file, penulisan komentar, dan fungsi sistem.
- b. Standarisasi Dokumentasi Pengembangan Sistem
Dalam standarisasi dokumentasi pengembangan sistem, akan dibahas terkait dengan kerangka laporan pengembangan sistem mulai dari latar belakang, maksud tujuan, metodologi pengembangan yang digunakan, diagram-diagram pendukung (DFD/UML, ERD), progress pengembangan, dokumentasi testing, dan hal-hal yang dirasa penting untuk dilampirkan.

BAB III STANDARISASI TEKNIS PENGEMBANGAN SISTEM

Pasal 3

Standarisasi Teknis Pengembangan Sistem, yaitu :

a. bahasa Pemrograman

Penggunaan bahasa pemrograman dapat ditentukan oleh pengembang yang telah memiliki pengalaman dalam bahasa pemrograman yang dipilih. Tentunya dengan memperhatikan kualitas dari bahasa pemrograman tersebut.

kriteria standarisasi untuk pemilihan bahasa pemrograman:

1. mengutamakan bahasa pemrograman yang telah populer di kalangan pengembang sistem. Hal ini bertujuan untuk menanggulangi resiko kegagalan dalam pengembangan sistem dan mempermudah transisi proyek untuk pengembang yang lain.
2. mengutamakan bahasa pemrograman yang bersifat *Open Source*.

3. versi yang...

3. versi yang digunakan menggunakan versi terakhir (terbaru) yang telah dinyatakan stabil.

Sebagai contoh, untuk membangun aplikasi berbasis web, pengembang dapat menggunakan bahasa pemrograman PHP/Python atau yang lainnya selama memenuhi kriteria di atas.

- b. *Penggunaan Framework*
Framework dapat dipilih sesuai dengan bahasa pemrograman yang dipilih. Umumnya, framework yang dipilih haruslah framework yang telah populer di kalangan pengembang sistem. Sebagai contoh, apabila bahasa yang dipilih adalah PHP, maka gunakan *Laravel/CodeIgniter*. Apabila menggunakan Python, maka gunakan *Django/Flask*. Versi *framework* yang digunakan juga versi terakhir (terbaru) yang telah dinyatakan stabil.
- c. *Pemilihan Basis Data*
Pemilihan basis data dapat ditentukan oleh pengembang sistem. Pemilihan database biasanya tergantung dari kebutuhan sistem itu sendiri. Pengembang sistem harus menentukan dan memperkirakan jenis basis data yang paling sesuai untuk suatu aplikasi. Pengembang diharapkan dapat mengutamakan basis data yang bersifat *Open Source*. Sebagai contoh, MySQL, PostgreSQL, dan lainnya.
- d. *Penamaan File Program*
Dalam penamaan file program, nama yang diberikan haruslah dapat memberikan representasi tentang maksud dari file tersebut. Bahasa yang diperbolehkan untuk digunakan adalah bahasa Inggris atau Indonesia. Sebagai contoh, file program untuk menampilkan data laporan dapat di namai dengan `ViewReport.php` atau `LihatReport.php`. Pengembang harus memperhatikan konsistensi dari penggunaan bahasa. Penamaan yang baik adalah penamaan yang konsisten terhadap bahasa.
- e. *Penamaan Variabel dan Fungsi*
Variabel yang dibuat harus bersifat deskriptif dan terdiri dari satu lebih dari satu kata. Penamaan variabel dapat menggunakan format *camelCase* (awalan kata menggunakan huruf kapital kecuali kata pertama, `Co: getValue, setData, dll`) atau dapat menggunakan *underscores* sebagai pemisah (`Co: get_value, set_data`). Seperti halnya penamaan file, maka penamaan variabel dan fungsi juga harus memperhatikan konsistensi penggunaan bahasa.
- f. *Pemberian Komentar pada Fungsi*
Pemberian komentar pada setiap fungsi yang telah dibuat. Hal ini memiliki tujuan agar kode program lebih dapat dimengerti dengan baik untuk proses pengembangan ke versi selanjutnya. Komentar yang ditulis diharapkan dapat dibaca dan dimengerti dengan baik oleh pembaca kode.

g. *Konsistensi...*

g. Konsistensi Dalam Pemberian Indentasi

Pengembang harus konsisten dalam menggunakan jarak indentasi pada setiap baris kode. Hal ini dapat mempermudah dalam pembacaan suatu kode program. (Contoh: Dalam PEAR Coding Standards <http://pear.php.net/manual/en/standards.php>, bracket opening “{” harus dalam baris yang sama dengan control structure, kecuali dalam penulisan function definition).

Sebagai contoh, pengembang dapat menggunakan 2 (dua) gaya koding di bawah ini:

STYLE 1	STYLE 1
<pre>function foo() { if (\$maybe) { do_it_now(); again(); } else { abort_mission(); } finalize(); }</pre>	<pre>function foo() { if (\$maybe){ do_it_now(); again(); } else { abort_mission(); } finalize(); }</pre>

h. Penggunaan Prinsip *DRY/DIE*

DRY (Don't Repeat Yourself) atau dapat dikenal dengan *DIE (Duplication is Evil)* adalah suatu prinsip agar pengembang sistem dapat menerapkan *usability* dalam kode program yang dibuat. Sebagai contoh, dalam pengembangan website, beberapa komponen seperti header dan footer akan memiliki kode yang sama untuk setiap halaman, maka pengembang dapat memisahkan kode header dan footer tersebut, sehingga halaman lain yang membutuhkannya dapat menggunakan header dan footer yang sama.

i. Menghindari *Deep Nesting*

Deep Nesting dalam suatu control structure dapat mengakibatkan kode yang ditulis menjadi sulit untuk dipahami. Maka, sebisa mungkin pengembang sistem diharapkan dapat menghindari penggunaan *Deep Nesting*. Sebagai contoh:

TERLALU...

TERLALU BANYAK LEVEL DALAM STRUKTUR KONTROL

```
function do_stuff() {
// ...
  if (is_writable($folder)) {
    if ($fp = fopen($file_path,'w')) {
      if ($stuff = get_some_stuff()) {
        if (fwrite($fp,$stuff)) {

          // ...
        } else {
          return false;
        }
      } else {
        return false;
      }
    } else {
      return false;
    }
  } else {
    return false;
  }
} else {
  return false;
}
}
```

Kode di atas memiliki banyak level struktur kontrol. Semakin banyak level nesting dari struktur kontrol, maka kode program akan semakin sulit untuk dibaca. Pengembang dapat menyederhanakan kode program yang menggunakan deep nesting menjadi kode program yang lebih mudah untuk dibaca. Seperti kode program berikut ini:

```
function do_stuff() {
// ...
if (!is_writable($folder)) {
return false;
}
if (!$fp = fopen($file_path,'w')) {
return false;
}
if (!$stuff = get_some_stuff()) {
return false;
}
if (fwrite($fp,$stuff)) {
// ...
} else {
return false;
}
```

j.Pembatasan...

j. Pembatasan *Line Length*

Penulisan kode yang terlalu panjang dalam satu baris secara horizontal membuat kode kurang efisien ketika dibaca. Biasanya, beberapa code-editor seperti Sublime Text, Atom.io, Netbeans, Visual Studio, dan lainnya telah memiliki fitur untuk memberikan garis batas untuk mengukur panjangnya suatu kode dalam satu baris secara horizontal.

CONTOH PENULISAN KODE YANG TERLALU PANJANG YANG DITULIS SECARA HORIZONTAL

```
$my_email->set_from('test@email.com')-  
>add_to('programming@gmail.com')-  
>set_subject('Methods Chained')-  
>set_body('Some long message')->send();
```

CONTOH PENULISAN KODE TELAH DISEDERHANAKAN

```
$my_email  
->set_from('test@email.com')  
->add_to('programming@gmail.com')  
->set_subject('Methods Chained')  
->set_body('Some long message')  
->send();
```

k. Peletakan File dan Folder

Secara teknis, pengembang dapat membuat suatu program dalam 1 (satu) direktori file. Namun hal tersebut sangat tidak efektif. Maka pengembang dapat memecah kode program jadi beberapa bagian dan memasukkannya ke dalam folder dengan penamaan yang baik. Salah satu solusi untuk hal ini adalah dengan menggunakan framework programming. *Framework* biasanya telah memiliki struktur folder yang telah menjadi standar format framework itu sendiri.

BAB IV

STANDARISASI DOKUMENTASI PENGEMBANGAN SISTEM

Bagian Kesatu
Umum

Pasal 4

- (1) Dokumentasi adalah suatu hal yang pertama-tama harus ditentukan dan diselesaikan.
- (2) Hal yang penting agar dokumentasi sebagaimana dimaksud ayat (1) dapat disusun dengan sukses adalah, dilakukan dengan cara mengintegrasikan dokumentasi ini dengan metodologi, sehingga proses dokumentasi dilakukan ketika setiap langkah development dilakukan, daripada melakukannya setelah selesai.

Pasal 5...

Pasal 5

- (1) Dokumentasi yang dibuat sebagaimana dimaksud pada Pasal 4, harus dapat menjawab: *who, what, when, where, why and how* dari pelaksanaan proyek tersebut. Perangkat bantu untuk melaksanakan tugas ini disebut dengan *Statement of the Works (SOW)*.
- (2) SOW sebagaimana dimaksud pada ayat (1) adalah kesepakatan antar client dan developer yang ditulis berdasarkan perspektif bisnis dan teknis yang berisi topik-topik seperti pengantar (misal informasi latar belakang), tujuan dan obyektif (misal cost, jadwal, dan kualitas), scope (misal, aplikasi HTML atau VRML), asumsi (misal kemampuan penanganan peningkatan traffic jaringan), user, sumber daya (misal spesialis jaringan, programmer), milestone untuk penjadwalan (misal waktu akhir testing), pembiayaan, amandement (definisi ulang dari penyerahan pekerjaan), tanda tangan (manajemen senior atau stakeholder terkait).

Bagian Kedua
Standarisasi Dokumentasi Pengembangan Sistem

Pasal 6

Standarisasi Dokumentasi Pengembangan Sistem adalah sebagai berikut :

a. *User Requirement*

Bagian ini menjelaskan kebutuhan dari sistem baik, dari sisi pengguna, fungsi maupun teknologi. Bagian ini terdiri dari:

1) *Requirement Definition*

a. Bagian ini mendefinisikan kebutuhan dan hal-hal yang berkaitan dengan kebutuhan akan sistem secara menyeluruh. Dokumen ini harus disetujui oleh pemberi kerja dan pelaksana kerja. Bagian yang dibahas dalam dokumen ini adalah:

- *Purposeful Requirement*
Bagian ini menjelaskan mengapa kebutuhan tersebut perlu dipenuhi.
- *Functional Requirement*
Bagian ini menjelaskan tentang fungsi yang dibutuhkan oleh user dari sistem yang akan dikembangkan.
- *Nonfunctional Requirement*
Bagian ini menjelaskan tentang kebutuhan lain diluar dari fungsi inti yang biasanya berkaitan dengan operasi sistem, ketersediaan dan portabilitas sistem, dan disain user interface.
- *User Profile*
Bagian ini menjelaskan tentang pengguna dari sistem ini, terutama yang berkaitan dengan tingkat pengenalan terhadap teknologi yang akan diterapkan.

2) *Requirement...*

2) *Requirement Analysis*

Bagian ini akan menganalisis kebutuhan yang dijabarkan pada bagian sebelumnya serta kaitannya dengan hal-hal yang mempengaruhi pelaksanaan ataupun pemilihan solusi.

- *Requirement Prioritization*

Bagian ini menjelaskan tentang kebutuhan mana yang terpenting dari kebutuhan-kebutuhan yang telah terdefiniskan untuk sistem yang akan dibuat tersebut.

- *Requirement Prioritization*

Bagian ini menjelaskan tentang halangan dan resiko yang ada untuk mengimplementasikan model dan mengaplikasikan sistem ini.

- *Trade-off Analysis*

Bagian ini menjelaskan tentang hal-hal yang saling bertentangan dalam perancangan, pengimplementasian serta pengaplikasian dari solusi yang ditawarkan.

3) *Requirement Model*

Bagian ini menyusun model kebutuhan tersebut menjadi suatu bentuk model kebutuhan. Model ini digunakan untuk mengurangi ketidak pastian kebutuhan sistem, sehingga tercapai kesepakatan antara pemberi kerja dan pelaksana kerja.

- *Functional Model*

Bagian ini digunakan untuk menggambarkan fungsi apa saja yang dibutuhkan dalam pembangunan suatu sistem. (Dapat direpresentasikan dengan UseCase Diagram).

- *Exploratory Prototype*

Bagian ini digunakan untuk menjelaskan deskripsi fungsi secara umum dengan beberapa bagian dari prototype yang telah dibuat (bila ada).

- *Conceptual Model*

Bagian ini digunakan untuk menggambarkan konseptual desain dari sistem yang akan dibangun.

b. *System Specification*

Pada bagian ini dijabarkan spesifikasi detail dari sistem yang harus dibuat, spesifikasi ini meliputi hal-hal berikut ini:

1) *System Operating Cycle Specification*

Dalam bagian ini dijelaskan siklus penggunaan sistem ini, tujuan untuk mendapatkan spesifikasi teknis terutama akan berguna pada bagian implementasi. Sistem fault tolerant 24 jam akan berbeda dengan sistem yang beroperasi 1 jam sehari. Begitu juga siklus kerja sistem, atau langkah kerja satu demi satu di spesifikasikan.

2)Functional...

2) *Functional Specification*

Pada bagian ini dijabarkan fungsi dari sistem yang akan dibuat.

- *Essential Capabilities*
Bagian ini akan menjelaskan fungsi utama dari sistem. Ini merupakan fungsi minimal yang harus dipenuhi oleh sistem yang dibuat.
- *Additional Capabilities*
Bagian ini akan mendeskripsikan fungsi tambahan yang timbul karena dipenuhi dan digunakannya sistem yang akan dibangun.
- *Future Capabilities*
Menjelaskan fungsi pada masa datang dari sistem ini. Penjelasan ini erat kaitannya dengan perkembangan teknologi dan pengembangan fitur agar sistem dapat selalu memenuhi kebutuhan pengguna.

3) *System Component*

- *Hardware*
Bagian ini menjelaskan tentang perangkat keras yang dibutuhkan untuk mengimplementasikan sistem yang akan dibangun.
- *Software*
Bagian ini menjelaskan tentang perangkat lunak yang dibutuhkan untuk mengimplementasikan sistem yang akan dibangun.
- *User*
Bagian ini menjelaskan tentang pengguna yang akan berhubungan langsung dengan sistem.

c. *Performance Specification*

Pada bagian ini dijelaskan performansi/unjuk kerja yang ingin dicapai serta kemungkinan keterbatasan di dalam penggunaan sistem. Setiap kebutuhan diusahakan dispesifikasikan dengan jelas, termasuk karakteristik elektrik dan fisis.

1) *Characteristics and Constraints*

Bagian ini menjelaskan tentang karakteristik tiap komponen pendukung sistem, termasuk karakteristik fisis, elektrik, maupun karakteristik perangkat lunak (misal user friendly, interaktif). Bagian ini juga menjelaskan keterbatasan di dalam penggunaan sistem.

2) *Physical Characteristics*

Bagian ini menjelaskan tentang bentuk luar dari sistem secara keseluruhan. Misal ukuran berat dan lain-lain.

3) *Environmental Characteristics*

Bagian ini menjelaskan lingkungan tempat pengoperasian sistem yang akan dibangun.

4) *Human Factors*

Bagian ini menjelaskan pengaruh manusia di dalam pengoperasian sistem yang akan dibangun, baik sebagai penentu operasi, misal dalam decision making system, atau supervised control system. Juga pengaruh manusia dalam menimbulkan ketidakakuratan atau kesalahan sistem.

d. Design System...

d. Design System

Bagian ini akan menjelaskan tentang penjabaran metoda dari solusi yang digunakan untuk memenuhi kebutuhan yang telah dispesifikasikan. Yang dijelaskan dalam bagian ini adalah langkah perancangan dari solusi yang ditawarkan. Sedapat mungkin harus dipisahkan antara perancangan dan implementasi. Pada perancangan diusahakan sedapat mungkin yang dilakukan adalah memodelkan solusi secara logika atau secara algoritmis. Tanpa terkait erat dengan teknologi yang digunakan dalam proses implementasi dari model. Keterkaitan implementasi hanyalah menjadi constraint dari model bukan menjadi dasar disain. Langkah-langkah disain akan sangat bergantung pada model dari sistem yang dibuat. Pada dasarnya akan meliputi:

1) *Diagram Blok*

Pada bagian ini akan digambarkan blok pembangunan sistem secara keseluruhan juga digambarkan interface antara kedua bagian tersebut. Blok diagram terdiri dari 2 bagian:

- Representasi blok bagian luar sistem yang berhubungan dengan sistem.
- Representasi blok bagian dalam sistem.

2) *Representation of Data Flow*

Bagian ini akan menggambarkan aliran data serta transformasi yang dialami data tersebut dalam sistem.

3) *Entity Relationship Diagram*

Diagram ini akan menggambarkan hubungan antar entitas yang saling terlibat berikut dengan setiap atributnya yang akan digunakan sebagai acuan pembangunan desain basis data dari suatu sistem yang akan dibangun

4) *Database Diagram*

Diagram ini akan menggambarkan detail dari setiap entitas yang telah digambarkan dalam Entity Relationship Diagram berikut dengan tipe data, panjang karakter, dan keterangan tambahan (bila dibutuhkan).

5) *Activity Diagram*

Diagram ini akan menggambarkan aktifitas yang dapat dilakukan pengguna dalam penggunaan fungsi yang akan dibuat dalam sistem. Bagian ini dapat digunakan untuk merepresentasikan aktifitas bisnis proses maupun aktifitas penggunaan fungsi sistem.

6) *State-machine Diagram*

Diagram ini akan menggambarkan suatu proses perubahan kondisi/status yang perlu untuk digambarkan.

7) *Component Diagram*

Diagram ini akan menggambarkan komponen-komponen apa saja yang dibutuhkan dalam pengimplementasian sistem yang akan dibangun. Bagian ini dapat digunakan untuk menggambarkan komponen perangkat lunak diluar sistem yang berhubungan langsung dengan sistem yang akan dibangun.

8) *Deployment...*

8) *Deployment Diagram*

Diagram ini akan menggambarkan komponen-komponen perangkat keras yang dibutuhkan dalam pengimplementasian sistem yang akan dibangun. Diagram ini terdiri dari beberapa node yang terhubung dengan node yang lainnya yang bertujuan untuk menggambarkan secara fisik bagaimana sistem yang dibangun ini diimplementasikan.

e. *Testing Documentation (Use-Case, Black Box)*

Bagian ini menunjukkan kerja dari sistem baik untuk masukan yang bersifat normal, ataupun untuk masukan yang di luar ambang normal. Setiap pengujian dilakukan dokumentasi sebagai bukti otentik kemampuan sistem. Pada dasarnya disamping testing yang bersifat testing dalam kondisi operasi, akan baik pula bila dilakukan testing berikut ini:

1) *Recovery Testing*

Pengujian dilakukan untuk mengetahui kemampuan sistem, untuk mengembalikan ke kondisi normal setelah suatu masukan atau kondisi di luar dari yang dispesifikasikan. Untuk sistem-sistem yang bersifat fault tolerant, jenis pengujian ini merupakan suatu kewajiban.

2) *Stress Testing*

Pengujian ini dilakukan untuk mengetahui kemampuan sistem dalam menangani beban kerja yang berat, sangat baik untuk mengetahui kemampuan maksimal dari sistem.

3) *Security Testing*

Testing ini dilakukan untuk jenis-jenis aplikasi yang berkaitan dengan keamanan sistem, baik dari pengguna yang melakukan kesalahan tidak sengaja, ataupun sengaja.

Pasal 7

Setelah dokumentasi sistem telah terdefiniskan dengan baik, maka pengembang sistem berkewajiban untuk membuat manual penggunaan yang isinya dapat terdiri dari:

a. Pengantar

- Tujuan sistem
Bagian ini menjelaskan tujuan sistem ini dibuat.
- Lingkungan operasi
Bagian ini menjelaskan tentang lingkungan pengoperasian dari sistem yang dibuat.
- Fungsi sistem secara umum
Bagian ini menjelaskan tentang fitur dari sistem yang telah dibuat secara umum.
- Fitur khusus
Bagian ini menjelaskan tentang fitur khusus yang dapat diakses oleh masing-masing level pengguna dengan batasan fungsi tertentu.

Keterbatasan...

- Keterbatasan
Bagian ini menjelaskan tentang hal-hal yang tidak dapat dilakukan dengan sistem yang telah dibuat.
- Keterangan tambahan
Keterangan tambahan dapat dibuat sesuai dengan kebutuhan penjelasan dari suatu sistem.

b. Installasi

- *Minimum requirement specification*
Bagian ini menjelaskan tentang spesifikasi minimal suatu server dapat digunakan sebagai perangkat implementasi dari sistem yang telah dibuat.
- Menyalin dan memback-up
Bagian ini menjelaskan tentang bagaimana pengguna dapat melakukan proses back-up dan menyalin data.
- Proses instalasi
Bagian ini menjelaskan tentang bagaimana proses instalasi sistem yang telah dibuat ini dilakukan.
- Konfigurasi
Bagian ini menjelaskan beberapa konfigurasi yang ada dalam sistem (bila ada).

c. Tutorial

Penjelasan langkah demi langkah dengan contoh
Bagian ini menjelaskan langkah demi langkah suatu fitur dapat digunakan oleh pengguna sistem. Bagian ini akan dibagi sesuai dengan level akses pengguna di dalam sistem.

d. Instruksi Detail

- Output dari sistem
Bagian ini menjelaskan tentang keluaran yang dapat digenerate oleh sistem. Output dapat berupa file dokumen maupun file back-up sistem.
- Input dari sistem
Bagian ini menjelaskan tentang inputan yang mungkin dimasukkan ke dalam sistem.
- Pengoperasian sistem
Bagian ini menjelaskan tentang bagaimana cara mengoperasikan sistem secara umum.
- Penanganan error
Bagian ini menjelaskan tentang penyebab sistem memberikan pesan error ketika terjadi kesalahan beserta dengan penjelasan dan cara menanganinya.
- Fungsi khusus
Bagian ini menjelaskan tentang fitur khusus yang mungkin terdapat dalam sistem serta penjelasan seputar cara pengoperasiannya.

e. Detail Teknis

- Fitur lanjutan
Bagian ini berisikan tentang fitur-fitur yang dapat dijadikan sebagai acuan pengembangan dalam sistem yang telah dibuat.
- Algoritma utama yang digunakan
Bagian ini berisikan tentang algoritma khusus yang digunakan dalam suatu sistem (bila ada).

Struktur...

- Struktur data utama
Bagian ini berisikan tentang struktur data khusus yang digunakan dalam algoritma utama (bila ada).
- Cara memperoleh dukungan teknis dan informasi lanjutan
Bagian ini berisi seputar bagaimana pengguna dapat menghubungi pihak pengembang sistem untuk mendapatkan informasi lanjutan.

Pasal 8

Peraturan Bupati ini mulai berlaku pada tanggal diundangkan.

Agar setiap orang mengetahuinya, memerintahkan pengundangan Peraturan Bupati ini dengan penempatannya dalam Berita Daerah Kabupaten Tangerang.

Ditetapkan di Tigaraksa
pada tanggal 10 Desember 2018

BUPATI TANGERANG,



A. ZAKI ISKANDAR

Diundangkan di Tigaraksa
pada tanggal 10 Desember 2018

SEKRETARIS DAERAH
KABUPATEN TANGERANG,



MOCH. MAESYAL RASYID

BERITA DAERAH KABUPATEN TANGERANG TAHUN 2018 NOMOR 56