|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\Admin\Pictures\JATA KPM.png  **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | | |
| **KOD DAN NAMA PROGRAM NOSS** | | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP** | | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | | **CU2/WA4 – WRITE MODULE CODE** | |
| **KOD DAN NAMA PROGRAM KV** | | TEKNOLOGI SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | | KPD 2023 WEB PROGRAMMING | |
| **NO.DAN TAJUK STANDARD KANDUNGAN KSKV** | | K4 WRITE MODULE CODE | |
| **NO. KOD** | **NO. KOD NOSS** | IT-010-3:2016-C02/P(58/70) | Muka Surat : 1 Drp : 13 |
| **NO. KOD KSKV** | KPD 2023 / P(10/11) |

**TAJUK/***TITLE***:**

**JENIS DOKUMEN SPESIFIKASI FUNGSI, ANTARAMUKA PENGGUNA DAN PENGALAMAN PENGGUNA, APLIKASI *MOCK-UP*, PENGUJIAN UNIT DAN PROSEDUR *DEBUG***

**TUJUAN/***PURPOSE***:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

1. Senarai jenis-jenis dokumen spesifikasi fungsi
2. Terang maksud antaramuka pengguna dan pengalaman pengguna
3. Terang aplikasi *Mock up*
4. Terang maksud Ujian Unit
5. Terang procedur *debugging.*

**4.12 DEFINITION OF USER INTERFACE AND USER EXPERIENCE**

**User Experience (UX)**

UX merujuk kepada merekabentuk antaramuka pengguna pada tahap pengalaman pengguna yang bermatlamatkan pada kefungsian iaitu sejauh mana aplikasi itu menepati matlamat pengguna, mudah digunakan dan mudah dipelajari oleh pengguna akhir.

**User Interface (UI)**

UI pula bermatlamatkan pada sudut pandangan iaitu penghasilan paparan atau bentuk aplikasi atau laman web yang mencakupi hal tatacara penyusunan atau layout dan rekabentuk visualnya contohnya penggunaan penempatan butang-butang arahan, penggunaan warna dan font agar paparan yang sesuai dapat ditampilkan kepada pengguna.

**4.13 APLIKASI MOCKUP**

*Mock up* adalah proses merekabentuk peringkat sederhana yang boleh dikatakan sebagai rekabentuk pra-keluaran yang di datangkan dengan paparan yang sebenar dan menggambarkan antara mukasebenar aplikasi yang mengandungi warna dan gaya (*style*) namun belum berfungsi. *Mock up* mengandungi struktur maklumat, memaparkan kandungan dan mendemonstrasikan fungsi yang asas dengan cara yang statik kerana ia tidak benar-benar berfungsi contohnya aplikasi tersebut tidak mampu menyimpan sebarang input dari pengguna kerana belum dihubungkan dengan pangkalan data. Antara perisian yang boleh membangunkan *mock up* ialah *Mokingbird, Mockup builder, proto.io* (untuk aplikasi mudah alih) dan banyak lagi.

**KELEBIHAN** **PELAKSANAAN APLIKASI MOCK UP**

1. Mudah memahami keperluan sebenar pelanggan apabila pengguna terlibat secara aktif dalam pembangunan
2. Pembangun dapat membangunkan aplikasi atau menghasilkan produk akhir yang menepati kehendak pengguna
3. Oleh kerana di dalam kaedah ini menyediakan simulasi kepada pengguna, pengguna akan dapat pemahaman yang lebih baik daripada sistem yang dibangunkan.
4. Kesilapan boleh dikesan lebih awal.
5. Maklum balas pengguna yang lebih cepat boleh didapati membawa kepada penyelesaian yang lebih baik.
6. Fungsi yang hilang boleh dikenal pasti dengan mudah
7. Fungsi mengelirukan atau sukar dapat dikenal pasti
8. Dengan simulasi prototaip mungkin akan mencetus beberapa idea tambahan yang tidak dijangkakan malah lebih diperlukan

**KEKURANGAN PELAKSANAAN APLIKASI MOCK UP**

1. Mengabaikan struktur pembangunan yang lebih terancang dengan terus ke tahap pelaksanaan kod dan terus melakukan penambahbaikan. Analisis masalah tidak dilakukan dengan jelas.
2. Kaedah ini boleh meningkatkan kerumitan sistem di mana skop sistem itu boleh berkembang di luar rancangan asal.
3. Proses mungkin lebih rumit disebabkan banyak melibatkan pengguna yang kadangkala tidak memahami konsep sebenar sesuatu aplikasi.
4. Pelanggan mungkin mahu terus menggunakan aplikasi prototaip tanpa menunggu aplikasi sebenar dihasilkan.
5. Usaha yang lebih daripada kumpulan pembangun untuk menghasilkan aplikasi tersebut dan terpaksa membuat penambahbaikan dari masa ke semasa.

**4.14 PENGUJIAN UNIT (*UNIT TESTING*)**

Pengujian unit dilakukan terhadap unit terkecil yang dinamakan modul. Modul boleh terdiri daripada satu fungsi atau prosedur. Setiap modul perlu diuji dari beberapa aspek seperti pengendali ralat, antaramuka, laluan logik, struktur data, nilai sempadan dan kelas data.

* Pengendali ralat : menangani ralat daripada berlaku semasa pelaksanaan
* Antaramuka : menyemak parameter masuk dan keluar daripada modul atur cara
* Laluan logik : memastikan setiap penyataan diuji sekurang-kurangnya sekali
* Struktur data : Data yang tersimpan dapat mengekalkan integrasi semasa perlaksanaan
* Nilai sempadan dan kelas data : Memastikan modul berfungsi dengan baik pada nilai sempadan. Kelas data yang sah dan tidak sah juga diuji.

**4.15 PENGUJIAN KEBOLEHGUNAAN (*USABILITY TESTING*)**

Kebolehgunaan terhadap aplikasi yang dibangunkan hanya boleh ditentukan apabila pengujian kebolehgunaan dilaksanakan. Pembangun boleh melihat pengguna berinteraksi dengan aplikasi dan pengguna akan menjawab soalan-soalan yang disediakan seperti ;

* Adakah pengguna boleh menggunakan aplikasi tanpa bantuan dan arahan?
* Adakah rekabentuk aplikasi mudah dan ringkas?
* Adakah pengguna mengalami gangguan teknikal semas menggunakan aplikasi ini?
* Adakah pengguna boleh mengawal sepenuhnya fungsi dalam aplikasi ini?
* Adalah laras bahasa atau penggunaan simbol cukup difahami oleh pengguna?

Jika banyak soalan yang dijawab oleh pengguna adalah ‘Ya’ ia boleh dikatakan sebagai kebolehgunaan telah tercapai. Pengujian ini penting untuk mengukur tahap kepuasan pengguna dan mengurangkan kos sokongan dan juga dokumentasi.

Pengujian kebolehgunaan boleh dilakukan untuk menguji fungsi dan penerimaan pengguna terhadap antaramuka. Pengujian antaramuka perlu mengambilkira 5 prinsip asas rekabentuk interaksi iaitu konsistensi, kebolehan membuat pemerhatian, boleh dipelajari, kebolehan untuk menjangka dan maklumbalas.

**Prinsip asas reka bentuk interaksi**

* **Konsistensi**

Konsistensi merujuk kepada kedudukan setiap elemen dalam setiap halaman. Semua elemen perlu kekal pada kedudukan yang sama supaya pengalaman pengguna tidak terganggu untuk menggunakan aplikasi tersebut. Sebagai contoh apabila merujuk kepada papan cerita kajian kes yang diberi tadi, kedudukan untuk KELUAR sistem adalah sama bagi kedua-dua modul dan akan konsisten pada setiap halaman yang berkaitan.

* **Kebolehan membuat pemerhatian**

Kebolehan membuat pemerhatian merujuk kepada sejauh mana pengguna berasa selesa dan selamat untuk menggunakan sebarang elemen tanpa ragu-ragu. Contohnya elemen untuk menerima dan menolak permohonan oleh pensyarah jelas dan mesejnya mudah sampai kepada pengguna tersebut. Oleh yang demikian, pengguna dapat menggunakan aplikasi tanpa sebarang keraguan.

* **Boleh dipelajari**

Boleh dipelajari penting bagi pengalaman pengguna buat kali pertama menggunakan aplikasi. Sejauh mana pengguna boleh mengingati cara untuk mengendalikan aplikasi tersebut perlu diambilkira.

* **Kebolehan untuk menjangka**

Kebolehan untuk menjangka pula ialah pengguna dapat menjangka apa yang akan berlaku apabila menggunakan sebarang elemen dalam aplikasi. Sebagai contoh apabila pengguna memilih fungsi untuk Cadang Tajuk, pengguna dapat menjangka untuk mengisi sesuatu bagi mencadangkan tajuk projek akhir.

Pengguna akan dibawa ke halaman untuk memasukkan tajuk Projek Tahun Akhir yang akan dipaparkan dalam senarai yang akan dipohon oleh pelajar nanti.

* **Maklumbalas**

Maklumbalas bertujuan untuk melakukan penambahbaikan pada rekabentuk interaksi yang dihasilkan.

**KAEDAH PENILAIAN**

Antara kaedah penilaian yang boleh digunakan dalam menjalankan pengujian sistem ialah soal selidik. Soal selidik boleh dibangunkan untuk menilaian kebolehgunaan produk berdasarkan prinsip asas rekabentuk di atas. Pilihan jawapan dalam soal selidik boleh ditampilkan dalam dua jenis iaitu bentuk Skala Likert ataupun bentukSkala Guttmen.

Skala Likert digunakan untuk melihat sejauh mana persetujuan pengguna terhadap produk yang dibangunkan. Jenis ini menampilkan skala yang mewakili maksud seperti

5 = Amat setuju

4 = Setuju

3 = Tidak setuju

2 = Sangat tidak setuju

1 = Tidak pasti

Skala Guttman pula bertujuan untuk mendapatkan data yang lebih jelas, tegas dan konsisten iaitu menampilkan pilihan ‘Ya’ yang mewakili setuju dan ‘Tidak’ untuk tidak atau kurang bersetuju.

**4. 16 PROSEDUR PENYAHPIJATAN**

Terdapat 12 prosedur untuk menghasilkan laporan pengujian dan pepijat yang efektif. Prosedur ini merangkumi apa yang perlu ada didalam laporan pengujian. Prosedur yang dimaksudkan adalah:

1. Tujuan penghasilan laporan pengujian

Keterangan ringkas mengenai objektif penghasilan laporan. Contohnya: ***“Dokumen ini menerangkan pelbagai aktiviti pengujian sistem yang dilakukakan keatas Sistem Pengurusan Pelajar”.***

1. Pengenalan kepada sistem yang diuji

***“Sistem Pengurusan Pelajar ada sistem berasaskan web. Semua maklumat pelajar akan dipaparkan kedalam sistem seperti maklumat peribadi, maklumat kewangan, maklumat pendaftaran pelajaran, maklumat peperiksaan dan maklumat aktiviti. Ada beberapa modul yang ada iaitu modul Pelajar, Kewangan, Subjek dan Aktiviti”.***

1. Skop pengujian

Skop pengujian terbahagi kepada 3 iaitu *in-scope, out-of-scope* dan item yang tidak diuji. *In-scope* bermaksud semua kefungsian sistem perlu diuji bagi memastikan ia memenuhi keperluan pelanggan. *Out-of-scope* pula menerangkan bahagian sistem yang tidak diuji. Contohnya prestasi sistem tidak akan diuji. Item yang tidak diuji pula contohnya Sistem Maklumat Pelajar memerlukan *third-party system* untuk melaksanakan transaksi yuran iaitu Sistem Bendahari. Sistem Bendahari tidak akan diuji kerana sistem itu adalah sistem luaran.

1. Metriks

Metrik akan menjadikan keputusan pengujian akan lebih difahami. Contohnya pada Rajah 3.1 – Rajah 3.3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Kes Pengujian** | **Kes kajian yang diuji** | **Kes kajian Berjaya** | **Kes kajian gagal** |
| 80 | 75 | 70 | 5 |

**Rajah 3.1:** Jumlah Kes pengujian yang berjaya/gagal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Kritikal | Major | Medium | Low | Jumlah |
| Tutup | 25 | 15 | 20 | 0 | 60 |
| Buka | 0 | 0 | 0 | 5 | 5 |

**Rajah 3.2:** Jumlah Kes pengujian yang gagal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Pelajar** | **Kewangan** | **Subjek** | **Aktiviti** | **Jumlah** |
| **Kritikal** | 6 | 7 | 8 | 5 | 26 |
| **Major** | 4 | 5 | 2 | 4 | 15 |
| **Medium** | 6 | 8 | 2 | 4 | 20 |
| **Low** | 1 | 2 | 1 | 1 | 5 |
| **Jumlah** | 17 | 22 | 13 | 14 | 66 |

**Rajah 3.3:** Jumlah semua kegagalan kes pengujian

1. Jenis-jenis pengujian yang dilakukan

Perlu dinyatakan jenis-jenis pengujian yang telah dijalankan seperti *smoke* dan *sanity testing, regression testing* dan *Alpha testing.*

1. Persekitaran pengujian

Menyatakan dengan jelas persekitaran pengujian dimana pengujian sistem dilakukan. Contohnya Rajah 3.4.

|  |  |
| --- | --- |
| URL | http://smp.edu.my |
| IP pelayan | 192.168.133.22 |
| Pangkalan data | Oracle 12g |
| HP QC/ALM | 192.168.137.22 |

**Rajah 3.4:** Persekitaran Pengujian

1. Maklumbalas daripada hasil pengujian

Menerangkan isu-isu yang kritikal semasa proses pengujian dilakukan dan cara penyelesaiannya. Maklumbalas ini perlu didokumenkan secara formal agar dapat membantu penguji semasa proses pengujian seterusnya dengan mengelakkan melakukan kesalahan yang sama.

1. Cadangan penambahbaikkan

Cadangan penambahbaikkan dari segi proses pengujian. Sebagai contoh, penguji boleh memberi cadangan untuk menggunakan *tools* bagi menghasilkan kes pengujian secara automatik.

1. Praktik yang efektif bagi pengujian

Penguji mendokumenkan praktik pengujian yang efektif berdasarkan pengalamannya melakukan proses pengujian.

1. Ciri-ciri penamatan pengujian sistem

Bahagian ini sangat penting bagi menentukan bila pengujian sistem akan ditamatkan. Ini kerana, adalah mustahil untuk memastikan sistem 100% bebas daripada pepijat. Antara ciri-ciri yang boleh diambil kira adalah :

i) Semua kes pengujian telah diuji

ii) Semua pepijat yang kritikal telah dibuat penambahbaikkan.

1. *Sign off*

Perjanjian diantara penguji bagi menentukan samaada proses pengujian perlu ditamatkan atau tidak. Ianya banyak bergantung kepada ciri-ciri penamatan pengujian sistem. Hal ini perlu mendapat persetujuan kepada semua penguji.

1. Definisi dan akronim yang digunakan

Mendokumenkan definisi dan akronim untuk dirujuk pada masa hadapan.

**4.16.1 HALAMAN RALAT TERSUAI**

Tujuan Halaman ralat tersuai dibina oleh pembangun sesebuah sistem adalah untuk menunjukkan pengunjung laman web tesebut terdapat kesalahan /ralat

pada page yang dikunjungi atau diminta. Halaman ralat tersuai juga boleh berlaku apabila terdapat kegagalan pada sesuatu sistem semasa beroperasi. Ia berfungsi dengan mana-mana tema apabila ia diaktifkan pada blog anda. Pemilik laman web boleh menyesuaikan *404 Not Found* dan halaman ralat lain di laman web mereka.

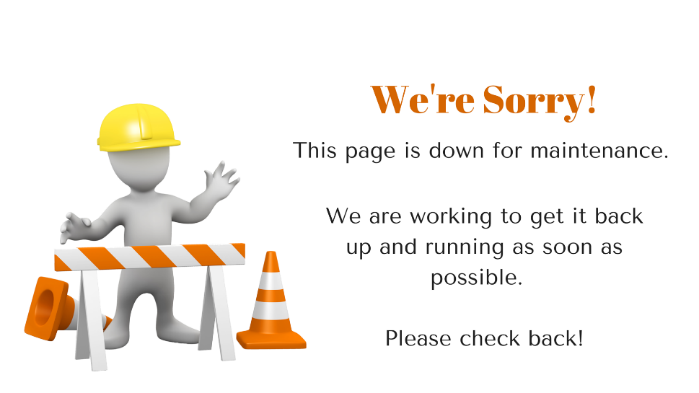
Sebagai contoh, di sini di *How-To Geek* mempunyai ralat khas *404 Page Not Found* yang diilhami oleh permainan Mario klasik. Kesalahan ini bermaksud perkara yang sama, tetapi disesuaikan untuk menjadi lebih mesra dan membantu mencari apa yang anda cari. Contohnya Rajah 3.5.

Rajah 3.5 Halaman ralat tersuai

**4.16.2 JENIS RALAT TERSUAI**

1. Sedang dalam Penyelenggaraan (*Under Maintenance)*

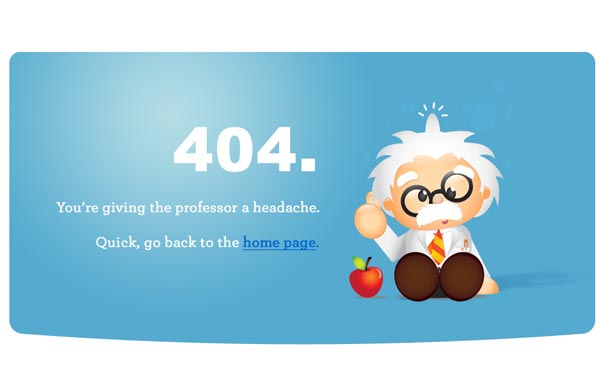
Halaman penyelenggaraan adalah pemegang tempat sementara apabila tapak atau aplikasi perlu diambil di luar talian untuk dikemas kini, dijadikan sandaran atau penyelenggaraan. Ia memberitahu pengguna bahawa laman web yang cuba di capai buat sementara tidak tersedia kerana dalam proses membaik pulih. Ia bukan adegan jenayah, hanya zon pembinaan sementara. Contohnya Rajah 3.6



Rajah 3.6 Sedang dalam Penyelenggaraan

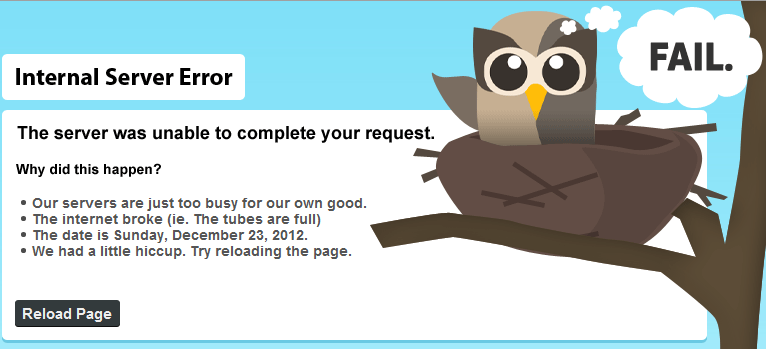
1. File tidak dijumpai *(file not found)*

Anda mungkin melihat pelbagai mesej pelayan web apabila mengakses laman web. Yang paling biasa ialah "404 Not Found," yang bermaksud anda cuba mengakses halaman yang tidak wujud. Sama ada halaman web telah dialih keluar atau anda sedang menaip di alamat dan salah taipkannya. Mesej ralat ini dijana oleh pelayan web jauh dan dihantar ke penyemak imbas anda. Jika anda melihatnya, semak semula alamat halaman web yang anda taipkan. Sekiranya anda mengklik pautan, pautan itu salah - atau halaman yang ditunjukkan olehnya telah dialih keluar. Contohnya Rajah 3.7



Rajah 3.7 Fail tidak dijumpai

1. Rangkaian hilang (Network lost)Ada sesuatu yang tidak kena pada pelayan web dan ia tidak dapat memproses permintaan atau kemas kini anda.



Rajah 3.8 Rangkaian hilang

**4.16.3 ALAT PENYAHPIJATAN**

Alat penyahpijatan digunakan untuk empat sebab terutamanya:

* Ujian pepijat. Kelemahan dengan mudah boleh dieksploitasi jika terdapat 'kesilapan' dalam kod anda.
* Menemui sumber pepijat itu
* Mampu mengasingkan pepijat dan kesilapan. Cuba mencari sumber apa yang menyebabkan pepijat.
* Menetapkan dan 'menampal' kesilapan atau logik yang menyebabkan kelemahan yang berpotensi

Antara perisian yang digunakan untuk penyahpijatan

* Advanced Debugger – Standard Penyahpijatan UNIX
* AQtime — profile dan memori/sumber penyahpijatan untuk *Windows*
* CodeView — adalah penyahpijatan untuk platform DOS
* DBG — PHP Debugger dan Profiler
* DEBUG — penyahpijatan terbina dalam DOS dan Microsoft Window
* WinDbg — penyahpijatan pelbagai guna untuk Windows
* GNU Debugger – penyahpijatan mudah alih yang berjalan pada sistem Unix dan kerja untuk bahasa pengaturcaraan termasuk Ada, C, C++, Objective-C, Free Pascal, Fortran, Go and Java

**4.16.4 MASA PUSING BALIK *(TURNAROUND TIME)***

Masa pusingan (TAT) bermaksud jumlah masa yang diambil untuk memenuhi permintaan terhadap sesuatu proses. Dalam pengkomputeran, masa pusingan adalah jumlah masa yang diambil antara penyerahan program / proses / tugas untuk pelaksanaan dan pengembalian output lengkap kepada pelanggan / pengguna. Ia mungkin berbeza-beza untuk pelbagai bahasa pengaturcaraan bergantung kepada perisian atau program.

**4.16.5 JENIS-JENIS PENGUJIAN**

1. Pengujian Kefungsian

Ujian Kefungsian merupakan proses pengujian sistem/perisian yang digunakan dalam pembangunan perisian di mana perisian diuji untuk memastikan ia memenuhi semua keperluan.

Ujian kefungsian dilaksanakan dengan cara menyemak sistem/perisian untuk memastikan bahawa ia mempunyai semua fungsi yang diperlukan yang ditentukan dalam keperluan fungsinya.

1. Pengujian Fail

Pengujian dilaksanakan berdasarkan saiz fail dan juga format pada fail tersebut samada disokong oleh sistem tersebut ataupun tidak.

Contohnya memuat naik fail dengan mengambil kira saiz dan format fail.

1. Pengujian Data Integriti

Mengesahkan data dalam pangkalan data yang tepat dan berfungsi seperti yang diharapkan. Data yang dimasukkan perlu diselaraskan dengan *data dictionary.*

Merujuk kepada proses manual atau automatik yang digunakan oleh pentadbir pangkalan data untuk memastikan ketepatan, kualiti dan fungsi data yang disimpan dalam pangkalan data.

**SOALAN/***QUESTION***:**

1. Berikan jenis-jenis ralat tersuai.
2. Terangkan kegunaan aplikasi mock up
3. Terangkan perbezaan antara UI dan UX
4. Terangkan jenis-jenis pengujian.

**RUJUKAN/***REFERRENCE***:**

1. Yogesh Singh, (2012). Software Testing. New York: Cambridge University Press. Mukasurat:95 hingga110,
2. Kaner, C. & Fiedler, R.L. (2013). Foundations of Software Testing. Context- Driven Press, Mukasurat:72 hingga 81
3. McCaffrey, J.D. (2009). Software Testing: Fundamentals Principles and Essential Knowledge. BookSurge Publishing. Mukasurat:62 hingga 72
4. Pezzá, M. & Young, M. (2008). Software Testing and Analysis: Process, Principles and T echniques. Hoboken: Wiley & Sons, Inc. Mukasurat:52 hingga 61.
5. Pressman, R.S. (2009). Software Engineering, A Practitioner’s Approach (7th Edition). New York: McGraw Hill. Mukasurat:47 hingga 62
6. https://en.wikipedia.org/wiki/Architectural\_pattern 11 Feb 2019 10.54
7. https://en.wikipedia.org/wiki/List\_of\_debuggers 7 Aug 2019 11.00