**BAB II**

**TINJAUAN PUSTAKA**

Pada bab kedua ini akan dibahas mengenai beberapa definisi yang berkaitan dengan judul tugas akhir dan juga teori serta istilah dalam penyusunan tugas akhir yang dimaksud. Dengan adanya pembahasan dalam bab kedua ini diharapkan tugas akhir yang disusun dapat lebih mudah dipahami.

**2.1 Pengertian Pembangunan Aplikasi**

**2.1.1 Pengertian Pembangunan**

Pengertian Pembangunan menurut Rogers adalah suatu proses perubahan sosial dengan partisipatori yang luas dalam suatu masyarakat yang dimaksudkan untuk kemajuan sosial dan material (termasuk bertambah besarnya kebebasan, keadilan dan kualitas lainnya yang dihargai) untuk mayoritas rakyat melalui kontrol yang lebih besar yang mereka peroleh terhadap lingkungan mereka.

**2.1.2 Pengertian Aplikasi**

Menurut **Kamus Kamus Besar Bahasa Indonesia** (KBBI) aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari user (pengguna).

**2.1.3 Pengertian Pembangunan Aplikasi**

Pembangunan aplikasi merupakan perancangan sistem yang dapat digunakan untuk mengolah suatu data menjadi informasi yang bermanfaat untuk dijadikan sebagai bahan penganmbil keputusan bagi organisasi/kelompok untuk mendukung pencapain suatu tujuan.

**2.2 Pengertian Parkir**

Menurut kamus besar bahasa Indonesia definisi parkir ialah menghentikan atau menaruh (kendaraan bermotor) untuk beberapa saat ditempat yang sudah disediakan . Pengertian diatas memiliki definisi dari penyedia jasa layanan parkir yaitu penyedia tempat untuk menerima penghentian atau penaruhan (kendaraan bermotor) untuk beberapa saat. Jika melihat fungsi dari perparkiran terdapat juga asumsi parkir digunakan sebagai tempat penitipan barang yang memiliki arti apabila seseorang menerima sesuatu barang dari orang lain, dengan syarat bahwa ia akan menyimpannnya dan mengembalikannya dalam wujud asalnya menurut Pasal 1694 KUH Perdata.

**2.3 QR Code**

**2.3.1 Pengertian QR Code**

Menurut Wikipedia *QR Code* adalah suatu jenis kode [matriks](https://id.wikipedia.org/wiki/Matriks) atau [kode batang](https://id.wikipedia.org/wiki/Kode_batang) dua dimensi yang dikembangkan oleh [*Denso Wave*](https://id.wikipedia.org/w/index.php?title=Denso_Wave&action=edit&redlink=1), sebuah divisi [*Denso Corporation*](https://id.wikipedia.org/w/index.php?title=Denso_Corporation&action=edit&redlink=1) yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun [1994](https://id.wikipedia.org/wiki/1994) dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh [pemindai](https://id.wikipedia.org/wiki/Pemindai) *QR* merupakan singkatan dari [*quick response*](https://id.wikipedia.org/w/index.php?title=Quick_response&action=edit&redlink=1) atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, *QR Code* mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis *QR Code* dapat menampung informasi yang lebih banyak daripada [kode batang](https://id.wikipedia.org/wiki/Kode_batang).

**2.3.2 Fungsi QR Code**

*QR Code* berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan [*URL*](https://id.wikipedia.org/wiki/URL), nomor telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, iklan, pada tanda-tanda bus, kartu nama ataupun media lainnya. Atau dengan kata lain sebagai penghubung secara cepat konten [daring](https://id.wikipedia.org/wiki/Daring) dan konten [luring](https://id.wikipedia.org/wiki/Luring). Kehadiran kode ini memungkinkan audiens berinteraksi dengan media yang ditempelinya melalui [ponsel](https://id.wikipedia.org/wiki/Ponsel) secara efektif dan efisien. Pengguna juga dapat menghasilkan dan mencetak sendiri *QR Code* untuk orang lain dengan mengunjungi salah satu dari beberapa ensiklopedia *QR Code*.

**2.3.3 Kelebihan *QR Code***

QR Code memiliki kapasitas tinggi dalam data pengkodean, yaitu mampu menyimpan semua jenis data , seperti data [*numerik*](https://id.wikipedia.org/w/index.php?title=Numerik&action=edit&redlink=1) , data [*alphabetis*](https://id.wikipedia.org/w/index.php?title=Alphabetis&action=edit&redlink=1) , [*kanji*](https://id.wikipedia.org/wiki/Kanji) , [*kana*](https://id.wikipedia.org/wiki/Kana) , [hiragana](https://id.wikipedia.org/wiki/Hiragana) , [simbol](https://id.wikipedia.org/wiki/Simbol) , dan [kode biner](https://id.wikipedia.org/w/index.php?title=Kode_biner&action=edit&redlink=1) . Secara spesifik , *QR Code* mampu menyimpan data jenis numerik sampai dengan 7.089 karakter, data *alphanumerik* sampai dengan 4.296 karakter, kode binari sampai dengan 2.844 *byte*, dan huruf *kanji* sampai dengan 1.817 karakter. Selain itu *QR Code* memiliki tampilan yang lebih kecil daripada kode batang. Hal ini dikarenakan *QR Code* mampu menampung data secara horizontal dan vertikal, oleh karena itu secara otomatis ukuran dari tampilannya gambar *QR Code* bisa hanya seperspuluh dari ukuran sebuah kode batang. Tidak hanya itu *QR Code* juga tahan terhadap kerusakan, sebab *QR Code* mampu memperbaiki kesalahan sampai dengan 30%. Oleh karena itu, walaupun sebagian simbol *QR Code* kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun sepanjang 360 derajat.

**2.4 Pengertian *Semantic UI***

[*Semantic UI*](http://semantic-ui.com) yang memiliki tagline "*User Interface is the language of the web*" merupakan sebuah *Front-end Framework* (Kerangka Kerja untuk membuat tampilan situs), yang dibuat oleh Tim *semantic*, *gratis*, *open source* dan memiliki [Lisensi MIT](http://semantic-ui.mit-license.org/).

**2.5 Perangkat Permodelan Sistem**

**2.5.1 Flowcart**

**2.5.1.1 Pengertian Flowcart**

Menurut Krismiaji (2010:71) dalam bukunya yang berjudul Sistem Informasi Akuntansi, menjelaskan bahwa bagan alir (*flowchart*) merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis. Bagan alir (*flowchart*) menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem.

simbol dari bagan alir ( *flowchart* ) adalah sebagai berikut ini :

Tabel 2.1Daftar Simbol *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Pengertian | Keterangan |
| 1. |  | Mulai / berakhir ( Terminal ) | Digunakan untuk memulai, mengakhiri, atau titik henti dalam sebuah proses atau program; juga digunakan untuk menunjukkan pihak eksternal. |
| 2. |  | Dokumen | Sebuah dokumen atau laporan; dokumen dapat dibuat dengan tangan atau dicetak oleh komputer. |
| 4. | T | Arsip | Arsip dokumen disimpan dan diambil secara manual. Huruf didalamnya menunjukkan cara pengurutan arsip: N = Urut Nomor; A = Urut Abjad; T = Urut Tanggal. |
| 5. |  | Input / Output; Jurnal / Buku Besar | Digunakan untuk menggambarkan berbagai media input dan output dalam sebuah bagan alir program. |
| 6. |  | Disk | Data disimpan secara permanen pada disk |
| 7. |  | Penghubung Pada Halaman Berbeda | Menghubungkan bagan alir yang berada dihalaman yang berbeda. |
| 8. |  | Pemasukan Data On Line | Entri data alat oleh on line seperti terminal CRT dan komputer pribadi. |
| 9. |  | Pemrosesan Komputer | Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi |
| 10. |  | Arus Dokumen atau Pemrosesan | Arus dokumen atau pemrosesan; arus normal adalah ke kanan atau ke bawah. |
| 11. |  | Keputusan | Sebuah tahap pembuatan keputusan |
| 12. |  | Penghubung Dalam Sebuah Halaman | Menghubungkan bagan alir yang berada pada halaman yang sama. |
| 13. |  | Dokumen Tunggal | Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas. |
| 14. |  | Dokumen Rangkap | Digambarkan dengan menupuk simbol dokumen dan pencetakan nomor dokumen dibagian depan dokumen pada bagian kiri atas. |
| 15. |  | File/ Disk | Simbol yang menyatakan input yang berasal dari file/disk atau disimpan ke disk. |
| 16. |  | Display | Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, print dan sebagainya. |

#### 2.5.1.2 Jenis-jenis Flowchart

Menurut Jogiyanto terdapat 5 macam bagan alir, yaitu:

1. Bagan Alir Sistem ( *System Flowchart* )

Merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem.

1. Bagan Alir Dokumen *( Document Flowchart )*

Bagan alir dokumen atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

1. Bagan Alir Skematik ( *Schematic Flowchart* )

Merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur didalam sistem. Perbedaanya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang pahan dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarnya.

1. Bagan Alir Program ( *Program Flowchart* )

Merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dapat terdiri dari dua macam, yaitu Bagan Alir Logika Program ( *Program Logic Flowchart* ) dan Bagan Alir Program Komputer terinci (*Detailed Computer Program Flowchart*).

1. Bagan Alir Proses ( *Process Flowchart* )

Merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses selain dapat menunjukkan kegiatan dan simpanan yang digunakan dalam suatu prosedur, dapat juga menunjukkan jarak kegiatan yang satu dengan yang lainnya serta waktu yang diperlukan oleh suatu kegiatan.

## 2.5.2 Diagram Konteks

Diagram konteks merupakan sebuah diagram yang terdiri darisatu proses yang mendiskripsikan interaksi langsung antara sistem yang dikaji dengan beberapa *entity* yang berada diluar sistem. Diagram konteks mencakup masukan-masukan dasar, sistem umum dan keluaran.

## *Data Flow Diagram* (DFD)

Data Flow Diagram digunakan untuk menggambarkan bagaimana data ditransformasikan pada perangkat lunak, serta fungsi-fungsi yang mentranformasikan data. Data Flow Diagram ini pertama kali diperkenalkan oleh Martin dan Extrim yang memperkenalkan suatu algoritma program dengan menggunakan simbol lingkaran dan panah untuk mewakili arus data (*Data Flow*).

Menurut buku karangan *Krismiaji* simbol dari *data flow diagram* adalah sebagai berikut ini :

**Tabel 2.2** Daftar Simbol *Data Flow Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| Data Store | Untuk menggambarkan data flow yang sudah disimpan atau diarsipkan. Atau dengan kata lain mempresentasikan sebuah penyimpanan data. |
| Entitas Eksternal | Simbol ini digunakan untuk menggambarkan asal data atau tujuan data. Dapat berupa orang, organisasi, sumber informasi lain atau penerima akhir daei suatu laporan |
|  | Digunakan untuk proses pengelolaan atau transformasi data |
| Aliran Data | Simbol ini digunakan untuk menggambarkan aliran data yang berjalan. |

**(sumber :**[www.ilkom.unsri.ac.id](http://www.ilkom.unsri.ac.id)**, 2010)**

Adapun simbol-simbol yang digunakan untuk membuat DFD adalah sebagai berikut:

1. Kesatuan Luar

Suatu sistem mempunyai batasan sistem (Boundaries) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya.

1. Arus Data

Arus data ditampilkan dengan tanda panah. Arus data menggambarkan transfer data antara penyimpanan data dan proses. Terjadinya arus data harus berisi data, dan semua arus data mungkin akan mengawali proses ataupun merupakan hasil dari proses tersebut.

1. Proses Data

Proses digambarkan dengan lingkaran (Yuordon dan De Marco) yang disebut gelombang atau transform dan digambarkan dengan persegi panjang yang mempunyai pojok-pojok yang bulat. Proses mentransformasi masukkan menjadi keluaran. Nama proses terdiri atas kata kerja dan objek. Nama proses yang dipilih tergantung pada tingkat rincian yang dibutuhkan pada DFD.

1. Simpanan Data

Simpanan data digunakan untuk tempat penyimpanan kumpulan data atau paket data.Simpanan data digambarkan dalam bentuk sejajar.

## 2.5.4 Process Specification

*Process Spesification* adalah deskripsi setiap fungsi yang disajikan pada DFD. (Roger S.Pressman. 2002:354). Tujuannya adalah mendefinisikan apa saja yang harus dilakukan dengan tujuan untuk mengubah *input* menjadi *output*. Tools yang dapat digunakan adalah *decision table, pre/post condition, flowchart, Nassi-shneiderman diagrams, structured English.* Dari tools-tools tersebut yang akan digunakan dalam penulisan tugas akhir ini adalah S*tructured English*.

## 2.5.5 Kamus Data

Definisi *Data Dictionary* atau Kamus Data menurut Al Bahra (2005:147) dalam buku yang berjudul *Konsep Sistem Basis Data dan Implementasinya* adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Selama penyusunan suatu sistem informasi, kamus data digunakan sebagai alat untuk:

1. Merancang input
2. Merancang Laporan-laporan *database*.

Kamus data digunakan untuk merancang input, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang terdapat pada *data flow diagram.*

Kamus data dirancang berdasarkan arus data - arus data yang berada pada Diagram Alir Data. Keterangan - keterangan lebih detail dapat dilihat pada Kamus Data.

Pada tahap analisa sistem kamus data dapat digunakan sebagai alat komunikasi antara sistem analis dengan pemakai sistem mengenai data yang mengalir pada sistem, data yang mengalir pada sistem meliputi:

1. Data yang masuk kedalam sistem
2. Informasi yang dibutuhkan oleh Pemakai Sistem

Kegunaan – kegunaan Kamus Data yang lain meliputi:

1. Untuk melakukan validasi terhadap Diagram Alir Data dalam hal kelengkapan dan keakuratan
2. Menyediakan suatu titik awal untuk mengembangkan tampilan Layar baik masukan dan keluaran dan Laporan-laporan yang diperlukan
3. Menentukan bentuk muatan data yang disimpan dalam file-file atau table –table.
4. Mengembangkan logika untuk proses-proses Diagram Alir Data

Notasi yang digunakan dalam kamus data adalah sebagai berikut:

**Tabel 2.3** Notasi Kamus Data

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Uraian** |
| 1 | = | Terdiri dari, mendefinisikan, diuraikan menjadi, artinya |
| 2 | + | Dan |
| 3 | () | Opsional (boleh ya atau boleh tidak) |
| 4 | {} | Pengulangan |
| 5 | [] | Memilih salah satu dari sejumlah alternatif, seleksi |
| 6 | \*\* | Komentar |
| 7 | @ | Indetifikasi atribut kunci |
| 8 | | | Perintah sejumlah alternatif pilih antara symbol [] |

**(sumber:** [www.ilkom.unsri.ac.id](http://www.ilkom.unsri.ac.id)**, 2010)**

### 2.5.6 *Entity Relational Diagram (ERD)*

*Entity Relational Diagram (ERD)* adalah Suatu penyajian data dengan menggunakan entitas dan hubungannya*. ERD* dapat mengambarkan hubungan antar entity dengan jelas dan mengambarkan batasan jumlah entity dan partisipasi antar entity sehingga mudah dimengerti oleh pemakai komponen-komponen *ERD*, yaitu:

1. Entitas *(Entity)*

Entitas adalah sesuatu yang dapat dibedakan dalam dunia nyata, dimana informasi yang berkaitan dengannya dikumpulkan. Simbol untuk *entity* adalah persegi panjang.

2. Hubungan *(Relationship)*

*Relationship* adalah asosiasi yang terjadi antara satu atau lebih entity. Hubungan digambarkan dengan bentuk belah ketupat. Tiap belah ketupat diberi label kata kerja.

*3. Attribute*

*Attribute* adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang entity atau relationship tersebut. Simbol *attribute* adalah bentuk oval atau elips.

4. *Cardinality Ratio*

*Cardinality Ratio* adalah batasan yang menjelaskan jumlah keterhubungan satu *entity* dengan *entity* lainnya. Jenis *Cardinality Ratio*antara lain:

a. *One to one relationship*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu, atau dapat pula sebaliknya *file* kedua dengan *file* pertama adalah satu berbanding satu.

b. *One to many relationship*

Hubungan antara *file* pertama dengan *file* kedua adalah berbanding banyakatau dapat pula dibalik banyak lawan satu.

c. *Many to many relationship*

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak atau dapat pula di balik *file* kedua dengan *file* pertama banyak berbanding banyak. Notasi-notasi dalam *Entity Relational Diagram (ERD)* adalah sebagai berikut:

Tabel 2.4 Notasi *Entity Relational Diagram (ERD)*

|  |  |  |
| --- | --- | --- |
| **No** | **Simbol** | **Keterangan** |
| 1 |  | Himpunan entitas E |
| 2 |  | Atribut a sebagai *key* |
| 3 |  | Himpunan relasi R |
| 4 |  | Link |

# Perancangan Basis Data

## 2.6.1 Pengertian Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan.Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.

Dan menurut Tata Sutabri (2005: 21) data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi.

Menurut Inmon (2005, p493), data adalah kumpulan dari fakta, konsep, atau instruksi pada penyimpanan yang digunakan untuk komunikasi, perbaikan dan diproses secara otomatis yang mempresentasikan informasi yang dapat di mengerti oleh manusia.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar tetapi belum diolah.

## 2.6.2 Pengertian Basis Data

Menurut Elmasri R (1994) data yang disimpan dalam komputer dan didefinisikan sebagai basis data adalah:

1. Basis data merupakan penyajian suatu aspek dari dunia nyata real (World atau mini world) misal basis data perbankan, perpustakaan, pertanahan perpajakan dan model basis data lainnya.
2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implicit. Sehingga data yang terkumpul secara acak dan tanpa arti tidak bisa disebut basis data.
3. Basis Data dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

Menurut Inmon (2005, p493), basis data adalah sekumpulan data yang saling berhubungan dan disimpan berdasarkan suatu skema.

Menurut Connolly dan Begg (2010, p65), basis data adalah kumpulan berbagai data logika terkait dan deskripsi, yang dirancang untuk memenuhi kebutuhan informasi organisasi.

Menurut O’Brien (2010, p173), basis data adalah kumpulan elemen data yang terintegrasi yang berhubungan secara logikal.

Berdasarkan teori para ahli diatas dapat disimpulkan bahwa, basis data adalah kumpulan data yang berhubungan secara logikal dan disimpan berdasarkan suatu skema untuk memperoleh informasi yang dibutuhkan oleh organisasi.

## Normalisasi

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data diantara relasi dalan perancangan logikal basis data. Format normalisasi terdiri dari 5 bentuk, yaitu:

1. **Form Normal Pertama (1NF)**

Suatu tabel dikatakan sudah 1NF jika telah memenuhi ketentuan sebagai berikut:

• Tidak ada atribut mempunyai nilai berulang atau nilai *array*

• Tidak mempunyai baris yang rangkap

1. **Form Normal Kedua (2NF)**

Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

• Key terdiri dari atribut tunggal

• Setiap atribut nonkey ketergantungan fungsional pada semua *key* atau tidak terjadinya ketergantungan pada *key* *composite*.

1. **Form Normal Ketiga (3NF)**

Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada *ketergantungan transitif* diantara atribut.

1. **Form Normal Keempat (4NF)**

Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivalue dependency*).

1. **Form Normal Kelima (5NF)**

Konsep pada bentuk ini adalah ketergantungan pada gabunganbeberapa atribut (*join dependency*).

* 1. **Teknologi Pendukung**

**2.7.1 CodeIgniter (CI)**

CodeIgniter (CI) adalah web application framework yang bersifat open source yang digunakan untuk membangun aplikasi web. Tujuan utama penegmbangan CI adalah untuk membantu developer untuk membangun aplikasi lebih cepat karena tidak menulis code dari awal. CI di pernalkan pada 28 Februari 2006 dengan mengunakan konsep Model-View-Controller development pattern (MVC). CI merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. CodeIgniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum CodeIgniter. Selain itu CodeIgniter juga memiliki fitur-fitur lainya yang sangat bermanfaat, antara lain:

* **Menggunakan *Pattern* *MVC*.** Dengan menggunakan *pattern* *MVC* ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.
* **URL *Friendly*.** *URL* yang dihasilkan sangat *url* *friendly*. Pada CodeIgniter diminimalisasi penggunaan ***$\_GET*** dan di gantikan dengan *URI*.
* **Kemudahan.** Kemudahan dalam mempelajari, membuat *library* dan *helper*, memodifikasi serta meng-integrasikan *Library* dan *Helper*.

Jika kita membandingkan antara CodeIgniter dengan framework-framework lainnya maka beberapa poin yang membuat CodeIgniter unggul adalah:

* **Kecepatan.** Berdasarkan hasil *benchmark* CodeIgniter merupakan salah satu *framework* PHP tercepat yang ada saat ini.
* **Mudah dimodifikasi dan beradaptasi.** Sangat mudah memodifikasi *behavior framework* ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi *library* lainya.
* **Dokumentasi lengkap dan jelas.** Bahkan tanpa tulisan ini pun CodeIgniter sebenarnya telah menyediakan sebuah panduan yang lengkap mengenai CodeIgniter. Semua informasi yang anda butuhkan tentang codeigniter ada disana.
* ***Learning Curve* Rendah.** CodeIgniter sangat mudah dipelajari. Dalam pemilihan *framework* hal ini sangat penting diperhatikan karena kita juga harus memperhatikan *skill* dari seluruh anggota *team*. Jika sebuah *framework* sangat sulit dipelajari maka akan beresiko untuk memperlambat *team development* anda.

**2.7.2 *Database MySql***

*MySQL* merupakan software sistem manajemen database. *MySQL* ini merupakan database yang paling populer digunakan untuk membangun suatu aplikasi yang menggunakan database sebagai sumber dan pengolahan datanya. *MySQL* mudah untuk digunakan, cepat secara *query* , dan mencukupi untuk kebutuhan *database* pada perusahaan-perusahaan yang berskala menengah maupun besar.

* + 1. ***XAMPP***

*XAMPP* ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program *MySQL* *database*, *Apache* *HTTP* *Server*, dan penerjemah ditulis dalam bahasa pemrograman *PHP* dan *Perl*.

Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan.Dulu *XAMPP* untuk Linux dinamakan *LAMPP*, sekarang diganti namanya menjadi *XAMPP* *FOR* *LINUX*.

XAMPP adalah kepanjangan dari :

X : *multi platform*, bisa *Windows*, *Linux*, *Mac* *OS*, maupun *Solaris*

A : *Apache HTTP Server*

M : *MySQL Database Server*

P : *PHP Scripting Language*

P : *Perl Scripting Language*