

SKRIPSI

**KONVERSI DIAGRAM DRAW.IO MENJADI LARAVEL
MIGRATION**



Daniel Alexander Kefas

NPM: 6181801001

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022**

UNDERGRADUATE THESIS

CONVERT DRAW.IO DIAGRAM TO LARAVEL MIGRATION



Daniel Alexander Kefas

NPM: 6181801001

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

LEMBAR PENGESAHAN

**KONVERSI DIAGRAM DRAW.IO MENJADI LARAVEL
MIGRATION**

Daniel Alexander Kefas

NPM: 6181801001

Bandung, 23 Juni 2022

Menyetujui,

Pembimbing

**Digitally signed
by Raymond
Chandra Putra**

Raymond Chandra Putra, M.T.

Ketua Tim Penguji

**Digitally signed
by Chandra
Wijaya**

Chandra Wijaya, M.T.

Anggota Tim Penguji

**Digitally signed
by Vania Natali**

Vania Natali, M.T.

Mengetahui,

Ketua Program Studi

**Digitally signed
by Mariskha Tri
Adithia**

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

KONVERSI DIAGRAM DRAW.IO MENJADI LARAVEL MIGRATION

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 23 Juni 2022



Daniel Alexander Kefas
NPM: 6181801001

ABSTRAK

Dalam membuat perangkat lunak yang fungsional, peran basis data dibutuhkan sebagai tempat untuk menyimpan data. Untuk dapat menentukan isi untuk setiap kumpulan data atau tabel diperlukan rancangan basis data. Salah satu model yang dapat digunakan untuk merancang basis data adalah *Entity Relationship Diagram*. *Entity Relationship Diagram*, atau ERD adalah sebuah model berbasis objek yang menyatakan hubungan antara entitas dengan lainnya. Untuk membuat ERD, pengembang dapat menggunakan perangkat lunak draw.io. Draw.io adalah sebuah perangkat lunak *open-source* yang digunakan untuk membuat berbagai diagram. Dengan menggunakan draw.io, pengguna dapat menyimpan diagram berbagai format data, salah satunya adalah dalam bentuk format *Extended Markup Language*, atau XML. Setelah ERD dibuat, pemetaan dalam basis data dapat dilakukan dengan menulis kueri SQL. Terdapat cara alternatif apabila pembangunan perangkat lunak menggunakan *framework* misalnya *Laravel*. Dalam *Laravel*, terdapat sebuah fitur yaitu *Laravel Migration* yang memungkinkan untuk memindahkan rancangan ERD dalam basis data tanpa kueri SQL.

Dalam penelitian ini, dibangun sebuah perangkat lunak yang mampu melakukan konversi diagram draw.io menjadi kode *Laravel Migration*. Perangkat lunak dapat melakukan konversi dengan memanfaatkan format XML yang didukung oleh draw.io dengan melakukan proses *parsing* dan membangun model yang kemudian dituliskan dalam kode *Migration*. Perangkat lunak ini dibangun dengan bahasa pemrograman PHP dan berbentuk *Laravel Package*.

Dilakukan pengujian untuk menguji kebenaran dalam perangkat lunak yang dibangun. Pengujian yang dilakukan meliputi pengujian fungsional yang melibatkan beberapa kasus sederhana yang melibatkan satu atau dua entitas dan pengujian eksperimental yang melibatkan lima kasus dengan ERD yang lengkap. Pengujian dilakukan dengan dua notasi yang didukung perangkat lunak yaitu notasi Chen dan notasi Crow. Berdasarkan hasil pengujian, perangkat lunak mampu melakukan konversi dari *file* XML menjadi kode *Laravel Migration* dengan baik. *File Migration* yang dihasilkan memiliki entitas, atribut, serta relasi yang sesuai dengan kasus uji yang diberikan.

Kata-kata kunci: *Entity Relationship Diagram*, *Laravel Migration*, Konversi, *Extended Markup Language*, *Laravel Package*, draw.io

ABSTRACT

To make a functional software, the role of the database is needed as a place to store data. To be able to determine the contents of each data set or table, a database design is required. One of the models that can be used to design a database is the Entity Relationship Diagram. Entity Relationship Diagram or ERD is an object-based model that states the relationship between an entity and the other entities. To create an ERD, developers can use a software called draw.io. Draw.io is an open-source software to create various diagrams. By using draw.io, users can save diagrams in various data formats, one of which is in the form of Extended Markup Language, or XML. After the ERD is created, the mapping in the database can be done by writing SQL queries. There is an alternative way if software development uses a framework such as Laravel. In Laravel, there is a feature called Laravel Migration which makes it possible to write ERD designs in the database without SQL queries.

In this research, we developed a software that can convert draw.io diagrams into Laravel Migration. The software can perform conversions using the XML format supported by draw.io by parsing and building the suitable models written in Migration code. This software was built using PHP language, and using Laravel Package as the main structure.

Several tests were done to ensure the developed software converted the draw.io diagram into Laravel Migration correctly. The test includes functional testing involving several simple cases that contain one or two entities and experimental testing involving five cases that contains a complete ERD. The test was done using two notations, Chen notation and Crow notation. Based on the test result, the developed software is able to convert from draw.io's XML to Laravel Migration correctly. The Migration files that are produced have entities, attributes, and relations that match the given test cases.

Keywords: Entity Relationship Diagram, Laravel Migration, Convert, Extended Markup Language, Laravel Package, draw.io

Dipersembahkan kepada Tuhan Yang Maha Esa, keluarga, para dosen, teman-teman, serta diri sendiri.

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus untuk pimpinan, rahmat dan kasih setia-Nya, penulis dapat menyelesaikan penelitian dan penulisan dokumen skripsi yang berjudul “Konversi Diagram draw.io menjadi Laravel Migration”. Skripsi ini dibuat untuk memberikan pengetahuan kepada pembaca mengenai proses konversi diagram menjadi potongan kode program *Laravel Migration*. Selain itu, skripsi ini juga dibuat untuk menerapkan ilmu yang sudah dipelajari menjadi bukti nyata untuk memudahkan proses pengembangan sebuah sistem informasi. Dibalik penulisan skripsi ini, penulis sadar bahwa penulisan skripsi ini tidak mungkin dilakukan tanpa dukungan orang-orang terdekat. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Keluarga yang selalu mendukung dan membantu penulis selama proses perkuliahan.
2. Bapak Raymond Chandra Putra, S.T., M.T. sebagai dosen pembimbing yang telah memimbing dan mendukung penulis dalam proses penyusunan dokumen skripsi.
3. Bapak Chandra Wijaya S.T., M.T. dan Ibu Vania Natali S.Kom, M.T. selaku penguji yang telah memberikan kritik dan saran yang membangun.
4. Ibu Mariskha Tri Adithia, S.Si, M.Sc, PDEng. selaku koordinator mata kuliah Skripsi dan kepala program studi Informatika.
5. Bapak Kristopher David Harjono, M.T. dan Ibu Natalia S.Si, M.Si. selaku dosen wali selama proses perkuliahan
6. Kerabat dekat penulis Geraldi Akira Surya, Lucyus M. Ardivan, Deddy Chandra, Wiranata Lim, Stevent Kent dan kerabat lainnya yang tidak dapat disebutkan, karena telah mendukung penulis selama proses penyusunan dokumen skripsi.
7. Seluruh angkatan Informatika 2018 sebagai teman-teman seperjuangan dalam menempuh perkuliahan
8. Serta seluruh pihak yang tidak dapat disebutkan yang telah membantu dan mendukung penulis selama proses perkuliahan dan penyusunan dokumen skripsi ini.

Akhir kata, penulis memohon maaf apabila terdapat kesalahan penulisan atau kekurangan pada penelitian ini dan menerima kritik dan saran untuk menyempurnakan penelitian ini. Penulis juga berharap agar skripsi ini dapat menjadi berkat dan bermanfaat bagi setiap pembaca yang hendak melakukan penelitian dan pengembangan lebih lanjut dan berbagai pihak yang berkepentingan.

Bandung, Juni 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxiii
DAFTAR KODE PROGRAM	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Draw.io	5
2.2 Entity Relationship Diagram (ERD)	6
2.2.1 ERD Notasi Chen	9
2.2.2 ERD Notasi Crow	12
2.3 Extended Markup Language	14
2.4 JavaScript Object Notation	14
2.5 Database-Management System (DBMS)	15
2.6 Laravel Framework	16
2.6.1 Fitur dalam Laravel	16
2.6.2 Laravel Migration	17
2.6.3 Struktur Aplikasi Laravel	18
2.6.4 Laravel Package	21
3 ANALISIS	23
3.1 Analisis Masalah	23
3.2 Analisis Pemilihan Notasi	24
3.2.1 Notasi Chen	24
3.2.2 Notasi Crow	24
3.3 Analisis XML draw.io	24
3.3.1 Bentuk Diagram	25
3.3.2 Entitas	29
3.3.3 Atribut	30
3.3.4 Relasi dan Garis	31
3.4 Analisis Proses Konversi	34

3.5	Analisis Spesifikasi Kebutuhan	34
3.6	Analisis Perangkat Lunak	35
3.6.1	Diagram <i>Use Case</i>	35
3.6.2	<i>Use Case Scenario</i>	35
3.6.3	Diagram Aktivitas	36
3.6.4	Diagram Kelas Sederhana	37
3.6.5	Analisis Masukan Perangkat Lunak	38
3.6.6	Analisis Keluaran Perangkat Lunak	40
4	PERANCANGAN	43
4.1	Diagram Kelas	43
4.1.1	Kelas <code>DrawverterServiceProvider</code>	44
4.1.2	Kelas <code>DrawverterInstallPackage</code>	44
4.1.3	Kelas <code>DrawverterProcessERD</code>	44
4.1.4	Kelas <code>Drawverter</code>	44
4.1.5	Kelas <code>Convert</code>	44
4.1.6	Kelas <code>ChenConvert</code> dan <code>CrowConvert</code>	45
4.1.7	Kelas <code>Entity</code>	45
4.1.8	Kelas <code>MigrationWriter</code>	46
4.2	Diagram Sekuens	50
4.2.1	Proses Inisiasi	50
4.2.2	Proses Konversi	51
5	IMPLEMENTASI DAN PENGUJIAN	53
5.1	Implementasi Perangkat Lunak	53
5.1.1	Lingkungan Pengembangan	53
5.1.2	Hasil Implementasi	53
5.1.3	<i>Deployment</i>	55
5.2	Pengujian	55
5.2.1	Pengujian Fungsional	55
5.2.2	Pengujian Eksperimental	60
5.2.3	Analisis Pengujian	62
6	KESIMPULAN	67
6.1	Kesimpulan	67
6.2	Saran	67
	DAFTAR REFERENSI	69
	A KODE PROGRAM	71
	B HASIL PENGUJIAN	83
B.1	Kasus ERD SI Rumah Sakit	83
B.1.1	Notasi Chen	83
B.1.2	Notasi Crow	87
B.2	Kasus ERD SI POS TeaTime	90
B.2.1	Notasi Chen	90
B.2.2	Notasi Crow	93
B.3	Kasus ERD SI Forum HiTani	95
B.3.1	Notasi Chen	95
B.3.2	Notasi Crow	99
B.4	Kasus ERD SI Matakuliah	101
B.4.1	Notasi Chen	101

B.4.2	Notasi Crow	105
B.5	Kasus ERD SI Event Organizer	107
B.5.1	Notasi Chen	107
B.5.2	Notasi Crow	112

DAFTAR GAMBAR

1.1	Contoh ERD sederhana	1
1.2	Tampilan antarmuka draw.io	2
2.1	Tampilan editor draw.io ¹	6
2.3	Contoh notasi IDE1FX	7
2.2	Contoh notasi barker ²	7
2.4	Contoh notasi panah ²	8
2.5	Contoh notasi UML ²	8
2.6	Contoh entitas dengan atribut	10
2.7	Entitas dalam notasi Crow ⁴	11
2.8	Bagian dalam indikator relasi	13
2.9	Struktur aplikasi <i>Laravel v8</i>	19
3.1	Alur proses konversi	24
3.2	Koleksi <i>Entity Relation</i> dalam <i>shape panel</i>	26
3.3	Entitas yang dihasilkan dari Kode 3.4	31
3.4	Contoh Entitas dari Kode 3.5	32
3.5	Contoh relasi dari Kode 3.6	32
3.6	Contoh relasi notasi Crow dari Kode 3.7	33
3.7	Diagram <i>use case</i>	36
3.8	Diagram Aktivitas	37
3.9	Diagram kelas sederhana	38
3.10	Dialog pilihan export draw.io	39
3.11	Contoh hubungan garis yang benar dan salah	40
4.1	Diagram Kelas	43
4.2	Diagram sekuens proses instalasi	50
4.3	Diagram sekuens proses konversi	51
5.1	Penggunaan perintah <code>drawverter:install</code> berhasil	53
5.2	Penggunaan perintah <code>dravterter:install</code> tidak berhasil	54
5.3	Dialog konfirmasi pada perintah <code>drawverter:start</code>	54
5.4	Pesan tidak ditemukan dari perintah <code>drawverter:start</code>	54
5.5	Pesan gagal dari perintah <code>drawverter:install</code>	55
5.6	Pesan berhasil dari perintah <code>drawverter:install</code>	55
5.7	Pengujian relasi satu-ke-satu	56
5.8	Hasil pengujian relasi satu-ke-satu	57
5.9	Pengujian relasi satu-ke-banyak	57
5.10	Hasil pengujian relasi satu-ke-banyak	58
5.11	Pengujian relasi banyak-ke-banyak	58
5.12	Hasil pengujian relasi banyak-ke-banyak	59
5.13	Hasil pengujian relasi banyak-ke-banyak dengan atribut	59
5.14	Pengujian entitas dengan atribut <i>multivalued</i>	60

5.15 Hasil pengujian entitas dan atribut <i>multivalued</i>	61
5.16 Hasil pengujian <i>multipage</i>	61
5.17 Hasil pengujian eksperimental SI Rumah Sakit	62
5.18 Pengujian eksperimental SI Rumah Sakit	63
5.19 Relasi yang keliru dalam kasus SI Event Organizer	64
5.20 Relasi pada entitas <i>History</i>	65
B.1 ERD SI Rumah Sakit dengan notasi chen	84
B.2 ERD SI Rumah Sakit dengan notasi crow	87
B.3 ERD SI POS TeaTime dengan notasi chen	91
B.4 ERD SI POS TeaTime dengan notasi crow	94
B.5 ERD SI Forum HiTani dengan notasi chen	96
B.6 ERD SI Forum HiTani dengan notasi crow	99
B.7 ERD SI Matakuliah dengan notasi chen	102
B.8 ERD SI Matakuliah dengan notasi chen	105
B.9 ERD SI Event Organizer dengan notasi chen	108
B.10 ERD SI Event Organizer dengan notasi crow	112

DAFTAR TABEL

2.1	Tabel Jenis Entitas	9
2.2	Tabel Jenis Atribut	10
2.3	Tabel Jenis Relasi	11
2.4	Tabel Jenis Opsionalitas Relasi	11
2.5	Tabel Jenis Kardinalitas	12
2.6	Tabel Jenis Kardinalitas	13
2.7	Tabel indikator relasi notasi Crow	13
3.1	<i>Shape Entity Relation</i> yang dapat digunakan untuk notasi Chen	26
3.2	<i>Shape Entity Relation</i> yang dapat digunakan untuk notasi Crow	28
3.3	Tabel atribut style pada entitas	30
3.4	Tabel atribut style pada atribut dalam notasi Chen	30
5.1	Tabel hasil pengujian eksperimental	62
5.2	Tabel hasil pengujian fungsional	64

DAFTAR KODE PROGRAM

2.1	Contoh file XML	14
2.2	Penulisan Atribut Dokumen XML	14
2.3	Contoh Dokumen JSON	15
2.4	Contoh file <i>Laravel Migration</i>	17
2.5	Kode SQL yang setara dengan <i>method up()</i>	17
2.6	Kode SQL yang setara dengan <i>method down()</i>	18
3.1	Struktur file XML draw.io	23
3.2	Kerangka XML draw.io	24
3.3	Kerangka mxCell yang memiliki mxGeometry	25
3.4	Contoh entitas dengan atribut pada notasi Crow	31
3.5	Kode entitas dan atribut pada notasi Chen	31
3.6	Kode relasi pada notasi Chen	32
3.7	Kode relasi dalam notasi Crow	33
3.8	Contoh file xml draw.io yang terenkripsi	39
3.9	Contoh file xml draw.io yang sesuai	39
3.10	Contoh entitas dalam file <i>Migration</i>	41
3.11	Contoh relasi dalam file <i>Migration</i>	42
4.1	Contoh struktur keluaran notasi chen	45
4.2	Contoh struktur keluaran notasi crow	46
A.1	Drawverter.php	71
A.2	Convert.php	71
A.3	ChenConvert.php	72
A.4	CrowConvert.php	73
A.5	MigrationWriter.php	74
A.6	DrawverterInstallPackage.php	81
A.7	DrawverterProcessERD.php	81
A.8	DrawverterServiceProvider.php	81
A.9	Entity.php	82
B.1	Hasil pengujian SI Rumah Sakit dengan notasi chen	83
B.2	Hasil pengujian SI Rumah Sakit dengan notasi crow	87
B.3	Hasil pengujian SI POS TeaTime dengan notasi chen	90
B.4	Hasil pengujian SI POS TeaTime dengan notasi crow	93
B.5	Hasil pengujian SI Forum HiTani dengan notasi chen	95
B.6	Hasil pengujian SI Forum HiTani dengan notasi crow	99
B.7	Hasil pengujian SI Matakuliah dengan notasi chen	101
B.8	Hasil pengujian SI Matakuliah dengan notasi crow	105
B.9	Hasil pengujian SI Event Organizer dengan notasi chen	107
B.10	Hasil pengujian SI Event Organizer dengan notasi crow	113

BAB 1

PENDAHULUAN

Bab ini berisi latar belakang dibuatnya perangkat lunak untuk konversi diagram draw.io menjadi *Laravel Migration*, rumusan masalah untuk penelitian ini, tujuan penelitian, batasan masalah, metodologi penelitian dan sistematika pembahasan dari dokumen ini.

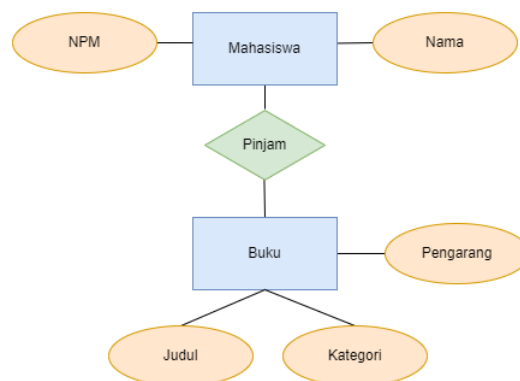
1.1 Latar Belakang

Dalam membuat sebuah perangkat lunak yang fungsional, peran basis data sangat dibutuhkan sebagai media penyimpanan data. Setiap data tersebut disimpan dalam bentuk tabel relasional. Saat membuat sistem informasi, pada umumnya diawali dengan membuat rancangan basis data. Perancangan basis data yang dimaksud adalah menentukan isi untuk setiap kumpulan data atau tabel yang dibutuhkan oleh sistem yang dibuat.

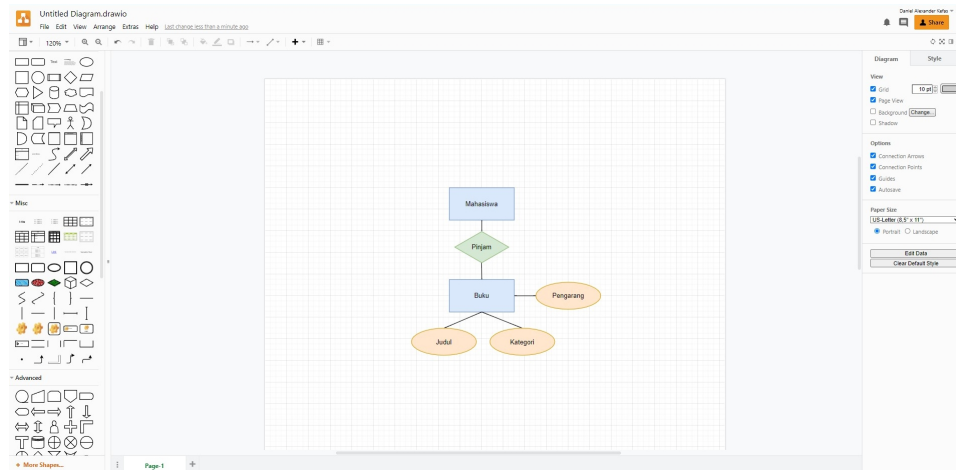
Salah satu model yang dapat digunakan untuk merancang basis data adalah *Entity Relationship Model* (ER-Model). ER-Model atau biasa dikenal dengan *Entity Relationship Diagram* (ERD) adalah sebuah model berbasis objek yang menyatakan hubungan antara entitas dengan entitas yang lainnya. ERD dikembangkan untuk memfasilitasi desain basis data dengan memuat spesifikasi yang mewakili keseluruhan struktur logis dari basis data. ERD menyatakan hubungan relasi dalam dunia nyata ke dalam model yang konseptual [1]. Dalam pembuatannya, ERD dikomposisi oleh beberapa simbol yang sudah ditetapkan penggunaannya sehingga pembuatannya memiliki standar yang sama. Sebagai contoh, pada notasi Chen simbol belah ketupat yang menyatakan relasi, simbol persegi panjang yang menyatakan entitas/objek, dan simbol elips yang menyatakan atribut dari entitas/objek. Gambar 1.1 merupakan contoh ERD sederhana dengan notasi chen.

Untuk membuat sebuah ERD, pengembang dapat menggunakan draw.io. Draw.io atau diagrams.net adalah sebuah perangkat lunak berbasis web *open source* yang berfungsi untuk membangun diagram aplikasi¹. Gambar 1.2 merupakan tampilan antarmuka dari draw.io. Perangkat lunak

¹<https://www.diagrams.net/about>, diakses pada 10 November 2021



Gambar 1.1: Contoh ERD sederhana



Gambar 1.2: Tampilan antarmuka draw.io

ini banyak digunakan karena perangkat lunak ini gratis dan tidak memerlukan proses instalasi. Draw.io dapat digunakan untuk membuat berbagai macam diagram, salah satunya yaitu ERD. Draw.io juga menyediakan berbagai macam *template* yang dapat digunakan. Selain itu, draw.io memberikan kebebasan untuk menyimpan diagram yang telah dibuat ke berbagai macam format *file* seperti JPEG/JPG, PNG, SVG, hingga XML. Dibalik setiap kemudahan yang diberikan, draw.io tidak lebih hanya sekedar tool untuk membuat gambar diagram sehingga tidak adanya pemeriksaan terhadap semantik dari diagram tersebut.

Setelah ERD selesai dibuat, untuk dapat digunakan pada perangkat lunak, ERD perlu diterjemahkan menjadi tabel dalam basis data melalui kode SQL. Pada basis data, entitas di ERD berubah menjadi sebuah tabel dan atribut yang terdapat pada entitas tersebut menjadi kolom dari tabel tersebut. Proses tersebut dibuat secara manual dalam bentuk kode SQL yang kemudian dijalankan di DBMS. Jika menggunakan *web framework* misalnya *Laravel*, terdapat cara alternatif untuk membuat tabel basis data yaitu dengan menggunakan fitur *Laravel Migration*.

Laravel adalah salah satu *web framework open source* untuk membangun aplikasi web yang menggunakan bahasa pemrograman PHP dan memakai arsitektur Model-View-Controller (MVC) [2, 3]. *Laravel* menyediakan berbagai kemudahan bagi pengguna karena mempunyai berbagai fitur. Salah satu fitur tersebut adalah *Laravel Migration* yang memiliki kemampuan membuat tabel basis data secara otomatis tanpa menggunakan kueri SQL. Untuk menggunakan fitur *Migration*, diperlukan sebuah perintah CLI yaitu Artisan yang merupakan *command* utama untuk memakai seluruh fitur yang dimiliki oleh *Laravel*. Sebagai contoh, *command* Artisan dapat dijalankan melalui terminal yang dijalankan pada direktori proyek *Laravel*. *Command* `php artisan make:migration <nama_entitas>` adalah perintah untuk membuat sebuah *file Migration* dengan format nama `yyyy_mm_dd_hhmmss_create_<nama_tabel>_table.php`. *File Migration* tersebut dapat dilengkapi dengan kolom untuk tabel dan nama tabel yang sesuai dengan ERD yang sudah ada. *File Migration* tersebut dapat dijalankan dengan menggunakan *command* Artisan yaitu `php artisan migrate` yang menyebabkan tabel yang sesuai dengan *file Migration* secara otomatis berada dalam DBMS. Pada pembuatan kode *Migration* diperlukan ketelitian karena perlu mencocokkan dengan rancangan ERD yang telah dibuat.

Untuk menyelesaikan masalah tersebut, diusulkan metode untuk membuat kode *Laravel Migration* secara otomatis dari diagram ERD. Hal ini dapat dimungkinkan karena draw.io dapat menyimpan ERD dalam bentuk XML (*Extensible Markup Language*). XML adalah sebuah bahasa markup yang digunakan untuk mengirim dan menyimpan data. Data yang disimpan dalam bentuk XML bersifat *universal*. Menurut Martin Bryan [4], XML menyediakan sintak formal untuk mendeskripsikan relasi antara entitas, elemen, dan atribut yang membuat XML dapat dibaca oleh komputer.

Pada penelitian ini, dibuat sebuah perangkat lunak yang dapat mengkonversi diagram draw.io dalam format XML menjadi *Laravel Migration* secara otomatis. Hal ini meliputi proses menerjemahkan ERD dengan membuat tabel sesuai entitas yang ada serta menyatakan relasi dengan tabel lain dalam bentuk kode PHP *Laravel Migration*. Perangkat lunak yang dibangun merupakan perangkat lunak berbasis web dan menggunakan *framework Laravel*. Perangkat lunak ini menerima masukan berupa *file XML* yang dikeluarkan oleh draw.io, kemudian memproses data dengan melakukan ekstraksi dengan teknik *parsing* yang sesuai untuk *file XML* dan memberikan keluaran berupa *file Laravel Migration* yang dapat dimasukkan ke dalam struktur *framework* serta dapat dijalankan tanpa *error*. Perangkat lunak dibuat dalam bentuk *Laravel Package* yang dibuat dengan bahasa PHP dan bantuan *framework Laravel*. *Laravel Package* adalah sekumpulan *source code* yang berfungsi untuk menambahkan fungsionalitas dari *framework Laravel*.

1.2 Rumusan Masalah

Berikut ini adalah beberapa rumusan masalah dari topik yang dikerjakan pada penelitian ini:

1. Bagaimana cara mengkonversi data dari *file XML* draw.io menjadi *file Laravel Migration*?
2. Bagaimana implementasi perangkat lunak yang mampu melakukan konversi diagram draw.io menjadi *Laravel Migration*?

1.3 Tujuan

Berikut adalah tujuan yang dicapai dalam proses penelitian ini:

1. Melakukan ekstraksi data terhadap *file XML* dengan teknik *parsing* dan membangun model yang sesuai kemudian mengubahnya menjadi *file Migration*.
2. Melakukan analisis dan perancangan terhadap perangkat lunak yang dibangun sesuai dengan model yang dihasilkan dan pemetaan ERD.
3. Membangun sebuah *Laravel Package* yang dapat melakukan konversi data dari *file XML* draw.io menjadi *Laravel Migration*.

1.4 Batasan Masalah

Berikut adalah beberapa batasan masalah yang ditetapkan dalam penelitian ini:

1. ERD yang dapat digunakan untuk proses konversi hanya ERD dengan notasi Chen dan notasi Crow.
2. Penggunaan *shape* untuk ERD dibatasi pada *shape-shape* yang sudah ditentukan dan tidak dilakukan pemeriksaan terhadap bentuk diagram yang lain.

1.5 Metodologi

Metodologi yang dilakukan pada penelitian adalah sebagai berikut:

1. Melakukan studi literatur terhadap ERD dan notasi yang digunakan, penggunaan aplikasi draw.io, XML, JSON, *framework Laravel* dan *Laravel Package*.
2. Membuat hasil analisis terhadap *file XML* draw.io berdasarkan notasi yang digunakan.
3. Membuat perancangan perangkat lunak yang dibuat berdasarkan hasil analisis.
4. Melakukan implementasi pembuatan perangkat lunak berdasarkan rancangan yang dibuat dalam bentuk *Laravel Package*.
5. Melakukan pengujian fungsional dan eksperimental terhadap perangkat lunak yang dibuat.
6. Menarik kesimpulan berdasarkan proses penelitian dan pengujian.
7. Melaporkan hasil penelitian dalam bentuk dokumen skripsi.

1.6 Sistematika Pembahasan

Pembahasan penelitian dilakukan secara sistematis dengan detail sebagai berikut:

1. Bab Pendahuluan

Bab 1 berisi latar belakang dibuatnya perangkat lunak untuk konversi diagram draw.io menjadi *Laravel Migration*, rumusan masalah, tujuan, batasan masalah, metodologi dan sistematika pembahasan.

2. Bab Dasar Teori

Bab 2 berisi teori yang digunakan dalam penelitian ini yaitu aplikasi draw.io, ERD, XML, JSON, dan *framework Laravel*.

3. Bab 3 Analisis

Bab 3 berisi pembahasan dari analisis masalah, analisi pemilihan notasi, analisis xml draw.io, analisis kebutuhan, dan analisis perangkat lunak.

4. Bab Perancangan

Bab 4 berisi pembahasan dari perancangan perangkat lunak yang mampu melakukan konversi diagram draw.io menjadi *Laravel Migration*. Dalam bab ini dibahas kelas diagram dan diagram urutan yang terkait dengan perangkat lunak yang dibangun

5. Bab Implementasi dan Pengujian

Bab 5 berisi implementasi dari perangkat lunak yang mampu melakukan konversi diagram draw.io menjadi *Laravel Migration*. Bab ini akan membahas implementasi perangkat lunak, pengujian, dan hasil dari pengujian yang dilakukan.

6. Bab Kesimpulan

Bab 6 berisi kesimpulan dari seluruh teori, implementasi, dan pengujian yang telah dilakukan serta saran untuk penelitian lebih lanjut.