

# Bab 19

## Daftar Referensi

### Sub-CPMK yang Dicakup dalam Bab Ini:

- **Sub-CPMK 1.1-6.2:** Menunjukkan kemampuan literasi informasi dan penggunaan referensi ilmiah dalam bidang teknik kompilasi

### 19.1 Buku Referensi Utama

#### 19.1.1 Compiler Design Fundamentals

- **Compilers: Principles, Techniques, and Tools** (2nd Edition)  
Aho, Alfred V., Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman  
Pearson/Addison-Wesley, 2007  
ISBN: 978-0321486813
- **Modern Compiler Implementation in C**  
Appel, Andrew W.  
Cambridge University Press, 1998  
ISBN: 978-0521583909
- **Engineering a Compiler** (2nd Edition)  
Cooper, Keith D. and Linda Torczon  
Morgan Kaufmann, 2011  
ISBN: 978-0120884788
- **Programming Language Pragmatics** (4th Edition)  
Scott, Michael L.  
Morgan Kaufmann, 2015  
ISBN: 978-0124104099

### 19.1.2 Advanced Compiler Topics

- **Advanced Compiler Design and Implementation**

Muchnick, Steven S.

Morgan Kaufmann, 1997

ISBN: 978-1558603204

- **Optimizing Compilers for Modern Architectures**

Allen, Randy and Ken Kennedy

Morgan Kaufmann, 2001

ISBN: 978-1558602863

- **The Dragon Book: Compilers**

Aho, Alfred V., Ravi Sethi, and Jeffrey D. Ullman

Addison-Wesley, 1986

ISBN: 978-0201100884

## 19.2 Buku Praktis dan Implementasi

### 19.2.1 Lexical Analysis and Parsing

- **flex & bison: Text Processing Tools**

Levine, John R.

O'Reilly Media, 2009

ISBN: 978-0596155971

- **The Definitive ANTLR 4 Reference**

Parr, Terence

Pragmatic Bookshelf, 2013

ISBN: 978-1934356599

- **Building a Parser with JavaCC**

Grune, Dick and Ceriel J.H. Jacobs

Addison-Wesley, 2008

ISBN: 978-0321316324

### 19.2.2 Code Generation and Optimization

- **The LLVM Compiler Infrastructure**

Lattner, Chris and Vikram Adve

ACM SIGPLAN Notices, 2004

- **Linkers and Loaders**

Levine, John R.

Morgan Kaufmann, 2000

ISBN: 978-1558604964

- **Computer Systems: A Programmer's Perspective**

Bryant, Randal E. and David R. O'Hallaron

Pearson, 2015

ISBN: 978-0134092669

## 19.3 Jurnal dan Paper Akademik

### 19.3.1 Seminal Papers

- **A Fast Algorithm for Finding Dominators in a Flowgraph**

Lengauer, Thomas and Robert E. Tarjan

ACM Transactions on Programming Languages and Systems, 1979

- **Register Allocation via Graph Coloring**

Chaitin, Gregory J.

ACM SIGPLAN Notices, 1982

- **SSA-Based Optimizations**

Cytron, Ron, et al.

ACM Transactions on Programming Languages and Systems, 1991

- **Linear Scan Register Allocation**

Poletto, Massimiliano and Vivek Sarkar

ACM SIGPLAN Notices, 1999

### 19.3.2 Recent Research

- **LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation**

Lattner, Chris and Vikram Adve

International Symposium on Code Generation and Optimization, 2004

- **The LLVM Instruction Set and Compilation Strategy**

Lattner, Chris

University of Illinois at Urbana-Champaign, 2002

- **Automatic Vectorization for Free**

Nuzman, Dorit, et al.

ACM SIGPLAN Notices, 2006

## 19.4 Dokumentasi Teknis

### 19.4.1 Compiler Tools Documentation

- **GNU Compiler Collection Internals**

Free Software Foundation

<https://gcc.gnu.org/onlinedocs/gccint/>

- **LLVM Language Reference Manual**

LLVM Project

<https://llvm.org/docs/LangRef.html>

- **Flex Manual**

Free Software Foundation

<https://westes.github.io/flex/manual/>

- **Bison Manual**

Free Software Foundation

<https://www.gnu.org/software/bison/manual/>

### 19.4.2 Architecture Specifications

- **x86-64 Architecture Programmer's Manual**

Intel Corporation

<https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html>

- **ARM Architecture Reference Manual**

ARM Limited

<https://developer.arm.com/documentation/ddi0487/latest>

- **RISC-V ISA Manual**

RISC-V Foundation

<https://riscv.org/technical/specifications/>

## 19.5 Online Resources

### 19.5.1 Course Materials

- **CS4120/CS4121: Compilers**

Cornell University

<https://www.cs.cornell.edu/courses/cs4120/2018fa/>

- **CS143: Compilers**

Stanford University

<https://www.stanford.edu/class/cs143/>

- **15-213: Introduction to Computer Systems**

Carnegie Mellon University

<https://www.cs.cmu.edu/~213/>

### 19.5.2 Tutorials and Examples

- **LLVM Tutorial: My First Language Frontend**

LLVM Project

<https://llvm.org/docs/tutorial/MyFirstLanguageFrontend/index.html>

- **Let's Build a Compiler**

Jack Crenshaw

<https://compilers.iecc.com/crenshaw/>

- **Writing a Simple Compiler**

Carl Burch

<https://www.cburch.com/cs/330/>

## 19.6 Standar dan Spesifikasi

### 19.6.1 Programming Language Standards

- **ISO/IEC 9899:2018 - C Programming Language**

International Organization for Standardization

- **ISO/IEC 14882:2020 - C++ Programming Language**

International Organization for Standardization

- **ISO/IEC 23660:2020 - Programming Language Specifications**

International Organization for Standardization

## 19.6.2 Compiler Standards

- **DWARF Debugging Information Format**

DWARF Standards Committee

<http://dwarfstd.org/>

- **ELF: Executable and Linkable Format**

UNIX System Laboratories

<https://refspecs.linuxfoundation.org/elf/>

- **COFF: Common Object File Format**

Microsoft Corporation

## 19.7 Software dan Tools

### 19.7.1 Open Source Compilers

- **GCC (GNU Compiler Collection)**

Free Software Foundation

<https://gcc.gnu.org/>

- **Clang**

LLVM Project

<https://clang.llvm.org/>

- **TinyCC (Tiny C Compiler)**

Fabrice Bellard

<https://bellard.org/tcc/>

### 19.7.2 Compiler Development Tools

- **ANTLR**

University of San Francisco

<https://www.antlr.org/>

- **Flex**

Free Software Foundation

<https://github.com/westes/flex>

- **Bison**

Free Software Foundation

<https://www.gnu.org/software/bison/>

## 19.8 Historical References

### 19.8.1 Classic Papers

- **Recursive Functions of Symbolic Expressions and Their Computation by Machine**  
McCarthy, John  
Communications of the ACM, 1960
- **A Method for Translating Algol 60**  
Bauer, Friedrich L. and Klaus Samelson  
Communications of the ACM, 1960
- **A Formal System for Specification and Verification of Compilers**  
McCarthy, John and James Painter  
Communications of the ACM, 1967

### 19.8.2 Historical Books

- **The Design of an Optimizing Compiler**  
Aho, Alfred V. and Jeffrey D. Ullman  
Elsevier, 1977
- **Compiler Construction: Theory and Practice**  
Gries, David  
Wiley, 1971
- **A Compiler Generator**  
Feldman, Joel and David Gries  
Prentice-Hall, 1970

## 19.9 Catatan Penggunaan Referensi

Referensi-referensi ini disusun untuk mendukung pembelajaran Teknik Kompilasi secara komprehensif:

- **Buku Utama:** Digunakan sebagai referensi fundamental untuk konsep dan teori
- **Buku Praktis:** Memberikan panduan implementasi dan penggunaan tools
- **Paper Akademik:** Menyajikan penelitian terkini dan algoritma modern
- **Dokumentasi Teknis:** Referensi spesifik untuk tools dan arsitektur

- **Online Resources:** Materi tambahan dan tutorial interaktif
- **Standar:** Spesifikasi resmi untuk bahasa dan format
- **Software:** Tools yang dapat digunakan untuk praktikum
- **Historical:** Konteks perkembangan compiler construction

Semua referensi ini dapat diakses melalui perpustakaan universitas, online repositories, atau pembelian langsung. Mahasiswa disarankan untuk memilih referensi yang sesuai dengan kebutuhan pembelajaran dan proyek yang sedang dikerjakan.

## Aktivitas Pembelajaran

1. **Reference Deep Dive:** Pilih satu buku referensi utama dan rangkum satu bab.
2. **Paper Analysis:** Baca paper seminal tentang register allocation dan analisis algoritmanya.
3. **Tool Investigation:** Eksplorasi dokumentasi LLVM atau GCC untuk fitur optimasi tertentu.
4. **Bibliography Management:** Buat database referensi pribadi menggunakan BibTeX.
5. **Resource Mapping:** Petakan referensi ke setiap Sub-CPMK yang ada dalam mata kuliah.

## Latihan dan Refleksi

1. Temukan perbedaan pendekatan antara Dragon Book dan Engineering a Compiler!
2. Cari paper terbaru (3 tahun terakhir) tentang compiler optimization di ACM Digital Library!
3. Bandingkan spesifikasi x86 dan ARM dalam hal calling conventions dari manual teknis!
4. Identifikasi kontribusi John McCarthy dalam sejarah teknik kompilasi!
5. Susun rencana membaca referensi untuk mendukung proyek portofolio Anda!
6. **Refleksi:** Bagaimana keragaman referensi membantu Anda memahami kompleksitas teknik kompilasi?

## Asesmen (Evaluasi Kinerja)

### Instrumen Evaluasi Kemampuan Literasi

#### A. Review Literatur

1. Kualitas sitasi dalam laporan proyek
2. Kedalaman analisis perbandingan antar referensi
3. Ketepatan penggunaan standar teknis

#### B. Presentasi Referensi

1. Kemampuan menjelaskan konsep dari paper akademik
2. Relevansi referensi yang dipilih dengan masalah teknis yang dihadapi

**Rubrik Penilaian:** Lihat Lampiran A

## Checklist Pencapaian Kompetensi

*Centang item berikut setelah Anda yakin telah menguasainya:*

- Saya mengenal buku-buku referensi utama dalam bidang teknik kompilasi
- Saya dapat menelusuri paper akademik untuk algoritma spesifik
- Saya mahir membaca dan mengikuti dokumentasi teknis compiler tools
- Saya memahami standar dan spesifikasi arsitektur serta format file
- Saya dapat menggunakan online resources dan tutorial secara efektif
- Saya memahami konteks sejarah dan perkembangan teknologi kompilator

## Rangkuman

Bab ini menyajikan daftar referensi komprehensif yang mencakup buku teks, paper akademik, dokumentasi teknis, dan online resources. Mahasiswa belajar memanfaatkan berbagai sumber informasi untuk memperdalam pemahaman teknik kompilasi.

### Poin Kunci:

- Literatur klasik memberikan fondasi teori yang kuat
- Dokumentasi modern memberikan panduan praktis implementasi

## BAB 19. DAFTAR REFERENSI

---

- Research papers menunjukkan arah perkembangan teknologi masa depan
- Standar teknis memastikan interoperabilitas dan kepatuhan sistem
- Pembelajaran mandiri didorong melalui eksplorasi berbagai media belajar

**Kata Kunci:** *Referensi, Literatur, Paper Akademik, Dokumentasi, Standar, Historical Context*